

Documentation 6.0

ZABBIX

17.05.2024

Contents

Zabbix マニュアル	6
著作権表示	6
1. はじめに	6
1 マニュアルの構成	6
2 Zabbix とは	7
3 Zabbix の機能	7
4 Zabbix の概要	8
5 Zabbix 6.0.0 の新機能	9
6 Zabbix 6.0.1 の新機能	20
7 Zabbix 6.0.2 の新機能	20
8 Zabbix 6.0.3 の新機能	21
9 Zabbix 6.0.4 の新機能	21
10 Zabbix 6.0.5 の新機能	22
11 Zabbix 6.0.6 の新機能	23
12 Zabbix 6.0.7 の新機能	23
13 Zabbix 6.0.8 の新機能	24
14 Zabbix 6.0.9 の新機能	24
15 Zabbix 6.0.10 の新機能	24
16 Zabbix 6.0.11 の新機能	25
17 Zabbix 6.0.12 の新機能	25
18 Zabbix 6.0.13 の新機能	25
19 Zabbix 6.0.14 の新機能	27
20 Zabbix 6.0.15 の新機能	27
21 Zabbix 6.0.16 の新機能	28
22 Zabbix 6.0.17 の新機能	28
23 Zabbix 6.0.18 の新機能	28
24 Zabbix 6.0.19 の新機能	29
25 Zabbix 6.0.20 の新機能	29
26 Zabbix 6.0.21 の新機能	30
27 Zabbix 6.0.22 の新機能	30
28 Zabbix 6.0.23 の新機能	31
29 Zabbix 6.0.24 の新機能	32
30 Zabbix 6.0.25 の新機能	32
31 Zabbix 6.0.26 の新機能	32
32 Zabbix 6.0.27 の新機能	33
33 Zabbix 6.0.28 の新機能	34
34 What's new in Zabbix 6.0.29	34
35 What's new in Zabbix 6.0.30	34
2. 定義	35
3. Zabbix のプロセス	37
1 Zabbix サーバー	37
2 Zabbix エージェント	44
3 Zabbix エージェント 2	47
4 Zabbix プロキシ	49
5 Zabbix Java ゲートウェイ	52
6 zabbix_sender コマンド	56
7 zabbix_get コマンド	57
8 zabbix_js コマンド	57
9 Zabbix Web サービス	58
4. インストール	58

1 Zabbix の入手	58
2 要件	59
3 ソースコードからのインストール	73
4 パッケージインストール	83
5 コンテナでのインストール	99
6 Web インタフェースのインストール	106
7 アップグレード手順	113
8 既知の問題点	124
9 テンプレート変更点	130
10 6.0.0 へのアップグレードノート	133
11 6.0.1 へのアップグレードノート	135
12 6.0.2 へのアップグレードノート	136
13 6.0.3 へのアップグレードノート	136
14 6.0.4 へのアップグレードノート	136
15 6.0.5 へのアップグレードノート	136
16 6.0.6 へのアップグレードノート	136
17 6.0.7 へのアップグレードノート	136
18 6.0.8 へのアップグレードノート	136
19 6.0.9 へのアップグレードノート	137
20 6.0.10 へのアップグレードノート	137
21 6.0.11 へのアップグレードノート	137
22 6.0.12 へのアップグレードノート	137
23 6.0.13 へのアップグレードノート	137
24 6.0.14 へのアップグレードノート	138
25 6.0.15 へのアップグレードノート	138
26 6.0.16 へのアップグレードノート	138
27 6.0.17 へのアップグレードノート	138
28 6.0.18 へのアップグレードノート	139
29 6.0.19 へのアップグレードノート	139
30 Upgrade notes for 6.0.20	139
31 Upgrade notes for 6.0.21	139
32 Upgrade notes for 6.0.22	139
33 Upgrade notes for 6.0.23	139
34 Upgrade notes for 6.0.24	140
35 Upgrade notes for 6.0.25	140
36 Upgrade notes for 6.0.26	140
37 Upgrade notes for 6.0.27	140
38 Upgrade notes for 6.0.28	141
39 Upgrade notes for 6.0.29	141
5. クイックスタート	141
1 ログインとユーザー設定	141
3 新規アイテム	146
4 新規トリガー	148
5 障害発生時の通知	150
6 新規テンプレート	154
6. Zabbix アプライアンス	155
7. 設定	158
1 ホストとホストグループ	166
2 アイテム	176
3 トリガー	418
4 イベント	436
5 イベント相関	440
6 タグ付け	447
7 視覚化	450
8 テンプレート	477
9 すぐに使えるテンプレート (Templates out of the box)	478
10 イベント時の通知	505
11 マクロ	551
12 ユーザーとユーザーグループ	561
13 secrets の保存	569
14 スケジュールレポート	570
8. サービス監視	574
1 サービスツリー	575

2 サービスアクション	578
3 SLA	580
4 セットアップ例	582
9. Web 監視	587
1 Web 監視アイテム	596
2 シナリオの実例	598
10. 仮想マシン監視	606
1 仮想マシンディスクカバリキーフィールド	611
11. メンテナンス	616
12. 正規表現	621
13. 障害イベントの確認	625
14. 設定のエクスポート/インポート	627
1 ホストグループ	629
2 テンプレート	629
3 ホスト	657
4 ネットワークマップ	681
5 メディアタイプ	690
15. ディスカバリ	698
1 ネットワークディスクカバリ	698
2 アクティブエージェントの自動登録	708
3 ローレベルディスクカバリ	711
16 分散監視	763
1 プロキシ	764
17. 暗号化	768
1 証明書の利用	775
2 事前共有鍵の使用	781
3 トラブルシューティング	783
18 Web インターフェース	786
1 メニュー	786
2 フロントエンドセクション	787
3 ユーザー設定	945
4 グローバルサーチ	949
5 フロントエンドメンテナンスモード	951
6 ページパラメーター	952
7 定義	953
8 独自テーマ作成	953
9 デバッグモード	954
10 Zabbix 用 Cookie	955
11 タイムゾーン	956
13 パスワードのリセット	957
19. API	957
メソッドリファレンス	962
6.0 での Zabbix API 変更内容	1523
付録 1. 参考文献の解説	1525
付録 2. 5.4 から 6.0 への変更点	1530
20. モジュール	1534
21. 付録	1540
1 FAQ / トラブルシューティング	1540
2 インストールとセットアップ	1541
3 プロセスの設定	1578
4 プロトコル	1635
5 アイテム	1660
6 サポートされている関数	1693
7 マクロ	1720
8 単位と記号	1750
9 日時フォーマット	1751
10 コマンドの実行	1752
11 バージョン間の互換性	1753
12 データベースエラーハンドリング	1753
13 Windows 用の Zabbix sender ダイナミックリンクライブラリ	1754
14 Python library for Zabbix API	1754
14 サービスモニタリングのアップグレード	1755
15 その他の問題	1755

16 Agent と agent 2 の比較	1756
18 Escaping examples	1757
Zabbix リファレンスマニュアル	1759
zabbix_agent2	1759
名前	1759
概要	1759
説明	1759
オプション	1760
ファイル	1760
参照	1761
著者	1761
目次	1761
zabbix_agentd	1761
名前	1761
概要	1761
説明	1761
オプション	1761
ファイル	1762
参照	1762
著者	1763
目次	1763
zabbix_get	1763
名前	1763
概要	1763
説明	1763
オプション	1763
例	1764
参照	1764
著者	1765
目次	1765
zabbix_js	1765
名前	1765
概要	1765
説明	1765
オプション	1765
例	1766
参照	1766
目次	1766
zabbix_proxy	1766
名前	1766
概要	1766
説明	1766
オプション	1767
ファイル	1767
参照	1767
著者	1767
目次	1768
zabbix_sender	1768
名前	1768
概要	1768
説明	1769
オプション	1769
終了ステータス	1770
例	1771
参照	1771
著者	1771
目次	1771
zabbix_server	1772
名前	1772
概要	1772
説明	1772
オプション	1772

ファイル	1773
参照	1773
著者	1773
目次	1774
zabbix_web_service	1774
名前	1774
概要	1774
説明	1774
オプション	1774
ファイル	1774
参照	1774
著者	1775
目次	1775

Zabbix マニュアル

Zabbix のユーザーマニュアルによろ。これらのページは、ユーザーが Zabbix を使用して、単純なことからより複雑なことまで、監視タスクを正常に管理できるようにするために作成されています。

著作権表示

Zabbix のドキュメントは、GPL ライセンスの下で配布されていません。Zabbix ドキュメントの使用には、次の条件が適用されます。

個人的な使用のみを目的として、このドキュメントの印刷コピーを作成することができます。実際の内容が変更または編集されない限り、他の形式への変換は許可されます。Zabbix が配布する方法と同様の方法（つまり、Zabbix Web サイトでダウンロードするために電子的にダウンロードする）または USB または同様の方法でドキュメントを配布する場合を除き、いかなる形式またはメディアでも、このドキュメントを公開または配布してはなりません。ただし、ドキュメンテーションがソフトウェアと一緒に同じ媒体で配布されることを条件とします。印刷物を配布したり、このドキュメントの全体または一部を別の出版物で使用したりするなど、その他の使用には、Zabbix の正式な代表者からの事前の書面による同意が必要です。Zabbix は、上記で明示的に付与されていない、このドキュメントに対するすべての権利を留保します。

1. はじめに

このセクションのコンテンツを参照するためには、サイドバーから項目を選択してください。

1 マニュアルの構成

構成

このマニュアルは、興味のあるトピックを探しやすくするために、セクションとサブセクションに分かれています。

各セクションに移動するときには、サイドバーを展開して、サブセクションや個々のページの内容がすべて見えるようにしてください。

ユーザーが関係する情報を見落とさないように、できる限り多くの関連するコンテンツ間のクロスリンクを貼るようにしています。

セクション

Zabbix とはで、現バージョンの Zabbix に関する一般的な情報を提供しています。このセクションを読むと、なぜ Zabbix を選ぶべきなのかがわかります。

定義では、Zabbix で使用される用語を解説し、Zabbix コンポーネントに関する詳細を提供します。

インストールとクイックスタートのセクションは、Zabbix を使い始めるためのガイドになっています。**Zabbix アプライアンス**で、Zabbix を使用感を試すこともできます。

設定は、このマニュアルの中で最大かつとても重要なセクションの一つです。ホストのセットアップから、重要なデータの取得、データの閲覧、通知の設定、障害発生時に実施されるリモートコマンドなど、あなたの環境を監視するための Zabbix のセットアップ方法について不可欠なアドバイスがたくさん掲載されています。

サービス監視では、あなたの監視環境の高レベルな概要把握のための Zabbix の使い方の詳細を説明します。

Web 監視では、Web サイトの有効性を監視する方法を学ぶことができます。

仮想マシン監視では、VMware 環境の監視の設定方法を紹介します。

メンテナンス、正規表現、イベント通知とXMLのエクスポート/インポートは、Zabbix の様々な機能について、使用方法を説明するセクションです。

ディスカバリには、ネットワークデバイス、アクティブなエージェント、ファイルシステム、ネットワークインターフェースなどの自動ディスカバリの設定手順が書かれています。

分散監視では、より大規模で複雑な環境で Zabbix を使用方法について説明しています。

暗号化は、Zabbix コンポーネント間の通信暗号化を説明しています。

Web インターフェースには、Zabbix の Web インターフェースを使用するための情報が書かれています。

APIセクションでは、Zabbix の API を使用する際の詳細情報が書かれています。

技術情報の詳細なリストは**付録**に掲載されています。付録には、FAQ のセクションも掲載されています。

2 Zabbix とは

概要

Zabbix は Alexei Vladishev によって作成され、現在は Zabbix SIA が積極的に開発およびサポートを行っています。

Zabbix はエンタープライズに対応した監視ソリューションであり、オープンソースで配布されています。

Zabbix は、多数のネットワークのパラメータおよびサーバー、仮想マシン、アプリケーション、サービス、データベース、Web サイト、クラウドなどの稼働状態と整合性を監視するためのソフトウェアです。Zabbix は柔軟な通知メカニズムを備え、すべてのイベントに対して電子メールによるアラートを設定できます。これによって、サーバーの問題に迅速に対応できます。Zabbix は、保存されたデータをもとに優れたレポート機能とデータ視覚化機能を提供し、キャパシティプランニングに最適なソフトウェアとなっています。

Zabbix は、ポーリングとトラッピングの両方をサポートします。すべての Zabbix のレポート、統計情報、および設定パラメーターは、Web インターフェースからアクセスできます。Web インターフェースは、どのような場所からでも、あなたのネットワークとサーバーの状態を確認することができます。適切に設定することで、IT インフラストラクチャを監視する重要なツールとして Zabbix を活用することができます。少数のサーバーを有する小規模な組織から、大量のサーバーを有する大企業まで、同じように利用できます。

Zabbix は無料です。Zabbix は GPL (General Public License) バージョン 2 の下で作成および配布されています。ソースコードは無料で配布され、誰でも自由に利用できます。

Zabbix 社と世界中のパートナーが提供する [商用サポート](#) を利用することも可能です。

詳細は、[Zabbix の機能](#)をご覧ください。

Zabbix のユーザー

世界中のあらゆる規模の多くの企業や組織が、メインの監視プラットフォームとして Zabbix を採用しています。

3 Zabbix の機能

概要

Zabbix は、高度に統合されたネットワーク監視ソリューションであり、1 つのパッケージで多様な機能を提供します。

データの収集

- 可用性とパフォーマンスのチェック
- ポーリングとトラッピング双方の SNMP、IPMI、JMX、VMware 監視のサポート
- チェックのカスタマイズ
- カスタマイズした間隔での詳細データの収集
- サーバーまたはプロキシによる実行と、エージェントによる実行

柔軟な閾値定義

- バックエンドデータベースからの値を参照して、トリガーと呼ばれる非常に柔軟な障害の閾値を定義可能

高度に設定可能なアラーム機能

- 通知スケジュール、受信者、メディアなどをカスタマイズして通知可能
- マクロ変数を使用することにより、通知内容を分かりやすくすることが可能
- リモートコマンドを含む自動実行

リアルタイムグラフ

- 監視アイテムは組み込みのグラフ機能を使用して瞬時にグラフ化

Web 監視機能

- Web サイト上でのマウスクリックの動きを追跡し、その機能と応答時間をチェック可能

幅広い可視化オプション

- 複数のアイテムを 1 つのビューに表示できるカスタムグラフ
- ネットワークマップ
- ダッシュボードスタイルで概要を表示するスライドショー
- レポート
- 監視リソースの高度な (ビジネス向け) 表示機能

履歴データのストレージ

- データは 1 つのデータベースに格納
- 履歴の設定が可能
- 組み込みのハウスキューピング処理

わかりやすい設定

- 監視対象デバイスをホストとして追加
- データベースに登録されるとホストは監視対象とされる
- 監視対象デバイスへのテンプレート適用

テンプレートの使用

- テンプレート内での監視項目のグループ化
- テンプレートは他のテンプレートを継承可能

ネットワークディスカバリ

- ネットワークデバイスのオートディスカバリ
- エージェントの自動登録
- ファイルシステム、ネットワークインターフェース、SNMP OID のディスカバリ

高速な Web インターフェース

- PHP で記述された Web ベースのインターフェース
- どこからでもアクセス可能
- クリックだけですべて実行可能
- 監査ログ

Zabbix API

- 大規模な操作やサードパーティのソフトウェア開発、その他の目的のために Zabbix をプログラムから操作できるインターフェースを提供

パーミッション管理

- セキュリティで保護されたユーザー認証
- 特定のユーザー用の限定されたビュー

高機能かつ容易に拡張可能なエージェント

- 監視対象上に配置
- Linux と Windows の両方に配置可能

バイナリデーモン

- C 言語で記述されているため、高いパフォーマンスと少ないメモリ占有
- 移植が容易

複雑な環境への対応

- Zabbix プロキシを使用することでリモート監視を容易化

4 Zabbix の概要

アーキテクチャー

Zabbix は、いくつかの主なソフトウェアコンポーネントから構成されています。各コンポーネントの役割を以下に示します。

サーバー

Zabbix サーバーは、エージェントから可用性や整合性の情報や統計情報を収集する中核のコンポーネントです。サーバーは中核のレポジトリで、すべての設定や統計データ、運用データを保存します。

データベースストレージ

Zabbix によって収集されたデータと同様に、すべての設定情報もデータベースに保存されます。

Web インターフェース

どこからでもどんなプラットフォームからでも、Zabbix に簡単にアクセスできるように、Web インターフェースが提供されています。インターフェースは Zabbix サーバーの一部であり、通常はサーバーと同じ物理サーバー上で動作しますが、別サーバーで動作させることもできます。

プロキシ

Zabbix プロキシは、Zabbix サーバーの代わりにパフォーマンスと可用性データを収集します。プロキシは Zabbix のオプション部分ですが、1 台の Zabbix サーバーの負荷を分散するのにとても有益です。

エージェント

Zabbix エージェントは、監視対象上にインストールされ、ローカルリソースやアプリケーションを能動的に監視し、収集したデータを Zabbix サーバーに送信します。Zabbix 4.4 からはエージェントとして 2 つの種類があります：**Zabbix エージェント** (C 言語で実装され、低負荷で多くのプラットフォームをサポート) と **Zabbix エージェント 2** (Go 言語で実装され、柔軟でプラグインで拡張しやすい) です。

データフロー

加えて、Zabbix 全体のデータフローを俯瞰的に見るのが非常に重要です。データを収集するアイテムを作成するためには、まず最初にホストを作成する必要があります。Zabbix で処理させるためには、最初にトリガーを作成するためのアイテムが必要です。アクションを作成するためには、トリガーが必要です。このように、例えばサーバー X で CPU 負荷が高すぎる時に通知を受け取りたい場合は、まずサーバー X のホストを作成し、続いて CPU を監視するためのアイテムを作成し、それから、CPU 負荷が高い場合に作動するトリガーを作成し、あなたにメールを送信するアクションを作成します。沢山のステップがあるように感じるかもしれませんが、テンプレートを使用すれば実はそんなことはありません。この設計であるため、非常に柔軟な構成が可能になっています。

5 Zabbix 6.0.0 の新機能

このバージョンでの**重大な変更点**はこちらを参照してください。

Zabbix サーバの高可用性 (HA) クラスタリング

新バージョンでは、Zabbix サーバの高可用性ソリューションをネイティブで提供します。

このソリューションは、複数の `zabbix_server` インスタンスまたはノードで構成されます。一度にアクティブ (現用ノード) にできるのは 1 ノードのみで他のノードはスタンバイ状態にあり、現用ノードの停止や障害発生時にスタンバイノードに引き継ぐことができます。

参照: [高可用性 \(HA\) クラスタリング](#)

サービス監視

サービス監視について、いくつかのアップデートが行われました。サービス監視では、Zabbix で監視しているインフラストラクチャの概要を提供します。

タグベースの障害へのサービスのマッピング

サービスは、旧バージョンの Zabbix ではトリガーとその状態に依存していました。新しいバージョンでは、この問題はタグベースのマッピングに置き換えられ、各サービスの問題に対応するようになりました。

サービス構成では、ハード依存とソフト依存はもはや存在しません。代わりに、1 つのサービスは複数の親サービスを持つことができます。

サービスメニュー

Zabbix に 4 つのメニューセクションからなる新しいサービスメニューが追加されました。

- **サービス** - サービス概要とサービス構成用 (監視から サービスへ移動)
- **サービスアクション** - サービスアクション (新しいアクションタイプ)
- **SLA** - SLA の構成用
- **SLA レポート** - SLA レポート用 (ダッシュボードウィジェットとしても利用可能)

Name	Status
Availability 2	High
Disc space	OK
Example service	OK

サービスの設定のための別のセクション、設定はもう存在しないことに注意してください。

ステータス計算と伝搬ルール

直接の子サービスのステータスや重みから、親サービスのステータスを計算するための新しいステータス計算ルールと柔軟な追加ルールが用意されました。また、親サービスに状態を伝搬するためのルールを柔軟に設定できるようになりました。

サービスのパーミッション設定

サービスに対する柔軟なパーミッションが**ユーザーの役割**のレベルで実装されました。(サービス名やタグに基づいて) 選択されたサービスに対して、読み書きまたは読み取り専用アクセス権限を与えることができます。

根本原因の分析

新しい根本原因列には、サービス状態に直接的または間接的に影響を与える根本的な問題が列挙されています。

Services

Name	Status	Root cause
Availability 2	High	Nodata trigger, Nodata trigger 1h

監視データ → 障害で障害名をクリックすればその障害の詳細を見ることができます。

サービスステータス変更時のアラート

トリガーの状態変化に関するアラートと同様に、サービスの状態変化に関する自動的なアラートを受け取ることができるようになりました。

Zabbix の他のアクションと同じように、新しく**サービスアクション**機能が追加されました。サービスアクションには、サービスに関連する問題解決、復旧、更新操作の手順が含まれます。サービスアクションは指定された受信者にメッセージを送信するアクションと指定された受信者にリモートでアクションを実行するアクションの 2 種類を設定することが可能です。

トリガーアクションと同様に、サービスアクションは問題の**エスカレーション**シナリオをサポートします。

新しいメッセージテンプレート サービス、サービスの復旧、サービスの更新が**メディアタイプ**に追加されたため、サービスアクションの通知を正しく送信できるように定義する必要があります。

サービスの複製

サービスの複製が可能になりました。複製ボタンが、サービスの**設定フォーム**に追加されました。サービスの複製を作成すると、その親リンクは保持されますが、子リンクは保持されません。

プライマリーキー

新規インストール時に、履歴テーブルを含むすべてのテーブルでプライマリーキーを使用するようになりました。

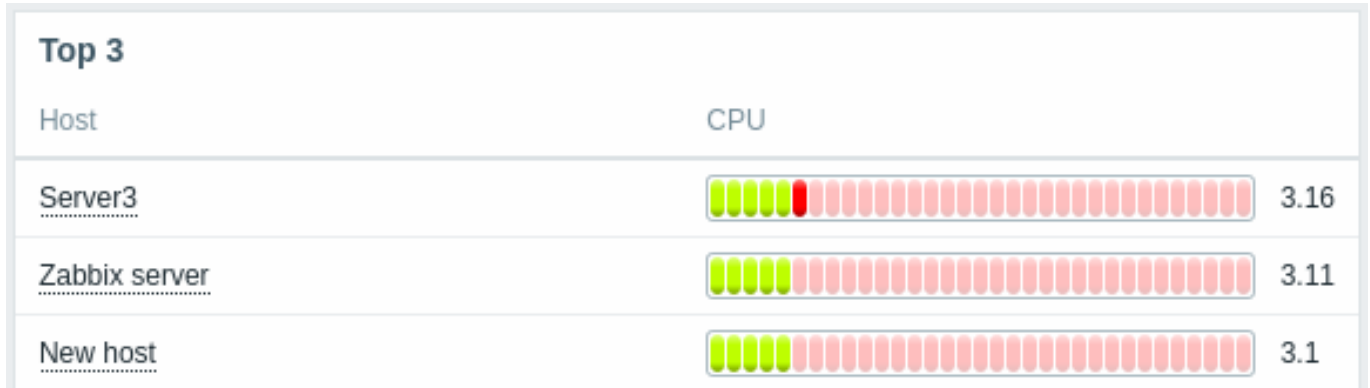
既存環境への主キーの自動アップグレードはありません。履歴テーブルを主キーに 手動でアップグレードする方法については次のリンクを参照してください。 [MySQL/MariaDB](#)、[PostgreSQL](#)、[TimescaleDB v1、v2](#)、[Oracle](#)

ウィジェット バージョン 6.0.0 では、いくつかのダッシュボードウィジェットが追加されました。

上位ホストウィジェット

ダッシュボードウィジェットに上位ホストウィジェットが追加されました。このウィジェットは、現在非推奨となっているデータの概要ウィジェットを置き換えるために設計されています。

上位ホストウィジェットでは、データ概要のためのカスタムテーブルを作成することができます。これは、上位 N ホストのようなレポートや、キャパシティ・プランニングに役立つプログレスバー形式のレポートを作成できます。



詳細は、[上位のホストウィジェット](#)をご覧ください。

アイテムの値ウィジェット

ダッシュボードウィジェットに、アイテムの値ウィジェットが追加されました。

このタイプのウィジェットは、単一の項目の値を目立つように表示するのに便利です。さまざまな表示スタイルが可能です：



詳細は、[アイテムの値ウィジェット](#)をご覧ください。

マクロ 新しいマクロ

トリガー式のデバッグと内部動作のための新しいマクロがサポートされました。

デバッグ用マクロは、トリガー条件式のデバッグを簡素化します：

- {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN} - これらは、項目ベースの関数のみに適用され、部分的に評価されたトリガー式またはリカバリトリガー式に展開されます。
- {FUNCTION.VALUE<1-9>}, {FUNCTION.RECOVERY.VALUE<1-9>} - これらは、その時の N 番目の項目ベースの関数の結果に展開されます。

内部アクションのマクロは、アイテム、LLD ルール、トリガーがサポートされなくなった理由を含んでいます。

- {ITEM.STATE.ERROR} - アイテムベースの内部通知用
- {LLDRULE.STATE.ERROR} - LLD ルールベースの内部通知用
- {TRIGGER.STATE.ERROR} - トリガーベースの内部通知用

詳しくは、[サポートされるマクロ](#)を参照してください。

単純なマクロを式マクロに置き換える

トリガーと計算アイテムのための新しい式の構文が、[Zabbix 5.4](#) で導入されました。しかし、単純なマクロでは、まだ古い構文が使われていました。新しいバージョンでは、単純なマクロの機能が式マクロに移行され、新しい式の構文を採用しました。以下の比較を参照してください：

Zabbix 6.0	Zabbix 6.0 以前
<code>{?avg(/host/key, 1h)}</code>	<code>{host:key.avg(1h)}</code>
新しいバージョンでの式のサンプル	前のバージョンでの単純なマクロのサンプル

以前の単純なマクロは、アップグレード中に式のマクロに変換されます。式のマクロの範囲は、単純なマクロで提供されていたものと同じです。したがって、式マクロは次のような場合で使用できます：

- 障害の通知とコマンド実行
- 障害の更新の通知とコマンド実行
- マップの要素のラベル
- マップ内のリンクのラベル
- マップの図形のラベル
- グラフの名前

位置指定マクロはサポートされなくなりました

アイテム名の位置指定マクロ (`$1`, `$2...$9`) のサポートは Zabbix 4.0 から非推奨となり、完全に削除されました。

アイテム名のユーザーマクロはサポートされなくなりました。

Zabbix 4.0 から非推奨となっていたアイテム名 (ディスカバリールール名を含む) のユーザーマクロのサポートは、完全に廃止されました。

Prometheus メトリクスの一括処理

保存前処理キューに依存項目の一括処理を導入し、Prometheus メトリクスの検索性能を向上させました。

詳しくは [Prometheus のチェック](#) を参照してください。

Prometheus パターンに対する結果処理

保存前処理で Prometheus パターンのステップを実行すると、複数の行がマッチすることがあります。この状況に対応するために、新しい結果処理 **パラメータ** が Prometheus パターンの保存前処理のステップに追加されました。sum、min、max、avg、count などの関数を導入することで、複数のマッチング行のデータを集約することができます。

関数 Prometheus ヒストグラム用関数

[Prometheus のメトリクス](#) を Zabbix で収集することができるようになりました。しかし、メトリクスの中には作業がしづらいものもあります。具体的に言うと、ヒストグラムタイプのメトリクスは、Zabbix では同じキー名で異なるパラメータを持つ複数のアイテムとして表示することができます。しかし、これらの項目は論理的に関連しており、同じデータを表しているにもかかわらず、専用の機能がなければ、収集されたデータを分析することは困難でした。新バージョンでは、この機能ギャップをカバーするために、**rate()** と **histogram_quantile()** 関数が追加され、PromQL の対応するものと同じ結果を生成します。

この機能を補完する他の新しい追加機能は、**bucket_rate_foreach()** と **bucket_percentile()** 関数です。詳細については以下を参照してください：

- **ヒストリ関数 (rate())** 参照
- **集計関数 (histogram_quantile(), bucket_percentile())** 参照
- **Foreach 関数 (bucket_rate_foreach())** 参照

単調変化

新しい **monoinc()** または **monodec()** **ヒストリ関数** を使用して、項目値の単調増加、単調減少をチェックできるようになりました。

変更回数

新しい **履歴関数** として隣接する値間の変更回数を数えることができる **changecount()** が追加されました。

この機能は、次の 3 つのモードをサポートしています。すべての変化、減少のみ、または増加のみをカウントします。例としてユーザー数の変化や、システム稼働時間の減少を追跡するために使用することができます。

エンティティ数

foreach 関数で返却される、特定のホストや項目のカウントを簡略化するための新しい **関数** が追加されました。

集計関数:

- **count** - foreach 関数が返す配列内の値の合計 (整数を返す)

- **item_count** - 現在有効な項目のうち、フィルタ条件に一致するものの総数 (整数を返す)

Foreach 関数:

- **exists_foreach** - 現在有効な項目のうち、フィルター条件に一致するものの数 (配列を返す)

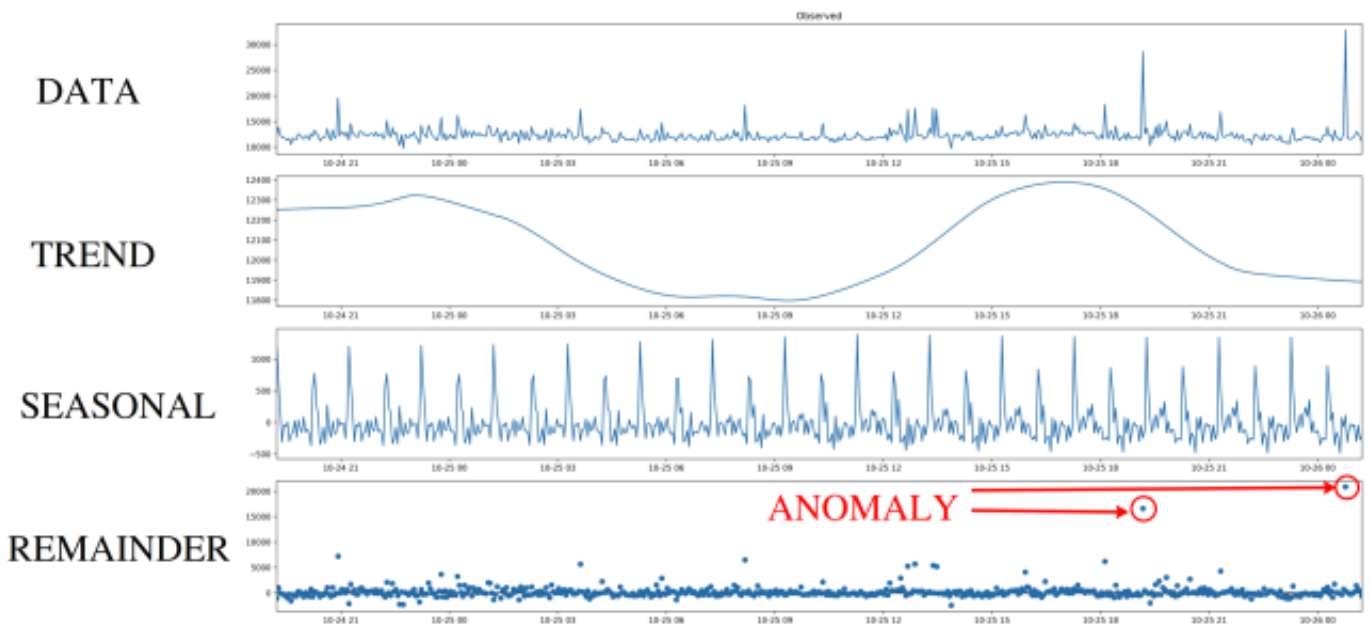
アノマリー検知

Zabbix 5.2 では、ベースライン監視に役立つ新しいトレンド関数が導入されました。しかし、この機能でも相対的な閾値を定義する必要があります。(例: 2021 年 9 月の Web トラフィックが 2020 年 9 月に比べて 2 倍以下であることをチェックする)。実際のところ、このような閾値の定義が困難なケースもあります。例えば、新しいが非常に人気のあるウェブサイトのトラフィックは、1 年間で何倍にも有機的に成長するが、その成長率は未知数です。しかし、DDOS 攻撃による急激なトラフィックの増加は、有機的なトラフィックの増加とは無関係にアラートを発生させなければいけません。

異常値検出アルゴリズムは、まさにこれを行うもので、他の値のコンテキストの中で正常に見えないデータ (異常値) を見つけるものです。

新しい履歴関数 **trendstl()** が追加され、異常率を計算するために 'decomposition' メソッドを使用するようになりました。この関数は、1 つの時系列を 3 つの時系列に分割します。

- **トレンドシーケンス**: 元データの大きな変化のみを含む (例: ウェブサイトのトラフィックが増加している)
- **シーズンシーケンス**: 季節的な変化のみを含む (例: 夏にはウェブサイトのトラフィックが少なく、秋には多くなる)
- **トレンドの一部とも季節の一部とも解釈できない残差のみを含むシーケンス**



異常検出は余りの列で動作し、余りの値の大多数から離れすぎている (far from) 値があるかどうかをチェックします。"離れすぎている (Far)" とは、余りの配列からの絶対値が、標準偏差または平均偏差の N 倍であることを意味します。

文字列関数

文字列関数 **concat** は、2 つ以上のパラメータを連結して使用できるようになりました。文字列と値を異なる組み合わせで結合したり、2 つ以上の値を互いに付加したりするのに利用できます。数値データ型もサポートされています。

アイテム 自動タイプ選択機能

アイテム設定画面で、選択されたアイテムキーが特定のタイプのデータのみを返す場合、一致する情報のタイプを自動的に提案するようになりました。(例えば、**log[]** アイテムは、データ型: ログであることが必要) データ型パラメーターがアイテムタブの Key パラメーターの下に配置され、少なくとも 1 つの保存前処理が定義されている場合は、保存前処理タブに複製されます。選択したタイプの情報とキーが一致しない可能性がある場合、警告アイコンがデータ型フィールドの横に表示されます。

新しいエージェントのアイテム

Zabbix エージェント/エージェント 2 にいくつかの新しいアイテムが追加されました。

- **agent.hostmetadata** - ホストのメタデータを返却
- **kernel.openfiles** - オープンしているファイルディスクリプタの数を返す
- **net.tcp.socket.count[]** - パラメーターに合致した TCP ソケットの数を返す
- **net.udp.socket.count[]** - パラメーターに合致した UDP ソケットの数を返す
- **vfs.dir.get[]** - ディレクトリにあるファイルのリストを JSON で返す
- **vfs.file.get[]** - ファイルに関する情報を JSON で返す

- **vfs.file.owner[]** - ファイルの所有者を返す
- **vfs.file.permissions[]** - ファイルの UNIX パーミッションを 8 進 4 桁数字文字列で返す

さらに:

- **vfs.file.cksum[]** 第 2 パラメータの mode で (crc32, md5, sha256) をサポート
- **vfs.file.size[]** 第 2 パラメータの mode で (bytes or lines) をサポート
- **vfs.fs.discovery** と **vfs.fs.get** が Windows 上で {#FSLABEL} を (ボリューム名と一緒に) 返す

詳細については、[Zabbix エージェントのアイテム](#)を参照してください。

計算アイテムのデータ型

計算アイテムは、数値だけでなく、テキスト、ログ、文字型の情報にも対応しました。

エージェントの再起動を伴わないユーザーパラメータの再読み込み

ユーザーパラメータを設定ファイルから再読み込みできるようになりました。これを行うには、新しい `userparameter_reload` ランタイムコントロールオプションを実行します。

```
zabbix_agentd -R userparameter_reload
```

または

```
zabbix_agent2 -R userparameter_reload
```

UserParameter は、このコマンドで再読み込みされる唯一のエージェントの設定オプションです。

BSD 系 OS におけるランタイム制御について

これまで、Zabbix サーバーと Zabbix プロキシの実行時制御オプションは、BSD ベースのシステムには対応していませんでした。ランタイムコマンドの転送方法を変更することで、この制限を撤廃することができました。FreeBSD、NetBSD、OpenBSD、その他 ***BSD** 系の OS でほとんどのコマンドをサポートすることができるようになりました。

正確なリストは、ランタイム制御の[サーバー向け](#) または [プロキシ向け](#) を参照してください。

Zabbix エージェント 2 プラグイン

外部プラグインローダー

これまで、プラグインは Zabbix エージェント 2 にのみコンパイルして組み込むことができ、利用可能なプラグインのセットを変更する必要があるたびに、エージェントを再コンパイルする必要がありました。外部プラグインローダーが追加されたことで、プラグインをエージェント 2 に直接統合する必要がなくなり、個別の外部アドオン (ロード可能なプラグイン) として追加できるようになったため、新しい監視メトリクスを収集するための追加プラグインの作成プロセスが容易になりました。

ロード可能な外部プラグインの導入に伴い、以下の設定パラメータを変更しました:

- `Plugins.<PluginName>.Path` パラメーターは `Plugins.<PluginName>.System.Path` へ移動しました。
- `Plugins.<PluginName>.Capacity` パラメーターは、まだサポートされていますが、非推奨となりました。代わりに `Plugins.<PluginName>.System.Capacity` をご使用ください。

パスワードの要件

Zabbix のパスワードの複雑さの要件を内部[認証方法](#)で指定できるようになりました。Zabbix ユーザーが脆弱なパスワードを設定することを防ぐため、以下のような制限をかけることができます:

- パスワードの最小の長さを設定する。
- パスワードには、大文字と小文字、数字、特殊文字を組み合わせたものを要求する。大文字、小文字、数字、特殊文字を含むパスワードを設定する。
- 最も一般的で容易に推測可能なパスワードの使用を禁止する。

データベース 最適なユーザーエクスペリエンスを実現し、様々な実環境で最高の Zabbix のパフォーマンスを確保するために一部の古いデータベースリリースのサポートは終了しました。これは主に、サポート期間の終了に近いデータベースバージョンと通常のパフォーマンスに支障をきたす可能性のある未修整の問題があるバージョンに適用されます。

Zabbix 6.0 から、公式にサポートしているバージョンの[データベース](#)は以下の通りです。

- MySQL/Percona 8.0.X
- MariaDB 10.5.X - 10.6.X
- PostgreSQL 13.X - 14.X
- Oracle 19c - 21c
- TimescaleDB 2.0.1 - 2.3
- SQLite 3.3.5 - 3.34.X

デフォルトでは、サポートされていないデータベースのバージョンが検出された場合 Zabbix server と proxy は起動しません。ただし推奨はしませんが、**サーバーやプロキシ**の AllowUnsupportedDBVersions 設定パラメータを変更することで、DB バージョンをチェックしないようにすることができます。

MySQL の utf8mb4 対応について

MySQL/MariaDB でのインストール時の、utf8mb4 エンコーディングと utf8mb4_bin 照合順序がサポートされました。

以前は utf8 エンコーディングのみがサポートされていましたが、MySQL の場合、これは utf8mb3 encoding の略で適切な UTF-8 文字のサブセットのみをサポートしていました。新しいバージョンでは、**すべての utf8mb4**サポートが追加されました。utf8mb3 を使用している古いインストールはそのまま維持され、そのエンコードを使用し続けることができます。

Zabbix 6.0 へのアップグレード後の**utf8mb4 への変換**の実行手順も参照してください。

プロセス Zabbix get と Zabbix sender のタイムアウト

Zabbix get と Zabbix sender ユーティリティは、-t <seconds> または--timeout <seconds> のタイムアウトパラメータをサポートするようになりました。有効な範囲は以下のとおりです：

- Zabbix get は 1-30 秒 (デフォルト: 30 秒)
- Zabbix sender は 1-300 秒 (デフォルト: 60 秒)

SNMP ゲートウェイの機能拡張

SNMP ゲートウェイが障害状態にあるトリガーに関する情報を提供し、トリガーの詳細でホスト情報を明らかにできるようになりました。

さらに、SNMP ゲートウェイから送信される SNMP トラップのレートを制限できるようになりました。

サポートされる OID のリストは、トリガーホスト名のカンマ区切りのリスト用の新しい OID **.10** で拡張されました。

SNMP ゲートウェイ設定ファイルに新しいパラメーターが追加されました：

- ProblemBaseOID - 障害トリガーテーブルの OID;
- ProblemMinSeverity - 最小の深刻度、それより低い深刻度のトリガーは含まれません;
- ProblemHideAck - 指定した場合、未確認の障害があるトリガーのみが含まれます;
- ProblemTagFilter - 指定した場合、指定したタグ名を持つトリガーのみが含まれます;
- TrapTimer - 設定されている場合、Zabbix は指定された時間内に深刻度が最も高いトラップを 1 つしか送信しません。

詳細は[Zabbix SNMP Gateway](#) を参照してください。

Web 監視における圧縮コンテンツ

Zabbix の Web 監視に圧縮コンテンツの処理機能が追加されました。**libcurl** でサポートされているすべてのエンコーディング形式がサポートされます。

保存前処理 Prometheus のクエリ

Zabbix の Prometheus 保存前処理での**クエリ言語**で、以下の 2 つのラベルマッチングオペレータがサポートされるようになりました。

- != -- 指定された文字列と等しくないラベルを選択します
- !~ -- 指定された文字列と正規表現一致しないラベルを選択します

JavaScript のメソッド

HTTP メソッド PATCH, HEAD, OPTIONS, TRACE, CONNECT を JavaScript エンジンに追加しました。また、新しい JS メソッド XMLHttpRequest.customRequest を使って、カスタム HTTP メソッドのリクエストを送信できるようになりました。

追加の JavaScript オブジェクトもあわせて参照してください。

監査ログ

監査レコード

監査ログに、LLD ルール、ネットワークディスカバリーアクション、自動登録アクション、スクリプト実行の結果として発生した変更を含む、すべての Zabbix オブジェクトの設定変更に関するレコードが監査ログに含まれるようになりました。これまで、Zabbix サーバーからの設定変更 (例：ディスカバリールールの実行) は記録されませんでした。今回、このようなオブジェクトの変更は、System ユーザーに起因する監査レコードとして保存されます。

レコードフィルター

レコードを、そのエントリーの原因となった Web インターフェースの操作でフィルタリングする機能が追加されました。例えばテンプレートのリンク/アンリンクなど、1 つの操作の結果として複数のログレコードが作成された場合、そのレコードは同じ記録 ID を持つことになります。

監査設定

新しい**セクション** 管理 → 一般設定メニューに 監査ログが追加され、監査ログの有効化と無効化ができるようになりました。これまでデータの保存期間セクションにあった監査用の保存期間設定も、新しい監査ログセクションに移動されました。

PCRE2 のサポート

RHEL 7 以降、SLES(全バージョン)、Debian 9 以降、Ubuntu 16.04 以降の Zabbix インストールパッケージが PCRE2 のサポートが追加されました。PCRE は引き続きサポートされていますが、Zabbix は PCRE または PCRE2 のどちらか一方のライブラリでコンパイルする必要があり、両方を同時に使用することはできません。

ODBC チェックの分離処理

ODBC チェックの処理は、通常の poller プロセスから、独立したサーバーとプロキシのプロセスである ODBC poller に移動しました。この変更により、poller プロセスで作成されるデータベースへの接続数を制限することができます。以前は、ODBC チェックは Zabbix エージェントアイテムや SSH チェックなどでも動作する通常の poller プロセスによって実行されていました。

新しい設定パラメータ StartODBCPollers が Zabbix**サーバー** と **プロキシ** の設定ファイルに追加されました。

内部項目 zabbix[process,<type>] を使用すると、ODBC poller の負荷を監視できます。

Webhook 統合

Zabbix の通知から **webhook** メディアタイプを使用して **Github issues** を作成できるような統合処理を行うことができるようになりました。

テンプレート 新しい監視のための公式テンプレートが利用できます。

Kubernetes

- Kubernetes nodes by HTTP
- Kubernetes cluster state by HTTP
- Kubernetes API server by HTTP
- Kubernetes Controller manager by HTTP
- Kubernetes Scheduler by HTTP
- Kubernetes kubelet by HTTP

Kubernetes の監視を有効にするには、新しいツール **Zabbix Helm Chart** を使用し、Kubernetes クラスタに Zabbix プロキシと Zabbix エージェントをインストールする必要があります。

テンプレートの設定の詳細については、**HTTP テンプレートの操作** を参照してください。

Mikrotik

- MikroTik <device model> SNMP - MikroTik のイーサネットルーターとスイッチの様々なモデルを監視するための 53 の新しいモデル固有のテンプレートです。 **完全なリスト** を参照してください。
- Mikrotik SNMP - MikroTik デバイスを監視するための汎用テンプレートです。

これらのテンプレートは以下の方法で取得できます:

- 新規インストールの場合は 設定 → テンプレートで利用可能です。
- 旧バージョンからアップグレードする場合、最新のテンプレートは **Zabbix の Git リポジトリ** からダウンロードし、手動で Zabbix の設定 → テンプレートセクションにインポートします。同じ名前のテンプレートが既に存在する場合、インポートする前に、存在しない場合に削除オプションをチェックし、クリーンなインポートを実現します。この方法では、更新されたテンプレートから除外された項目が削除されます。(ただし、削除された項目の履歴は失われます)

通知 テンプレートリンクの視認性向上

テンプレートのリンクがより見やすくなるように、ホスト、ホストプロトタイプ、テンプレート設定フォーム、およびホスト/テンプレート一括更新フォームの最初のタブに配置されました。

* Host name

Visible name

Templates	Name	Action
	Linux OS agent	Unlink Unlink and clear
	App Zabbix Server	Unlink Unlink and clear

その結果、すべてのフォームから、テンプレートリンクのための独立したタブが削除されました。

これに関連して、ホストプロトタイプの設定において、ホストグループ/ホストグループプロトタイプの選択フィールドも、別のタブから最初のタブに移動されました。

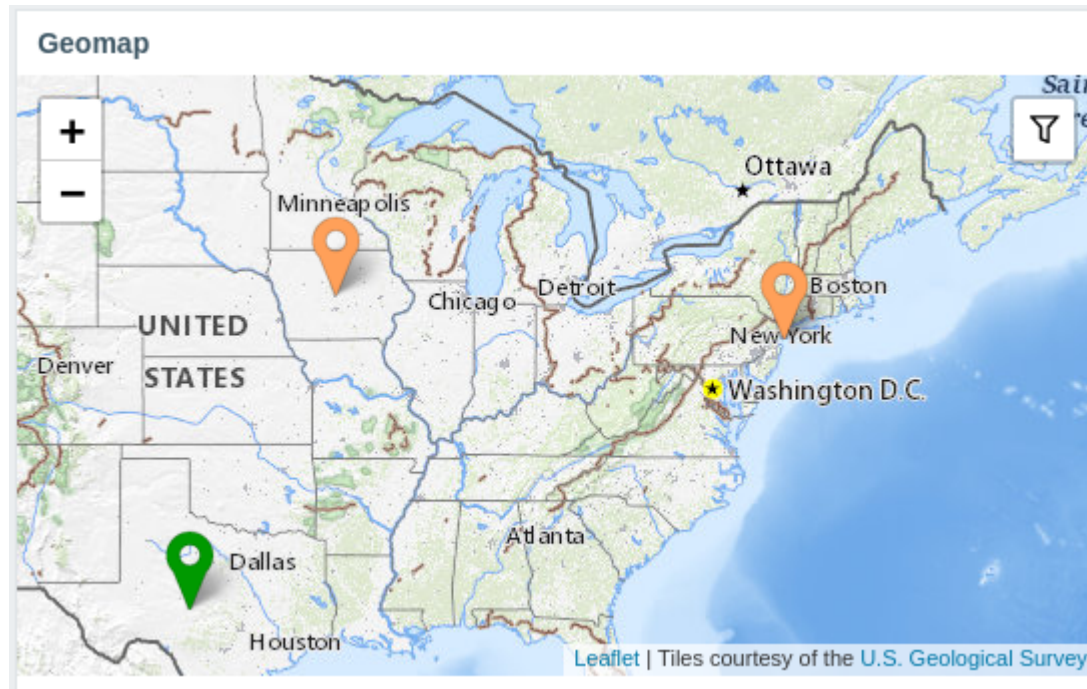
ランタイムコマンド発行

Zabbix サーバーとプロキシのランタイムコマンドは、Unix シグナルではなく、ソケット経由で発行されるようになりました。この変更により、ランタイムコントロールオプションの操作性が向上しました：

- コマンド実行の結果がコンソールに表示されるようになりました。
- ノード番号の代わりに HA ノード名など、より長い入力パラメータを送信することができます。

Web インターフェイス 地理マップウィジェット

ダッシュボードに新しい地理マップウィジェットが導入され、ホストを地理的な地図上に表示できるようになりました。詳しくは地理マップ [ダッシュボードウィジェット](#) と [地理マップ](#) を参照してください。



最新データにサブフィルターが追加されました

最新データセクションにサブフィルターが追加されました。このサブフィルターは、関連する項目のグループにワンクリックでアクセスするのに便利です。ワンクリックでアクセスできます。

サブフィルターには、クリックできるリンクが表示され、以下のことが可能です。共通のエンティティ（ホスト、タグ名、タグの値）に基づいてアイテムをフィルタリングすることができます。エンティティがクリックされるとすぐに、アイテムがフィルタリングされます。

詳細は、[最新データ](#) セクションを参照してください。

カスタムグラフのユーザビリティ向上

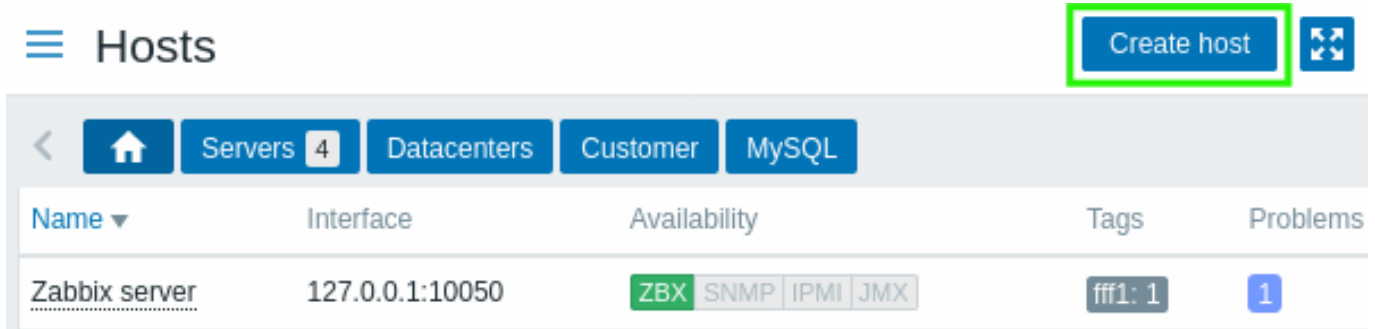
監視データ → ホスト → グラフのグラフページで、いくつかの操作性が改善されました。

- グラフページに 20 個のグラフの制限がなくなりました。
- 共通のタグやタグの値に基づいて、関連するグラフのグループをすばやく選択できるサブフィルタが追加されました。
- ホスト用のシンプルなグラフをカスタムグラフと一緒に表示することができます。

詳細は、[グラフ](#) ページを参照してください。

監視データからのホスト定義作成

監視データ → [ホスト](#) から新しいホスト定義を作成することもできるようになりました。



ホスト作成ボタンは、管理者と特権管理者のユーザーが使用できます。

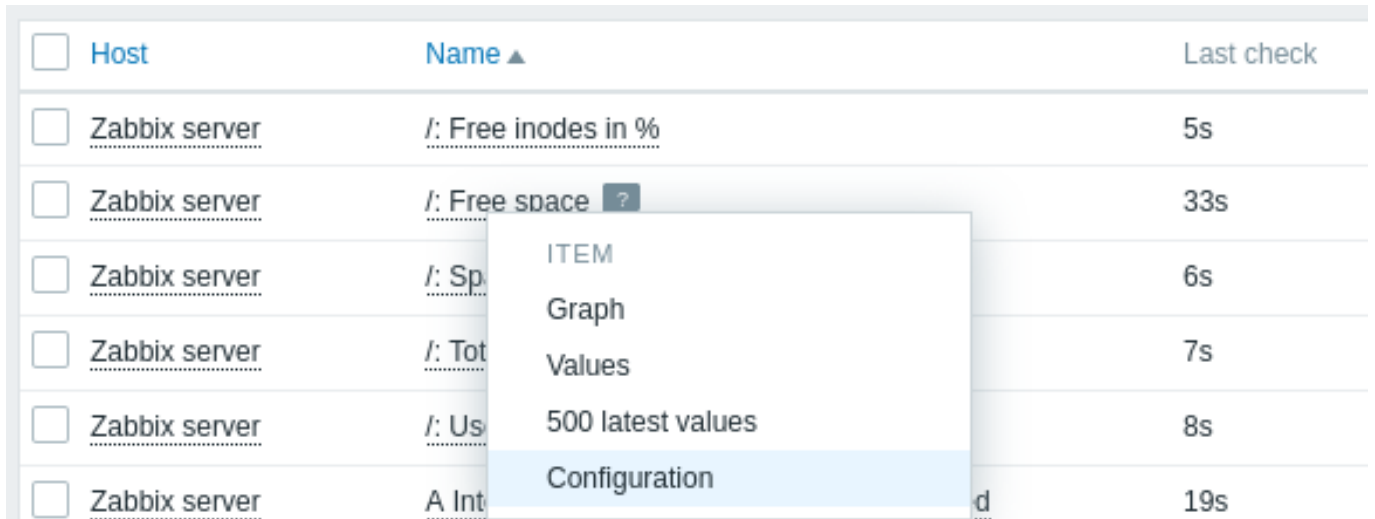
ホスト定義編集をポップアップで表示

ホストメニューやホスト設定への直接リンクがある、設定 → ホスト、監視データ → ホスト、および任意のページで、ホスト定義の作成と編集のためのフォームがモーダル（ポップアップ）ウィンドウで開かれます。

ホスト編集ページへの直接リンクはまだ機能しており、全画面でホスト編集ページを開きます。

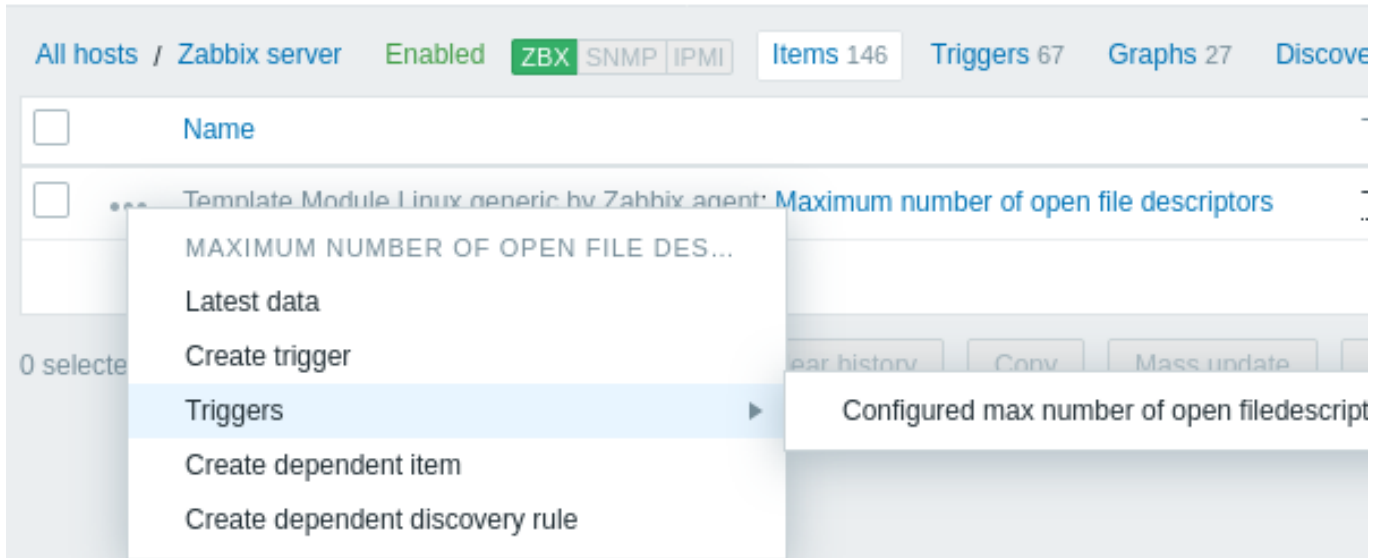
アイテム設定と最新データ間のナビゲーションの改善

最新データに、アイテムのための新しいコンテキストメニューが導入され、アイテムの構成と利用可能なグラフにアクセスできるようになりました。



逆に、[ホストのアイテム一覧](#) の設定メニューに新しいコンテキストメニューが導入され、最新のデータにアクセスできるようになりました。他にも便利なオプションがあります。

≡ Items



このメニューは、以前のバージョンのウィザードオプションを置き換えたものです。同様のメニューは、[テンプレートのアイテム](#)にも導入されています。[アイテムのプロトタイプ](#)にも同様のメニューが導入されました。

エスカレーションのキャンセル時の通知について

[アクションの実行内容](#)の設定時に、該当するオプションのチェックボックスをオフにすると、キャンセルされたエスカレーションに関する通知をキャンセルすることができるようになりました。

監視データ → 最新データ更新

最新データセクションにいくつかの改良が加えられました:

- アイテムの最終実行時間の代わりに、最終チェックからの時間 (例: 1m20s) が表示されるようになりました。
- アイテムの最後の値にカーソルを合わせると、単位や値のマッピングが適用されていない生の値が表示されます。
- ホストがメンテナンス中の場合、ホスト名の横にオレンジ色のレンチのアイコンが表示されます。

監視データ → 概要を削除

監視データメニューの概要セクションが完全に削除されました。データの概要とトリガーの概要のダッシュボード[ウィジェット](#)を使用して、同じ機能にアクセスすることができます。

その他

- 次のフィールドの最大フィールドサイズが増加しました:
 - [アイテムの前処理](#)のパラメーター
 - [メディアタイプ](#) message
- Zabbix ウェブインターフェースのデフォルト言語が、イギリス英語からアメリカ英語に変更されました。イギリス英語のサポートは終了しました。
- メインメニューの共有リンクはインテグレーションリンクに変更され、Zabbix ウェブサイトの[インテグレーション](#)ページが表示されます。
- Zabbix Web インタフェースを Zabbix ウェブサイトで利用可能な言語で開いている場合、インテグレーションリンクをクリックすると該当する言語でインテグレーションページが表示されます。それ以外の言語では英語で表示されます。
- [アクションの設定](#)で条件計算のために使用するカスタム式は、最大 1024 文字まで使用できるようになりました (従来は 255 文字)。
- 監視データ-> ホストセクションは、現在発生中の障害がない場合でも、ホストの障害画面へのリンクを表示するようになりました。

重大な変更点 監査ログ

[監査ログ機能](#)の変更を実装するために、既存のデータベース構造を再構築する必要がありました。アップグレードによって、auditlog と auditlog_detailsDB テーブルは、異なるフォーマットの新しいテーブル auditlog に置き換えられます。既存の監査ログのレコードは削除されます。

サポートされる DB バージョンの確認

Zabbix [サーバー](#)と[プロキシ](#)は、起動前にデータベースのバージョンを確認し、サポート範囲外のバージョンの場合は起動しないようになりました。詳細については、[データベース](#)を参照してください。

PCRE2 サポート

Zabbix は PCRE と PCRE2 の両方をサポートするようになりました。RHEL 7 以降、SLES(全バージョン)、Debian 9 以降、Ubuntu 16.04 以降の Zabbix パッケージは、PCRE の代わりに PCRE2 でコンパイルするように更新されました。ソースからコンパイルする場合、ユーザーは "--with-libpcre" または "--with-libpcre2" フラグを選択することができます。既存のインストールをアップグレードする場合、PCRE を PCRE2 に変更すると、いくつかの正規表現の動作が変わることがあります。詳細については、[既知の問題点](#) を参照してください。

設定ファイルの分割

Zabbix エージェント 2 プラグインに、個別の**設定ファイル**が追加されました。デフォルトでは、これらのファイルは ./zabbix_agent2.d/plugins.d/ ディレクトリに配置されます。このパスは、エージェント 2 設定ファイルの Include パラメータで指定し、**zabbix_agent2.conf** または **zabbix_agent2.win.conf** ファイルの場所を指定します。

ベースライン監視

ベースライン監視のオプションが拡張され、**baselinedev** と **baselinewma** の 2 つの関数が追加されました。

- **baselinedev** - 直近のデータ期間と、それ以前のシーズンの同じデータ期間を比較し、偏差の値を返します。
- **baselinewma** - 加重移動平均アルゴリズムを使用して、複数の等しい期間（「シーズン」）の同じタイムフレームのデータを平均化することによりベースラインを計算します。

これらの機能において、「シーズン」という用語は、時間、日、週、月、年単位で設定可能な時間枠を指します。シーズンの長さとして分析するシーズン数は、関数のパラメータで設定します。

詳しくは、[ヒストリ関数](#)を参照してください。

6 Zabbix 6.0.1 の新機能

Zabbix エージェント 2 のアイテム

- **アイテム net.dns** および **net.dns.record** のネイティブサポートが追加されました。これらのアイテムは Zabbix エージェント 2 で使用され、同時チェック処理に対応するようになりました。Windows では、ip パラメータにカスタム DNS IP アドレスが許可され、timeout と count パラメータは無視されなくなりました。
- S.M.A.R.T. プラグインでサポートされている smart.disk.discovery と smart.attribute.discovery **アイテム** は更新されて {#DISKTYPE} マクロ値を小文字で戻すようになりました。

無効化された **systemd** ユニットの検知 systemd.unit.discovery アイテムキーを使用して、無効化された systemd ユニットの検出ができるようになりました。Zabbix エージェント 2 が対応しています。無効化された systemd ユニットのプロトタイプから作成されたアイテムやトリガーを使用する場合は、{#UNIT.ACTIVESTATE} と {#UNIT.UNITFILESTATE} マクロの禁止する LLD フィルタを調整（または削除）する必要がありますことに注意してください。

詳しくは[systemd サービスのディスカバリ](#)を参照してください。

暗号化された接続での **SNI** サポート Zabbix エージェントと Zabbix サーバーまたはプロキシ間の暗号化された TCP 接続が SNI に対応しました。

LDAP シンプルチェックにおけるソース **IP** アドレスのサポート LDAP **シンプルチェック** に SourceIP のサポートが追加されました。注意 OpenLDAP では、バージョン 2.6.1 以降が必要です。

7 Zabbix 6.0.2 の新機能

Zabbix エージェント 2 のアクティブチェック設定

Zabbix エージェント 2 に新しいオプションの**設定パラメータ** ForceActiveChecksOnStart が追加されました。このパラメータを ForceActiveChecksOnStart=1 に設定すると、アクティブチェックのアイテムデータは Zabbix エージェント再起動時に直ちに収集されます。ただし、定期設定された**監視間隔のカスタマイズ**を設定したアイテムは例外です。そうでなければ、エージェント再起動後の最初のデータ収集は、リソース使用量のスパイクを防ぐために、アイテムの更新間隔より短いランダムな時間に起こります。

また、Plugins. <PluginName>.System.ForceActiveChecksOnStart を使用して、特定のプラグインにのみこのオプションを設定することができます（例:Plugins.Uptime.System.ForceActiveChecksOnStart=1）設定された場合、プラグインレベルのパラメータは、グローバルな設定を上書きします。

JMX モニタリング テンプレート Generic Java JMX は、メモリプールとガベージコレクタのローレベルディスカバリのための検知ルールを含むようになりました。

キーボードによるナビゲーション フロントエンドのインフォメーションアイコンにキーボードコントロールが実装されました。これによりキーボードで情報アイコンにフォーカスし、ヒントを開くことができるようになりました。

8 Zabbix 6.0.3 の新機能

PostgreSQL のメトリクス

Zabbix エージェント 2 の PostgreSQL プラグインに新しい **アイテム** が追加されました。メトリック **pgsql.queries** は、クエリの実行時間を監視するために使用されます。

テンプレート

新しいテンプレート OpenWeatherMap by HTTP が利用可能になりました。セットアップ方法については、**HTTP テンプレートの操作** を参照してください。

既存のテンプレートにおいて、以下の変更を行いました。

- Windows services by Zabbix agent、Windows services by Zabbix agent active、Windows by Zabbix agent、Windows by Zabbix agent active の `{SERVICE.NAME.NOT_MATCHES}` マクロ値が更新され、拡張サービスリストのフィルタリングが可能になりました。
- テンプレート PostgreSQL by Zabbix agent 2 は、スロークエリ数をチェックし、その数が閾値を超えると障害を生成するようになりました。

これらのテンプレートは以下の手順で入手できます。

- 新規インストールの場合は **設定** → **テンプレート** で利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを Zabbix **Git リポジトリ** からダウンロードすることができます。または、ダウンロードした最新バージョンの `zabbix/templates` ディレクトリにあります。その後、**設定** → **テンプレート** で、手動で Zabbix にインポートすることができます。

9 Zabbix 6.0.4 の新機能

上位ホストウィジェットのテキストデータ **上位ホスト** ウィジェットで、任意の情報 (Character, Text, Log) を含む項目を選択可能になりました。例えば、このウィジェットを使って、各ホストで動作している Zabbix エージェントのバージョンを表示できるようになりました。

OpenSSL 3.0 対応 OpenSSL 3.0.x をサポートしました。この変更は、フロントエンドの暗号化 (独自の `openssl-php` パッケージを使用) および監視対象の Java ゲートウェイ JMX 暗号化接続 (独自の Java 暗号化ライブラリを使用) には影響しないことに注意してください。

テンプレート 新しいテンプレートが利用可能です:

- TrueNAS SNMP - SNMP を使用して、TrueNAS ストレージ OS を監視します。
- Proxmox VE by HTTP - **HTTP テンプレート** のセットアップ手順を参照してください。

以下のテンプレートに、仮想ファイルシステム監視のために、ファイルシステム使用率の警告または致命的な障害に対応する閾値を定義できる新しいマクロが追加されました。HOST-RESOURCES-MIB storage SNMP Linux by Prom Linux filesystems SNMP Linux filesystems by Zabbix agent active Linux filesystems by Zabbix agent Mellanox SNMP PFSense SNMP Windows filesystems by Zabbix agent active Windows filesystems by Zabbix agent ファイルシステム使用率トリガーが更新され、これらのマクロを使用できるようになりました。

これらのテンプレートは以下の方法で取得できます:

- 新規インストールの場合は **設定** → **テンプレート** で利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを Zabbix **Git リポジトリ** からダウンロードすることができます。または、ダウンロードした最新バージョンの `zabbix/templates` ディレクトリにあります。その後、**設定** → **テンプレート** で、手動で Zabbix にインポートすることができます。

GLPi インテグレーション 新しい **GLPi integration** では、**webhook** メディアタイプを使用して、Zabbix の障害通知に基づいて GLPi アシスタンスセクションに問題を作成することが可能です。

S.M.A.R.T. モニタリング Zabbix エージェント 2 でサポートされるスマートプラグインは、より効率的にディスクを検出し、検出されたすべてのディスクではなく、特定のディスクに関する情報を返すことができるようになりました。Zabbix エージェント 2 の **アイテム smart.disk.discovery** と **smart.disk.get** が更新されました。テンプレート SMART by Zabbix agent 2 と SMART by Zabbix agent 2 (active) も新機能を組み込むために修正されました。

10 Zabbix 6.0.5 の新機能

テンプレート 新しいテンプレートが利用できます。

- CockroachDB by HTTP
- Envoy Proxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP

HTTP テンプレート のセットアップ方法を参照してください。

これらのテンプレートは以下の方法で取得できます

- 新規インストールの場合は 設定 → テンプレート で利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを Zabbix [Git リポジトリ](#) からダウンロードすることができます。または、ダウンロードした最新バージョンの zabbix/templates ディレクトリにあります。その後、設定 → テンプレート で、手動で Zabbix にインポートすることができます。

Prometheus の保存前処理における **NaN** 値の取り扱いについて NaN 値を扱う (スキップする) ための新しい動作があります。つまり、データセットが有効な数値と NaN で構成されている場合、NaN 値はスキップされ、以下ようになります。

- 'avg', 'max', 'min', 'sum' は有効な値から計算された結果を戻します。
- 'count' は有効な値の数を戻します。

データセット内の全ての値が NaN の場合、'avg', 'max', 'min', 'sum' は「データなし」(少なくとも 1 つの値が必要) のエラーを返し、'count' は 0 を戻します。

以前は、NaN がデータセットの最初の値であった場合、次のように振舞いました。
 - avg', 'max', 'min', 'sum' は "Value "NaN" of type "string" is not suitable for value type "Numeric (float)" error" というエラーを戻していました。 - count は値の数 (NaN を含む) を戻していました。

- また、以前は NaN がデータセットの最初の値でない場合、
- 'avg', 'sum' は "Value "NaN" of type "string" is not suitable for value type "Numeric (float)" というエラーを戻していました。
- 'max' は最初の NaN が発生するまでの値の最大値を戻していました。
- 'min' は最初の NaN が発生するまでの値の最小値を戻していました。
- 'count' は NaN を含む値の個数を戻していました。

ホストの最新データリンクに数値が表示される **監視データ -> ホスト** にあるホストの最新データリンクは、最新データのアイテム数を表示するようになりました。

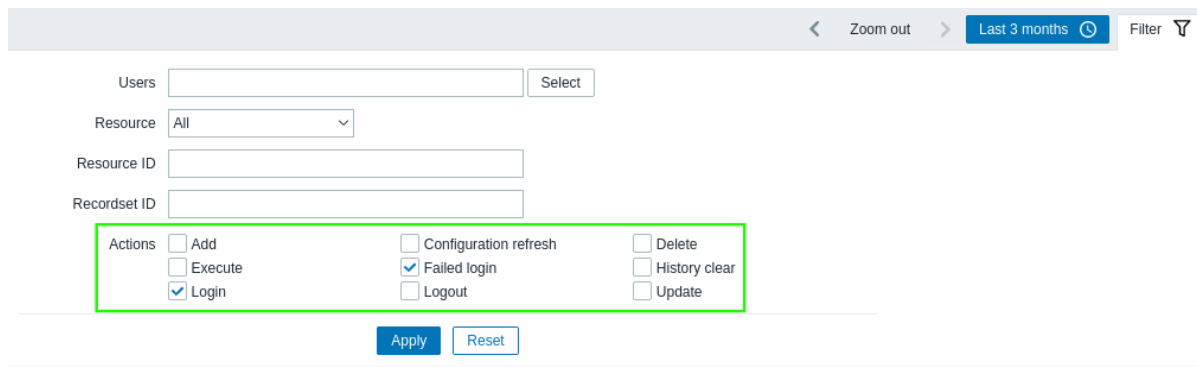
Web インターフェースの言語 Web インターフェースでドイツ語とベトナム語が使用できるようになりました。

最新データサブフィルターの拡張可能なリスト **最新データ** のサブフィルターで、拡張可能なリストが導入されました。

- 各エンティティグループ (例: タグ、ホスト) について、最大 10 行のエンティティが表示されるようになりました。
 それ以上のエンティティがある場合、このリストは最大 1000 エントリまで拡張できます。
 (**Web インターフェースの定義** にある SUBFILTER_VALUES_PER_GROUP の値) を、
 末尾に表示される 3 つの点のアイコンをクリックすることで設定できます。以前は、拡張不可能な最大 100 項目が制限されていました。
- タグの値のリストで、タグ名が最大 10 行まで表示されるようになりました。
 より多くのタグ名と値がある場合、このリストは、下部に表示される 3 つの点のアイコンをクリックすることで
 最大 200 件まで拡張することができます。従来は、拡張できない最大 20 行のタグ名が限界でした。

各タグ名に対して、最大 10 行の値が表示されます (**Web インターフェースの定義** の SUBFILTER_VALUES_PER_GROUP の値で最大 1000 件まで拡張可能)

監査ログフィルター レポート -> **監査** にある監査ログフィルターで複数のアクションを選択できるようになりました。



これは、関連するすべてのアクション (たとえば、Web インターフェースへのログインの成功や失敗) を監査リストで確認するのに便利です。

11 Zabbix 6.0.6 の新機能

PHP 8 のサポート PHP 8.0 および 8.1 がサポートされるようになりました。

MariaDB 10.7 のサポート 現在、MariaDB でサポートされている最大バージョンは 10.7.X です。

ローダブル **MongoDB** プラグイン MongoDB [プラグイン](#) は、Zabbix agent 2 の一部ではなくなり、代わりにロード可能なプラグインとして提供されるようになりました。サポートされる MongoDB バージョンが 2.6-5.3 に拡張されました。

プラグインの機能とサポートされる **アイテム** のセットは変更されていません。

テンプレート 新しいテンプレート

新しいテンプレートが利用可能になりました:

- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP

HTTP テンプレート のセットアップ方法を参照してください。

これらのテンプレートは以下の方法で取得できます:

- 新規インストールの場合は **設定** → **テンプレート** で利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを Zabbix [Git リポジトリ](#) からダウンロードすることができます。または、ダウンロードした最新バージョンの zabbix/templates ディレクトリにあります。その後、**設定** → **テンプレート** で、手動で Zabbix にインポートすることができます。

ExpressMS messenger webhook の **API** が変更されました。 ExpressMS messenger webhook の API バージョンが v4 に変更されました。

12 Zabbix 6.0.7 の新機能

MariaDB 10.8 をサポート MariaDB の最大サポートバージョンは、現在 10.8.X です。

TimescaleDB 2.6 のサポート 現在、TimescaleDB でサポートされている最大バージョンは 2.6 です。

テンプレート 新しいテンプレート

新しいテンプレート HPE Synergy by HTTP が利用可能です。

HTTP テンプレート の設定方法を参照してください。

このテンプレートは以下の方法で取得することができます。

- 新規インストールの場合、設定 → テンプレートで利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートは Zabbix [Git リポジトリ](#) からダウンロードするか、ダウンロードした最新バージョンの Zabbix の zabbix/templates ディレクトリから入手することができます。その後、設定 → テンプレートで、手動で Zabbix にインポートすることができます。

テンプレートの更新

PostgreSQL Agent 2 テンプレートが更新されました。

チェックサム障害を検知するトリガーが [PostgreSQL Agent 2 テンプレート](#) の Dbstat アイテムに追加されました。

このテンプレートは以下の方法で取得できます：

- 新規インストールの場合は設定 → テンプレートで利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを [Zabbix Git リポジトリ](#) からダウンロードすることができます。または、ダウンロードした最新バージョンの templates ディレクトリにあります。その後、設定 → テンプレートで、手動で Zabbix にインポートすることができます。

13 Zabbix 6.0.8 の新機能

月の省略文字が大文字に “月” がフロントエンドでは大文字の“M” で省略されるようになりました。以前は小文字の“m” で省略されていたので、“分”の省略と重複していました。

TimescaleDB 2.7 サポート TimescaleDB でサポートされる最大バージョンは 2.7 になりました。

テンプレート [新しいテンプレート](#) OPNsense by SNMP が利用可能になりました。

このテンプレートを使用するには：

- 新規インストールの場合は設定 → テンプレート
- 以前のバージョンからアップグレードする場合は、[Zabbix Git リポジトリ](#) から新しいテンプレートをダウンロードするか、ダウンロードした最新バージョンの zabbix/templates ディレクトリで見つけることができます。次に設定 → テンプレートで、手動で Zabbix にインポートすることができます。

RHEL パッケージの名前が変更されました RHEL パッケージは、名前に“リリース”という単語が追加されました。

ネーミング	パッケージ名
旧	zabbix-agent-6.0.7-1.e19.x86_64.rpm
新	zabbix-agent-6.0.8-release1.e19.x86_64.rpm

この変更に伴う機能の変更はありません。

これは、6.0.9 で開始される予定のマイナーバージョン (つまり 6.0.x) リリース候補のパッケージを提供するための準備として必要です。命名の変更により、システムで安定したリポジトリと不安定なリポジトリの両方が有効になっている場合、リポジトリの更新が正しい順序で受信されるようになります。この命名の変更は、RHEL パッケージのみを対象としています。

14 Zabbix 6.0.9 の新機能

式マクロ {ITEM.KEY<1-9>} マクロが**式マクロ**でサポートされるようになりました。

パッケージ SQL スクリプトは、Zabbix パッケージの /usr/share/doc ディレクトリから /usr/share に移動されました。

15 Zabbix 6.0.10 の新機能

フィルター設定の記憶

いくつかの監視データページ (障害、ホスト、最新データ) で、現在のフィルター設定がユーザープロファイルに記憶されるようになりました。ユーザーがページを再び開いたとき、フィルター設定は維持されます。

また、変更した（保存していない）お気に入りフィルターのマークは、フィルター名の横にイタリック体ではなく、緑色のドットが表示されるようになりました。

TimescaleDB 2.8 のサポート 現在、サポートしている TimescaleDB の最大バージョンは 2.8 です。

PostgreSQL 15 対応 PostgreSQL 15 に対応しました。ただし、TimescaleDB はまだ PostgreSQL 15 をサポートしていません。

Zabbix エージェント 2 をオフラインでビルド可能 Zabbix エージェント 2 をオフラインでビルドすることができます。ソースの tarball 内に `src/go/vendor` ディレクトリが含まれるようになりました。golang は依存モジュールを自動的にダウンロードする必要がありません。go mod tidy か go get コマンドを使用することで最新のモジュールにアップデートすることができます。

PostgreSQL プラグインがロード可能に PostgreSQL **プラグイン** が Zabbix エージェント 2 でロード可能になりました。(以前は組み込み)

参照: [PostgreSQL のロード可能なプラグイン](#) リポジトリ

Web インターフェース その他

- ヒストリやトレンドのテーブルに圧縮されたチャンクが含まれている場合、TimescaleDB の不適切な housekeeping の設定に関する警告が表示されるようになりましたが、アイテムの履歴の保存期間の上書きまたはアイテムのトレンドの保存期間の上書きのオプションが無効になっています。詳しくは、[TimescaleDB のセットアップ](#)を参照してください。

16 Zabbix 6.0.11 の新機能

inode がゼロのファイルシステムを報告する `vfs.fs.get` エージェントアイテムは、inode カウントがゼロに等しいファイルシステムを報告できるようになりました。これは、動的 inode を持つファイルシステム (例: btrfs) の場合です。

さらに、「`vfs.fs.inode`」アイテムは、モードが「`pfree`」または「`pusd`」に設定されている場合でもサポートされなくなることはありません。代わりに、そのようなファイルシステムの `pfree/pused` 値は、それぞれ「100」と「0」として報告されます。

最適化された **API** クエリ `hosts` テーブルと `items` テーブルで名前を検索するときに作成される API のデータベースクエリが最適化され、より効率的に処理されるようになりました。

17 Zabbix 6.0.12 の新機能

history syncers のパフォーマンス改善 `history syncer` のパフォーマンスは、新しい read-write ロックを導入することで改善されました。これにより、構成キャッシュにアクセスしながら共有読み取りロックを使用して、`history syncer`、`trapper`、`proxy poller` 間のロックが削減されます。新しいロックは、構成キャッシュリロードを実行する `configuration syncer` によってのみロックされた書き込みができます。

18 Zabbix 6.0.13 の新機能

重大な変更 読み込み可能なプラグインのバージョン管理

Zabbix エージェント 2 で **読み込み可能なプラグイン** のバージョン管理システムは、Zabbix 本体と同様のものを採用するようになりました。これに伴うバージョン変更は次の通りです:

- MongoDB 1.2.0 -> MongoDB 6.0.13
- PostgreSQL 1.2.1 -> PostgreSQL 6.0.13

これらのプラグインは、Zabbix 6.0 のすべてのマイナーバージョンでサポートされています。各プラグインのソースコードのレポジトリは、専用の `release/6.0` ブランチをもつようになりました。(変更前は、`master` ブランチのみ)

MariaDB 10.10 のサポート 現在サポートされている MariaDB の最大バージョンは、10.10.X です。

設定のインポート 以前は、インポート可能な設定情報 (ホストグループ、アイテム、グラフなど) の UUID の不一致により、インポートの処理が失敗することがありました。例えば、同じ名前のホストグループがホストに既に存在している場合、ホストグループをインポートすることができませんでした。

新しいバージョンでは、UUID の不一致が原因でインポートが失敗することはありません。代わりに、エンティティはエンティティ ID (名前) などの一意性条件によって照合されます。エンティティがインポートされると、UUID はインポートされたエンティティの UUID に更新されます。

別の改善点として、インポートされたテンプレートまたはホストを介してテンプレートのリンクを削除 (テンプレートのリンクの Delete missing オプション) すると、これらのエンティティがインポートファイルから欠落し、かつ特定のエンティティの Delete missing オプションがマークされていない限り、リンクされていないテンプレートの継承されたエンティティは削除されなくなります (テンプレートはリンク解除され、リンク解除およびクリアされません)。

この変更により、テンプレートのリンクの Delete missing オプションを選択した場合の警告メッセージは表示されなくなります。

Zabbix エージェント 2 を使用した **Oracle** データベース内の個別のテーブルスペースのクエリ Oracle プラグインでサポートされている以下の **Zabbix エージェント 2 のアイテム** に、オプションパラメーターが追加されました。

- oracle.diskgroups.stats[<existingParameters>,<diskgroup>]
- oracle.archive.info[<existingParameters>,<destination>]
- oracle.cdb.info[<existingParameters>,<database>]
- oracle.pdb.info[<existingParameters>,<database>]
- oracle.ts.stats[<existingParameters>,<tablespace>,<type>]

これらのパラメータを使用すると、すべてのデータではなくデータの個別のインスタンスをクエリできるため、パフォーマンスが向上します。

`docker.container_info[]` で追加情報取得 `docker.container_info[]` **Zabbix エージェント 2 のアイテム** は、Docker コンテナに関する部分的 (短い) または完全な低レベルの情報を取得するオプションをサポートするようになりました。

プロファイリングのランタイムコマンド プロファイリング用のランタイムコマンドが、Zabbix サーバーと Zabbix プロキシに追加されました。

- prof_enable - プロファイリングを有効化
- prof_disable - プロファイリングを無効化

プロファイリングは、サーバーとプロキシのプロセスごとに有効にできます。プロファイリングを有効にすると、関数名ごとにすべての rwlocks/mutex の詳細が提供されます。

参照:

- [Zabbix サーバーのランタイムコマンド](#)
- [Zabbix プロキシのランタイムコマンド](#)

JavaScript 用の **HMAC** 関数 JavaScript エンジンに新しい関数が追加され、HMAC ハッシュを返すことができるようになりました:

- hmac('<hash type>',key,string)

これは、リクエストの署名にハッシュベースのメッセージ認証コード (HMAC) が必要な場合に役立ちます。MD5 および SHA256 ハッシュタイプがサポートされています。例:

- hmac('md5',key,string)
- hmac('sha256',key,string)

テンプレート 新しいテンプレートが使用できます:

- AWS EC2 by HTTP
- AWS by HTTP
- AWS RDS instance by HTTP
- AWS S3 bucket by HTTP
- Azure by HTTP
- Control-M server by HTTP
- Control-M enterprise manager by HTTP
- Veeam Backup Enterprise Manager by HTTP
- Veeam Backup and Replication by HTTP

HTTP テンプレートのセットアップ方法を参照してください。

テンプレート **Oracle by Zabbix agent 2** は、**Zabbix エージェント 2 のアイテム** への変更に従って更新されました (複数の静的アイテムが削除され、複数のアイテムプロトタイプが追加されました)。

更新の詳細については、[テンプレートの変更](#)を参照してください。

これらのテンプレートは以下の方法で取得できます：

- 新規インストールの場合は [設定](#) → テンプレートで利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを [Zabbix Git リポジトリ](#) からダウンロードすることができます。または、ダウンロードした最新バージョンの `zabbix/templates` ディレクトリにあります。その後、[設定](#) → テンプレートで、手動で Zabbix にインポートすることができます。

TimescaleDB 2.9 のサポート 現在サポートされている TimescaleDB の最大バージョンは 2.9 です。

Webhook 統合 新しいメディアタイプ LINE が追加され、[webhook](#) の機能を使って Zabbix イベントの通知を LINE メッセンジャーに送信できるようになりました。

Web インターフェースの言語 Web インターフェースでカタロニア語とルーマニア語が使用できるようになりました。

Windows 用 **Go** 言語ライブラリの更新 Zabbix エージェント 2 で Windows を監視するために、MongoDB や PostgreSQL プラグインとともに使用されている Go 言語ライブラリは、[github.com/Microsoft/go-winio](#)、バージョン 0.6.0 です。(変更前は [github.com/natefinch/npiper](#)) **Go 言語ライブラリ**、**MongoDB プラグイン依存関係**、**PostgreSQL プラグイン依存関係** も併せてご確認ください。

Zabbix エージェント 2 のオープンファイルディスクリプタの制限の緩和 Zabbix agent 2 パッケージでやり取りされる `systemd` サービスファイルのオープンファイルディスクリプタの制限は、8196 になりました。変更前は、システムのデフォルトの制限として、1024 が使用されていました。新しい制限は、Zabbix エージェント 2 のデフォルト設定に十分なものとなっています。追加のプラグインや拡張機能を使用しているなど、Zabbix エージェント 2 が標準の設定でない場合は、この制限を手動で拡張する必要があります。その場合は、`systemd` ユニットファイルの `LimitNOFILE` パラメーターを調整してください。

19 Zabbix 6.0.14 の新機能

ローダブルプラグイン 暗号化された MongoDB プラグインの接続

MongoDB プラグインは、名前付きセッションを使用して MongoDB に接続するときに TLS 暗号化をサポートするようになりました。

Zabbix 6.0.14 以降、更新されたプラグイン (MongoDB plugin 1.2.1) が Zabbix 公式パッケージに含まれています。この MongoDB 用のプラグインはローダブルプラグインであり、パッケージまたはソースから個別にインストールできることに注意してください。このプラグインは、Zabbix 6.0 のすべてのマイナーバージョンで動作します。詳細については、[MongoDB プラグイン](#) を参照してください。

PHP のサポート PHP のサポートバージョンとして 8.2 までをサポート。

プリプロセッシングにおける **JavaScript** オブジェクトの制限 プリプロセッシングでは、[JavaScript オブジェクト](#) に以下の制限があります。

- `log()` メソッドでログ化できるメッセージの総サイズは、スクリプト実行一回あたり 8 MB までです。
- 複数の `HttpRequest` オブジェクトの初期化は、スクリプト実行一回あたり 10 個までです。
- `addHeader()` メソッドでひとつの `HttpRequest` オブジェクトに追加できるヘッダーの長さは、128 Kbytes (特殊文字とヘッダー名を含む) までです。

20 Zabbix 6.0.15 の新機能

MariaDB 10.11 サポート MariaDB でサポートされているバージョンの最新版は 10.11.X です。

TimescaleDB 2.10 サポート TimescaleDB でサポートされているバージョンの最新版は 2.10 です。

Oracle プラグインの接続オプション Zabbix エージェント 2 でサポートされている Oracle プラグインで、ログインオプションとして `as sysdba`、`as sysoper`、`as sysasm` を指定できるようになりました。このオプションは、ユーザーアイテムキーのパラメーター、または `user as sysdba` フォーマットのプラグイン設定パラメーター `Plugins.Oracle.Sessions.<SessionName>.User` に追加することができます (ログインオプションは大文字と小文字を区別せず、末尾にスペースを含めることはできません)。

RS256 を使用したデジタル署名 JavaScript 関数 `sign(hash,key,data)` が新たに実装され、RS256 の暗号化アルゴリズムを使用したデジタル署名が可能になりました。

詳細については、[Additional JavaScript objects](#) を参照してください。

21 Zabbix 6.0.16 の新機能

Oracle 構成の同期最適化

Oracle を使用した Zabbix のインストールで、データベースパッチを適用することで、アイテムおよびアイテムの前処理データベースフィールドタイプを `nclob` から `nvarchar2` に手動で変更できるようになりました。

パッチを適用すると、アイテム数とアイテムの前処理ステップが多い環境での構成の同期速度が向上する可能性があります。ただし、一部のアイテムパラメーターでは、最大フィールドサイズの制限が 65535 バイトから 4000 バイトに減少します。詳細については、[既知の問題](#) を参照してください。

22 Zabbix 6.0.17 の新機能

Webhook 統合 Zabbix 通知を [Event-Driven Ansible](#) にプッシュするための新しい **webhook** メディアタイプが追加されました。

Zabbix エージェント 2 プラグインでのアイテムキーとセッションパラメーターの混在 Zabbix エージェント 2 で、アイテムキーのパラメーターに新しい値を指定することで、**名前付きセッション** パラメーターを上書きできるようになりました。変更前は、名前付きセッションとアイテムキーのどちらで接続文字列値を指定するかを、ユーザーが選択する必要がありました。名前付きセッションが使用される場合、関連するアイテムキーのパラメーターは空である必要がありました。新しいバージョンでは、名前付きセッションを使用する場合、最初のパラメーター (通常は URI) のみ名前付きセッションで指定する必要がありますが、他のパラメーターは名前付きセッション、アイテムキーのどちらでも指定することができます。

Geomap アトリビューションの HTML サポートの終了 **Geomap ダッシュボードウィジェット** のアトリビューションテキストはプレーンテキストのみ対応になりました。HTML のサポートは終了しました。

Geographical maps の設定では、管理 → 一般セクションで、アトリビューション欄は、タイルプロバイダーがその他に設定されている場合のみ表示されるようになりました。

23 Zabbix 6.0.18 の新機能

アイテム `docker.container_stats`

Zabbix エージェント 2 の `docker.container_stats` アイテムも、コンテナ上の現在のプロセス/スレッド数とともに `pids_stats` プロパティを返すようになりました。

設定のエクスポートの改善 Zabbix エンティティ設定のエクスポート中に生成された YAML ファイルには、配列内のエンティティ間に空の行が含まれなくなりました。これにより、ファイルが短くなり、操作がより便利になりました。アップデート後からのエクスポートの例は、[設定のエクスポート/インポート](#) をご覧ください。

設定のインポートの **UTF-8 BOM** 対応 [設定のインポート](#) で、UTF-8 バイトオーダーマーク (BOM) を持つファイルをサポートするようになりました。

Cosmos DB 監視 テンプレート `Azure by HTTP` が、Azure Cosmos DB for MongoDB でも動作するようになりました。

テンプレートの取得方法:

- 新規インストールでは、設定 → テンプレート
- 前のバージョンから更新している場合は、Zabbix [Git リポジトリ](#) からダウンロードするか、最新の Zabbix バージョンの `zabbix/templates` ディレクトリから探してください。設定 → テンプレートから手動で Zabbix にインポートできます。

プロキシ履歴のハウスキーピング プロキシ履歴のハウスキーピングの際に、プロキシのデータベースから削除されるサイクルあたりのデータ量の制限がなくなりました。

変更前は、`housekeeper` は `HousekeepingFrequency` 4 倍時間分までしか、古い情報の削除をしませんでした。例えば、`HousekeepingFrequency` が "1" に設定されている場合、最大 4 時間分の古い情報が (古いエントリーから順番に) 削除されました。プロキシが `ProxyOfflineBuffer` で設定されたよりも古いデータを絶えず受信する場合、過剰なデータ蓄積が発生する可能性があります。

この制限が取り除かれたことで、より効果的なプロキシ履歴ハウスキーピングが行えるようになりました。

Google クラウドプラットフォーム監視 [テンプレート](#) Google Cloud Platform by HTTP (GCP by HTTP) が新たに利用できるようになりました。

設定方法については [HTTP templates](#) をご覧ください。

テンプレートの取得方法:

- 新規インストールでは、設定 → テンプレート
- 前のバージョンから更新している場合は、Zabbix [Git リポジトリ](#) からダウンロードするか、最新の Zabbix バージョンの `zabbix/templates` ディレクトリから探してください。設定 → テンプレートから手動で Zabbix にインポートできます。

Zabbix エージェント 2 のデフォルト値 Zabbix エージェント 2 プラグインで、監視対象へ接続するためのデフォルト値を設定ファイルで定義できるようになりました。アイテムキーや名前付きセッションで、値が何も指定されていない場合、プラグインは対応するデフォルトパラメーターで定義された値を使用します。新しいパラメーターは `Plugins.<PluginName>.Default.<Parameter>` という構造になります。- 例えば、`Plugins.MongoDB.Default.Uri=tcp://localhost:27017` のようになります。詳細についてはこちらのセクションをご覧ください。:

- [プラグインの設定](#)
- [プラグイン設定ファイルパラメーター](#)

24 Zabbix 6.0.19 の新機能

集計関数 `count_foreach` 関数で、配列内の一致する項目に対して、項目のデータが存在しない場合、またはデータがフィルターと一致しない場合に、その項目に対して '0' を返すようになりました。変更前は、そのような項目は無視されていました (集計にデータが追加されませんでした)。

TimescaleDB 2.11 サポート TimescaleDB version 2.11 がサポートされるようになりました。

MQTT プラグインで設定可能な **TLS** および接続パラメータ Zabbix エージェント 2 の **MQTT プラグイン** に設定オプションが追加されました。追加されたオプションは、プラグイン設定ファイルで、**名前付きセッション** または **デフォルト** パラメーターとして定義することが可能です。

- 接続関連のパラメーター: ブローカー URL、トピック、ユーザー名、パスワード
- TLS 暗号化パラメータ: 最上位の CA 証明書、MQTT 証明書または証明書チェーンの場所、プライベートキー

新しいパラメーターはすべて省略可能です。

JavaScript プリプロセッシング スクリプトのヒープ制限が 64 メガバイトから 512 メガバイトに引き上げられました。

サポートされているプラットフォーム Debian 12 (Bookworm) のサポートが追加され、公式パッケージは [Zabbix Web サイト](#) からダウンロードできます。

25 Zabbix 6.0.20 の新機能

テンプレート 新しいテンプレートが利用可能になりました:

- [AWS ECS Cluster by HTTP](#) (along with its [Serverless Cluster version](#))
- [Cisco SD-WAN by HTTP](#)
- [OpenStack by HTTP](#), which includes OpenStack Nova by HTTP template for monitoring OpenStack Nova service
- [PostgreSQL by ODBC](#)

これらのテンプレートは以下の方法で取得できます:

- 新規インストールの場合、設定 → テンプレートで利用可能です。
- 以前のバージョンからアップグレードする場合、新しいテンプレートを [Zabbix Git リポジトリ](#) からダウンロードすることができます。または、ダウンロードした最新バージョンの `zabbix/templates` ディレクトリにあります。その後、設定 → テンプレートで、手動で Zabbix にインポートすることができます。

Web インターフェイス 説明文以外のテキストエリアで、スペルチェックが無効になります

Spellcheck has been disabled for the text areas in which non-descriptive text is entered, such as scripts, expressions, macro values, etc. スクリプト、式、マクロ値など、説明文でないテキストが入力されるテキストエリアでは、スペルチェックが無効になりました。

その他 SLES 12 の MySQL のデータベース TLS 接続

SUSE Linux Enterprise Server バージョン 12 にサーバー/プロキシをインストールするためのパッケージは、MariaDB Connector/C ライブラリを使用して構築されるようになり、DBTLSConnect [パラメータ](#)を使用した MySQL への接続の暗号化が可能になります。サポートされている暗号化値は「required」と「verify_full」です。

26 Zabbix 6.0.21 の新機能

MySQL 8.1 をサポート

MySQL のサポートされているバージョンの最大が 8.1.X になりました。

MariaDB 11.0 をサポート

MariaDB のサポートされているバージョンの最大が 11.0.X になりました。

ログファイルの監視

`log[]`、`logrt[]`、`log.count[]`、`logrt.count[]` 項目の正規表現ランタイムエラーが Zabbix エージェントログファイルに記録されるようになりました。詳細を参照してください。

アイテム Zabbix エージェント 2 の新しいアイテム

Zabbix エージェント 2 の MySQL プラグインに新しいアイテムが追加されました。この新しいアイテム `mysql.custom.query` は、カスタム MySQL クエリを実行するためのものです。

テンプレート 新しいテンプレートが利用可能です:

- [AWS Cost Explorer by HTTP](#)

テンプレートの入手方法:

- 新規インストールの場合は、設定 → テンプレートにあります。
- 以前のバージョンからアップグレードする場合は、Zabbix の [Git リポジトリ](#) からダウンロードしてくるか、最新バージョンの Zabbix の `zabbix/templates` ディレクトリから入手できます。そして設定 → テンプレートから、手動で Zabbix にインポートすることができます。

通知

Webhook の統合 Zabbix 通知を [Mantis Bug Tracker](#) に届けるための新しい **webhook** メディアタイプが追加されました。

インストール ARM64/AArch64 をサポート

ARM64/AArch64 インストールパッケージが、Debian、RHEL 8、9 およびその派生製品、および SLES/OpenSUSE Leap 15 で利用できるようになりました。

27 Zabbix 6.0.22 の新機能

集計関数

last_foreach 関数は、次の **集計関数** でもサポートされるようになりました: `kurtosis`, `mad`, `skewness`, `stddevpop`, `stddevsamp`, `sumofsquares`, `varpop`, and `varsamp`

戻り値の制限

スクリプトや他のプログラムなどの外部ソースからデータを受信する場合の戻り値の制限が 16MB に引き上げられました。以下に影響します:

- Zabbix エージェントのアイテム、`system.run[]`と`vfs.file.contents[]`
- `ユーザーパラメータ`で定義されたカスタムエージェントチェック
- SSH エージェント、外部からのチェックアイテム、web インターフェースの`scripts`アイテム
- リモートコマンド

テンプレート 新しいテンプレートが利用可能になりました:

- [Acronis Cyber Protect Cloud by HTTP](#)
- [HashiCorp Nomad by HTTP](#)
- [MantisBT by HTTP](#)

テンプレートの入手方法:

- 新規インストールの場合、設定 → テンプレートにあります。
- 以前のバージョンからアップグレードの場合、Zabbix の [Git リポジトリ](#) からダウンロードしてくるか、最新バージョンの Zabbix をダウンロードし、その `zabbix/templates` ディレクトリから入手します。そして設定 → テンプレートから、手動で Zabbix にインポートすることができます。

28 Zabbix 6.0.23 の新機能

このバージョンでの **重大な変更点** はこちらをご覧ください。

データベース サポートされているバージョン

PostgreSQL**16** と MariaDB**11.1** をサポートしました。詳細はこちら: [要件](#)。

プラグイン Zabbix エージェント 2 の PostgreSQL プラグインに新しいアイテム

Zabbix エージェント 2 の PostgreSQL プラグインに `pgsql.version` という **アイテム** が追加されました。このアイテムは PostgreSQL のバージョンを返すのに使います。

テンプレート 新しいテンプレート

新しいテンプレートが利用可能になりました:

- [FortiGate by HTTP](#)
- [FortiGate by SNMP](#)
- [Nextcloud by HTTP](#)

テンプレートの入手方法:

- 新規インストールの場合、設定 → テンプレートにあります。
- 以前のバージョンからアップグレードの場合、Zabbix の [Git リポジトリ](#) からダウンロードしてくるか、最新のバージョンの Zabbix をダウンロードし、`zabbix/templates` ディレクトリから入手します。そして設定 → テンプレートから、手動で Zabbix にインポートすることができます。

テンプレートの更新

[PostgreSQL by ODBC](#) テンプレートと [PostgreSQL by Zabbix agent 2](#) テンプレートに PostgreSQL バージョンを監視するためのアイテムとトリガーが含まれるようになりました。

[Cisco Meraki organization by HTTP](#) テンプレートにアイテム、アイテムプロトタイプ、LLD ルール、および認証やライセンス、ネットワーク、SAML ロール、VPN ステータスに関連するマクロが追加されました。

Web インターフェイス その他

設定 → ホスト → **アイテム**にあるヒストリを削除ボタンはその機能をより正確に表すために、アイテムの**構成画面**にあるものと同じヒストリとトレンドを削除という名前に変更されました。

トリガーアクション設定画面のトリガー名という条件タイプをその機能をよりよく表現するためにイベント名に改名しました。**トリガーの設定**でカスタムイベント名が設定されていない限り、デフォルトではイベント名とトリガー名は一致することに注意してください。

29 Zabbix 6.0.24 の新機能

データベース TimescaleDB 2.12 サポート

TimescaleDB のバージョン 2.12 のサポートが利用可能になりました。

プラグイン Zabbix エージェント 2 プラグインの新しいアイテム

データベースサーバーのバージョンを返すアイテムが [MongoDB plugin \(mongodb.version\)](#) と [Oracle Database plugin \(oracle.version\)](#) で利用可能になりました。

アイテム コンテンツの UTF-8 への変換

HTTP エージェントアイテム、web シナリオ、web チェック、および JavaScript アイテムが、HTTP ヘッダーまたは HTTP メタタグで指定された文字セットから UTF-8 に変換されるように改善されました。

テンプレート 新しいテンプレート

新しいテンプレートが利用可能になりました:

- [HPE iLO by HTTP](#)

テンプレートの入手方法:

- 新規インストールの場合、設定 → テンプレートにあります。
- 以前のバージョンからアップグレードの場合、Zabbix の [Git リポジトリ](#) からダウンロードしてくるか、最新のバージョンの Zabbix をダウンロードし、zabbix/templates ディレクトリから入手します。そして設定 → テンプレートから、手動で Zabbix にインポートすることができます。

テンプレートの更新

OpenShift との統合が [Kubernetes cluster state by HTTP](#) テンプレートに追加されました。

30 Zabbix 6.0.25 の新機能

TimescaleDB 2.13 サポート

TimescaleDB のバージョン 2.13 のサポートが利用可能になりました。

アイテム 追加の ssh.run オプション

`ssh.run[]` アイテムが更新され、アイテムキーの一部として追加の SSH オプションを渡すことができるようになりました。これらのオプションは、バージョン 0.9.0 以降の libssh または libssh2 を使用する場合にのみサポートされます。サポートされるオプションのキーと値は、SSH ライブラリによって異なります。詳細については [SSH チェック](#) を参照してください。

プラグイン PostgreSQL プラグインのキャッシュモードパラメータ

PostgreSQL プラグイン [設定](#) にデフォルトまたはセッション名レベルでキャッシュモードを制御するための新しいパラメータが追加されました:

- `Plugins.PostgreSQL.Default.CacheMode`
- `Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode`

キャッシュモードパラメータには、`prepare` (デフォルト) または `describe` という 2 つの許容値のいずれかを指定できます。“describe” は主に、PgBouncer のような接続プーリングサーバーを実行している場合など、環境で準備されたステートメントが許可されていない場合に便利です。

31 Zabbix 6.0.26 の新機能

テンプレート 新しいテンプレート

Azure by HTTPテンプレートのセットは、Azure Cost Management by HTTP テンプレートで補足されました。

テンプレートの入手方法:

- 新規インストールの場合、設定 → テンプレートにあります。
- 以前のバージョンからアップグレードの場合、Zabbix のGit [リポジトリ](#)からダウンロードしてくるか、最新のバージョンのZabbix をダウンロードし、zabbix/templates ディレクトリから入手します。そして設定 → テンプレートから、手動で Zabbix にインポートすることができます。

更新されたテンプレート

MSSQL by ODBC テンプレートが更新され、フェールオーバークラスターインスタンス (FCI) や可用性グループ (AG) などの AlwaysOn 機能を使用できるようになりました。このテンプレートを使用して、クラスター内のホスト、スタンドアロンのホスト、およびクラスター名によるホストを監視できるようになりました。インスタンス名のマクロは使用されなくなりました。マスターが切り替えられたときに、マクロを変更する必要はありません。

- Quorum と Quorum メンバーの新しい LLD ルールとメトリックが追加されました。
- LLD ルールのタイプが「データベースモニター」から「依存項目」に変更されました。
- db.odbc.discovery キーを持つアイテムは、db.odbc.get アイテムに依存するアイテムに変更されました
- 新しいアイテムが追加されました - MSSQL DB '#{DBNAME}': Recovery model。データベースディスクカバリでデータベースリカバリモデルを返します。
- 新しいマクロ、{\$MSSQL.BACKUP_FULL.USED}、{\$MSSQL.BACKUP_DIFF.USED}、{\$MSSQL.BACKUP_LOG.USED} が追加されました - これらは、特定のデータベースのバックアップ期間トリガーを無効にするために使用できます。

Web インターフェイス PHP のサポート

PHP のサポートバージョンが 8.3 までになりました。

X-Frame-Options HTTP ヘッダー

X-Frame-Options header パラメータは Use X-Frame-Options header に改名され、チェックボックスと入力フィールド（「null」を指定する代わりにチェックボックスのマークを外すことでヘッダーを無効にできる）、追加の値のサポートで構成されています。

他のセキュリティパラメータも同じ構造に従います。詳細は、管理 → 一般設定にある [セキュリティ](#) のパラメーターを参照してください。

データベース MySQL 8.2 をサポート

MySQL のサポートされているバージョンの最大が 8.2.X になりました。

32 Zabbix 6.0.27 の新機能

データベース MySQL 8.3 をサポート

MySQL のサポートされているバージョンの最大が 8.3.X になりました。

MariaDB 11.2 をサポート

MariaDB のサポートされているバージョンの最大が 11.2.X になりました。

プラグイン MSSQL

Zabbix エージェント 2 による MSSQL の直接監視用の新しいプラグインが追加されました。

詳細は以下をご覧ください:

- [MSSQL プラグイン readme](#)
- [Zabbix エージェント 2 のアイテム](#)
- [MSSQL プラグインのパラメーター](#)
- [Zabbix エージェント 2 のインストール](#)

テンプレート 新しいテンプレート

新しいテンプレートが利用可能になりました:

- [YugabyteDB by HTTP](#), 各 YugabyteDB クラスターを監視するための YugabyteDB Cluster by HTTP テンプレートが含まれます。

テンプレートの入手方法:

- 新規インストールの場合、設定 → テンプレートにあります。
- 以前のバージョンからアップグレードの場合、Zabbix の [Git リポジトリ](#) からダウンロードしてくるか、最新のバージョンの Zabbix をダウンロードし、zabbix/templates ディレクトリから入手します。そして設定 → テンプレートから、手動で Zabbix にインポートすることができます。

プラットフォーム Windows での Zabbix エージェント 2 のサポート

重大なセキュリティ脆弱性を防ぐため、Zabbix エージェント 2 の最小 Windows バージョンが Windows 10/Windows Server 2016 へ引き上げられました。詳細については、[サポートするプラットフォームの Note](#) を参照してください。

33 Zabbix 6.0.28 の新機能

データベース TimescaleDB 2.14 をサポート

TimescaleDB で [サポートされているバージョン](#) の最大が 2.14.X になりました。

テンプレート New templates

A new template is available:

- [MSSQL by Zabbix agent 2](#)

You can get this template:

- In Configuration → Templates in new installations;
- If you are upgrading from previous versions, you can download new templates from the Zabbix [Git repository](#) or find them in the zabbix/templates directory of the latest Zabbix version you have downloaded. Then, while in Configuration → Templates, you can import them manually into Zabbix.

34 What's new in Zabbix 6.0.29

Databases MariaDB 11.3 support

The maximum [supported version](#) for MariaDB is now 11.3.X.

Templates New templates

A new template is available:

- [Oracle Cloud by HTTP](#), a master template that discovers various Oracle Cloud Infrastructure (OCI) services and resources.

You can get this template:

- In Configuration → Templates in new installations;
- If you are upgrading from previous versions, you can download new templates from the Zabbix [Git repository](#) or find them in the zabbix/templates directory of the latest Zabbix version you have downloaded. Then, while in Configuration → Templates, you can import them manually into Zabbix.

Updated templates

- [FortiGate by SNMP](#) template has been supplemented with metrics regarding VPN, high availability (HA), wireless termination points (WTPs), SD-WAN health checks, and HW sensors.

35 What's new in Zabbix 6.0.30

Frontend Frontend languages

Dutch, Georgian, and Spanish languages are now enabled in the frontend.

Plugins Ember+

A new plugin for direct monitoring of Ember+ by Zabbix agent 2 has been added.

For more information, see:

- [Ember+ plugin readme](#)
- [Agent 2 items](#)
- [Ember+ plugin parameters](#)
- [Agent 2 installation](#)

Templates New templates

The AWS ELB template set has been supplemented with the template [AWS ELB Network Load Balancer by HTTP](#).

You can get this template:

- In Configuration → Templates in new installations;
- If you are upgrading from previous versions, you can download new templates from the Zabbix [Git repository](#) or find them in the zabbix/templates directory of the latest Zabbix version you have downloaded. Then, while in Configuration → Templates, you can import them manually into Zabbix.

Updated templates

The [OS templates](#) (agent, SNMP, and Prometheus-based) have been given a mounted filesystem update. In mounted filesystem discovery, the "Space is low" and "Space is critically low" triggers no longer have the absolute threshold and burst condition.

2. 定義

概要 このセクションでは、Zabbix で一般的に使用するいくつかの用語の意味を学ぶことができます。

定義 ホスト

- 物理または仮想デバイスやアプリケーション、サービス、またはその他の監視対象パラメーターの論理的に関連するコレクション。

ホストグループ

- 論理的なホストの集まりです; ホストグループには、ホストとテンプレートが含まれます。ホストグループ内のホストとテンプレートは、互いにリンクされていません。ホストグループは、様々なユーザーグループにホストへのアクセス権を与える際に使用されます。

アイテム

- ホストから受信したい特定のデータ、データのメトリクスです。

値の保存前処理

- データベースに保存する前に、受信したメトリクス値を変換します。

トリガー

- 障害の閾値を定義し、アイテムとして取得したデータを評価するために使用される論理式です。

取得したデータが閾値を超えた場合、トリガーの状態は「OK」から「障害」に変わります。取得したデータが閾値を超えていない場合は、トリガーの状態は「OK」のまま、あるいは「OK」に戻ります。

イベント

- トリガーの状態が変化したり、ディスクバリアやエージェントの自動登録が実行されるなど、注意すべき何かが発生したことを示します。

イベントタグ

- イベントに対して事前に定義された印。イベントの相関関係や権限の細分化などに使用できます。

イベントの相関関係

- 発生した障害をそれらの解決策に柔軟にかつ正確に関連付ける方法です。

例えば、あるトリガーが別のデータ収集方法を使用した別のトリガーによって解決するようなトリガーを定義するかもしれません。

障害

- トリガーによって「障害」の状態を示します。

障害の更新

- Zabbix では、発生した障害の管理オプションとして、コメントを追加したり、確認済にしたり、深刻度を変更したり、手動でクローズしたりすることができます。

アクション

- 発生したイベントに反応して行う処理を事前に定義したものです。

アクションは、操作（通知の処理など）と条件（処理が実行される時）で構成されます。

エスカレーション

- アクション内で操作を実行するためのカスタムシナリオで、通知の送信やリモートコマンド実行などの一連の処理の流れです。

メディア

- 通知を配信する手段、つまり配信チャネルです。

通知

- 選択したメディアチャネルを使用してユーザーに送信されるイベントに関するメッセージです。

リモートコマンド

- ある状態になった時に監視しているホスト上で自動的に実行するための事前に定義したコマンドです。

テンプレート

- 1 つまたは複数のホストに対して適用できるよう用意されたエンティティ（アイテム、トリガー、グラフ、ダッシュボード、ローレベルディスクバリのルール、Web 監視設定）のセットです。

テンプレートの役割は、ホストへの監視設定の展開を高速化するためにあります。また、監視設定の大量の変更を簡単に適用できるようにします。そして、テンプレートは、個別のホストに直接リンクされています。

Web シナリオ

- Web サイトの可用性を確認するための 1 つまたは複数の HTTP リクエストです。

Web インターフェース

- Zabbix によって提供される Web インターフェースです。

ダッシュボード

- ウィジェットと呼ばれる表示用部品を使用して Web インターフェース上に要約や視覚化できるようカスタマイズします。

ウィジェット

- ダッシュボード内で使用できる特定のものや情報ソースの情報 *（要約、マップ、グラフ、時計など）を表示する表示用の部品です。

Zabbix API

- Zabbix API を使うことで、JSON RPC プロトコルで Zabbix 上の情報（ホスト、アイテム、グラフなど）を作成、更新、情報取得したり、その他のタスク処理を行うことができます。

Zabbix サーバー

- 監視を行うために、Zabbix プロキシや Zabbix エージェントと通信し、トリガーの条件式を判断し、通知の処理を行ったりする Zabbix のソフトウェアの中でも中心となるプロセスです。データを集約する中心の機能となります。

Zabbix プロキシ

- Zabbix サーバーに代わってデータを収集し、Zabbix サーバーの処理の負荷を軽減するプロセスです。

Zabbix エージェント

- ローカルのリソースやアプリケーションを能動的に監視するために、監視対象上で起動されるプロセスです。

Zabbix エージェント 2

- ローカルのリソースとアプリケーションを能動的に監視する新世代の Zabbix エージェントです。監視のためにカスタムプラグインを使用することができます。

Attention:

Zabbix エージェント 2 は、Zabbix エージェントと多くの機能を共有しています。ドキュメントに記載されている「Zabbix エージェント」という用語は、同じ機能が用意されている場合は、Zabbix エージェントと Zabbix エージェント 2 の両方を指します。Zabbix エージェント 2 は、異なる機能である場合にのみ、具体的に名前を記載するようにしています。

暗号化

- TLS プロトコルを使用して、Zabbix の各コンポーネント (server、proxy、agent、zabbix_sender、zabbix_get などのユーティリティ) 間は暗号化することができます。

ネットワークディスカバリ

- ネットワークデバイスの自動的なディスカバリ機能です。

ローレベルディスカバリ

- 特定のデバイス上の詳細なエンティティ (ファイルシステム、ネットワークインターフェースなど) の自動的なディスカバリ機能です。

ローレベルディスカバリルール

- デバイス上の自動的なローレベルディスカバリ処置の定義です。

アイテムのプロトタイプ

- 変数として特定のパラメーターを持ちローレベルディスカバリで使用されるメトリクスです。ローレベルディスカバリ後に、変数は実際にディスカバリされたパラメーターに自動的に置き換えられます。そして、自動的にそのメトリクスデータの収集を開始します。

トリガーのプロトタイプ

- 変数として特定のパラメーターを持ちローレベルディスカバリで使用されるトリガーです。ローレベルディスカバリ後に、変数は実際にディスカバリされたパラメーターに置き換えられます。そして、自動的にそのトリガーによってデータの評価を開始します。

他のローレベルディスカバリで使用されるプロトタイプとしては、グラフのプロトタイプ、ホストのプロトタイプ、ホストグループのプロトタイプがあります。

エージェントによる自動登録

- Zabbix エージェント自身で登録する自動化されたプロセスによってホストとして登録されて監視が開始されます。

3. Zabbix のプロセス

サイドバーを使用して Zabbix のプロセスの各コンテンツにアクセスしてください。

1 Zabbix サーバー

概要

Zabbix サーバーは、Zabbix のソフトウェアの中でも中核となるプロセスです。

Zabbix サーバーは、データのポーリングとトラッピングを処理し、それをトリガーによって判断し、ユーザーに対して通知を行います。Zabbix サーバーは、Zabbix エージェントや Zabbix プロキシからシステムの可用性や整合性のデータを受け取る中核となるコンポーネントです。Zabbix サーバー自体は、シンプルなサービスチェックの機能を使用して、ネットワークに接続されたサービス (Web サーバーやメールサーバーなど) をリモートからチェックすることができます。

Zabbix サーバーは、すべての設定、統計、運用データが保存される中核となるリポジトリで、監視システムのいずれかで障害が発生したときに、管理者に対して能動的に警告を通知する役割を担います。

Zabbix サーバーの基本的な機能は、3 つの別々のコンポーネントに分けられます。その 3 つとは、Zabbix サーバー、Web インターフェース、データベースストレージです。

Zabbix のすべての設定情報はデータベース上に保存されています。そして、サーバーと Web インターフェースの両方がそのデータベースと通信します。例えば、Web インターフェース (または API) を使用して新しいアイテムを作成した時には、データベース内の items テーブルに追加されます。次に、Zabbix サーバーは約 1 分に 1 回、有効なアイテムの一覧を取得するため items テーブルを検索し、Zabbix サーバー内のキャッシュに保存します。これが、Zabbix の Web インターフェースで行った更新が Zabbix サーバーに反映され、最新データとして見る事ができるまでに最大 2 分かかる場合がある理由です。

Zabbix サーバーの起動

パッケージを使用してインストールした場合

Zabbix サーバーは、デーモンプロセスとして起動されます。サーバーの起動には以下のように実行します：

```
shell> service zabbix-server start
```

これは、ほとんどの GNU/Linux システムで動きます。他のシステムでは、以下のように実行する必要があるかもしれません：

```
shell> /etc/init.d/zabbix-server start
```

同様に、停止、再起動、ステータス表示には以下のように実行します：

```
shell> service zabbix-server stop
shell> service zabbix-server restart
shell> service zabbix-server status
```

手動での起動

上記の方法で起動できない場合は、手動で起動する必要があります。zabbix_server のバイナリを見つけて、以下のように実行します：

```
shell> zabbix_server
```

Zabbix サーバーでは、以下のコマンドラインパラメーターを使用できます：

-c --config <file>	設定ファイルを絶対パスで指定します。(デフォルトは/usr/local/etc/zabbix_s
-f --foreground	Zabbixサーバーをフォアグラウンドで実行します。
-R --runtime-control <option>	管理用のランタイムコマンドを実行します。
-h --help	ヘルプを表示します。
-V --version	バージョン情報を表示します。

Zabbix サーバーのコマンドラインパラメータの使用例です：

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf
shell> zabbix_server --help
shell> zabbix_server -V
```

ランタイムコントロール

ランタイムコントロールオプション

オプション	説明	ターゲット
config_cache_reload	設定キャッシュをリロードします。キャッシュが現在ロード中の場合は無視されます。	
diaginfo[=<target>]	サーバーログファイルで診断情報を収集します。	historycache - 履歴キャッシュの統計 valuecache - 値キャッシュの統計 preprocessing - 前処理マネージャー統計 alerting - アラートマネージャの統計 lld - LLD マネージャーの統計 locks - ミューテックスのリスト (**BSD* system では空です)
ha_status	高可用性 (HA) クラスターの状態をログに記録します。	
ha_remove_node=<name>	名前または ID で指定された高可用性 (HA) ノードを削除します。アクティブ/スタンバイノードは削除できないことに注意してください。	ターゲット - ノードの名前または ID (ha_status を実行して取得できます)
ha_set_failover_delay=<time>	高可用性 (HA) フェイルオーバー遅延を設定します。タイムサフィックスがサポートされています。例: 10s, 1m.	
secrets_reload	Vault からシークレットをリロードします。	
service_cache_reload	サービスマネージャーのキャッシュを再読み込みします。	
snmp_cache_reload	SNMP キャッシュをリロードし、すべてのホストの SNMP プロパティ (エンジン時間、エンジンブート、エンジン ID、認証情報) をクリアします。	
housekeeper_execute	ハウスキーピング手順を開始します。ハウスキーピング手順が現在進行中の場合は無視されます。	

オプション	説明	ターゲット
trigger_housekeeper_execute	ハウスキーピング手順を開始します。トリガーハウスキーピングプロシージャが現在進行中の場合は無視されます。	
log_level_increase[=<target>]	=<target>を上げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。 **BSD* システムではサポートされていません。	process type - 指定されたタイプのすべてのプロセス (ポーラーなど) サーバープロセスタイプ をすべて表示 process type,N - プロセスの種類と数 (例: poller,3) pid - プロセス ID(1 から 65535). より大きな値の場合は、ターゲットを 'process type,N' として指定します。
log_level_decrease[=<target>]	=<target>を下げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。 **BSD* システムではサポートされていません。	

ランタイム制御を使用してサーバー構成キャッシュをリロードする例:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R config_cache_reload
```

ランタイム制御を使用して診断情報を収集する例:

サーバー ログ ファイルで利用可能なすべての診断情報を収集:

```
shell> zabbix_server -R diaginfo
```

サーバー ログ ファイルで履歴キャッシュ統計を収集:

```
shell> zabbix_server -R diaginfo=historycache
```

ランタイム制御を使用して SNMP キャッシュをリロードする例:

```
shell> zabbix_server -R snmp_cache_reload
```

ランタイム制御を使用してハウスキーパーの実行をトリガーする例:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R housekeeper_execute
```

ランタイムコントロールを使用してログレベルを変更する例:

すべてのプロセスのログ レベルを上げる:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase
```

2 番目のポーラー プロセスのログ レベルを上げる:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2
```

PID 1234 でプロセスのログ レベルを上げる:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234
```

すべての http ポーラー プロセスのログレベルを下げる:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

HA フェイルオーバー遅延を最小 10 秒に設定する例:

```
shell> zabbix_server -R ha_set_failover_delay=10s
```

プロセス起動ユーザー

Zabbix サーバーは、非 root ユーザーで動作するように設計されています。起動されたどんな非 root ユーザーでも起動すれば動作します。したがって、非 root ユーザーで問題なくサーバーを実行できます。

root ユーザーで起動した場合には、ハードコードされた zabbix ユーザーに切り替わります。このユーザーは、システム上に**存在する必要**があります。サーバーの設定ファイル内の「AllowRoot」パラメーターを変更した時のみ、root ユーザーで実行させることができます。

Zabbix サーバーと**エージェント**が同じマシン上で動作する場合、サーバーの実行とエージェントの実行に、異なるユーザーを使用することをお勧めします。そうしないと、同じユーザーとしてサーバーとエージェントの両方が実行された場合、エージェントはサーバーの設定ファイルにアクセスでき、例えば、Zabbix 内のどの管理レベルのユーザーでも、極めて容易にデータベースのパスワードを取得することができてしまいます。

設定ファイル

zabbix_server の設定の詳細は、[設定ファイル](#)を参照してください。

起動スクリプト

スクリプトは、システムの起動/シャットダウン中の Zabbix プロセスの自動起動/停止に使用されます。スクリプトは、misc/init.d ディレクトリの下にあります。

サーバープロセスの種類

- alert manager - アラート・キュー・マネージャー
- alert syncer - アラート DB ライター
- alerter - 通知の送信プロセス
- availability manager - ホスト可用性の更新プロセス
- configuration syncer - 設定データのメモリ内キャッシュを管理するプロセス
- discoverer - デバイス検出プロセス
- escalator - アクションエスカレーションのプロセス
- history poller - データベース接続を必要とする計算された内部チェックを処理するためのプロセス
- history syncer - 履歴 DB ライター
- housekeeper - 古い履歴データの削除プロセス
- http poller - ウェブ監視ポラー
- icmp pinger - icmping チェックのポラー
- ipmi manager - IPMI ポラーマネージャー
- ipmi poller - IPMI チェックのポラー
- java poller - Java チェックのポラー
- lld manager - ローレベルディスクバリタスクのマネージャプロセス
- lld worker - ローレベルディスクバリタスクのワーカプロセス
- odbc poller - ODBC チェックのポラー
- poller - パッシブチェック用の通常のポラー
- preprocessing manager - 前処理タスクのマネージャー
- preprocessing worker - データ前処理のプロセス
- problem housekeeper - 削除されたトリガーの問題を解決するためのプロセス
- proxy poller - パッシブプロキシのポラー
- report manager - スケジュールされたレポート生成タスクのマネージャー
- report writer - スケジュールされたレポートを生成するプロセス
- self-monitoring - 内部サーバー統計を収集するためのプロセス
- snmp trapper - SNMP トラップのトラッパー
- task manager - 他のコンポーネントによって要求されたタスクをリモートで実行するためのプロセス (例: 障害のクローズ、障害の確認、項目値の即時チェック、リモートコマンド機能)
- timer - メンテナンス処理用タイマー
- trapper - アクティブチェック、トラップ、プロキシ通信のトラッパー
- unreachable poller - 到達不能デバイスのポラー
- vmware collector - VMware サービスからのデータ収集を担当する VMware データコレクター

サーバーログファイルを使用して、これらのプロセスタイプを監視できます。

zabbix[process,<type>,<mode>,<state>] internal **item**を使用して、さまざまなタイプの Zabbix サーバープロセスを監視できます。

サポートされているプラットフォーム

サーバーの処理のセキュリティ要件やミッションクリティカルな性質から、必要なパフォーマンス、フォールトトレランス、および復元力を一貫して提供できる OS は、UNIX だけです。Zabbix は、市場に出回っているバージョンで動作します。

Zabbix サーバーは、以下のプラットフォーム上での動作を確認済みです：

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server
- Tru64/OSF1

Note:

Zabbix は、他の UNIX 系 OS でも同様に動作します。

ロケール

いくつかのテキスト形式のアイテムを正しく処理できるようにするため、サーバーには UTF-8 のロケールが必要であることに注意してください。最近のほとんどの UNIX ライクなシステムにはデフォルトで UTF-8 ロケールがありますが、システムによっては具体的に設定しなければ使用することができません。

1 高可用性 (HA)

概要

実質的にダウンタイムを発生させられない重要なインフラストラクチャーでは、高可用性 (HA) が求められます。したがって、停止してしまう可能性があるサービスについては、現在のサービスが停止してしまったときに処理を引き継げるようにするためのフェールオーバーオプションが必要です。

Zabbix は、セットアップが簡単で、以前の HA の専門知識を必要としない組み込まれた高可用性のソリューションを提供します。組み込まれた Zabbix の HA は、Zabbix サーバーのソフトウェア/ハードウェアの障害に対する保護やメンテナンスに伴うダウンタイムを削減するのに役に立つでしょう。

Zabbix の高可用性モードでは、複数の Zabbix サーバーがクラスター内のノードとして実行されます。クラスター内の 1 台の Zabbix サーバーがアクティブである間、他のサーバーはスタンバイ状態となり、必要に応じて引き継ぐことができるよう準備されます。



Zabbix の HA への切り替えは戻すことが可能で、いつでもスタンダロン環境に戻すことができます。

こちらも参照: [実装の詳細](#)

高可用性の有効化

Zabbix サーバーをクラスターノードとして起動

Zabbix サーバーをクラスターノードとして起動するには、サーバー設定で 2 つのパラメーターを設定する必要があります。

- **HANodeName** パラメータは、HA クラスターノードとなる Zabbix サーバごとに指定する必要があります。

これは、エージェントおよびプロキシ構成でサーバーが参照される一意のノード識別子 (例: zabbix-node-01) です。HANodeName が指定されていない場合、サーバーはスタンダロンモードで起動されます。

- **NodeAddress** パラメーターは、ノードごとに指定する必要があります。

The NodeAddress parameter (address:port) will be used by Zabbix frontend to connect to the active server node. NodeAddress must match the IP or FQDN name of the respective Zabbix server. NodeAddress パラメータ (アドレス: ポート番号) は、Zabbix フロントエンドがアクティブなサーバーノードに接続するために使用されます。NodeAddress は、それぞれの Zabbix サーバーの IP または FQDN 名と一致する必要があります。

設定ファイル変更後、すべての Zabbix サーバーを再起動します。これでクラスターノードとして機能します。サーバーの最新ステータスは、レポート → [システム情報](#) で確認できます。

```
zabbix_server -R ha_status
```

上記ランタイムコマンドは、現在の HA クラスターの状態を Zabbix サーバーログ (および stdout) に記録します。

```
Failover delay: 60 seconds
```

```
Cluster status:
```

#	ID	Name	Address	Status	Last Access
1.	ckzxxqg7u00011sropenyzh3m	zabbix-node-01	64.227.66.193:10051	standby	0s
2.	ckzxyqo1k00013frpq539e1jp	zabbix-node-02	64.227.74.25:10051	active	3s

Web インターフェースの準備

Web インターフェースの設定ファイル (Web インターフェースのファイルがあるディレクトリ内または/etc/zabbix/web ディレクトリ内にある conf/zabbix.conf.php) 内で、Zabbix サーバーのアドレス: ポート番号が設定されていないことを確認してください。

```
// Uncomment and set to desired values to override Zabbix hostname/IP and port.  
// $ZBX_SERVER                = '';  
// $ZBX_SERVER_PORT           = '';
```

Zabbix の Web インターフェースは、Zabbix データベースのノードテーブルから設定を読み取って、アクティブノードを自動的に検出します。アクティブノードのノードアドレスが Zabbix サーバーのアドレスとして使用されます。

Zabbix プロキシの設定

HA クラスターノード (サーバー) は、パッシブまたはアクティブモードの Zabbix プロキシの設定ファイルにリスト化されている必要があります。

パッシブプロキシの場合、ノード名をプロキシ設定ファイルの **Server** パラメータにカンマ区切りでリスト化する必要があります。

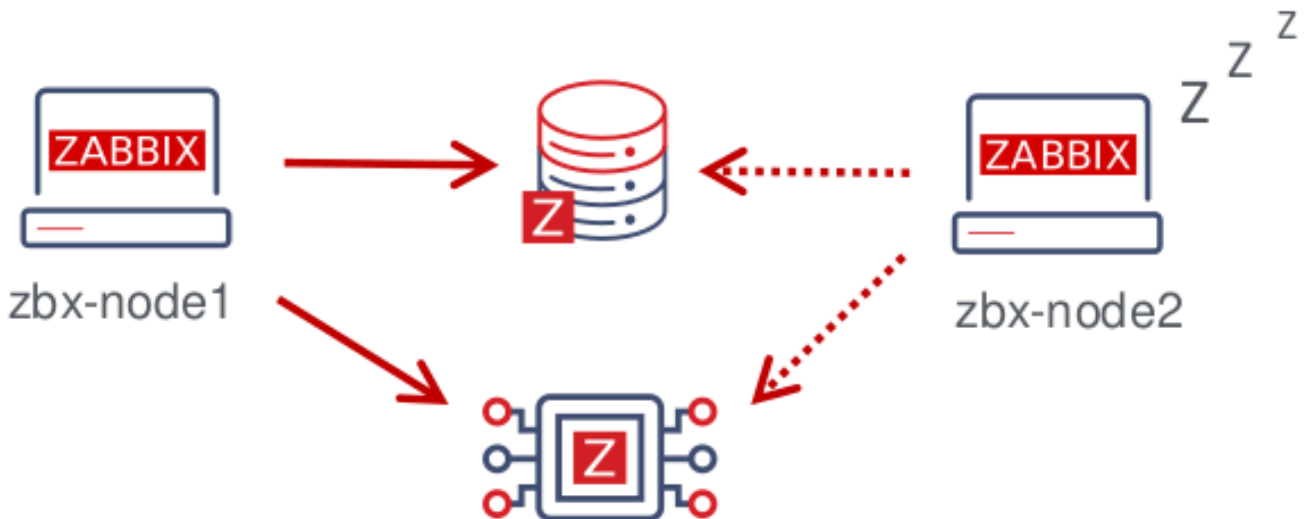
```
Server=zabbix-node-01,zabbix-node-02
```

アクティブプロキシの場合、ノード名をプロキシ設定ファイルの **Server** パラメータにセミコロン区切りでリスト化する必要があります。

```
Server=zabbix-node-01;zabbix-node-02
```

Zabbix エージェントの設定

HA クラスターのノード (サーバー) は、Zabbix エージェントまたは Zabbix エージェント 2 の設定ファイルに並べて記述する必要があります。



パッシブチェックを行うのならば、ノード名を**Server** パラメータにカンマ区切りで設定する必要があります。

```
Server=zabbix-node-01,zabbix-node-02
```

アクティブチェックを行うのならば、ノード名を**ServerActive** パラメータにセミコロン区切りで設定する必要があります。例:

```
ServerActive=zabbix-node-01;zabbix-node-02
```

スタンバイノードへのフェイルオーバー

アクティブなノードが停止すると、Zabbix は自動的に別のノードにフェイルオーバーします。フェイルオーバーが発生するには、少なくとも 1 つのノードがスタンバイ状態である必要があります。

フェイルオーバーはどのくらいの速さで実行されるか。すべてのノードは 5 秒ごとに最終アクセス時刻 (および変更されている場合はステータス) を更新します。よって: - アクティブノードがシャットダウンし、そのステータスが“ 停止” と報告された場合、別のノードが 5 秒以内に引き継ぎます。

- アクティブノードがシャットダウンするか、ステータスを更新できずに使用できなくなった場合、スタンバイノードはフェイルオーバー遅延 + 5 秒待機して引き継ぎます。

フェイルオーバー遅延は構成可能で、サポートされる範囲は 10 秒から 15 分 (デフォルトでは 1 分) です。フェイルオーバー遅延の変更は次で実行できます。

```
zabbix_server -R ha_set_failover_delay=5m
```

HA クラスターの管理

HA クラスターの現在のステータスは、専用の **ランタイムコントロール** オプションを使用して管理できます。HA クラスターの現在のステータスは、専用の **ランタイムコントロール** オプションを使用して管理できます。The current status of the HA cluster can be managed using the dedicated **runtime control** options:

- `ha_status` - Zabbix サーバーログ (および標準出力) に HA クラスターのステータスを記録します。
- `ha_remove_node=target` - リスト内のノードの番号 (この番号は、`ha_status` を実行した出力から取得できます) を `<target>` で指定して HA ノードを削除します。例:

```
zabbix_server -R ha_remove_node=2
```

アクティブ/スタンバイノードは削除できないことに注意してください。

- `ha_set_failover_delay=delay` - HA フェイルオーバー遅延の設定 (10 秒から 15 分まで。タイムサフィックスがサポートされています。例: 10s, 1m)

ノードのステータスを監視できます。

- レポート → **システム情報**
- システム情報のダッシュボードウィジェット
- サーバーの `ha_status` ランタイム制御オプション (上記を参照)

zabbix [クラスター, ディスカバリ, ノード] の内部アイテムは、高可用性ノード情報を含む JSON を返すため、ノード検出に使用できます。

高可用性の無効化

高可用性クラスターを無効にするには:

- 設定ファイルのバックアップコピーを作成する
- スタンバイノードを停止する
- アクティブなプライマリサーバーから `HANodeName` パラメータを削除
- プライマリサーバーを再起動 (スタンドアロンモードで起動します)

実装の詳細

高可用性 (HA) クラスターはオプトインソリューションであり、Zabbix サーバーでサポートされています。ネイティブ HA ソリューションは、簡単に使用できるように設計されており、複数のサイトで機能し、Zabbix が認識するデータベースに関する特定の要件はありません。ユーザーは、環境の高可用性要件に最適なものにに応じて、ネイティブの Zabbix HA ソリューションまたはサードパーティの HA ソリューションを自由に使用できます。

このソリューションは複数の `zabbix_server` インスタンスまたはノードで構成されています。

ノード毎に: - 個別に構成 - 同じデータベースを使用 - いくつかのモードがあります: アクティブ、スタンバイ、使用不可、停止

一度にアクティブ (稼働) できるノードは 1 つだけです。スタンバイノードは、HA マネージャーという 1 つのプロセスのみを実行します。スタンバイノードは、データの収集、処理、またはその他の通常のサーバーアクティビティを行いません。ポートをリッスンしません。データベース接続は最小限です。

アクティブノードとスタンバイノードの両方が、最終アクセス時刻を 5 秒ごとに更新します。各スタンバイノードは、アクティブノードの最終アクセス時刻を監視します。アクティブノードの最後のアクセス時間が 'failover delay' 秒を超えている場合、スタンバイノードは自身をアクティブノードに切り替え、以前のアクティブノードに '使用不可' ステータスを割り当てます。

アクティブノードは自身のデータベース接続を監視します。接続が `failover delay-5` 秒を超えて失われた場合は、すべての処理を停止してスタンバイモードに切り替えます。アクティブノードは、スタンバイノードのステータスも監視します。スタンバイノードの最終アクセス時間が 'failover delay' 秒を超えている場合、スタンバイノードには 'unavailable' ステータスが割り当てられます。

ノードは、Zabbix のマイナーバージョン間で互換性があるように設計されています。

2 Zabbix エージェント

概要

ローカルリソースやアプリケーション（ハードディスクドライブ、メモリ、プロセッサの統計など）を能動的に監視するために、Zabbix エージェントを監視対象上にインストールします。

Zabbix エージェントは、稼働情報をローカルで収集し、さらに処理するためにデータを Zabbix サーバーに報告します。障害（ハードディスクがいっぱいになっている、サービスプロセスがクラッシュしているなど）が発生した場合、Zabbix サーバーは障害が発生したマシンの管理者に対して、障害が発生したことを能動的に通知することができます。

Zabbix エージェントは、統計情報を収集するためにネイティブなシステムコールを使用するため、極めて効率的に動作します。

パッシブチェックとアクティブチェック

Zabbix エージェントは、パッシブチェックとアクティブチェックを行うことができます。

パッシブチェックでは、エージェントはデータの要求に対して応答します。Zabbix サーバー（またはプロキシ）が、例えば CPU 負荷のデータを要求すると、Zabbix エージェントが結果を返却します。

アクティブチェックでは、より複雑な処理が求められます。Zabbix エージェントは、独立した処理のために、最初に Zabbix サーバーからアイテムのリストを取得します。その後、定期的に新しい値を Zabbix サーバーに送信します。

パッシブチェック、アクティブチェックのどちらで動作するかは、それぞれの監視対象の**アイテムのタイプ**を選択することによって設定されます。Zabbix エージェントは、アイテムの「Zabbix エージェント」または「Zabbix エージェント (アクティブ)」のタイプを処理します。

サポートしているプラットフォーム

Zabbix エージェントは、以下の OS での動作をサポートしています：

- Linux
- IBM AIX
- FreeBSD
- NetBSD
- OpenBSD
- HP-UX
- Mac OS X
- Solaris: 9, 10, 11
- Windows: XP 以降のバージョンのすべてのデスクトップとサーバー

UNIX 系 OS 上の Zabbix エージェント

UNIX 系 OS 上の Zabbix エージェントは、監視対象のホストで実行されます。

インストール

パッケージを使用して Zabbix エージェントをインストールする方法については、**パッケージインストール**のセクションを参照してください。

また、パッケージを使用しない場合は**ソースコードからのインストール**の手順を参照してください。

Attention:

一般的に、32 ビット版 Zabbix エージェントは 64 ビットのシステム上で動作しますが、場合によっては失敗する可能性があります。

パッケージでインストールした場合

Zabbix エージェントは、デーモンプロセスとして実行されます。エージェントは以下のように実行して起動します：

```
shell> service zabbix-agent start
```

これは、ほとんどの GNU/Linux で動きます。他のシステムでは、以下のように実行します：

```
shell> /etc/init.d/zabbix-agent start
```

同様に、Zabbix エージェントを停止、再起動、状態表示するためには、以下のコマンドを使用します：

```
shell> service zabbix-agent stop
shell> service zabbix-agent restart
shell> service zabbix-agent status
```

手動での起動

手動で起動しようとして上記の方法で起動できない場合は、zabbix_agentdの実行ファイルのパスを確認して以下のように実行します：

```
shell> zabbix_agentd
```

Windows 上での Zabbix エージェント

Windows 上の Zabbix エージェントは、Windows 上のサービスとして実行されます。

準備

Zabbix エージェントは、zip アーカイブとして配布されます。アーカイブをダウンロードしたら展開する必要があります。Zabbix エージェント本体と設定ファイルを保存するフォルダを選択してください。例：

```
C:\zabbix
```

bin\zabbix_agentd.exe と conf\zabbix_agentd.conf files を c:\zabbix にコピーしてください。

c:\zabbix\zabbix_agentd.conf ファイルを必要に応じて編集し、特に"Hostname" パラメータは正しいことを確かめてください。

インストール

これが完了したら、次のコマンドを使用して Zabbix エージェントをインストールします。Windows サービス：

```
C:\> c:\zabbix\zabbix_agentd.exe -c c:\zabbix\zabbix_agentd.conf -i
```

これで、「Zabbix agent」サービスをその他の Windows サービスと同様、通常どおりに構成できるはずです。

続きを見る[詳細](#) Windows での Zabbix エージェントのインストールと実行について。

その他のエージェントオプション

ホスト上でエージェントは複数のインスタンスを実行することができます。単一のインスタンスは、デフォルトの設定ファイルまたはコマンドラインで指定された設定ファイルを利用できます。複数の場合には、各エージェントインスタンス毎に独自の設定ファイルが必要です。(1つのインスタンスはデフォルトの設定ファイルを使用できます)

次のコマンドラインパラメータは、Zabbix エージェントで使用できます。

パラメータ	説明
UNIX および Windows	
エージェント	
-c --config <config-file>	設定ファイルへのパス。 このオプションを使用して、デフォルト以外の設定ファイルを指定できます。 UNIX では、デフォルトは /usr/local/etc/zabbix_agentd.conf または コンパイル時 変数で設定されたもの。--sysconfdir または --prefix Windows では、デフォルトは c:\zabbix_agentd.conf
-p --print	既知の項目を出力して終了します。 注: ユーザーパラメータ の結果も返すには、設定ファイルを指定する必要があります。(デフォルトの場所がない場合)
-t --test <item key>	指定されたアイテムをテストして終了します。 注: ユーザーパラメータ の結果も返すには、設定ファイルを指定する必要があります。(デフォルトの場所がない場合)
-h --help	ヘルプ情報を表示
-V --version	バージョン番号を表示
UNIX エージェントのみ	
-R --runtime-control <option>	管理機能を実行します。 ランタイムコントロール を参照してください。
Windows エージェントのみ	
-m --multiple-agents	複数のエージェントインスタンスを使用します (-i、-d、-s、-x ファクションと共に)。 インスタンスのサービス名を区別するために、各サービス名には設定ファイルで指定された Hostname の値が含まれます。
Windows エージェントのみ (ファンクション)	
-i --install	Zabbix Windows エージェントをサービスとしてインストール
-d --uninstall	Zabbix Windows エージェントサービスをアンインストール
-s --start	Zabbix Windows エージェントサービスを開始
-x --stop	Zabbix Windows エージェントサービスを停止

コマンドラインパラメーターを使用する具体的な例:

- すべてのビルトインエージェントアイテムを値とともに出力します。
- 指定した設定ファイル内で定義された「mysql.ping」キーを使用したユーザーパラメータのテスト
- Windows 用の「Zabbix Agent」サービスをインストールします。デフォルトの設定ファイルへのパス c:\zabbix_agentd.conf
- Windows 用の「Zabbix Agent [Hostname]」サービスをインストールします。エージェントが実行可能な、同じ場所にある設定ファイルフォルダの zabbix_agentd.conf を Hostname の値で拡張してサービス名を一意にします。

```
shell> zabbix_agentd --print
shell> zabbix_agentd -t "mysql.ping" -c /etc/zabbix/zabbix_agentd.conf
shell> zabbix_agentd.exe -i
shell> zabbix_agentd.exe -i -m -c zabbix_agentd.conf
```

ランタイム制御

ランタイム制御オプションを使用すると、エージェントプロセスのログレベルを変更できます。

オプション	説明	ターゲット
log_level_increase[=target]	ログレベルを上げます。 ターゲットが指定されていない場合、すべてのプロセスが影響を受けます。	ターゲットは次のように指定できます: プロセスタイプ - 指定されたタイプ (例 listener) のすべてのプロセス すべてのエージェントプロセスタイプを参照してください。 プロセスタイプ,N - プロセスのタイプと数 (例 listener,3) pid - プロセス ID (1 ~ 65535)。より大きな値の場合は、'プロセスタイプ,N' の指定方法にしてください。
log_level_decrease[=target]	ログレベルを下げます。 ターゲットが指定されていない場合、すべてのプロセスが影響を受けます。	
userparameter_reload	現在の設定ファイルからユーザーパラメータを再読み込みします。 UserParameter のみエージェントの設定オプションから再読み込みされます。	

例:

- すべてのプロセスのログレベルを上げる
- 3 番目の listener プロセスのログレベルを上げる
- PID 1234 でプロセスのログレベルを上げる
- すべてのアクティブなチェックプロセスのログレベルを下げる

```
shell> zabbix_agentd -R log_level_increase
shell> zabbix_agentd -R log_level_increase=listener,3
shell> zabbix_agentd -R log_level_increase=1234
shell> zabbix_agentd -R log_level_decrease="active checks"
```

Note:

ランタイム制御は、OpenBSD、NetBSD、および Windows ではサポートしません。

エージェントプロセスタイプ

- active checks - アクティブチェックを実行するためのプロセス
- collector - データ収集のプロセス
- listener - パッシブチェックを受け付けるプロセス

エージェントログファイルを利用して、これらのプロセスタイプを観察できます。

プロセスユーザー

UNIX 上の Zabbix エージェントは、root 以外のユーザーとして実行するように設計されています。起動時から root 以外のユーザーとして実行されます。なので、root 以外のユーザーでエージェントを実行しても何も問題ありません。

単純に「root」として実行しようとすると、システムから提供されている「zabbix」ユーザーに切り替わるようハードコードされています。エージェントの設定ファイルの「AllowRoot」パラメーターを適切に変更することによって「root」としてエージェントを実行できます。

設定ファイル

Zabbix エージェントの設定の詳細については、設定ファイルのオプションを参照してください。[Zabbix エージェント \(UNIX\)](#) または [Zabbix エージェント \(Windows\)](#)。

ロケール

文字型のアイテムにて期待される値を返すには、UTF-8 ロケールがエージェントに必要であることに注意してください。最新の Unix ライクなシステムのほとんどはデフォルトで UTF-8 ロケールを持っていますが、一部のシステムでは具体的に設定する必要があるかもしれません。

終了コード

バージョン 2.2 より前の Zabbix エージェントは、正常に終了した場合に 0 を返し失敗した場合は 255 を返しました。バージョン 2.2 以降の Zabbix エージェントは、正常に終了した場合は 0 を返し、失敗した場合は 1 を返します。

3 Zabbix エージェント 2

概要

Zabbix エージェント 2 は新世代の Zabbix エージェントであり、Zabbix エージェントの代わりに使用できます。Zabbix エージェント 2 は次の目的で開発されました。

- TCP 接続の数を減らす
- チェックの同時実行性改善
- プラグイン拡張の簡易化。プラグインは次のようなことが可能です
 - 数行の単純なコードのみで構成される簡単なチェック
 - 長時間実行されるスクリプトと、データの定期的な返送を伴うスタンドアロンのデータ収集で構成される複雑なチェック
- Zabbix エージェントのドロップイン代替品 (以前のすべての機能をサポートするという点で)

Zabbix エージェント 2 は Go 言語で記述されています (Zabbix エージェントの C コードが再利用されています)。Zabbix エージェント 2 をビルドするには、現在サポートされている [Go バージョン](#) で構成された Go 環境が必要です。

Zabbix エージェント 2 には、Linux 向け組み込みデーモン化サポートがありません。Windows では [Windows サービス](#) として実行できます。

パッシブチェックは Zabbix エージェントと同様に機能します。アクティブチェックは、スケジュールされた柔軟な間隔設定をサポートし、アクティブサーバー内の同時実行性をチェックします。

Note:

デフォルトでは、Zabbix エージェント 2 は、リソース使用量の急増を防ぐために、アイテムの更新間隔内の条件付きランダム時間にアクティブチェックの最初のデータ収集をスケジュールします。エージェントの再起動直後に予定された [更新確認間隔](#) がないアクティブチェックを実行するには、[構成ファイル](#) で ForceActiveChecksOnStart パラメーター (グローバルレベル) または Plugins.<プラグイン名>.System.ForceActiveChecksOnStart (特定のプラグインチェックにのみ影響します) を設定します。プラグインレベルのパラメータが設定されている場合、グローバルパラメータを上書きします。Zabbix 6.0.2 以降、起動時にアクティブチェックを強制することがサポートされています。

同時実行性を確認してください

異なるプラグインからのチェックを同時に実行できます。1 つのプラグイン内の同時チェック数は、プラグイン容量設定によって制限されます。各プラグインにはハードコードされた容量設定 (デフォルトは 100) があり、プラグイン設定 [パラメーター](#) の Plugins.<プラグイン名>.System.Capacity=N を使用して下げることができます。このパラメータの以前の名前 Plugins.<PluginName>.Capacity は引き続きサポートされていますが、Zabbix 6.0 では非推奨になっています。

参照: [プラグイン開発ガイドライン](#)

サポートされているプラットフォーム

Zabbix エージェント 2 は以下のプラットフォームでサポートされています:

- Windows (XP 以降のすべてのデスクトップとサーバーのバージョン; [コンパイル済み実行バイナリ](#) で提供)
- Linux ([ディストリビューション用パッケージ](#) で提供)

インストール

Zabbix エージェント 2 はコンパイル済みの Zabbix パッケージが用意されています。Zabbix エージェント 2 を [ソース](#) からコンパイルするときには、configure 実行時に --enable-agent2 オプションを指定してください。

オプション

Zabbix エージェント 2 では次のコマンドラインパラメータを使用できます。

パラメーター	説明
-c --config <config-file>	設定ファイルへのパスを指定します。 このオプションを使用してデフォルトではない設定ファイルを指定できます。 UNIX の場合、デフォルトは /usr/local/etc/zabbix_agent2.conf であるか、コンパイル時の変数 --sysconfdir または --prefix で設定されます。
-f --foreground	Zabbix エージェントをフォアグラウンドで実行します (デフォルト: true)
-p --print	既知のアイテムを出力して終了します。 注: ユーザーパラメーターの結果も返すには、(デフォルトの場所がない場合は) 構成ファイルを指定する必要があります。
-t --test < アイテムキー >	指定されたアイテムをテストして終了します。 注: ユーザーパラメーターの結果も返すには、(デフォルトの場所がない場合は) 構成ファイルを指定する必要があります。
-h --help	ヘルプを出力して終了します
-v --verbose	デバッグ情報を出力します。このオプションは -p および -t オプションとともに使用します。
-V --version	エージェントのバージョン番号を出力して終了します。
-R --runtime-control <option>	管理機能を実行します。ランタイム制御を参照してください。

コマンドラインパラメータの使用の具体的な例：

- すべての組み込みエージェントアイテムの値を出力します
- 指定された設定ファイルで定義された "mysql.ping" キーを使用してユーザーパラメータをテストします

```
shell> zabbix_agent2 --print
shell> zabbix_agent2 -t "mysql.ping" -c /etc/zabbix/zabbix_agentd.conf
```

ランタイムコントロール

ランタイムコントロールは、リモートコントロールのためにいくつかのオプションを提供します。

オプション	説明
log_level_increase	ログレベルを上げます
log_level_decrease	ログレベルを下げてます
metrics	利用可能なメトリックを一覧表示します
version	エージェントのバージョンを表示します
userparameter_reload	現在の設定ファイルからユーザーパラメータをリロードします。 リロードされるエージェント設定オプションは UserParameter のみであることを注意してください。
help	ランタイムコントロールのヘルプを表示します

例:

- エージェント 2 のログレベルを上げます
- ランタイムコントロールオプションを出力します

```
shell> zabbix_agent2 -R log_level_increase
shell> zabbix_agent2 -R help
```

設定ファイル

エージェント 2 の設定パラメータは、いくつかの例外を除いて、Zabbix エージェントと互換性があります。

新しいパラメータ	説明
ControlSocket	ランタイムコントロールソケットパス。エージェント 2 はランタイムコマンドにコントロールソケットを使用します。
EnablePersistentBuffer, PersistentBufferFile, PersistentBufferPeriod	これらのパラメータは、アクティブなアイテムのエージェント 2 の永続ストレージを設定するために使用されます。

新しいパラメータ	説明
ForceActiveChecksOnStart	エージェントがアクティブチェックを再起動直後に実行するか、時間経過で均等に分散実行するかを決定します。Zabbix6.0.2以降でサポートされています。
Plugins	プラグインは <code>Plugins.<プラグイン名>.<Parameter>=<value></code> の形式で独自のパラメータを設定できます。一般的なプラグインパラメータは <code>System.Capacity</code> で、同時に実行できるチェックの制限を設定します。
StatusPort	ポートエージェント 2 は、HTTP ステータスリクエストをリッスンし、設定されたプラグインといくつかの内部パラメーターのリストを表示します。
削除されたパラメータ AllowRoot, User	説明 デーモン化がサポートされていないため、サポートされていません。
LoadModule, LoadModulePath StartAgents	ロードモジュールはサポートされていません。 このパラメータは、Zabbix エージェントではパッシブチェックの同時実行性を高めたり無効にしたりするために使用されていました。エージェント 2 では、同時実行はプラグインレベルで設定され、容量設定によって制限できます。一方、パッシブチェックの無効化は現在サポートされていません。
HostInterface, HostInterfaceltem	まだサポートされていません。

詳細については `zabbix_agent2` の設定ファイルオプションを参照してください。

終了コード

バージョン 4.4.8 以降、Zabbix エージェント 2 は古い OpenSSL バージョン (1.0.1、1.0.2) でコンパイルすることもできます。

この場合、Zabbix は OpenSSL でロックするためのミューテックスを提供します。ミューテックスのロックまたはロック解除が失敗した場合、エラーメッセージが標準エラーストリーム (STDERR) に出力され、エージェント 2 はそれぞれリターンコード 2 または 3 で終了します。

4 Zabbix プロキシ

概要

Zabbix プロキシは、1 つ以上の監視対象から監視データを収集し、その情報を Zabbix サーバーに送信するプロセスであり、基本的にサーバーに代わって機能します。収集されたすべてのデータはローカルにバッファリングされてから、プロキシが属する Zabbix サーバーに転送されます。

プロキシのデプロイはオプションですが、単一の Zabbix サーバーの負荷を分散するのに非常に役立つ場合があります。プロキシがデータを収集する場合、サーバーの CPU とディスク I/O の負荷が少なくなります。

Zabbix プロキシは、ローカル管理者がいなくても、リモートロケーション、ブランチ、ネットワークを一元的に監視するための理想的なソリューションです。

Zabbix プロキシにはサーバーとは別のデータベースが必要です。

Attention:

Zabbix プロキシでサポートされているデータベースは SQLite、MySQL、PostgreSQL であることに注意してください。Oracle の使用は自己責任であり、たとえば、低レベルの検出ルールの戻り値のように、いくつかの制限が含まれる場合があります。

参照：[分散環境でのプロキシの使用](#)

プロキシの起動

パッケージインストールされている場合

Zabbix プロキシはデーモンプロセスとして実行されます。プロキシは以下を実行することで開始できます。

```
shell> service zabbix-proxy start
```

これはほとんどの GNU/Linux システムで機能します。他のシステムでは以下の場合もあります。

```
shell> /etc/init.d/zabbix-proxy start
```

同様に Zabbix プロキシのステータスを停止/再開/表示するには、次のコマンドを使用します。

```
shell> service zabbix-proxy stop
shell> service zabbix-proxy restart
shell> service zabbix-proxy status
```

手動起動

サービス起動が機能しない場合は、手動起動する必要があります。zabbix_proxy バイナリへのパスを見つけて、以下を実行します。

```
shell> zabbix_proxy
```

Zabbix プロキシでは次のコマンドラインパラメータを使用できます。

-c --config <file>	設定ファイルパスを指定
-f --foreground	フォアグラウンドでZabbixプロキシを実行する
-R --runtime-control <option>	管理機能実行
-h --help	ヘルプの表示
-V --version	バージョン番号の表示

コマンドラインパラメータを使用して Zabbix プロキシを実行する例:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf
shell> zabbix_proxy --help
shell> zabbix_proxy -V
```

ランタイムコントロール

ランタイムコントロールオプション:

オプション	説明	ターゲット
config_cache_reload	設定キャッシュをリロードします。キャッシュが現在ロードされている場合は無視されます。アクティブな Zabbix プロキシは Zabbix サーバーに接続し、設定データを要求します。	
diaginfo[=<target>]	プロキシログファイルに診断情報を収集します	** historycache-履歴キャッシュ統計 preprocessing-前処理マネージャー統計 locks-ミューテックスリスト (BSD * システムでは空)
snmp_cache_reload	SNMP キャッシュをリロードし、すべてのホストの SNMP プロパティ (engine time, engine boots, engine id, 資格情報) をクリアします。	
housekeeper_execute	ハウスキーピングを開始します。ハウスキーピングが現在進行中の場合は無視されます。	
log_level_increase[=<target>]	ログレベルを上げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 **BSD* システムではサポートされていません。	process type - 指定されたタイプのすべてのプロセス (例: poller) すべての プロキシプロセスタイプ を参照してください。 process type,N - プロセスタイプと番号 (例: poller,3) pid - プロセス識別子 (1~65535)。値が大きい場合は、ターゲットを'process type,N' として指定します。
log_level_decrease[=<target>]	ログレベルを下げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 BSD システムではサポートされていません。	

ランタイム制御を使用してプロキシ設定キャッシュをリロードする例:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R config_cache_reload
```

ランタイム制御を使用して診断情報を収集する例:

プロキシログファイルに利用可能なすべての診断情報を収集する場合:

```
shell> zabbix_proxy -R diaginfo
```

プロキシログファイルに履歴キャッシュ統計を収集する場合:

```
shell> zabbix_proxy -R diaginfo=historycache
```

ランタイムコントロールを使用してSNMPキャッシュをリロードする例:

```
shell> zabbix_proxy -R snmp_cache_reload
```

ランタイムコントロールを使用してハウスキーパーの実行をトリガーする例:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R housekeeper_execute
```

ランタイムコントロールを使用してログレベルを変更する例:

すべてのプロセスのログレベルを上げる場合:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase
```

2番目のpollerプロセスのログレベルを上げる場合:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2
```

PID1234のプロセスのログレベルを上げる場合:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234
```

すべてのhttp pollerプロセスのログレベルを下げる場合:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

プロセスユーザー

Zabbix プロキシは root 以外のユーザーとして実行するように設計されていて、起動された root 以外のユーザーとして実行されます。したがって root 以外のユーザーで問題なくプロキシを起動できます。

'root' として実行しようとする、ハードコードされた 'zabbix' ユーザーに切り替わります。これはシステムに存在している必要があります。プロキシ設定ファイルの AllowRoot パラメータを変更した場合にのみ、「root」としてプロキシを実行できます。

設定ファイル

zabbix_proxy の詳細設定については[設定ファイルオプション](#)を参照してください。

Proxy process types

- availability manager - ホスト可用性更新プロセス
- configuration syncer - 設定データメモリ内キャッシュ管理プロセス
- data sender - プロキシデータ sender
- discoverer - デバイス発見プロセス
- heartbeat sender - プロキシハートビート sender
- history poller - データベース接続を必要とする計算、集約、内部チェックを処理するプロセス
- history syncer - 履歴 DB 書き込み
- housekeeper - 古い履歴データを削除するプロセス
- http poller - ウェブモニタリング poller
- icmp pinger - icmp ping チェック poller
- ipmi manager - IPMI poller マネージャー
- ipmi poller - IPMI チェック poller
- java poller - Java チェック poller
- odbc poller - ODBC チェック poller
- poller - バッブチェック用通常ポーラー
- preprocessing manager - 前処理タスクマネージャー
- preprocessing worker - データ前処理プロセス
- self-monitoring - 内部サーバー統計収集プロセス
- snmp trapper - SNMP traps のトラッパー
- task manager - 他のコンポーネントによって要求されたタスクのリモート実行プロセス (例: 問題のクローズ、問題の確認、現在のアイテム値の確認、リモートコマンド機能)
- trapper - アクティブチェック、traps、プロキシ通信用のトラッパー
- unreachable poller - 使用不能デバイスの poller
- vmware collector - VMware サービスからのデータ収集を担当する VMware データコレクタ

プロキシログファイルを使用して、これらのプロセスタイプを監視できます。

zabbix[process,<type>,<mode>,<state>] 内部アイテムを使用して、さまざまなタイプの Zabbix プロキシプロセスを監視できます。

サポートされているプラットフォーム

Zabbix プロキシは、Zabbix サーバーと同じサーバーがサポートしているプラットフォームのリストに記載されているプラットフォームで実行可能です。

ロケール

一部のテキスト項目を正しく解釈できるように、プロキシには UTF-8 ロケールが必要です。最近のほとんどの Unix ライクなシステムには、デフォルトで UTF-8 ロケールがありますが、それを具体的に設定する必要があるシステムもあります。

5 Zabbix Java ゲートウェイ

概要

JMX アプリケーションを監視するためのネイティブサポートは、“Zabbix Java ゲートウェイ”と呼ばれる Zabbix デモン形式で存在し、Zabbix 2.0 以降で利用可能になりました。Zabbix Java ゲートウェイは、Java で書かれたデモンです。ホスト上の特定の JMX カウンター値を見つけるために、Zabbix サーバーは Zabbix Java ゲートウェイでクエリを実行します。Zabbix Java ゲートウェイは、[JMX 管理 API](#) を使用して対象のアプリケーションヘリモートでクエリを実行します。アプリケーションに追加のソフトウェアをインストールする必要はありません。コマンドラインで `-Dcom.sun.management.jmxremote` オプションを使用して起動するだけです。

Java ゲートウェイは、Zabbix サーバーまたはプロキシからの着信接続を受け入れ、“パッシブプロキシ”としてのみ使用できます。Zabbix プロキシとは違い、Zabbix プロキシからも使用できます。(Zabbix プロキシは zabbix プロキシにチェーンできません) 各 Java ゲートウェイへのアクセスは、Zabbix サーバーまたはプロキシ設定ファイルで直接設定されるため、Zabbix サーバーまたは Zabbix プロキシごとに設定できる Java ゲートウェイは 1 つだけです。ホストに **JMX** エージェントタイプのアイテムとその他のタイプのアイテムがある場合、**JMX** エージェントアイテムのみが取得のために Java ゲートウェイに渡されます。

Java ゲートウェイ経由でアイテムを更新する必要がある場合、Zabbix サーバーまたはプロキシは Java ゲートウェイに接続して値を要求し、Java ゲートウェイはその値を取得してサーバーまたはプロキシに返します。そのため、Java ゲートウェイは値をキャッシュしません。

Zabbix サーバーまたはプロキシには、オプション `*StartJavaPollers` によって制御される、**Java** ゲートウェイに接続する特定のタイプのプロセスがあります。内部的には、**Java** ゲートウェイは `START_POLLERS` オプションによって制御される複数のスレッドを開始します。サーバー側では、接続に `Timeout` 秒以上かかると終了しますが、**Java** ゲートウェイは **JMX** カウンターから値を取得するためにまだビジーである可能性があります。これを解決するために、**Java** ゲートウェイには `TIMEOUT**` オプションがあり、**JMX** ネットワーク操作のタイムアウトを設定できます。

Zabbix サーバーまたはプロキシは、単一の JMX ターゲットへのリクエストを可能な限りプールし(項目間隔の影響を受けます)、パフォーマンスを向上させるためにそれらを単一の接続で Java ゲートウェイに送信しようとします。

StartJavaPollers を `START_POLLERS` 以下にすることをお勧めします。そうしないと、Java ゲートウェイで着信要求を処理するスレッドを使用できない状況が発生する可能性があります。このような場合、Java ゲートウェイは `ThreadPoolExecutor.CallersRunsPolicy` を使用します。これは、メインスレッドが着信要求を処理し、一時的に新しい要求を受け入れないことを意味します。

When an item has to be updated over Java gateway, Zabbix server or proxy will connect to the Java gateway and request the value, which Java gateway in turn retrieves and passes back to the server or proxy. As such, Java gateway does not cache any values.

Zabbix server or proxy has a specific type of processes that connect to Java gateway, controlled by the option **StartJavaPollers**. Internally, Java gateway starts multiple threads, controlled by the **START_POLLERS** option. On the server side, if a connection takes more than **Timeout** seconds, it will be terminated, but Java gateway might still be busy retrieving value from the JMX counter. To solve this, there is the **TIMEOUT** option in Java gateway that allows to set timeout for JMX network operations.

Zabbix server or proxy will try to pool requests to a single JMX target together as much as possible (affected by item intervals) and send them to the Java gateway in a single connection for better performance.

It is suggested to have **StartJavaPollers** less than or equal to **START_POLLERS**, otherwise there might be situations when no threads are available in the Java gateway to service incoming requests; in such a case Java gateway uses `ThreadPoolExecutor.CallersRunsPolicy`, meaning that the main thread will service the incoming request and temporarily [label](#) will not accept any new requests.

If you are trying to monitor Wildfly-based Java applications with Zabbix Java gateway, please install the latest `jboss-client.jar` available on the [Wildfly download page](#).

Java ゲートウェイの取得

[Zabbix Web サイト](#) からダウンロードしたソースまたはパッケージから Java ゲートウェイをインストールできます。

以下のリンクを使用すると、Zabbix Java ゲートウェイを取得して実行する方法、JMX 監視に Zabbix Java ゲートウェイを使用するよう Zabbix サーバー (または Zabbix プロキシ) を設定する方法、および特定の JMX カウンターに対応する Zabbix フロントエンドで Zabbix アイテムを設定する方法に関する情報にアクセスできます。

インストール元	手順	手順
ソース	インストール	セットアップ
RHEL/CentOS パッケージ	インストール	セットアップ
Debian/Ubuntu パッケージ	インストール	セットアップ

1 ソースからのセットアップ

概要

次の情報はソースでインストールする場合の Zabbix Java ゲートウェイセットアップに役立ちます。

ファイルの概要

ソースから Java ゲートウェイを取得した場合、シェルスクリプト、JAR、および設定ファイルのコレクションが \$PREFIX/sbin/zabbix_java の下にあるはずですが、これらのファイルの役割を以下にまとめます。

bin/zabbix-java-gateway-\$VERSION.jar

Java ゲートウェイ JAR ファイル本体

lib/logback-core-0.9.27.jar
lib/logback-classic-0.9.27.jar
lib/slf4j-api-1.6.1.jar
lib/android-json-4.3_r3.1.jar

Java ゲートウェイの依存関係: [Logback](#)、[SLF4J](#)、および [Android JSON](#) ライブラリ。

lib/logback.xml
lib/logback-console.xml

Logback の構成ファイル。

shutdown.sh
startup.sh

Java ゲートウェイを開始および停止するための便利なスクリプト。

settings.sh

上記の起動スクリプトとシャットダウンスクリプトによって提供される設定ファイル。

Java ゲートウェイの設定と実行

デフォルトでは、Java ゲートウェイはポート 10052 でリスンします。別のポートで Java ゲートウェイを実行する予定がある場合は、settings.sh スクリプトで指定できます。このオプションやその他のオプションの指定方法については、[Java ゲートウェイ設定ファイルの説明](#)を参照してください。

Warning:

ポート 10052 は [IANA registered] (<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>) ではありません。

設定完了後、スタートアップスクリプトを実行して Java ゲートウェイを起動できます。

```
$ ./startup.sh
```

同様に Java ゲートウェイが不要になったら、シャットダウンスクリプトを実行して停止します。

```
$ ./shutdown.sh
```

サーバーやプロキシとは異なり、Java ゲートウェイは軽量であり、データベースを必要としません。

Java ゲートウェイで使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これは [サーバー設定ファイル](#) で JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりに [プロキシ設定ファイル](#) で接続パラメータを指定します。

```
JavaGateway=192.168.3.14
```

```
JavaGatewayPort=10052
```

デフォルトでは、サーバーは JMX モニタリングに関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Java ゲートウェイに問題がある場合、またはフロントエンドのアイテムについて表示されるエラーメッセージが十分でない場合は、Java ゲートウェイのログファイルを確認することをお勧めします。

デフォルトでは、Java ゲートウェイはそのアクティビティを /tmp/zabbix_java.log ファイルにログレベル"info" で記録します。ですがその情報だけでは不十分な場合、ログレベル"デバッグ" の情報を取得する必要があります。ログレベルを上げるには、lib/logback.xml ファイルを変更し、<root> タグの level 属性を"debug" に変更します。

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Zabbix サーバーや Zabbix プロキシとは異なり、logback.xml ファイルを変更した後に Zabbix Java ゲートウェイを再起動する必要はありません。logback.xml の変更は自動的に取得されます。デバッグが完了したら、ログレベルを"info" に戻すことができます。

別のファイルまたはデータベースのようなまったく別のメディアにログを記録する場合は、必要に応じて logback.xml ファイルを調整します。詳細については、[Logback マニュアル](#) を参照してください。

デバッグ目的の場合、デーモンではなくコンソールアプリケーションとして Java ゲートウェイを起動すると便利な場合があります。これを行うには、settings.sh で PID_FILE 変数をコメントアウトします。PID_FILE を省略した場合、startup.sh スクリプトは Java ゲートウェイをコンソールアプリケーションとして起動し、代わりに Logback に lib/logback-console.xml ファイルを使用させます。これにより、コンソールにログが記録されるだけでなく、ログレベル"debug" も有効になります。

最後に、Java ゲートウェイはロギングに SLF4J を使用するため、適切な JAR ファイルを lib ディレクトリに配置することで、Logback を選択したフレームワークに置き換えることができます。詳細については、[SLF4J マニュアル](#) を参照してください。

JMX モニタリング

詳しくは[JMX モニタリングページ](#)をご覧ください。

2 RHEL パッケージでのセットアップ

Overview

If **installed** from RHEL packages, the following information will help you in setting up Zabbix **Java gateway**.

Configuring and running Java gateway

Configuration parameters of Zabbix Java gateway may be tuned in the file:

```
/etc/zabbix/zabbix_java_gateway.conf
```

For more details, see Zabbix Java gateway configuration **parameters**.

To start Zabbix Java gateway:

```
# service zabbix-java-gateway restart
```

To automatically start Zabbix Java gateway on boot:

RHEL 7 and later:

```
# systemctl enable zabbix-java-gateway
```

RHEL prior to 7:

```
# chkconfig --level 12345 zabbix-java-gateway on
```

Configuring server for use with Java gateway

With Java gateway up and running, you have to tell Zabbix server where to find Zabbix Java gateway. This is done by specifying JavaGateway and JavaGatewayPort parameters in the **server configuration file**. If the host on which JMX application is running is monitored by Zabbix proxy, then you specify the connection parameters in the **proxy configuration file** instead.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

By default, server does not start any processes related to JMX monitoring. If you wish to use it, however, you have to specify the number of pre-forked instances of Java pollers. You do this in the same way you specify regular pollers and trappers.

```
StartJavaPollers=5
```

Do not forget to restart server or proxy, once you are done with configuring them.

Debugging Java gateway

Zabbix Java gateway log file is:

```
/var/log/zabbix/zabbix_java_gateway.log
```

If you like to increase the logging, edit the file:

```
/etc/zabbix/zabbix_java_gateway_logback.xml
```

and change level="info" to "debug" or even "trace" (for deep troubleshooting):

```
<configuration scan="true" scanPeriod="15 seconds">
[...]  
  <root level="info">  
    <appender-ref ref="FILE" />  
  </root>  
  
</configuration>
```

JMX モニタリング

詳しくは[JMX モニタリングページ](#)をご覧ください。

3 Debian/Ubuntu パッケージでのセットアップ

概要

次の情報は Debian/Ubuntu パッケージでインストールする場合の Zabbix Java ゲートウェイ セットアップに役立ちます。

Java ゲートウェイの設定と実行

Java ゲートウェイ設定は、次のファイルで調整できます。

```
/etc/zabbix/zabbix_java_gateway.conf
```

詳細については Zabbix Java ゲートウェイ設定 [パラメーター](#) を参照してください。

Zabbix Java ゲートウェイを起動するには:

```
# service zabbix-java-gateway restart
```

サーバー起動時に Zabbix Java ゲートウェイを自動的に起動するには:

```
# systemctl enable zabbix-java-gateway
```

Java ゲートウェイで使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これは [サーバー設定ファイル](#) で JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりに [プロキシ設定ファイル](#) で接続パラメータを指定します。

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

デフォルトでは、サーバーは JMX モニタリングに関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポラーのインスタンスの数を指定する必要があります。これは通常のポラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです。

```
/var/log/zabbix/zabbix_java_gateway.log
```

ログの内容を増やしたい場合は、下記ファイルを編集します。

```
/etc/zabbix/zabbix_java_gateway_logback.xml
```

level="info" を "debug" または "trace" に変更します (詳細なトラブルシューティング用):

```
<configuration scan="true" scanPeriod="15 seconds">
[...]
```

```
  <root level="info">
    <appender-ref ref="FILE" />
  </root>
</configuration>
```

JMX モニタリング

詳しくは [JMX モニタリングページ](#) をご覧ください。

6 zabbix_sender コマンド

概要

Zabbix sender はパフォーマンスデータを Zabbix サーバーに送信して処理するために使用できるコマンドラインユーティリティです。

このユーティリティは通常、可用性とパフォーマンスのデータを定期的に送信するために、長時間実行されるユーザースクリプトで使用されます。

結果を Zabbix サーバーまたはプロキシに直接送信するには **trapper item** タイプを設定する必要があります。

Zabbix sender の実行

Zabbix UNIX sender の実行例:

```
shell> cd bin
shell> ./zabbix_sender -z zabbix -s "Linux DB3" -k db.connections -o 43
```

解説:

- z - Zabbix サーバーのホスト (IP アドレスも使用可能)
- s - 監視対象ホストの名称 (Zabbix フロントエンドに登録されているもの)
- k - アイテムキー
- o - 送信値

Attention:

空白を含むオプションは、ダブルクォーテーションで囲む必要があります。

Zabbix sender を使用して入力ファイルから複数の値を送信できます。詳細については、[Zabbix sender のマニュアルページ](#) を参照してください。

設定ファイルが指定されている場合、Zabbix sender はエージェントの ServerActive 設定パラメータで定義されているすべてのアドレスを使用してデータを送信します。バッチデータの送信が 1 つのアドレスへの送信で失敗した場合、他のアドレスへの送信を試み、次のバッチはこのアドレスに送信されません。

Zabbix sender はファイルの最初にバイト順マーク (BOM) がない UTF-8 エンコーディング (UNIX ライクなシステムと Windows の両方) の文字列を受け入れます。

Windows 上の Zabbix sender も同様に実行できます。

```
zabbix_sender.exe [options]
```

Zabbix 1.8.4 以降、zabbix_sender のリアルタイム送信シナリオが改善され、渡された複数の値が連続して収集され、単一の接続でサーバーに送信されるようになりました。前の値から 0.2 秒以上離れていない値を同じスタックに入れることができますが、最大ブーリング時間は 1 秒のままです。

Note:

指定された設定ファイルに無効な (parameter=value 表記に従わない) パラメータエントリが存在する場合、Zabbix sender は終了します。

7 zabbix_get コマンド

概要

zabbix_get コマンドは、Zabbix エージェントと通信して、エージェントから必要な情報を取得するために使用できるコマンドラインユーティリティです。

このユーティリティは通常、Zabbix エージェントのトラブルシューティングに使用されます。

zabbix_get コマンドの実行

UNIX で zabbix_get コマンドを実行して、エージェントからプロセッサのロードアベレージを取得する例：

```
shell> cd bin
shell> ./zabbix_get -s 127.0.0.1 -p 10050 -k system.cpu.load[all,avg1]
```

zabbix_get コマンドを実行して Web サイトから文字列をキャプチャする別の例：

```
shell> cd bin
shell> ./zabbix_get -s 192.168.1.1 -p 10050 -k "web.page.regex[www.example.com,,,\"USA: ([a-zA-Z0-9.-]+)\\"
```

ここのアイテムキーにはスペースが含まれているため、シェルにアイテムキーを引き渡すためにコーテーションが使用されていることに注意してください。コーテーションはアイテムキーの一部ではありません。それらはシェルによってトリミングされ、Zabbix エージェントに渡されません。

zabbix_get コマンドは、次のコマンドラインパラメーターを受け入れます：

-s --host <host name or IP>	ホスト名またはホストのIPアドレスを指定します。
-p --port <port number>	ホスト上で動作しているエージェントのポート番号を指定します。デフォルト
-I --source-address <IP address>	ソースのIPアドレスを指定します。
-t --timeout <seconds>	タイムアウトを指定します。有効な範囲：1-30 秒（デフォルト：30 秒）
-k --key <item key>	値を取得したいアイテムのキーを指定します。
-h --help	ヘルプを表示します。
-V --version	バージョン番号を表示します。

詳細は [zabbix_get コマンドのマニュアル](#) を参照してください。

Windows 上で zabbix_get コマンドは同様に実行できます：

```
zabbix_get.exe [options]
```

8 zabbix_js コマンド

概要

zabbix_js は、埋め込みスクリプトのテストに使用できるコマンドラインユーティリティです。

このユーティリティは、文字列パラメータを使用してユーザースクリプトを実行し結果を出力します。スクリプトは、埋込みの zabbix スクリプトエンジンを使用して実行されます。

コンパイルまたは実行エラーの場合、zabbix_js はエラーを stderr に出力し、コード 1 で終了します。

使用法

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

zabbix_js は、次のコマンドラインパラメーターを使用可能です。

-s, --script script-file	実行するスクリプトのファイル名を指定します。 '-'がファイル名として指定
-i, --input input-file	入力パラメーターのファイル名を指定してください。ファイル名として「-」
-p, --param input-param	入力パラメーターを指定
-l, --loglevel log-level	ログレベル指定
-t, --timeout timeout	タイムアウトを秒単位で指定
-h, --help	ヘルプの表示
-V, --version	バージョン番号の表示

例:

```
zabbix_js -s script-file.js -p example
```

9 Zabbix Web サービス

概要

Zabbix Web サービスは、外部 Web サービスとの通信に使用されるプロセスです。定期レポートの生成および送信のためにも使用されます。

Zabbix サーバーは HTTP(S) 経由で Web サービスに接続します。Zabbix Web サービスは GoogleChrome が同じホストにインストールされている必要があります。一部のディストリビューションでは Chromium でも機能する場合があります ([既知の問題](#))

インストール

Zabbix Web サービスは [Zabbix Web サイト](#) からダウンロードできるコンパイル済みの Zabbix パッケージで利用できます。ソースから [Zabbix Web サービス](#) をコンパイルするには `--enable-webservice` 設定オプションを指定します。

参照:

- [Zabbix web サービスの設定ファイルオプション](#)
- [定期レポートの設定](#)

4. インストール

サイドバーからインストールセクションのコンテンツにアクセスしてください。

1 Zabbix の入手

概要

Zabbix を入手するには 4 つの方法があります。

- [配布パッケージ](#)からのインストール
- 最新のソースアーカイブをダウンロードして [自分でコンパイル](#)
- [コンテナ](#)でインストール
- [仮想アプライアンス](#)をダウンロード

最新の配布パッケージ、コンパイル済みのソース、または仮想アプライアンスについては、[Zabbix ダウンロードページ](#)にアクセスしてください。最新バージョンへのリンクが掲示されています。

Zabbix ソースコードの取得

Zabbix ソースコードを取得する方法はいくつかあります。

- Zabbix の公式ウェブサイトから安定版リリースを [ダウンロード](#)できます
- Zabbix の公式ウェブサイト開発者ページからナイトリービルドを [ダウンロード](#)できます
- Git ソースコードリポジトリから最新の開発バージョンを入手できます
 - 完全なリポジトリの場所 <https://git.zabbix.com/scm/zbx/zabbix.git>
 - マスターリリースとサポートされているリリースも Github にミラーリングされています <https://github.com/zabbix/zabbix>

リポジトリのクローンを作成するには、Git クライアントをインストールする必要があります。公式のコマンドライン Git クライアントパッケージは、一般的に **git** と呼ばれます。たとえば、Debian / Ubuntu にインストールするには、次のコマンドを実行します。

```
sudo apt-get update
sudo apt-get install git
```

すべての Zabbix ソースを取得するには、配置するディレクトリに移動し、次のコードを入力して実行します。

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

2 要件

ハードウェア要件

メモリ要件

Zabbix には、物理メモリとディスク容量の両方が必要です。必要なディスク容量は、監視対象のホストとパラメータの数によって異なります。監視対象のパラメータの履歴を長期間保存することを計画している場合は、データベースに履歴を保存するのに十分なスペースとして、少なくとも数ギガバイトを確保することを検討する必要があります。各 Zabbix デモンプロセスには、データベースサーバーへの複数の接続が必要です。接続に割り当てられるメモリの量は、データベースエンジンの構成によって異なります。

Note:

物理メモリが多ければ多いほど、データベース (したがって Zabbix) の動作が速くなります!

CPU 要件

Zabbix、特に Zabbix データベースは、監視するパラメータの数と選択したデータベースエンジンによっては、かなりの CPU リソースを必要とする場合があります。

その他のハードウェア要件

Zabbix で SMS 通知サポートを使用するには、シリアル通信ポートとシリアル GSM モデムが必要です。USB からシリアルへのコンバーターも同様に有効です。

ハードウェア構成例

この表は **Linux/BSD/Unix** プラットフォームを想定したハードウェア構成の例を示しています。

まずはサイズとハードウェア構成の例です。Zabbix のインストールはそれぞれ固有のものです。ステージング環境または開発環境で Zabbix システムのパフォーマンスをベンチマークして、Zabbix インストールを本番環境に展開する前に要件を完全に理解できるようにしてください。

インストールサイズ	監視メトリック ¹	CPU/vCPU コア	メモリ (GiB)	データベース	Amazon EC2 ²
小	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
中	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
大	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL, Oracle	m6i.4xlarge/m6g.4xlarge
特大	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL, Oracle	m6i.8xlarge/m6g.8xlarge

¹ 1 メトリック = 1 アイテム + 1 トリガー + 1 グラフ
 ² ARM64 または x86_64 アーキテクチャを使用する Amazon 汎用 EC2 インスタンスの例では、本番環境にインストールする前に、Zabbix のインストール評価およびテスト中にコンピュート/メモリ/ストレージの最適化などの適切なインスタンスタイプを選択する必要があります。

Note:

実際の構成は、アクティブなアイテムの数とリフレッシュレートに大きく依存します (詳細については、このページの**データベースのサイジング** セクションを参照してください)。大規模なインストールの場合は、データベースを別のサーバーで実行することを強くお勧めします。

サポートするプラットフォーム

セキュリティ要件と監視サーバーのミッションクリティカルな性質から、UNIX は必要な性能と耐障害性、回復力を一貫して提供できる唯一の OS です。Zabbix は市場をリードするバージョンで動作します。

Zabbix のコンポーネントは以下のプラットフォームで動作確認されています：

プラットフォーム	サーバー	エージェント	エージェント 2
Linux	x	x	x
IBM AIX	x	x	-
FreeBSD	x	x	-
NetBSD	x	x	-
OpenBSD	x	x	-
HP-UX	x	x	-
Mac OS X	x	x	-
Solaris	x	x	-
Windows	-	x	x

Note:

Zabbix サーバー/エージェントは他の Unix 系 OS でも動作する可能性があります。

Zabbix エージェントは Windows XP 以降の全ての Windows デスクトップとサーバーの OS バージョンでサポートされています。

Zabbix エージェント 2 の重大なセキュリティ脆弱性をために、it is compiled only with **サポートされている Go リリース**のみでコンパイルされています。Go 1.21 では、**Windows バージョンの最小要件**が引き上げられています。したがって、Zabbix 6.0.27 以降、Zabbix エージェント 2 における Windows バージョンの最小要件は Windows 10/Server 2016 です。

Attention:

Zabbix は、暗号化してコンパイルされている場合はコアダンプを無効にし、システムがコアダンプの無効化を許可していない場合は起動しません。

ソフトウェア要件

Zabbix は最新の Web サーバー、主要なデータベースエンジン、および PHP スクリプト言語を組み合わせで動作します。

サードパーティの外部ソフトウェア

必須ステータスと記載されているソフトウェア/ライブラリは常に必要です。オプションのものは、特定の機能をサポートするのに必要です。

ソフトウェア	必須ステータス	サポートされているバージョン	コメント
MySQL/Percona	どれか一つ	8.0.30-8.3.X	MySQL(または Percona) を Zabbix バックエンドデータベースとして使用する場合に必要です。InnoDB エンジンが必要です。 MySQL のサポートバージョンの追加: - 8.1 since Zabbix 6.0.21; - 8.2 since Zabbix 6.0.26; - 8.3 since Zabbix 6.0.27. サーバー/プロキシの構築には C API (libmysqlclient) ライブラリを使用することをお勧めします。 MySQL バージョン 8.0.0-8.0.28 については、 Possible deadlocks を参照してください。

ソフトウェア	必須ステータス	サポートされているバージョン	コメント
MariaDB		10.5.00-11.3.X	<p>InnoDB エンジンが必要です。</p> <p>バージョンは 10.5 をお勧めします。</p> <p>サーバー/プロキシの構築には MariaDB Connector/C ライブラリを使用することをお勧めします。</p> <p>MariaDB のサポートバージョンの追加:</p> <ul style="list-style-type: none"> - 10.7.X since Zabbix 6.0.6; - 10.8.X since Zabbix 6.0.7; - 10.9.X and 10.10.X since Zabbix 6.0.13; - 10.11.X since Zabbix 6.0.15; - 11.0.X since Zabbix 6.0.21; - 11.1.X since Zabbix 6.0.23; - 11.2.X since Zabbix 6.0.27; - 11.3.X since Zabbix 6.0.29.
Oracle		19c - 21c	<p>Possible deadlocks もご覧ください。</p> <p>Oracle を Zabbix バックエンドデータベースとして使用する場合に必要です。</p>
PostgreSQL		13.0-16.X	<p>PostgreSQL を Zabbix バックエンドデータベースとして使用する場合に必要です。</p> <p>インストールサイズに応じて、PostgreSQL の work_mem の構成プロパティを増やす必要があるかもしれません (デフォルト値は 4MB)。これにより、データベースが特定の操作に使用するメモリの量が十分になり、クエリの実行に時間がかかりすぎなくなります。</p> <p>PostgreSQL のサポートバージョンの追加:</p> <ul style="list-style-type: none"> - 15.X since Zabbix 6.0.10; - 16.X since Zabbix 6.0.23.
TimescaleDB for PostgreSQL		2.1.0-2.14.X	<p>TimescaleDB を PostgreSQL データベース拡張機能として使用する場合に必要です。圧縮をサポートする TimescaleDB Community Edition を必ずインストールしてください。</p> <p>PostgreSQL 15 は TimescaleDB 2.10 以降でサポートされていることに注意してください。</p> <p>PostgreSQL と TimescaleDB のバージョンの互換性に関する詳細については、公式ドキュメント を参照することもできます。</p> <p>TimescaleDB のサポートバージョンの追加:</p> <ul style="list-style-type: none"> - 2.6 since Zabbix 6.0.7; - 2.7 since Zabbix 6.0.8; - 2.8 since Zabbix 6.0.10; - 2.9 since Zabbix 6.0.13; - 2.10 since Zabbix 6.0.15; - 2.11 since Zabbix 6.0.19; - 2.12 since Zabbix 6.0.24; - 2.13 since Zabbix 6.0.25; - 2.14 since Zabbix 6.0.28.
SQLite	オプション	3.3.5-3.34.X	<p>SQLite は Zabbix プロキシでのみサポートされています。SQLite を Zabbix プロキシデータベースとして使用する場合に必要です。</p>
smartmontools		7.1 以降	<p>Zabbix エージェント 2 に必要です。</p>
who			<p>ユーザーカウントプラグインに必要です。</p>
dpkg			<p>system.sw.packages プラグインに必要です。</p>
pkgtool			<p>system.sw.packages プラグインに必要です。</p>
rpm			<p>system.sw.packages プラグインに必要です。</p>
pacman			<p>system.sw.packages プラグインに必要です。</p>

Note:

Zabbix はオペレーティングシステムで利用可能なデータベースで動作しますが、最高のエクスペリエンスを得るには、公式のデータベース開発者リポジトリからインストールされたデータベースを使用することをお勧めします。

Web インターフェース

Zabbix の Web インターフェースでサポートされる最小画面幅は 1200px です。

必須要件は常に必要です。特定の機能をサポートするには、オプションの要件が必要です。

ソフトウェア	必須ステータス	バージョン	コメント
Apache Nginx PHP	どれか一つ Yes	2.4 以降 1.20 以降 7.2.5 以降、 8.0-8.3	PHP 7.4 以降の使用を推奨します。 PHP のサポートバージョンの追加: - 8.0 and 8.1 since Zabbix 6.0.6; - 8.2 since Zabbix 6.0.14; - 8.3 since Zabbix 6.0.26.
PHP extensions: gd	Yes	2.0.28 以降	PHP GD 拡張機能は、PNG 画像 (--with-png-dir)、JPEG 画像 (--with-jpeg-dir)、および FreeType 2 (--with-freetype-dir) をサポートする必要があります。 Web インターフェースで選択した言語によっては、 グラフ内でテキストが重なるのを避けるために 、バージョン 2.3.0 以降が必要になります。
bcmath ctype libXML		2.6.15 以降	php-bcmath (--enable-bcmath) php-ctype (--enable-ctype) ディストリビュータによって個別のパッケージとして提供されている場合は、php-xml
xmlreader			ディストリビュータによって個別のパッケージとして提供されている場合は、php-xmlreader
xmlwriter			ディストリビュータによって個別のパッケージとして提供されている場合は、php-xmlwriter
session			ディストリビュータによって個別のパッケージとして提供されている場合は、php-session
sockets			php-net-socket (--enable-sockets)。ユーザースクリプトのサポートに必要です。
mbstring gettext ldap	No		php-mbstring (--enable-mbstring) php-gettext (--with-gettext)。翻訳機能のために必要です。 php-ldap。Web インターフェースで LDAP 認証が使用されている場合にのみ必要です。
openssl			php-openssl。Web インターフェースで SAML 認証が使用されている場合にのみ必要です。
mysqli			MySQL を Zabbix バックエンドデータベースとして使用する場合に必要です。
oci8			Oracle を Zabbix バックエンドデータベースとして使用する場合に必要です。
pgsql			PostgreSQL を Zabbix バックエンドデータベースとして使用する場合に必要です。

Zabbix で提供されるサードパーティの Web インターフェースライブラリ:

ライブラリ	必須ステータス	最小バージョン	コメント
jQuery JavaScript Library	Yes	3.6.0	クロスブラウザ開発のプロセスを簡素化する JavaScript ライブラリ
jQuery UI		1.12.1	jQuery 上に構築された一連のユーザーインターフェイスインタラクション、エフェクト、ウィジェット、およびテーマ。
OneLogin's SAML PHP Toolkit		3.4.1	Zabbix にサインインできるようにするための SAML 2.0 認証サポートを追加する PHP ツールキット。

ライブラリ	必須ステータス	最小バージョン	コメント
Symfony Yaml Component		5.1.0	YAML 形式で Zabbix 構成要素をエクスポートおよびインポートするためのサポートを追加。

Note:

Zabbix は、以前のバージョンの Apache、MySQL、Oracle、および PostgreSQL でも動作する可能性があります。

Attention:

デフォルトの DejaVu 以外のフォントの場合、PHP 関数 `imagerotate` が必要になる場合があります。不足している場合、グラフが表示されたときにこれらのフォントが正しくレンダリングされない可能性があります。この関数は、PHP がバンドルされた GD でコンパイルされている場合にのみ使用できます。これは、Debian やその他のディストリビューションには当てはまりません。

Zabbix の Web インターフェースのコード作成とデバッグに使用されるサードパーティライブラリ:

ライブラリ	必須ステータス	最小バージョン	説明
Composer	No	2.4.1	PHP ソフトウェアと必要なライブラリの依存関係を管理するための標準形式を提供する、PHP 用アプリケーションレベルパッケージマネージャー。
PHPUnit		8.5.29	Zabbix の Web インターフェースをテストするための PHP 単体テストフレームワーク。
SASS		3.4.22	カスタマイズスタイルシート (CSS) にコンパイルされるプリプロセッサスクリプト言語。

クライアント側の Web ブラウザ

Cookie と JavaScript を有効にする必要があります。

Google Chrome、Mozilla Firefox、Microsoft Edge、Apple Safari、Opera の最新安定バージョンに対応しています。

Warning:

IFrame の same-origin ポリシーが実装されました。つまり、Zabbix を別ドメインのフレームに配置することはできません。

それでも、フレームに配置されたページと Zabbix の Web インターフェースが同じドメインにある場合、Zabbix フレームに配置されたページは (JavaScript を介して) Zabbix の Web インターフェースにアクセスできます。
<http://secure-zabbix.com/cms/page.html> のようなページを <http://secure-zabbix.com/zabbix/> のダッシュボードに配置すると、JS から Zabbix に完全にアクセスすることができます。

サーバー/プロキシ

必須要件は常に必須です。特定の機能をサポートするには、オプションの要件が必要です。

要件	必須ステータス	説明
<code>libpcre/libpcre2</code>	どれか一つ	Perl 互換正規表現 (PCRE) のサポートには PCRE/PCRE2 ライブラリが必要です。GNU/Linux ディストリビューションによって、'libpcre3'、'libpcre1' などと名前が異なる場合があります。PCRE v8.x および PCRE2 v10.x (Zabbix 6.0.0 以降) がサポートされています。
<code>libevent</code> <code>libpthread</code>	Yes	一括メトリックサポートと IPMI 監視に必要です。バージョン 1.4 以降。ミューテックスおよび読み取り/書き込みロックのサポートに必要です (libc の一部である可能性があります)。
<code>libresolv</code> <code>libiconv</code>		DNS 解決に必要です (libc の一部である可能性があります)。テキストのエンコード/フォーマット変換に必要です (libc の一部である可能性があります)。Linux 上の Zabbix サーバーには必須です。
<code>libz</code> <code>libm</code>		圧縮のサポートに必要です。Math ライブラリ。Zabbix サーバーでのみ必要です。
<code>libmysqlclient</code>	どれか一つ	MySQL を使用する場合は必要です。

要件	必須ステータス	説明
libmariadb		MariaDB を使用する場合は必要です。
libclntsh		Oracle を使用する場合は必要です。libclntsh のバージョンは、Oracle データベースのバージョン以上である必要があります。
libpq5		PostgreSQL を使用する場合は必要です。libpq5 のバージョンは、PostgreSQL データベースのバージョン以上である必要があります。
libsqlite3		Sqlite を使用する場合は必要です。Zabbix プロキシにのみです。
libOpenIPMI	No	IPMI のサポートに必要です。Zabbix サーバーのみです。
libssh2 or libssh		SSH チェック に必要です。バージョン 1.0 以降 (libssh2)、0.6.0 以降 (libssh)。 libssh は Zabbix 4.4.6 以降でサポートされています。
libcurl		Web 監視、VMware 監視、SMTP 認証、Zabbix エージェントの アイテム である <code>web.page.*</code> 、HTTP エージェントアイテム、Elasticsearch (使用する場合) に必要です。バージョン 7.28.0 以降を推奨します。 Libcurl のバージョン要件: - SMTP 認証: バージョン 7.20.0 以降 - Elasticsearch: バージョン 7.28.0 以降
libxml2		VMware 監視と XML XPath の前処理に必要
net-snmp		SNMP のサポートに必要です。バージョン 5.3.0 以降 強力な暗号化プロトコル (AES192/AES192C、AES256/AES256C) のサポートは、net-snmp ライブラリ 5.8 以降で利用可能です。RHEL 8+ ベースのシステムでは、net-snmp 5.8.15 以降を使用することをお勧めします。
libunixodbc		データベース監視に必要です。
libgnutls or libopenssl		暗号化 を使用する場合に必要です。 最小バージョン: libgnutls - 3.1.18、libopenssl - 1.0.1
libldap		LDAP のサポートに必要です。
fping		ICMP ping items に必要です。

エージェント

要件	必須ステータス	説明
libpcre/libpcre2	どれか一つ	Perl 互換正規表現 (PCRE) のサポートには、PCRE/PCRE2 ライブラリが必要です。 'libpcre3' や 'libpcre1' など、GNU/Linux ディストリビューションによって名前が異なる場合があります。PCRE v8.x および PCRE2 v10.x (Zabbix 6.0.0 以降) がサポートされています。 ログの監視に必要です。Windows でも必要です。
libpthread	Yes	ミューテックスおよび読み書きロックのサポートに必要です (libc の一部である可能性があります)。Windows では不要です。
libresolv		DNS 解決に必要です (libc の一部である可能性があります)。Windows では不要です。
libiconv		ログアイテム、ファイルコンテンツ、ファイル regex および regmatch アイテム (libc の一部である可能性があります) での UTF-8 へのテキストエンコーディング/フォーマット変換に必要です。Windows では不要です。
libgnutls or libopenssl	No	暗号化 を使用する場合に必要です。 最小バージョン: libgnutls - 3.1.18、libopenssl - 1.0.1 Microsoft Windows では OpenSSL 1.1.1 以降が必要です。
libldap		LDAP を使用する場合は必須です。Windows ではサポートされていません。
libcurl		<code>web.page.*</code> という Zabbix エージェントの アイテム に必要です。Windows ではサポートされていません。 バージョン 7.28.0 以降を推奨します。
libmodbus		Modbus モニタリングを使用する場合にのみ必要です。 バージョン 3.0 以降。

Note:

バージョン 5.0.3 以降の Zabbix エージェントは、バージョン 6.1 TL07 / AIX 7.1 TL01 より前の AIX プラットフォームでは動作しません。

エージェント 2

要件	必須ステータス	説明
libpcre/libpcre2	どちらか一つ	Perl 互換正規表現 (PCRE) のサポートには PCRE/PCRE2 ライブラリが必要です。名前は 'libpcre3' または 'libpcre1' など、GNU/Linux ディストリビューションによって異なる場合があります。PCRE v8.x および PCRE2 v10.x (Zabbix 6.0.0 以降) がサポートされています。 ログ監視に必要です。Windows でも必要です。
libopenssl	No	暗号化を使用する場合に必要です。 UNIX プラットフォームでは OpenSSL 1.0.1 以降が必要です。 OpenSSL ライブラリでは PSK サポートが有効になっている必要があります。 LibreSSL はサポートされていません。 Microsoft Windows システムでは、OpenSSL 1.1.1 以降が必要です。

Golang ライブラリ

要件	必須ステータス	最小バージョン	説明
git.zabbix.com/ap/plugin/support	Yes	1.X.X	Zabbix 独自のサポートライブラリ。主にプラグイン用です。
github.com/BurntSushi/locker		0.0.0	名前付き読み取り/書き込みロック、アクセス同期。
github.com/chromedp/cdproto		0.0.0	Chrome DevTools プロトコルドメイン用に生成されたコマンド、タイプ、およびイベント。
github.com/chromedp/chromedp		0.6.0	Chrome DevTools プロトコルのサポート (レポート生成)。
github.com/dustin/gomemcached		0.0.0	go 用の memcached バイナリプロトコルツールキット。
github.com/eclipse/paho.mqtt.golang		1.2.0	MQTT 接続を処理するためのライブラリ。
github.com/fsnotify/fsnotify		1.4.9	Go のクロスプラットフォームファイルシステム通知。
github.com/go-ldap/ldap		3.0.3	GO プログラミング言語の基本的な LDAP v3 機能。
github.com/go-ole/go-ole		1.2.4	Golang の Win32 ole 実装。
github.com/godbus/dbus		4.1.0	D-Bus のネイティブ Go バインディング。
github.com/go-sql-driver/mysql		1.5.0	MySQL ドライバ。
github.com/godror/godror		0.20.1	Oracle DB ドライバ。
github.com/mattn/go-sqlite3		2.0.3	Sqlite3 ドライバ。
github.com/mediocregopher/radix/v3		3.5.0	Redis クライアント。
github.com/memcachier/mc/v3		3.0.1	バイナリ Memcached クライアント。
github.com/miekg/dns		1.1.43	DNS ライブラリ。
github.com/omeid/go-yarn		0.0.1	埋め込み可能なファイルシステムにマッピングされたキー文字列ストア。
github.com/goburrow/modbus		0.1.0	Modbus のフォールトトレラントな実装。
golang.org/x/sys		0.0.0	オペレーティングシステムとの低レベルの対話用の Go パッケージ。プラグインサポートライブラリでも使用されます。MongoDB と PostgreSQL プラグインで使用されます。
github.com/Microsoft/go-winio	On Windows.	0.6.0	Windows の名前付きパイプの実装。プラグインサポートライブラリでも使用されます。MongoDB と PostgreSQL プラグインで使用されます。
github.com/goburrow/serial	Yes, indirect ¹	0.1.0	Modbus 用のシリアルライブラリ。
golang.org/x/errors		0.0.0	エラーを操作する関数。
gopkg.in/asn1-ber.v1		1.0.0	ASN1 BER のエンコード/デコードライブラリ。
github.com/go-stack/stack	No, indirect ¹	1.8.0	
github.com/golang/snappy		0.0.1	
github.com/klauspost/compress		1.13.6	
github.com/xdg-go/pbkdf2		1.0.0	

要件	必須ステータス	最小バージョン	説明
github.com/xdg-go/scram		1.0.2	
github.com/xdg-go/stringprep		1.0.2	
github.com/youmark/pkcs8		0.0.0	

¹ "indirect" は、エージェントが使用するライブラリの 1 つで使用されることを意味します。Zabbix はパッケージを使用するライブラリを使用するため必要です。

ロード可能なプラグインの依存関係も参照してください:

- PostgreSQL
- MongoDB

Java ゲートウェイ

ソースリポジトリまたはアーカイブから Zabbix を取得した場合、必要な依存関係は既にソースツリーに含まれています。

ディストリビューションのパッケージから Zabbix を取得した場合、必要な依存関係はパッケージングシステムによって既に提供されています。

上記のどちらの場合でも、ソフトウェアを使用する準備ができており、追加のダウンロードは必要ありません。

ただし、これらの依存関係のバージョンを提供したい場合 (たとえば、一部の Linux ディストリビューション用のパッケージを準備している場合)、Java ゲートウェイが動作することがわかっているライブラリバージョンのリストを以下に示します。Zabbix は、これらのライブラリの他のバージョンでも動作する場合があります。

次の表に、元のコードで現在 Java ゲートウェイにバンドルされている JAR ファイルを示します。

ライブラリ	必須ステータス	コメント
android-json	Yes	Version 4.3r1 以降 JSON (JavaScript Object Notation) の軽量データ交換フォーマットです。これは、Android SDK から抽出された org.json 互換の Android 実装です。
logback-classic		Version 1.2.9 以降
logback-core		Version 1.2.9 以降
slf4j-api		Version 1.7.32 以降

Java ゲートウェイは、Oracle Java またはオープンソースの OpenJDK (バージョン 1.6 以降) を使用して構築できます。Zabbix が提供するパッケージは、OpenJDK を使用してコンパイルされます。以下の表は、Zabbix パッケージのビルドに使用される OpenJDK のバージョンに関する情報をディストリビューションごとに示しています。

ディストリビューション	OpenJDK バージョン
RHEL 8	1.8.0
RHEL 7	1.8.0
SLES 15	11.0.4
SLES 12	1.8.0
Debian 10	11.0.8
Ubuntu 20.04	11.0.8
Ubuntu 18.04	11.0.8

デフォルトのポート番号

次のリストは各コンポーネントのデフォルトポート番号です。

Zabbix コンポーネント	ポート番号	プロトコル	接続の種類
Zabbix エージェント	10050	TCP	オンデマンド
Zabbix エージェント 2	10050	TCP	オンデマンド
Zabbix サーバー	10051	TCP	オンデマンド
Zabbix プロキシ	10051	TCP	オンデマンド
Zabbix Java ゲートウェイ	10052	TCP	オンデマンド

Zabbix コンポーネント	ポート番号	プロトコル	接続の種類
Zabbix web サービス	10053	TCP	オンデマンド
Zabbix Web インターフェース	80	HTTP	オンデマンド
	443	HTTPS	オンデマンド
Zabbix トラッパー	10051	TCP	オンデマンド

Note:

Zabbix 通信を有効にするには、ファイアウォールでポートを開く必要があります。通常、発信の TCP 接続には明示的なファイアウォール設定は必要ありません。

データベースのサイジング

Zabbix の設定データを保存するために一定量のディスク容量が必要ですが、ほとんど増えることはありません。

Zabbix データベースのサイズは、主に保存される履歴データの量に関する以下の数量によって決まります。

- 1 秒あたりの処理件数

Zabbix サーバが 1 秒間に受信する新しい値の平均数です。例えば、監視対象アイテムが 3000 件、更新速度が 60 秒の場合、1 秒あたりの値数は $3000/60 = 50$ 個になります。これは、毎秒 50 個の新しい値が Zabbix データベースに追加されることを意味します。

- ヒストリデータ用の Housekeeper の設定

Zabbix は一定期間 (通常は数週間または数ヶ月間) 収集した値を保存します。新しい値を追加するたびに、データとインデックス用に一定量のディスクスペースが消費されます。

そのため、30 日分の履歴を残したい場合、1 秒間に 50 個の値を受信すると考えると、総数としては $(30 \times 24 \times 3600) \times 50 = 129,600,000$ 、つまり約 1 億 3 千万個の値を受信することになります。

使用するデータベースエンジン、受信する値の種類 (浮動小数点数、整数、文字列、ログファイルなど) により、1 つの値を保持するためのディスク容量は、40 バイトから数百バイトまで変化します。通常、数値アイテムの場合、1 つの値につき 90 バイトです²。

この場合、約 1 億 3 千万個の値を保存するためには、約 1 億 3 千万個 \times 90 バイト = **10.9GB** のディスク容量が必要ということです。

Note:

テキスト/ログアイテムの値のサイズを正確に予測することはできませんが、1 つの値につき 500 バイト程度を見込んでおくといでしょう。

- トレンド用の Housekeeper の設定

Zabbix は各項目の 1 時間の最大/最小/平均/カウントのセットをテーブル **trends** に保持します。このデータはトレンド解析や長周期グラフに使用されます。この 1 時間の期間は変更できません。

Zabbix データベースは、データベースの種類にもよりますが、それぞれの集計値につき約 90 バイトの容量が必要です。

例えばトレンドデータを 5 年分保存する場合、3000 アイテムの値には、1 年あたり $3000 \times 24 \times 365 \times 90 = 2.2\text{GB}$ が必要で、5 年間では **11GB** となります。

- イベント用の Housekeeper の設定

Zabbix の各イベントは約 250 バイトのディスクスペースを必要とします¹。

Zabbix が毎日生成するイベント数を見積もることは困難です。最悪のケースでは、Zabbix は 1 秒間に 1 つのイベントを生成すると仮定することができます。回復したイベントごとに `event_recovery` レコードが作成されます。通常ほとんどのイベントは回復されるので、1 つのイベントにつき 1 つの `event_recovery` レコードを想定します。つまり、1 イベントあたり 80 バイトのレコードが追加されます。

オプションとして、イベントにはタグを付けることができ、各タグレコードは約 100 バイトのディスクスペースを必要とします¹。1 イベントあたりのタグの数 (`#tags`) は設定によります。

そのため、1 つのタグ (`#tags`) につき、さらに 100 バイトのディスクスペースが必要です。

つまり、3 年分のイベントを保存する場合、 $3 \times 365 \times 24 \times 3600 \times (250 + 80 + \#tags \times 100) = \sim 30\text{GB} + \#tags \times 100\text{B}$ のディスクスペース² が必要になります。

Note:

¹ 非 ASCII のイベント名、タグ、値を持つ場合、より多くなります。

² サイズの概算は MySQL に基づいており、他のデータベースでは異なる可能性があります。

この表には、Zabbix システムに必要なディスク容量の計算式が含まれています:

パラメータ名	必要なディスクスペースの計算式 (バイト単位)
Zabbix の設定 ヒストリ	固定サイズ。通常 10MB 以下。 $days \times (items / refresh\ rate) \times 24 \times 3600 \times bytes$ items : アイテムの数 days : ヒストリ保存日数 refresh rate : アイテムのリフレッシュレートの平均 bytes : 単一の値を保持するために必要なバイト数。データベースエンジンによって変わり、通常、~90 bytes。
トレンド	$days \times (items / 3600) \times 24 \times 3600 \times bytes$ items : アイテムの数 days : ヒストリ保存日数 bytes : 単一のトレンドを保持するために必要なバイト数。データベースエンジンによって変わり、通常、~90 bytes。
イベント	$days \times events \times 24 \times 3600 \times bytes$ events : 1 秒あたりのイベントの数。最悪の場合のシナリオでは、1 秒あたり 1 つのイベント。 days : ヒストリ保存日数 bytes : 単一のトレンドを保持するために必要なバイト数。データベースエンジンによって変わり、通常、~330 + イベント 1 つあたりのタグの平均数 × 100 bytes。

そのため、必要なディスク容量の合計は次のように計算されます:

設定 + ヒストリ + トレンド + イベント

算出したディスク容量は、Zabbix をインストールした直後から必要となるわけではありません。データベースサイズは最初は増え続けますが、ある時点で一定の値を維持するようになります。どの時点で一定になるかは、Housekeeper の設定に依存します。

時刻同期

Zabbix が動作しているサーバで正確なシステム時刻を得ることは非常に重要です。
 ntpd はホストの時刻を他のマシンの時刻と同期させる最も有名なデーモンです。
 Zabbix のコンポーネントが動作している全てのシステムで時刻を同期させることを強く推奨します。

プラグイン

1 PostgreSQL プラグインの依存関係

概要

PostgreSQL ローダブルプラグインに必要なライブラリは、このページに一覧表示されています。

Go 言語ライブラリ

要求	必須ステータス	最小バージョン	説明
git.zabbix.com/ap/plugin-support	必須	1.X.X	Zabbix 自身でサポートしているライブラリ。プラグイン用。
github.com/jackc/pgx/v4		4.17.2	PostgreSQL 用ドライバー。
github.com/omeid/go-yarn		0.0.1	埋め込み可能なファイルシステムにマッピングされたキー文字列スト
github.com/jackc/chunkreader/v2	間接的 ¹	2.0.1	
github.com/jackc/pgconn		1.13.0	
github.com/jackc/pgio		1.0.0	
github.com/jackc/pgpassfile		1.0.0	
github.com/jackc/pgproto3/v2		2.3.1	
github.com/jackc/pgservicefile		0.0.0	
github.com/jackc/pgtype		1.12.0	
github.com/jackc/puddle		1.3.0	
github.com/natefinch/npipe		0.0.0	
golang.org/x/crypto		0.0.0	
golang.org/x/sys		0.0.0	
golang.org/x/text		0.3.7	

¹ "間接的" とは、エージェントが使用するライブラリの 1 つで使用されることを意味します。Zabbix はパッケージを使用するライブラリを使用するので必須です。

2 MongoDB プラグインの依存関係

概要

MongoDB ローダブルプラグインに必要なライブラリは、このページに一覧表示されています。

Go 言語ライブラリ

要件	必須ステータス	最小バージョン	説明
git.zabbix.com/ap/plugin-support	必須	1.X.X	Zabbix 自身でサポートしているライブラリ。プラグインで使用。
go.mongodb.org/mongo-driver		1.7.6	名前による読み書きとロック、そしてアクセスの同期。
github.com/go-stack/stack	間接的 ¹	1.8.0	mongo-driver ライブラリに必要なパッケージ。
github.com/golang/snappy		0.0.1	mongo-driver ライブラリに必要なパッケージ。
github.com/klauspost/compress		1.13.6	mongo-driver ライブラリに必要なパッケージ。
github.com/natefinch/npipe		0.0.0	mongo-driver ライブラリに必要なパッケージ。
github.com/pkg/errors		0.9.1	mongo-driver ライブラリに必要なパッケージ。
github.com/xdg-go/pbkdf2		1.0.0	mongo-driver ライブラリに必要なパッケージ。
github.com/xdg-go/scram		1.0.2	mongo-driver ライブラリに必要なパッケージ。
github.com/xdg-go/stringprep		1.0.2	mongo-driver ライブラリに必要なパッケージ。
github.com/youmark/pkcs8		0.0.0	mongo-driver ライブラリに必要なパッケージ。
golang.org/x/crypto		0.0.0	mongo-driver ライブラリに必要なパッケージ。
golang.org/x/sync		0.0.0	mongo-driver ライブラリに必要なパッケージ。
golang.org/x/sys		0.0.0	mongo-driver ライブラリに必要なパッケージ。
golang.org/x/text		0.3.7	mongo-driver ライブラリに必要なパッケージ。

¹ "間接的" とは、エージェントが使用するライブラリの 1 つで使用されることを意味します。Zabbix はパッケージを使用するライブラリを使用するので必須です。

概要

このセクションには、Zabbix を安全な方法で設定するために遵守すべきベストプラクティスが含まれています。

ここに含まれるプラクティスは、Zabbix の機能には必要ありません。システムのセキュリティを強化するために推奨されます。

アクセス制御

最小特権の原則

Zabbix では、常に最小特権の原則 (Principle of least privilege) を使用する必要があります。この原則は、ユーザアカウント (Zabbix フロントエンド) またはプロセスユーザ (Zabbix server /proxy または agent) は意図した機能を実行するために必要な権限のみを持つということを意味します。言い換えると、常にユーザアカウントは可能な限り少ない権限で実行する必要があります。

Attention:

"zabbix" ユーザに特別なパーミッションを与えると、設定ファイルへのアクセスや操作の実行が可能になり、インフラ全体のセキュリティが脅かされる可能性があります。

ユーザアカウントに最小権限原則を導入する場合、Zabbix の **フロントエンドユーザタイプ** を考慮する必要があります。"Admin" ユーザタイプは "Super Admin" ユーザタイプよりも少ない権限しか持ちませんが、管理者権限を持っており、設定の管理やカスタムスクリプトの実行が可能です。

Note:

一部の情報は、非特権ユーザーでも利用可能です。例えば、Administration → Scripts は "Super Admin" 以外では利用できませんが、スクリプト自体は Zabbix API を利用することで実行可能です。スクリプトで利用できる機密情報の漏洩を防ぐために、スクリプトのパーミッションを制限し、(アクセス認証などの) 機密情報を追加しないようにしてください。

Zabbix agent の実行ユーザ

デフォルトの設定では、Zabbix server と Zabbix agent のプロセスは、単一の "zabbix" ユーザを共有します。server 設定の機密情報 (例: データベースのログイン情報) に agent がアクセスできないようにしたい場合は、以下の手順で agent を別のユーザで実行する必要があります:

1. セキュアユーザーの作成
2. agent [設定ファイル](#) ("User" パラメータ) でこのユーザを指定する。
3. 管理者権限でエージェントを再起動します。指定したユーザには特権が与えられます。

UTF-8 エンコーディング

Zabbix がサポートするエンコーディングは UTF-8 のみです。セキュリティ上の欠陥なく動作することが知られています。ユーザーは他のエンコーディングを使用する場合、既知のセキュリティ上の問題があることを留意してください。

Windows installer paths

When using Windows installers, it is recommended to use default paths provided by the installer as using custom paths without proper permissions could compromise the security of the installation.

Zabbix セキュリティアドバイザリと CVE データベース

[Zabbix セキュリティアドバイザリと CVE データベース](#) を参照してください。

Zabbix の Web インターフェースの SSL 設定

RHEL の場合、mod_ssl パッケージをインストールします：

```
yum install mod_ssl
```

SSL 鍵用のディレクトリを作成します：

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

SSL 証明書を作成します：

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

プロンプトに対して適切な値を入力します。特に重要な行は、Common Name を指定する行です。サーバーに関連付けるドメイン名を入力する必要があります。ドメイン名がない場合は、代わりにパブリック IP アドレスを入力することができます。ここでは、example.com を使用します。

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Apache の SSL の設定を編集します：

```
/etc/httpd/conf.d/ssl.conf
```

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apate-selfsigned.key
```

変更を反映させるため Apache のサービスを再起動します：

```
systemctl restart httpd.service
```

Web サーバーの強化

URL のルートディレクトリで Zabbix を有効化する

Apache の設定にバーチャルホストを追加し、ドキュメントルートを Zabbix SSL URL に恒久的にリダイレクトするよう設定します。このとき、忘れずに example.com を実際のサーバ名に置き換えてください。

```
/etc/httpd/conf/httpd.conf
```

```
#Add lines
```

```
<VirtualHost *:*>
ServerName example.com
Redirect permanent / https://example.com
</VirtualHost>
```

設定変更を反映させるために、Apache を再起動します。
 `systemctl restart httpd.service`

Web サーバーで HTTP Strict Transport Security (HSTS) を有効にする

Zabbix フロントエンドをプロトコルダウングレード攻撃から保護するために、以下のことを推奨します。

HSTS ポリシーを有効にすることをお勧めします。

例えば、Apache の設定で Zabbix フロントエンドの HSTS ポリシーを有効にする場合:

```
/etc/httpd/conf/httpd.conf
```

バーチャルホストの設定に以下のディレクティブを追加してください。

```
<VirtualHost *:443>
Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

設定変更を反映させるため、Apache を再起動します。

```
systemctl restart httpd.service
```

Enabling Content Security Policy (CSP) on the web server

To protect Zabbix frontend against Cross Site Scripting (XSS), data injection, and similar attacks, we recommend enabling Content Security Policy on the web server. To do so, configure the web server to return the [HTTP header](#).

Attention:

The following CSP header configuration is only for the default Zabbix frontend installation and for cases when all content originates from the site's domain (excluding subdomains). A different CSP header configuration may be required if you are, for example, configuring the [URL](#) widget to display content from the site's subdomains or external domains, switching from OpenStreetMap to another map engine, or adding external CSS or widgets.

To enable CSP for your Zabbix frontend in Apache configuration, follow these steps:

1. Locate your virtual host's configuration file:

- `/etc/httpd/conf/httpd.conf` on RHEL-based systems
- `/etc/apache2/sites-available/000-default.conf` on Debian/Ubuntu

2. Add the following directive to your virtual host's configuration file:

```
<VirtualHost *:*>
  Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe-
```

3. Restart the Apache service to apply the changes:

```
#### On RHEL-based systems:
```

```
systemctl restart httpd.service
```

```
#### On Debian/Ubuntu
```

```
systemctl restart apache2.service
```

Web サーバの情報公開を無効にする

Web サーバのハードニングプロセスの一環として、すべての Web サーバのシグネチャを無効にすることをお勧めします。
 Web サーバは、デフォルトでソフトウェアのシグネチャを公開します。

▼ **Response Headers** [view source](#)

```
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 1160
Content-Type: text/html; charset=UTF-8
Keep-Alive: timeout=5, max=100
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
```

この署名は、Apache (例として使用) の設定ファイルに 2 行追加することで無効にすることができます。

```
ServerSignature Off  
ServerTokens Prod
```

PHP のシングネチャ (X-Powered-By HTTP ヘッダー) は、php.ini の設定ファイルを変更することで無効にできます (デフォルトではシングネチャは無効になっています)

```
expose_php = Off
```

設定ファイルの変更を反映させるためには、Web サーバの再起動が必要です。

Apache で mod_security (package libapache2-mod-security2) を使用することにより、さらにセキュリティレベルを上げることができます。
 mod_security によって、サーバシングネチャを削除することができます。インストール後に "SecServerSignature" を任意の値に変更できます。

ソフトウェアシングネチャの削除・変更方法については、お使いの Web サーバのドキュメントを参照してください。

デフォルトの Web サーバのエラーページを無効にする

情報漏洩を防ぐため、デフォルトのエラーページを無効化することをお勧めします。
 Web サーバは、デフォルトで組み込みのエラーページを使用しています:

Not Found

The requested URL /custom-text was not found on this server.

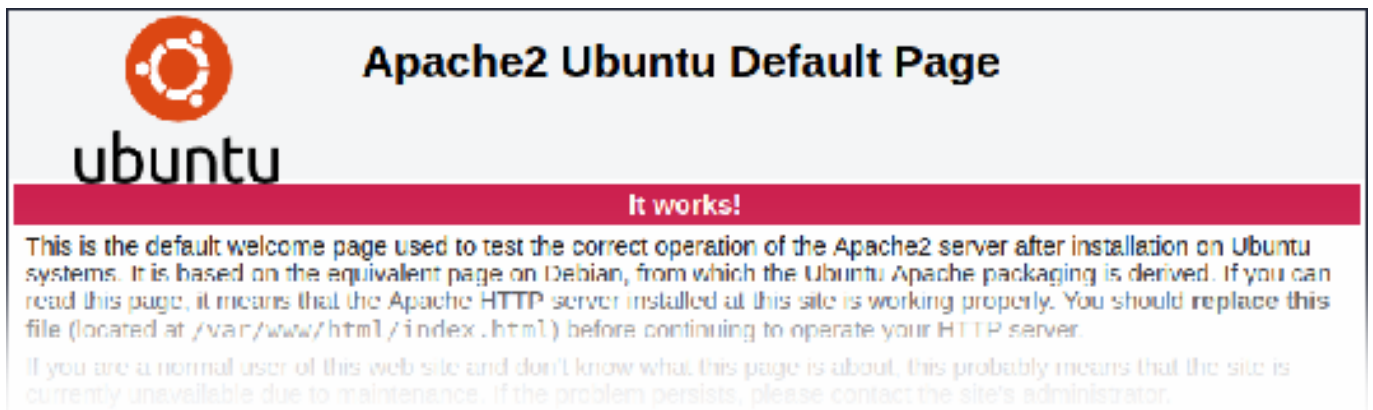
Apache/2.4.18 (Ubuntu) Server at localhost Port 80

デフォルトのエラーページは、Web サーバのハードニングプロセスの一環として、置き換え/削除する必要があります。
 "ErrorDocument" ディレクティブは (例として Apache) Web サーバ用のカスタムエラーページ/テキストを定義するために使用されます。

デフォルトのエラーページを置き換える/削除する方法については、お使いの Web サーバのドキュメントを参照してください。

Web サーバのテストページを削除する

情報漏洩を防ぐため、Web サーバのテストページは削除することをお勧めします。
 デフォルトでは、Web サーバの Webroot には、index.html と呼ばれるテストページが含まれています。
 (Ubuntu の Apache2 を例としています):



テストページは、Web サーバのハードニングプロセスの一環として削除するか、利用できないようにする必要があります。

X-Frame-Options HTTP 応答ヘッダーを設定する

デフォルトでは、Zabbix は X-Frame-Options HTTP response header が SAMEORIGIN に設定されて設定されています。つまり、ページ自体と同じオリジンを持つフレームにのみコンテンツをロードできます。

外部 URL (つまり URL ダッシュボードウィジェット) からコンテンツをプルする Zabbix フロントエンド要素は、すべてのサンドボックス制限が有効になっているサンドボックスで取得されたコンテンツを表示します。

これらの設定により、Zabbix フロントエンドのセキュリティが強化され、XSS およびクリックジャッキング攻撃に対する保護が提供されます。スーパー管理者は、必要に応じて iframe sandboxing および X-Frame-Options HTTP response header パラメータを**変更**できます。デフォルト設定を変更する前に、リスクと利点を慎重に検討してください。サンドボックスまたは X-Frame-Options を完全にオフにすることはお勧めしません。

Windows で SSL 構成ファイルへの書き込みアクセスを取り消す

OpenSSL でコンパイルされた Zabbix Windows エージェントは、c:\openssl-64bit にある SSL 構成ファイルにアクセスしようとします。ディスク C: の"openssl-64bit" ディレクトリは、非特権ユーザーでも作成できます。

そのため、セキュリティ強化のため、このディレクトリを手動で作成し、管理者以外のユーザーからの書き込みアクセスを取り消す必要があります。

32 ビット版と 64 ビット版の Windows では、ディレクトリ名が異なることに注意してください。

暗号化

共通パスワードのリストを含むファイルの隠蔽

パスワードのブルートフォースアタックの複雑性を高めるために、以下のような方法があります。
 ui/data/top_passwords.txt へのアクセスを制限することをお勧めします。

このファイルには、最も一般的なパスワードやコンテキスト固有のパスワードのリストが含まれており
 Avoid easy-to-guess passwords パラメータが [passwordpolicy] (/manual/web_interface/frontend_sections/administration/authentication#internal_authentication) で有効になっている場合、
 ユーザーがそのようなパスワードを設定できないようにするために使用されます。

例えば、NGINX では location ディレクティブを使用してファイルへのアクセスを制限することができます。

```
location = /data/top_passwords.txt {
deny all;
return 404;
}
```

Apache では、- .htaccess ファイルで:

```
<Files "top_passwords.txt">
Order Allow,Deny
Deny from all
</Files>
```

安全な **Zabbix** セットアップのベストプラクティス

3 ソースコードからのインストール

Zabbix の最新バージョンはソースからコンパイルすることで入手できます。

Zabbix をソースからインストールするためのステップバイステップのチュートリアルはこちらです。

1 Zabbix デーモンのインストール

1 ソースコードアーカイブをダウンロードする

[Zabbix ダウンロードページ](#)にアクセスし、ソースコードのアーカイブをダウンロードします。ダウンロード後、ソースを解凍してください。

以下のコマンドラインを実行してください:

```
$ tar -zxvf zabbix-6.0.0.tar.gz
```

Note:

コマンドに正しい Zabbix のバージョンを入力します。このコマンドはダウンロードしたアーカイブの名前と一致する必要があります。

2 ユーザーアカウントの作成

すべての Zabbix デーモンプロセスで、非特権ユーザーが必要です。非特権ユーザーアカウントから Zabbix デーモンを起動した場合、そのユーザーとして実行されます。

ただし、デーモンが"root" アカウントで起動された場合は、"zabbix" アカウントに切り替わるため、"zabbix" グループに所属するユーザーアカウント ("zabbix:zabbix") を作成する必要があります。

RedHat-based システムでは以下を実行:

```
groupadd --system zabbix
useradd --system -g zabbix -d /usr/lib/zabbix -s /sbin/nologin -c "Zabbix Monitoring System" zabbix
```

Debian-based システムでは以下を実行:

```
addgroup --system --quiet zabbix
adduser --quiet --system --disabled-login --ingroup zabbix --home /var/lib/zabbix --no-create-home zabbix
```

Attention:

Zabbix のプロセスにはホームディレクトリは必要ありません。そのため、作成することはお勧めしません。しかし、以下の様な機能を使用する場合は、ホームディレクトリを作成する必要があります。(例: MySQL 認証情報を \$HOME/.my.cnf に格納する)、

以下のコマンドを使用して自由に作成することができます。

RedHat-based システムでは以下を実行:

```
mkdir -m u=rwx,g=rwx,o= -p /usr/lib/zabbix
chown zabbix:zabbix /usr/lib/zabbix
```

Debian-based システムでは以下を実行:

```
mkdir -m u=rwx,g=rwx,o= -p /var/lib/zabbix
chown zabbix:zabbix /var/lib/zabbix
```

Zabbix フロントエンドのインストールには、個別のユーザアカウントは必要ありません。

Zabbix の **server** と **agent** が同じマシンで実行されている場合、**server** と **agent** は別のユーザで実行することをお勧めします。そうしないと、両方が同じユーザで実行された場合、**agent** が **server** の設定ファイルにアクセスし **agent** は **server** の設定ファイルにアクセスすることができ、Zabbix の Admin レベルのユーザは簡単に **server** の設定ファイル (例として、データベースパスワードなど) を取得することができてしまいます。

Attention:

Zabbix を root、bin、その他の特別な権限のアカウントで実行することはセキュリティ上のリスクがあります。

3 Zabbix データベースの作成

Zabbix **server** および **proxy** デーモン、および Zabbix フロントエンドには、データベースが必要です。Zabbix **agent** の実行にはデータベースは必要ありません。

SQL スクリプトが提供されていますので、データベーススキーマの作成とデータセットの挿入を行います。Zabbix proxy データベースにはスキーマのみ必要ですが、Zabbix server データベースにはスキーマ上にデータセットが必要です。

Zabbix データベースの作成が完了したら、次のステップに進みます。

4 ソースの構成

Zabbix server または proxy のソースコードを設定する場合、使用するデータベースの種類を指定する必要があります。一度にサーバまたはプロキシプロセスでコンパイルできるデータベースタイプは 1 つだけです。

サポートされているすべての設定オプションを確認するには、解凍した Zabbix のソースディレクトリ内で以下を実行してください。

```
./configure --help
```

Zabbix server と agent のソースを設定するには、以下のように実行します。

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --with-
```

Zabbix server (PostgreSQL など) のソースを設定するには、以下のコマンドを実行します。

```
./configure --enable-server --with-postgresql --with-net-snmp
```

Zabbix proxy のソースを設定する場合 (SQLite などを使用する場合) は、以下を実行します。

```
./configure --prefix=/usr --enable-proxy --with-net-snmp --with-sqlite3 --with-ssh2
```

Zabbix agent のソースを設定するには、以下を実行します。

```
./configure --enable-agent
```

Zabbix agent 2 のソースを設定するには、以下を実行します。

```
./configure --enable-agent2
```

Note:

現在サポートされているGoバージョンが、Zabbix エージェント 2 のビルドに必要です。golang.orgを参照してください。

コンパイルオプションに関する注意点:

- Command-line utilities zabbix_get and zabbix_sender are compiled if --enable-agent option is used.
- --with-libcurl and --with-libxml2 configuration options are required for virtual machine monitoring; --with-libcurl is also required for SMTP authentication and web.page.* Zabbix agent items. Note that cURL 7.20.0 or higher is required with the --with-libcurl configuration option.
- Zabbix always compiles with the PCRE library (since version 3.4.0); installing it is not optional. --with-libpcre=[DIR] only allows pointing to a specific base install directory, instead of searching through a number of common places for the libpcre files.
- You may use the --enable-static flag to statically link libraries. If you plan to distribute compiled binaries among different servers, you must use this flag to make these binaries work without required libraries. Note that --enable-static does not work in Solaris.
- Using --enable-static option is not recommended when building server. In order to build the server statically, you must have a static version of every external library needed. There is no strict check for that in configure script.
- Add optional path to the MySQL configuration file --with-mysql=//mysql_config to select the desired MySQL client library when there is a need to use one that is not located in the default location. It is useful when there are several versions of MySQL installed or MariaDB installed alongside MySQL on the same system.
- Use --with-oracle flag to specify location of the OCI API.

Attention:

./configure がライブラリの欠落やその他の理由で失敗した場合は、config.log ファイルを参照してください。config.log ファイルにエラーの詳細が記述されています。例えば、libssl が見つからない場合、直前のエラーメッセージは誤解を招く可能性があります。

```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
一方、config.log にはより詳細な記述があります:
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

こちらをご参照ください: - [Compiling Zabbix with encryption support for encryption support](#) - [Known issues with compiling Zabbix agent on HP-UX](#)

5 全体の Make とインストール

Note:

Zabbix Gitrepository からインストールする場合、以下のコマンドを最初に実行する必要があります。

```
$ make dbschema
```

make install

このステップは、十分なパーミッションを持つユーザーで実行する必要があります。(通常は root、または sudo を使用します)

make install を実行すると、デフォルトで daemon バイナリ (zabbix_server, zabbix_agentd, zabbix_proxy) が /usr/local/sbin に、client バイナリ (zabbix_get, zabbix_sender) が /usr/local/bin にインストールされます。

Note:

/usr/local 以外の場所を指定するには、ソースコードの構成のステップで --prefix キーを使用します。--prefix=/home/zabbix. この場合、daemon バイナリは <prefix>/sbin に、ユーティリティは <prefix>/bin にインストールされます。man ページは <prefix>/share にインストールされます。

6 設定ファイルの見直し・編集

- Zabbix agent configuration ファイルを編集します。/usr/local/etc/zabbix_agentd.conf

このファイルは、zabbix_agentd がインストールされているすべてのホストで設定する必要があります。

このファイルには、Zabbix server の IP アドレスを指定する必要があります。他のホストからの接続は拒否されます。

- Zabbix server configuration ファイルを編集します。/usr/local/etc/zabbix_server.conf

データベース名、ユーザー名、パスワード (使用している場合) を指定する必要があります。

残りのパラメータは、小規模な構成（監視するホストが 10 台まで）であれば、デフォルトのままでもよいでしょう。Zabbix server (または proxy) のパフォーマンスを最大化する場合は、デフォルトのパラメータを変更する必要があります。詳しくは [パフォーマンスチューニング](#) セクションを参照してください。

- Zabbix proxy をインストールしている場合、proxy configuration ファイルを編集します。file `/usr/local/etc/zabbix_proxy.conf`

server の IP アドレスと proxy のホスト名 (server が知っている必要があります)、およびデータベース名、ユーザー名、パスワード (使用している場合) を指定する必要があります。

Note:

SQLite の場合、データベースファイルへのフルパスが必要です。DB のユーザとパスワードは必要ありません。

7 デーモンの起動

サーバ側で zabbix_server を起動します。

```
shell> zabbix_server
```

Note:

共有メモリが 36MB (またはそれ以上) 確保できるシステムであることを確認してください。そうでない場合、server は起動せず、“Cannot allocate shared memory for <type cache>” というメッセージが表示されます。この現象は FreeBSD, Solaris 8 ja で発生する可能性があります。このページの一番下にある [“参照”](#) のセクションを参照して、共有メモリの設定方法を確認してください。

監視対象ホスト側で、zabbix_agentd を起動します。

```
shell> zabbix_agentd
```

Note:

システムが 2MB の共有メモリの割り当てを許可していることを確認してください。そうでない場合、agent は起動せず、agent のログファイルに “Cannot allocate shared memory for collector.” と表示されます。これは Solaris 8 で発生する可能性があります。

Zabbix proxy をインストールしている場合は、zabbix_proxy を起動します。

```
shell> zabbix_proxy
```

2 Zabbix web interface のインストール

PHP ファイルのコピー

Zabbix フロントエンドは PHP で書かれているため、実行するには PHP がサポートされた Web サーバが必要です。ファイルを ui ディレクトリから Web サーバの HTML ドキュメントディレクトリにコピーすることでインストールできます。

Apache ウェブサーバーにおける HTML ドキュメントディレクトリの一般的な場所が含まれます。

- /usr/local/apache2/htdocs (Apache をソースコードからインストールした場合のデフォルトディレクトリ)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL, CentOS)

HTML のルートではなくサブディレクトリを使用することを推奨します。サブディレクトリを作成し、Zabbix フロントエンドのファイルをそこにコピーするには、以下のコマンドを実行し、実際のディレクトリを置き換えます。

```
mkdir <htdocs>/zabbix
cd ui
cp -a . <htdocs>/zabbix
```

英語以外の言語を使用する場合は、[追加フロントエンド言語のインストール](#) の説明を参照してください。

フロントエンドのインストール

[ウェブインタフェースのインストール](#) のページで、Zabbix フロントエンドインストールウィザードの情報をご覧ください。

3 Java gateway のインストール

JMX アプリケーションを監視する場合のみ、Java gateway のインストールが必要です。Java gateway は軽量で、データベースを必要としません。

ソースからインストールする場合、まず [ダウンロード](#) をして、ソースのアーカイブを展開します。

Java gateway をコンパイルするには、`./configure` スクリプトを `--enable-java` オプション付きで実行します。`--prefix` オプションを指定して、デフォルトの `/usr/local` 以外のインストールパスを要求することが望ましいです。というのも、Java gateway をインストールすると、単一の実行ファイルだけでなく、ディレクトリツリー全体が作成されるからです。

```
$ ./configure --enable-java --prefix=$PREFIX
```

Java gateway をコンパイルして JAR ファイルにパッケージするには、`make` を実行します。注意このステップでは、パスの通った `javac` と `jar` の実行ファイルが必要であることに注意してください。

```
$ make
```

これで、`zabbix-java-gateway-$VERSION.jar` ファイルが以下の場所に作成されました。配布ディレクトリの `src/zabbix_java` から Java gateway を実行することに問題がない場合、Java gateway の設定と実行の説明に進むことができます。そうでない場合は、十分な権限を持っていることを確認し、`make install` を実行してください。

```
$ make install
```

`setup` で Java gateway の設定と実行に関する詳細を参照してください。

4 Zabbix web サービスのインストール

Zabbix web サービスは、[スケジュールレポート](#)を使用する場合のみインストールが必要です。

ソースコードからインストールする場合、まず[ダウンロード](#)をして、ソースコードのアーカイブを展開します。

Zabbix web サービスをコンパイルするには、`./configure` スクリプトを `--enable-webservice` オプションをつけて実行します。

Note:

設定済みのGoバージョン 1.13 以上の環境が必要です。

web サービスがインストールされているマシンで `zabbix_web_service` を実行します。

```
shell> zabbix_web_service
```

Scheduled reports の詳細については、[設定](#) を参照してください。

macOS での Zabbix エージェントのビルド

概要

このセクションでは、TLS の有無にかかわらず、ソースから Zabbix macOS エージェントバイナリをビルドする方法を示します。

前提条件

コマンドライン開発者ツール (Xcode は必要ありません)、Automake、pkg-config、および PCRE (v8.x) が必要です。TLS を使用してエージェントバイナリをビルドする場合は、OpenSSL または GnuTLS も必要になります。

Automake と pkg-config をインストールするには、<https://brew.sh/> の Homebrew パッケージマネージャーが必要です。インストールするには、ターミナルを開いて次のコマンドを実行します。

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

次に Automake と pkg-config をインストールします。

```
$ brew install automake
$ brew install pkg-config
```

PCRE、OpenSSL、および GnuTLS ライブラリの準備は、エージェントへのリンク方法によって異なります。

これらのライブラリが既にある macOS マシンでエージェントバイナリを実行する場合は、Homebrew が提供するコンパイル済みのライブラリを使用できます。これらは通常、Zabbix エージェントバイナリの構築やその他の目的で Homebrew を使用する macOS マシンです。

共有バージョンのライブラリがない macOS マシンでエージェントバイナリを使用する場合は、ソースから静的ライブラリをコンパイルし、Zabbix エージェントをリンクする必要があります。

共有ライブラリを使用したエージェントバイナリのビルド

PCRE をインストール

```
$ brew install pcre
```

TLS でビルドする場合は OpenSSL や GnuTLS をインストール

```
$ brew install openssl
```

```
$ brew install gnutls
```

Zabbix ソースをダウンロード

```
$ git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

TLS なしでエージェントをビルド

```
$ cd zabbix
```

```
$ ./bootstrap.sh
```

```
$ ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6
```

```
$ make
```

```
$ make install
```

OpenSSL を使用してエージェントをビルド

```
$ cd zabbix
```

```
$ ./bootstrap.sh
```

```
$ ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-openssl=/usr/local/opt
```

```
$ make
```

```
$ make install
```

GnuTLS を使用してエージェントをビルド

```
$ cd zabbix-source/
```

```
$ ./bootstrap.sh
```

```
$ ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-gnutls=/usr/local/opt
```

```
$ make
```

```
$ make install
```

TLS を使用しない静的ライブラリを使用したエージェントバイナリのビルド

\$HOME/static-libs に PCRE 静的ライブラリがインストールされると仮定します。PCRE 8.42 を使用します。

```
$ PCRE_PREFIX="$HOME/static-libs/pcre-8.42"
```

Unicode プロパティをサポートする PCRE をダウンロードしてビルドします。

```
$ mkdir static-libs-source
```

```
$ cd static-libs-source
```

```
$ curl --remote-name https://ftp.pcre.org/pub/pcre/pcre-8.42.tar.gz
```

```
$ tar xf pcre-8.42.tar.gz
```

```
$ cd pcre-8.42
```

```
$ ./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
```

```
$ make
```

```
$ make check
```

```
$ make install
```

Zabbix ソースとビルドエージェントをダウンロードします。

```
$ git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

```
$ cd zabbix
```

```
$ ./bootstrap.sh
```

```
$ ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre="$PCRE_PREFIX
```

```
$ make
```

```
$ make install
```

OpenSSL と静的ライブラリを使用してエージェントバイナリをビルドする

OpenSSL をビルドする場合は、ビルドが成功した後に `make test` を実行することをお勧めします。ビルドが成功した場合でも、テストが失敗することがあります。この場合は、続行する前に問題を調査して解決する必要があります。

PCRE と OpenSSL の静的ライブラリが \$HOME/static-libs にインストールされると仮定します。PCRE 8.42 と OpenSSL 1.1.1a を使用します。

```
$ PCRE_PREFIX="$HOME/static-libs/pcre-8.42"
```

```
$ OPENSSL_PREFIX="$HOME/static-libs/openssl-1.1.1a"
```

static-libs-source で静的ライブラリをビルド

```
$ mkdir static-libs-source
```

```
$ cd static-libs-source
```

Unicode プロパティをサポートする PCRE をダウンロードしてビルド

```
$ curl --remote-name https://ftp.pcre.org/pub/pcre/pcre-8.42.tar.gz
$ tar xf pcre-8.42.tar.gz
$ cd pcre-8.42
$ ./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
$ make
$ make check
$ make install
$ cd ..
```

OpenSSL をダウンロードしてビルド

```
$ curl --remote-name https://www.openssl.org/source/openssl-1.1.1a.tar.gz
$ tar xf openssl-1.1.1a.tar.gz
$ cd openssl-1.1.1a
$ ./Configure --prefix="$OPENSSL_PREFIX" --openssldir="$OPENSSL_PREFIX" --api=1.1.0 no-shared no-capieng m
$ make
$ make test
$ make install_sw
$ cd ..
```

Zabbix ソースとビルドエージェントをダウンロード

```
$ git clone https://git.zabbix.com/scm/zbx/zabbix.git
$ cd zabbix
$ ./bootstrap.sh
$ ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre="$PCRE_PREFIX
$ make
$ make install
```

GnuTLS と静的ライブラリを使用してエージェントバイナリをビルドする

GnuTLS は、Nettle 暗号バックエンドと GMP 演算ライブラリに依存しています。完全な GMP ライブラリを使用する代わりに、このガイドでは、Nettle に含まれている mini-gmp を使用します。

GnuTLS と Nettle をビルドするときは、ビルドが成功した後に `make check` を実行することをお勧めします。ビルドが成功した場合でも、テストが失敗することがあります。この場合は、続行する前に問題を調査して解決する必要があります。

PCRE、Nettle、および GnuTLS 静的ライブラリが `$HOME/static-libs` にインストールされると仮定します。PCRE 8.42、Nettle 3.4.1、および GnuTLS 3.6.5 を使用します。

```
$ PCRE_PREFIX="$HOME/static-libs/pcre-8.42"
$ NETTLE_PREFIX="$HOME/static-libs/nettle-3.4.1"
$ GNUTLS_PREFIX="$HOME/static-libs/gnutls-3.6.5"
```

`static-libs-source` で静的ライブラリをビルド

```
$ mkdir static-libs-source
$ cd static-libs-source
```

Nettle をダウンロードしてビルド

```
$ curl --remote-name https://ftp.gnu.org/gnu/nettle/nettle-3.4.1.tar.gz
$ tar xf nettle-3.4.1.tar.gz
$ cd nettle-3.4.1
$ ./configure --prefix="$NETTLE_PREFIX" --enable-static --disable-shared --disable-documentation --disabl
$ make
$ make check
$ make install
$ cd ..
```

GnuTLS をダウンロードしてビルド

```
$ curl --remote-name https://www.gnupg.org/ftp/gcrypt/gnutls/v3.6/gnutls-3.6.5.tar.xz
$ tar xf gnutls-3.6.5.tar.xz
$ cd gnutls-3.6.5
$ PKG_CONFIG_PATH="$NETTLE_PREFIX/lib/pkgconfig" ./configure --prefix="$GNUTLS_PREFIX" --enable-static --d
$ make
$ make check
$ make install
```

```
$ cd ..
```

Zabbix ソースとビルドエージェントをダウンロード

```
$ git clone https://git.zabbix.com/scm/zbx/zabbix.git
$ cd zabbix
$ ./bootstrap.sh
$ CFLAGS="-Wno-unused-command-line-argument -framework Foundation -framework Security" \
> LIBS="-lgnutls -lhogweed -lnettle" \
> LDFLAGS="-L$GNUTLS_PREFIX/lib -L$NETTLE_PREFIX/lib" \
> ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre="$PCRE_PREFIX"
$ make
$ make install
```

Windows 版 Zabbix agent 2 の構築

ここでは、Zabbix agent 2(Windows 版) をソースコードからビルドする方法を説明します。

MinGW コンパイラのインストール

1. MinGW-w64 with SJLJ (set jump/long jump) Exception Handling and Windows threads をダウンロードします。
 (例: x86_64-8.1.0-release-win32-sjlj-rt_v6-rev0.7z)

2. 解凍して、c:\mingw へ移動します。

3. 環境変数を設定します。

```
@echo off <br> set PATH=%PATH%;c:\mingw\bin <br> cmd <br>
```

コンパイル時には、MinGW が提供する MSYS 端末ではなく、Windows プロンプトを使用します。

PCRE 開発者用ライブラリのコンパイル

以下の手順で 64 ビット PCRE をコンパイルし、ライブラリを c:\dev\pcre に、
 32 ビットライブラリを c:\dev\pcre32 にそれぞれインストールします。

1. PCRE ライブラリバージョン 8.XX を pcre.org (<http://ftp.pcre.org/pub/pcre/>) からダウンロードし、解凍します。

2. cmd を開き、展開されたソースに移動します。

64bit 版 PCRE のビルド

1. 既存の configuration/cache がもし存在するならば削除します。
del CMakeCache.txt
 rmdir /q /s CMakeFiles

2. cmake を実行します。(CMake は <https://cmake.org/download/> から入手できます):
cmake -G "MinGW Makefiles" -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-O2 -g" -DCMAKE_CXX_FLAGS="-O2 -g" -DCMAKE_INSTALL_PREFIX=c:\dev\pcre
3. 次のコマンドを実行します:
mingw32-make clean
 mingw32-make install

32bit 版 PCRE のビルド

1. 次のコマンドを実行します:
mingw32-make clean
2. CMakeCache.txt を削除します:
del CMakeCache.txt
 rmdir /q /s CMakeFiles

3. cmake を実行します:
cmake -G "MinGW Makefiles" -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-m32 -O2 -g" -DCMAKE_CXX_FLAGS="-m32 -O2 -g" -DCMAKE_EXE_LINKER_FLAGS="-Wl,-mi386pe" -DCMAKE_INSTALL_PREFIX=c:\dev\pcre32
4. 次のコマンドを実行します:
mingw32-make install

OpenSSL 開発者用ライブラリのインストール

1. 32bit と 64bit のビルドを<https://curl.se/windows/>からダウンロードします。

2. ファイルを c:\dev\openssl32 and c:\dev\openssl に展開します。

3. MinGW は静的ライブラリより優先されるため、展開した *.dll.a (DLL コールラッパーライブラリ) を削除します。

Zabbix agent 2 のコンパイル

32 bit

MinGW 環境 (Windows コマンドプロンプト) を起動し、Zabbix ソースツリー内の build/mingw ディレクトリに移動します。

次のコマンドを実行します。

```
mingw32-make clean
mingw32-make ARCH=x86 PCRE=c:\dev\pcre32 OPENSSSL=c:\dev\openssl32
```

64 bit

MinGW 環境 (Windows コマンドプロンプト) を起動し、Zabbix ソースツリー内の build/mingw ディレクトリに移動します。

以下のコマンドを実行します:

```
mingw32-make clean
mingw32-make PCRE=c:\dev\pcre OPENSSSL=c:\dev\openssl
```

Note:

32 ビット版、64 ビット版ともに 64 ビットプラットフォームでビルドすることができますが、
 32 ビット・プラットフォーム上でビルドできるのは 32 ビット版のみです。
 32 ビット・プラットフォームで作業する場合は、64 ビット版と同じ手順を踏んでください。

Windows 版 Zabbix agent の構築

概要

このセクションでは、ソースコードからの Windows 版 Zabbix agent バイナリファイル作成方法 (TLS 対応有/無) について説明します。

OpenSSL のコンパイル

以下の手順は、MS の Windows 10 (64 ビット) でソースから OpenSSL をコンパイルするのに役立ちます。

1. OpenSSL をコンパイルするために、Windows マシンでは以下のものがが必要です: 1. C compiler (e.g. VS 2017 RC), 2. NASM (<https://www.nasm.us/>), 3. Perl (e.g. Strawberry Perl from <http://strawberryperl.com/>), 4. Perl module Text::Template (cpan Text::Template).
 2. OpenSSL のソースを<https://www.openssl.org/>から取得します。ここでは OpenSSL 1.1.1 を使います。
 3. OpenSSL のソースを、例えば E:\openssl-1.1.1 に解凍します。
 4. コマンドラインウィンドウを開きます。例: VS 2017 RC の x64 Native Tools Command Prompt
 5. OpenSSL のソースディレクトリ (例: E:\openssl-1.1.1) に移動します。1. NASM があることを確認する: 'e:\in openssl-1.1.1> nasm --version NASM バージョン 2.13.01 コンパイル日: 2017 年 5 月 1 日
 6. OpenSSL の設定を行います。例: e:\openssl-1.1.1> perl E:\openssl-1.1.1:\Configure VC-WIN64A no-shared no-capieng no-srp no-gost no-dgram no-dtls1-method no-dtls1_2-method --api=1.1.0 --prefix=C:\OpenSSL-Win64-111-static --openssldir=C:\OpenSSL-Win64-111-static
- オプション 'no-shared' に注意してください。
'no-shared' オプションを使用すると、OpenSSL スタティックライブラリ libcrypto.lib と libssl.lib は「自己充足 Zabbix バイナリには OpenSSL が含まれ、外部の OpenSSL DLL は必要ありません。

Advantage: Zabbix binaries can be copied to other Windows
 machines without OpenSSL libraries. Disadvantage: when a new
 OpenSSL bugfix version is released, Zabbix agent needs to
 recompiled and reinstalled.

- If 'no-shared' is not used, then the static libraries
 libcrypto.lib and libssl.lib will be using OpenSSL DLLs at
 runtime. Advantage: when a new OpenSSL bugfix version is
 released, probably you can upgrade only OpenSSL DLLs, without
 recompiling Zabbix agent. Disadvantage: copying Zabbix agent to
 another machine requires copying OpenSSL DLLs, too.

7. OpenSSL をコンパイルし、テストを実行し、インストールする。
 e:\openssl-1.1.1> nmake
 e:\openssl-1.1.1> nmake test
 ...
 All tests successful.
 Files=152, Tests=1152, 501 wallclock secs (0.67 usr + 0.61 sys = 1.28 CPU)
 Result: PASS
 e:\openssl-1.1.1> nmake install_sw
 "install_sw" installs only software components (i.e. libraries,
 header files, but no documentation). If you want everything, use "nmake install".

PCRE のコンパイル

1. PCRE または PCRE2(Zabbix6.0 以降サポート) ライブラリを pcre.org リポジトリからダウンロードします。
 (<https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.zip>)

2. ディレクトリ E:\pcre2-10.39 に解凍します。

3. CMake を<https://cmake.org/download/>からインストールします。
 cmake\bin にパスが通っていることを確認してください (バージョン 3.9.4 で確認しました)。

4. ソースディレクトリのサブディレクトリとして、空のビルドディレクトリを作成します。
 例: *E:\pcre2-10.39**build*.

5. コマンドラインウィンドウを開きます。
 例: x64 Native Tools Command Prompt を開き、そのシェル環境から cmake-gui を実行します。
 この時、Windows のスタートメニューから Cmake を起動しようとする、エラーが発生する可能性があります。

6. ソースディレクトリに E:\pcre2-10.39、ビルドディレクトリに E:\pcre2-10.39build を入力します。
7. "Configure" ボタンをクリックします。
8. このプロジェクトのジェネレータを指定する場合は、"NMake Makefiles" を選択します。
9. 空のインストールディレクトリを新規に作成します。
 例: E:\pcre2-10.39-install
10. GUI にいくつかの設定オプションが表示されます。以下のオプションが選択されていることを確認します。
 - **pcre_support_unicode_properties** on

 - **pcre_support_utf** on

 - **CMAKE_INSTALL_PREFIX** E:\pcre2-10.39-install

11. もう一度 "Configure" ボタンをクリックします。これで、隣の "Generate" ボタンが有効になります。
12. "Generate" ボタンをクリックします。
13. エラーが発生した場合、CMake ビルドプロセスを繰り返す前に CMake キャッシュを削除することをお勧めします。
 CMake GUI では、キャッシュは "File > Delete Cache" を選択することで削除できます。
14. ビルドディレクトリには、使用可能な Makefile が含まれているはずですが。
15. コマンドラインウィンドウ (例: VS 2017 の x64 Native Tools Command Prompt) を開き、上記の Makefile に移動します。
16. NMake コマンドを実行します。
 E:\pcre2-10.39build> nmake install

Zabbix のコンパイル

以下の手順で、Windows 10(64 ビット) 上でソースから Zabbix をコンパイルすることができます。
 Zabbix を TLS サポートありなしでコンパイルする場合の大きな違いは、手順 4 のみです。

1.
 - Linux マシン上で、以下のサイトからソースをチェックアウトしてください。
 git:\$ git clone <https://git.zabbix.com/scm/zbx/zabbix.git>
 \$ cd zabbix
 \$./bootstrap.sh
 \$./configure --enable-agent --enable-ipv6 --prefix=pwd'
 \$ make dbschema
 \$ make dist

2. アーカイブ (zabbix-4.4.0.tar.gz) を Windows マシンにコピーし、解凍します。
3. ここでは、ソースが e:\zabbix-4.4.0 にあると仮定します。コマンドラインウィンドウ
 (the x64 Native Tools Command Prompt for VS 2017 RC.) を開いて、E:\zabbix-4.4.0\project に移動します。
4. zabbix_get、zabbix_sender、zabbix_agent をコンパイルします。
 - without TLS: 'E:\zabbix-4.4.0\build\win32\project> nmake /K PCREINCDIR=E:\pcre2-10.39-install\include PCRELIBDIR=E:\pcre2-10.39-install\lib
 - with TLS: E:\zabbix-4.4.0\build\win32\project> nmake /K -f Makefile_get TLS=openssl TLSINCDIR=C:\OpenSSL-Win64-111-static\include TLSLIBDIR=C:\OpenSSL-Win64-111-static\lib PCREINCDIR=E:\pcre2-10.39-install\include PCRELIBDIR=E:\pcre2-10.39-install\lib E:\zabbix-4.4.0\build\win32\project> nmake /K -f Makefile_sender TLS=openssl TLSINCDIR="C:\OpenSSL-Win64-111-static\include TLSLIBDIR="C:\OpenSSL-Win64-111-static\lib" PCREINCDIR=E:\pcre2-10.39-install\include PCRELIBDIR=E:\pcre2-10.39-install\lib E:\zabbix-4.4.0\build\win32\project> nmake /K -f Makefile_agent TLS=openssl

```
TLSINCDIR=C:\OpenSSL-Win64-111-static\include TLSLIBDIR=C:\OpenSSL-Win64-111-static\lib PCREINCDIR=E:\pcre2-10.39-install\include PCRELIBDIR=E:\pcre2-10.39-install\lib'
```

- 新しいバイナリは e:\zabbix-4.4.0\bin\win64 に生成されます。OpenSSL は no-shared オプションでコンパイルされたため、Zabbix のバイナリは OpenSSL を含んでおり、OpenSSL を持たない他のマシンにコピーすることができます。

LibreSSL を使用した Zabbix のコンパイル

OpenSSL でコンパイルするのと同じような手順ですが、build\win32\project ディレクトリにあるファイルを少し変更する必要があります。

```
* In 'Makefile_tls' delete '/DHAVE_OPENSSL_WITH_PSK'. i.e. find <code> <br>
```

```
<br> CFLAGS = $(CFLAGS) /DHAVE_OPENSSL <br> /DHAVE_OPENSSL_WITH_PSK</code>and replace it with <br> CFLAGS = $(CFLAGS) /DHAVE_OPENSSL <br>
```

```
* In 'Makefile_common.inc' add '/NODEFAULTLIB:LIBCMT' i.e. find <code>
```

```
/MANIFESTUAC:"level='asInvoker' uiAccess='false'" /DYNAMICBASE:NO <br> /PDB:$(TARGETDIR)\$(TARGETNAME).pdb</code>and replace it with <br> /MANIFESTUAC:"level='asInvoker' uiAccess='false'" /DYNAMICBASE:NO /PDB:$(TARGETDIR)\$(TARGETNAME).pdb /NODEFAULTLIB:LIBCMT
```

4 パッケージインストール

Zabbix 公式リポジトリから

Zabbix SIA は、以下の公式 RPM および DEB パッケージを提供します。

- [Red Hat Enterprise Linux / CentOS](#)
- [Debian / Ubuntu / Raspbian](#)
- [SUSE Linux Enterprise Server](#)

各 OS 用のディストリビューションパッケージファイル yum/dnf、apt、zypper はリポジトリから利用可能です。 repo.zabbix.com。

一部の OS ディストリビューション (特に Debian ベースのディストリビューション) では注意してください。これらのサードパーティが提供するパッケージは Zabbix ではサポートされておらず、パッケージは古くなっている可能性があり、最新の機能やバグ修正が不足している可能性があります。 repo.zabbix.com の公式パッケージのみを使用することをお勧めします。以前に非公式の Zabbix パッケージを使用したことがある場合は [OS リポジトリからの Zabbix パッケージアップグレード](#) に関する注意事項を参照してください。

1 Red Hat Enterprise Linux

概要

Red Hat Enterprise Linux と Oracle Linux 用の公式な Zabbix 6.0 LTS パッケージは、 [Zabbix の Web サイト](#) から入手できます。

パッケージは、データベースとして MySQL か PostgreSQL、Web サーバーとして Apache か Nginx をサポートしています。

Zabbix エージェントパッケージとユーティリティである Zabbix get および Zabbix sender は、Zabbix 公式リポジトリから入手できます。 [RHEL 9](#)、[RHEL 8](#)、[RHEL 7](#)、[RHEL 6](#)、[RHEL 5](#)

Zabbix 公式リポジトリでは、fping、iksemel、libssh2 のパッケージも提供しています。これらのパッケージは、 [non-supported](#) というディレクトリにあります。

Attention:

EL9 用の EPEL リポジトリは、Zabbix のパッケージも提供しています。Zabbix 公式のリポジトリと EPEL リポジトリの両方がインストールされている場合、/etc/yum.repos.d/の下にある EPEL リポジトリの構成ファイルに次の設定を追加して EPEL の Zabbix パッケージを除外する必要があります：

```
[epel]
```

```
...
```

```
excludepkgs=zabbix*
```

インストール時の注意事項

[インストール方法](#) については、ダウンロードページのプラットフォームごとの情報を参照してください：

- リポジトリ情報のインストール
- サーバー/エージェント/ Web インターフェースのインストール

- 初期データベースの作成と初期データのインポート
- Zabbix サーバー用のデータベースの設定
- Zabbix の Web インターフェース用の PHP の設定
- サーバーとエージェントのプロセスの起動
- Zabbix の Web インターフェースの設定

Zabbix エージェントを root ユーザーで実行したい場合は、**root ユーザーによるエージェントの実行**を参照してください。

定期レポート生成に使用される Zabbix Web サービスプロセスには、Google Chrome ブラウザが必要です。Google Chrome ブラウザはパッケージに含まれていないので、手動でインストールする必要があります。

TimescaleDB 使用時のデータのインポート

TimescaleDB 使用时には、PostgreSQL のインポートコマンドに加えて、さらに以下のコマンドを実行します：

```
# cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb.sql | sudo -u zabbix psql zabbix
```

Warning:

TimescaleDB は、Zabbix サーバーでのみサポートされています。

PHP 7.2

Zabbix の Web インターフェースの動作には、**7.2.5** 以降が必要です。

SELinux の設定

Zabbix はソケットベースのプロセス間通信を使用します。SELinux が有効なシステムでは、Zabbix が SocketDir ディレクトリに UNIX ドメインソケットを作成/使用するための SELinux のルールの追加が必要となる場合があります。現在、ソケットファイルは、サーバー (alerter、preprocessing、IPMI) とプロキシ (IPMI) で使用されています。ソケットファイルは永続的であり、プロセスの実行中には常に存在します。

SELinux を enforcing モードで有効にした場合、次のコマンドを実行して Zabbix の Web インターフェースとサーバー間の通信を有効にする必要があります：

RHEL 7 以降：

```
# setsebool -P httpd_can_connect_zabbix on
```

ネットワーク経由でデータベースにアクセスする場合 (PostgreSQLで 'localhost' にアクセスする場合を含む) は、同

```
# setsebool -P httpd_can_network_connect_db on
```

RHEL 7 よりも前：

```
# setsebool -P httpd_can_network_connect on
```

```
# setsebool -P zabbix_can_network on
```

Web インターフェースと SELinux の設定変更後には、Apache Web サーバーを再起動してください：

```
# service httpd restart
```

さらに、Zabbix では、**RHEL 8** と **RHEL 7**用のパッケージとして zabbix-selinux-policy パッケージを提供しています。このパッケージは、基本的なデフォルトの SELinux のポリシーを設定し、Zabbix の各コンポーネントがソケットを作成して使用できるようにし、httpd から PostgreSQL にアクセス (Web インターフェースで使用) できるようにするものです。

ソース zabbix_policy.te ファイルには以下のルールが含まれています：

```
module zabbix_policy 1.2;
```

```
require {
    type zabbix_t;
    type zabbix_port_t;
    type zabbix_var_run_t;
    type postgresql_port_t;
    type httpd_t;
    class tcp_socket name_connect;
    class sock_file { create unlink };
    class unix_stream_socket connectto;
}
```

```
#===== zabbix_t =====
allow zabbix_t self:unix_stream_socket connectto;
allow zabbix_t zabbix_port_t:tcp_socket name_connect;
allow zabbix_t zabbix_var_run_t:sock_file create;
```

```
allow zabbix_t zabbix_var_run_t:sock_file unlink;
allow httpd_t zabbix_port_t:tcp_socket name_connect;
```

```
#===== httpd_t =====
allow httpd_t postgresql_port_t:tcp_socket name_connect;
```

このパッケージは、構成が複雑なためにユーザーが SELinux を無効にしてしまわないようにするために作成されました。これには、Zabbix の導入と設定を高速化するのに十分なデフォルトのポリシーが含まれています。セキュリティレベルを最大限に高めるには、SELinux の設定をカスタマイズすることをお勧めします。

プロキシのインストール

必要なリポジトリ情報が追加されていれば、以下を実行することで Zabbix プロキシをインストールすることができます：

```
# dnf install zabbix-proxy-mysql zabbix-sql-scripts
```

コマンド内の 'mysql' の部分を 'pgsql' に変更して PostgreSQL を使用するか、'sqlite3' に変更して SQLite3 (プロキシのみ) を使用することができます。

パッケージ 'zabbix-sql-scripts' には、Zabbix サーバーと Zabbix プロキシの両方でサポートされているすべてのデータベース管理システムのデータベーススキーマが含まれていて、初期データのインポート時に使用されます。

データベースの作成

Zabbix プロキシ用のデータベースは分けて作成します。

Zabbix サーバーと Zabbix プロキシは、同じデータベースを使用することができません。両者を同じホスト上にインストールする場合は、プロキシ用のデータベースは別の名前である必要があります。

データのインポート

初期のスキーマのインポートには以下を実行します：

```
# cat /usr/share/zabbix-sql-scripts/mysql/proxy.sql | mysql -uzabbix -p zabbix
```

PostgreSQL (または SQLite) を使用するプロキシ用には以下を実行します：

```
# cat /usr/share/zabbix-sql-scripts/postgresql/proxy.sql | sudo -u zabbix psql zabbix
# cat /usr/share/zabbix-sql-scripts/sqlite3/proxy.sql | sqlite3 zabbix.db
```

Zabbix プロキシ用のデータベース設定

zabbix_proxy.conf を編集します：

```
# vi /etc/zabbix/zabbix_proxy.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<password>
```

Zabbix プロキシ用の DBName は、Zabbix サーバーとは別のデータベースを使用します。

MySQL の場合は、DBPassword に Zabbix 用データベースのパスワードを指定します。PostgreSQL の場合は PostgreSQL 上のユーザーのパスワードを指定します。

PostgreSQL では DBHost= を使用します。デフォルト設定の DBHost=localhost (または IP アドレス) をそのまま使用するのであれば、PostgreSQL は Zabbix からネットワークソケットを使用して接続されます。手順については SELinux の設定を参照してください。

Zabbix プロキシプロセスの起動

Zabbix プロキシプロセスを起動し、システムブート時に起動させるためには以下のように実行します：

```
# service zabbix-proxy start
# systemctl enable zabbix-proxy
```

Web インターフェースの設定

Zabbix プロキシは、Web インターフェースを持ちません。Zabbix サーバーとのみ通信を行います。

Java ゲートウェイのインストール

JMX アプリケーションを監視したいときには、Java ゲートウェイをインストールする必要があります。Java ゲートウェイは軽量で、データベースを必要としません。

必要なリポジトリが追加されていたら、以下のように実行することで Zabbix Java ゲートウェイをインストールすることができます：

```
# dnf install zabbix-java-gateway
```

Java ゲートウェイの設定の実行の詳細については[設定](#)を参照してください。

debuginfo パッケージのインストール

Note:

Debuginfo パッケージは、現在、RHEL のバージョン 7、6、5 で利用可能です。

debuginfo リポジトリを有効にするには、`/etc/yum.repos.d/zabbix.repo` ファイルを編集します。zabbix-debuginfo リポジトリの `enabled=0` を `enabled=1` に変更します。

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo - $basearch
baseurl=http://repo.zabbix.com/zabbix/5.5/rhel/7/$basearch/debuginfo/
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
gpgcheck=1
```

これによって zabbix-debuginfo パッケージをインストールすることができます。

```
# yum install zabbix-debuginfo
```

このパッケージには、すべての Zabbix コンポーネントのデバッグ情報が含まれています。

2 Debian/Ubuntu/Raspbian

概要

Debian、Ubuntu、および Raspberry Pi OS (Raspbian) 用の公式 Zabbix 6.0 LTS パッケージは、[Zabbix Web サイト](#) で入手できます。

パッケージは、MySQL/PostgreSQL データベースと Apache/Nginx Web サーバーのいずれかをサポートしています。

インストール時の注意

以下のダウンロードページで、プラットフォームごとの [インストール手順](#) を参照してください。

- リポジトリのインストール
- サーバー/エージェント/フロントエンドのインストール
- 初期データベースの作成、初期データのインポート
- Zabbix サーバー用のデータベースの設定
- Zabbix フロントエンド用の PHP 設定
- サーバー/エージェントのプロセス開始
- Zabbix フロントエンドの設定

Zabbix エージェントを root として実行する場合は、[エージェントを root として実行する](#) を参照してください。

[スケジュールされたレポートの生成](#) に使用される Zabbix Web サービスプロセスには、Google Chrome ブラウザーが必要です。ブラウザはパッケージに含まれていないため、手動でインストールする必要があります。

Timescale DB でデータをインポートする

TimescaleDB では、PostgreSQL の `import` コマンドに加えて、次も実行します。

```
# cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb.sql | sudo -u zabbix psql zabbix
```

Warning:

TimescaleDB は、Zabbix サーバーでのみサポートされています。

PHP 7.2

Zabbix 5.0 以降のフロントエンドには、PHP バージョン **7.2** 以降が必要です。

PHP バージョン 7.2 未満のディストリビューションに Zabbix フロントエンドをインストールするには、[手順](#) を参照してください。

SELinux の設定

RHEL については、[SELinux の設定](#) を参照してください。

Web インターフェースと SELinux の設定が完了したら、Apache Web サーバーを再起動します：

```
# service apache2 restart
```

プロキシのインストール

必要なりポジトリが追加されたら、次を実行して Zabbix プロキシをインストールできます。

```
# apt install zabbix-proxy-mysql zabbix-sql-scripts
```

コマンドの'mysql'を'pgsql'に置き換えて PostgreSQL を、'sqlite3'に置き換えて SQLite3 を使用することもできます。

パッケージ'zabbix-sql-scripts'には、Zabbix サーバーと Zabbix プロキシの両方でサポートされているすべてのデータベース管理システムのデータベーススキーマが含まれており、データのインポートに使用されます。

データベースの作成

Zabbix プロキシ用に別のデータベースを**作成**します。

Zabbix サーバーと Zabbix プロキシは同じデータベースを使用できません。サーバーとプロキシが同じホストにインストールされている場合、プロキシデータベースは別の名前にする必要があります。

データのインポート

初期スキーマのインポート:

```
# cat /usr/share/zabbix-sql-scripts/mysql/proxy.sql | mysql -uzabbix -p zabbix
```

PostgreSQL (または SQLite) を使用するプロキシの場合:

```
# cat /usr/share/zabbix-sql-scripts/postgresql/proxy.sql | sudo -u zabbix psql zabbix
```

```
# cat /usr/share/zabbix-sql-scripts/sqlite3/proxy.sql | sqlite3 zabbix.db
```

Zabbix プロキシデータベースの設定

zabbix_proxy.conf を編集します。

```
# vi /etc/zabbix/zabbix_proxy.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<password>
```

Zabbix プロキシの DBName では、Zabbix サーバーとは別のデータベースを使用します。

DBPassword には、MySQL の Zabbix データベースパスワードを使用します。PostgreSQL の場合は PostgreSQL ユーザーのパスワードを使用します。

PostgreSQL では DBHost= を使用します。デフォルト設定の DBHost=localhost (または IP アドレス) を使用した場合、PostgreSQL は Zabbix への接続にネットワークソケットを使用します。手順については、RHEL/CentOS の**それぞれのセクション**を参照してください。

Zabbix プロキシのプロセス開始

Zabbix プロキシプロセスを開始し、システムの起動時に自動で開始するには:

```
# systemctl restart zabbix-proxy
# systemctl enable zabbix-proxy
```

フロントエンドの設定

Zabbix プロキシにはフロントエンドがありません。Zabbix サーバーと通信のみ行います。

Java ゲートウェイのインストール

JMX アプリケーションを監視したい場合のみ**Java ゲートウェイ**のインストールが必要です。Java ゲートウェイは軽量で、データベースを必要としません。

必要なりポジトリの追加後、次を実行して Zabbix Java ゲートウェイをインストールできます。

```
# apt install zabbix-java-gateway
```

Java ゲートウェイの設定と実行の詳細については、**セットアップ**に進んでください。

3 SUSE Linux Enterprise Server

概要

SUSE Linux Enterprise Server の公式 Zabbix 6.0 LTS パッケージは、[Zabbix Web サイト](#)で入手できます。

Zabbix エージェントパッケージとユーティリティ Zabbix get および Zabbix sender は、Zabbix 公式リポジトリで入手できます。
[SLES 15 SLES 12](#)

Note:

Verify CA 暗号化モード は、MySQL ライブラリが古いため、MySQL を使用する SLES 12 (すべてのマイナー OS バージョン) では機能しません。

Zabbix リポジトリの追加

リポジトリパッケージをインストールします。このパッケージには、yum (ソフトウェアパッケージマネージャー) 設定ファイルが含まれています。

SLES 15:

```
# rpm -Uvh --nosignature https://repo.zabbix.com/zabbix/6.0/sles/15/x86_64/zabbix-release-6.0-1.sles15.noarch.rpm
# zypper --gpg-auto-import-keys refresh 'Zabbix Official Repository'
```

SLES 12:

```
# rpm -Uvh --nosignature https://repo.zabbix.com/zabbix/6.0/sles/12/x86_64/zabbix-release-6.0-1.sles12.noarch.rpm
# zypper --gpg-auto-import-keys refresh 'Zabbix Official Repository'
```

スケジュールされたレポートの生成に使用される Zabbix Web サービスプロセスには、Google Chrome ブラウザーが必要です。ブラウザはパッケージに含まれていないため、手動でインストールする必要があります。

サーバー/フロントエンド/エージェントのインストール

MySQL をサポートする Zabbix サーバー/フロントエンド/エージェントをインストールするには、

```
# zypper install zabbix-server-mysql zabbix-web-mysql zabbix-apache-conf zabbix-agent
```

Nginx Web サーバー用のパッケージを使用する場合は、コマンドの 'apache' を 'nginx' に置き換えます。SLES 12/15 での Zabbix の Nginx セットアップも参照してください。

Zabbix エージェント 2 (SLES 15 SP1+ のみ) を使用している場合は、これらのコマンドで 'zabbix-agent' を 'zabbix-agent2' に置き換えます。

MySQL をサポートする Zabbix プロキシをインストールするには、

```
# zypper install zabbix-proxy-mysql zabbix-sql-scripts
```

Substitute 'mysql' in the commands with 'pgsql' to use PostgreSQL. PostgreSQL を使用する場合は、コマンドの 'mysql' を 'pgsql' に置き換えます。

パッケージ 'zabbix-sql-scripts' には、Zabbix サーバーと Zabbix プロキシの両方でサポートされているすべてのデータベース管理システムのデータベーススキーマが含まれており、データのインポートに使用されます。

データベースの作成

Zabbix サーバー およびプロキシ デーモンには、データベースが必要です。Zabbix エージェント を実行する必要はありません。

Warning:

Zabbix サーバーと Zabbix プロキシには個別のデータベースが必要です。同じデータベースを使用することはできません。したがってサーバーとプロキシが同じホストにインストールされている場合、それらが参照するデータベースは異なる名前で作成する必要があります!

MySQL または PostgreSQL の指示に従ってデータベースを作成します。

データのインポート

ここでは MySQL を使用してサーバーの初期スキーマとデータをインポートします。

```
# zcat /usr/share/packages/zabbix-sql-scripts/mysql/create.sql.gz | mysql -uzabbix -p zabbix
```

新しく作成したデータベースのパスワードを入力するよう求められます。

PostgreSQL の場合、

```
# zcat /usr/share/packages/zabbix-sql-scripts/postgresql/create.sql.gz | sudo -u zabbix psql zabbix
```

TimescaleDB では、前のコマンドに加えて、次も実行します。

```
# zcat /usr/share/packages/zabbix-sql-scripts/postgresql/timescaledb.sql.gz | sudo -u <username> psql zabbix
```


Warning:

TimescaleDB は、Zabbix サーバーでのみサポートされています。

プロキシの場合、初期スキーマをインポートします。

```
# zcat /usr/share/packages/zabbix-sql-scripts/mysql/schema.sql.gz | mysql -uzabbix -p zabbix
```

PostgreSQL を使用するプロキシの場合、

```
# zcat /usr/share/packages/zabbix-sql-scripts/postgresql/schema.sql.gz | sudo -u zabbix psql zabbix
```

Zabbix サーバー/プロキシのデータベースを構成する

/etc/zabbix/zabbix_server.conf (および zabbix_proxy.conf) を編集して、それぞれのデータベースを使用します。例：

```
# vi /etc/zabbix/zabbix_server.conf
```

```
DBHost=localhost
```

```
DBName=zabbix
```

```
DBUser=zabbix
```

```
DBPassword=<password>
```

DBPassword では、MySQL の Zabbix データベースパスワードを使用します。PostgreSQL の場合は PostgreSQL ユーザーパスワードです。

PostgreSQL の場合は DBHost= を使用します。デフォルト設定の DBHost=localhost (または IP アドレス) を使用したくなりますが、この場合 PostgreSQL が Zabbix への接続にネットワークソケットを使用することになります。

Zabbix フロントエンド設定

使用する Web サーバー (Apache/Nginx) に応じて、対応する Zabbix フロントエンドの設定ファイルを編集します。

- Apache の場合、設定ファイルは /etc/apache2/conf.d/zabbix.conf にあります。一部の PHP 設定はすでに構成されています。ただし "date.timezone" 設定のコメントを外して、[適切なタイムゾーンを設定](#)する必要があります。

```
php_value max_execution_time 300
```

```
php_value memory_limit 128M
```

```
php_value post_max_size 16M
```

```
php_value upload_max_filesize 2M
```

```
php_value max_input_time 300
```

```
php_value max_input_vars 10000
```

```
php_value always_populate_raw_post_data -1
```

```
# php_value date.timezone Europe/Riga
```

- zabbix-nginx-conf パッケージは、Zabbix フロントエンド用に別の Nginx サーバーをインストールします。その設定ファイルは /etc/nginx/conf.d/zabbix.conf にあります。Zabbix フロントエンドが機能するためには、listen および/または server_name ディレクティブのコメントを外して設定する必要があります。

```
# listen 80;
```

```
# server_name example.com;
```

- Zabbix は、Nginx で独自の php-fpm 接続プールを使用します。

設定ファイルは /etc/php7/fpm/php-fpm.d/zabbix.conf にあります。一部の PHP 設定はすでに構成されています。ただし [date.timezone](#) を適切に設定する必要があります。

```
php_value[max_execution_time] = 300
```

```
php_value[memory_limit] = 128M
```

```
php_value[post_max_size] = 16M
```

```
php_value[upload_max_filesize] = 2M
```

```
php_value[max_input_time] = 300
```

```
php_value[max_input_vars] = 10000
```

```
; php_value[date.timezone] = Europe/Riga
```

フロントエンドのインストール手順に進む準備ができました。これにより、新しくインストールした Zabbix にアクセスできるようになります。

Zabbix プロキシにはフロントエンドがなく、Zabbix サーバーとのみ通信することに注意してください。

Zabbix サーバー/エージェントプロセスの開始

Zabbix サーバーとエージェントのプロセスを開始し、システムの起動時に開始するようにします。

Apache Web サーバーの場合:

```
# systemctl restart zabbix-server zabbix-agent apache2 php-fpm
# systemctl enable zabbix-server zabbix-agent apache2 php-fpm
```

Nginx Web サーバーの場合、'apache2' を 'nginx' に置き換えます。

debuginfo パッケージのインストール

debuginfo リポジトリを有効にするには、/etc/zypp/repos.d/zabbix.repo ファイルを編集します。zabbix-debuginfo リポジトリの enabled=0 を enabled=1 に変更します。

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=http://repo.zabbix.com/zabbix/4.5/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=http://repo.zabbix.com/zabbix/4.5/sles/15/x86_64/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

これにより zabbix-**<component>**-debuginfo パッケージをインストールできます。

4 MSI ファイルによる Windows 用 agent のインストール

概要

Windows 版 Zabbix エージェントは、[ダウンロード](#)できる Windows MSI インストーラパッケージ (32bit 版と 64bit 版) を使用してインストール可能です。

Zabbix エージェント 2 の MSI インストール最小要件は Windows7 x32 です。

Zabbix の get および sender ユーティリティも、Zabbix エージェント/エージェント 2 と一緒または個別にインストールできます。

32bit 版のパッケージは 64bit 版の Windows にインストールすることはできません。

すべてのパッケージは TLS をサポートしていますが、TLS の設定はオプションです。

UI およびコマンドラインベースのインストールに対応しています。

Attention:

MSI インストーラパッケージからの Zabbix インストールは完全にサポートされていますが、適切なエラー処理のために少なくとも Microsoft .NET Framework 2 をインストールすることをお勧めします。[Microsoft Download .NET Framework](#) を参照してください。

インストール手順

インストールするには、ダウンロードした MSI ファイルをダブルクリックします。



ZABBIX

The Enterprise-class
Monitoring Solution
for Everyone

www.zabbix.com

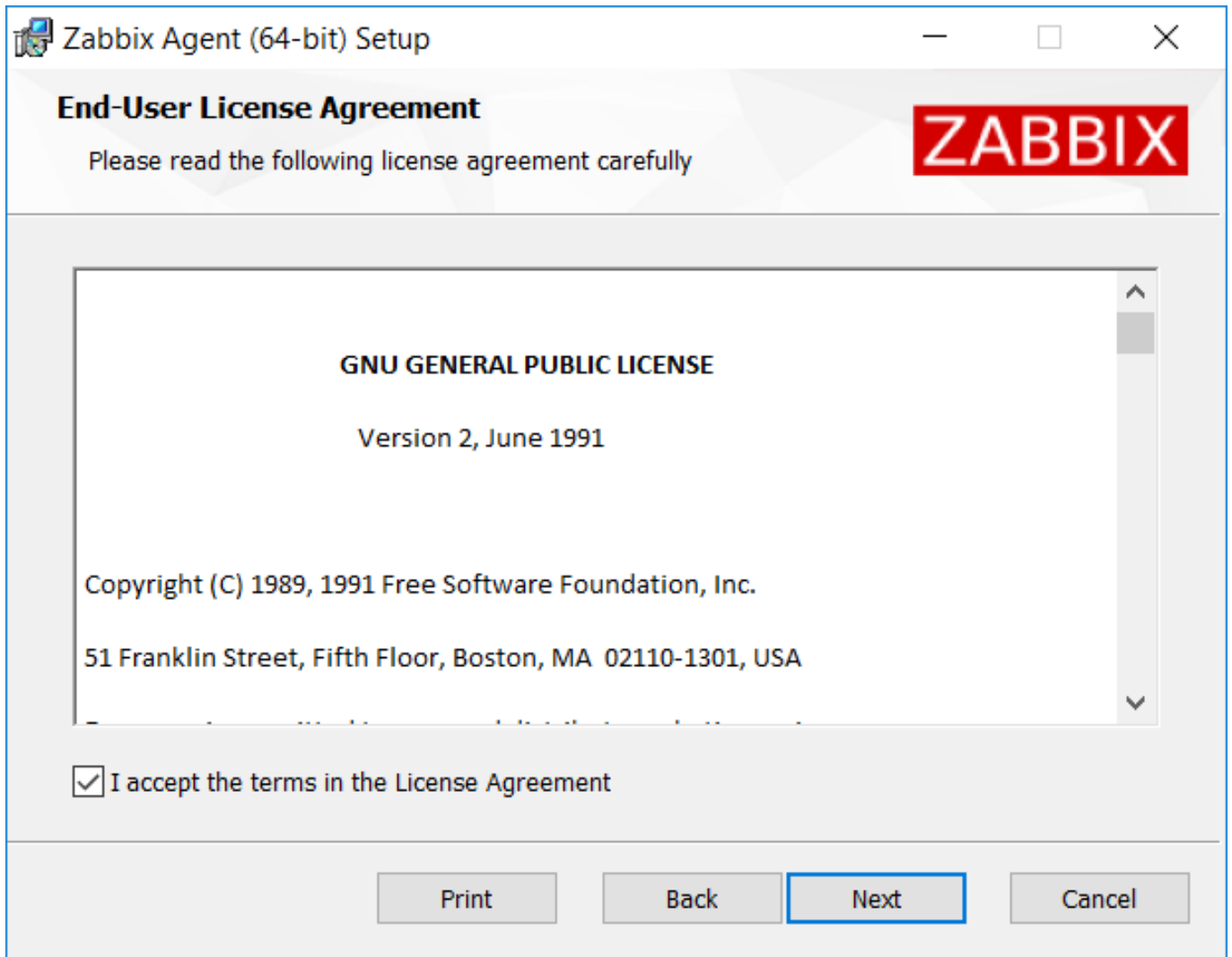
Welcome to the Zabbix Agent (64-bit) Setup Wizard

The Setup Wizard will install Zabbix Agent (64-bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

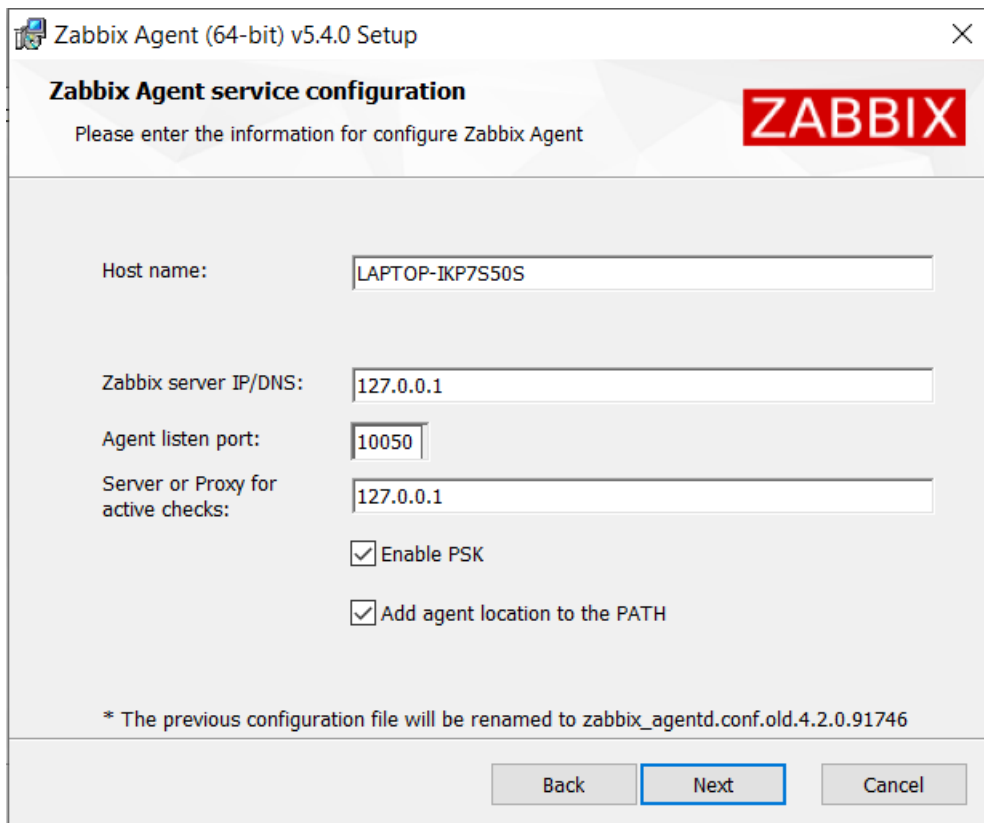
Back

Next

Cancel



ライセンスに「同意」して次のステップに進みます。



以下のパラメータを指定します。

パラメータ	説明
Host name	ホスト名を指定します
Zabbix server IP/DNS	Zabbix サーバーの IP/DNS を指定します
Agent listen port	エージェントのリッスンポートを指定します (デフォルトは 10050)
Server or Proxy for active checks	アクティブエージェントチェック用の Zabbix サーバー/プロキシの IP/DNS を指定します
Enable PSK	チェックボックスをマークして、事前共有キーによる TLS サポートを有効にします
Add agent location to the PATH	エージェントの場所を PATH 変数に追加します

Zabbix Agent (64-bit) PSK Setup

Zabbix Agent pre-shared key configuration

Please enter the PSK information for configure Zabbix Agent

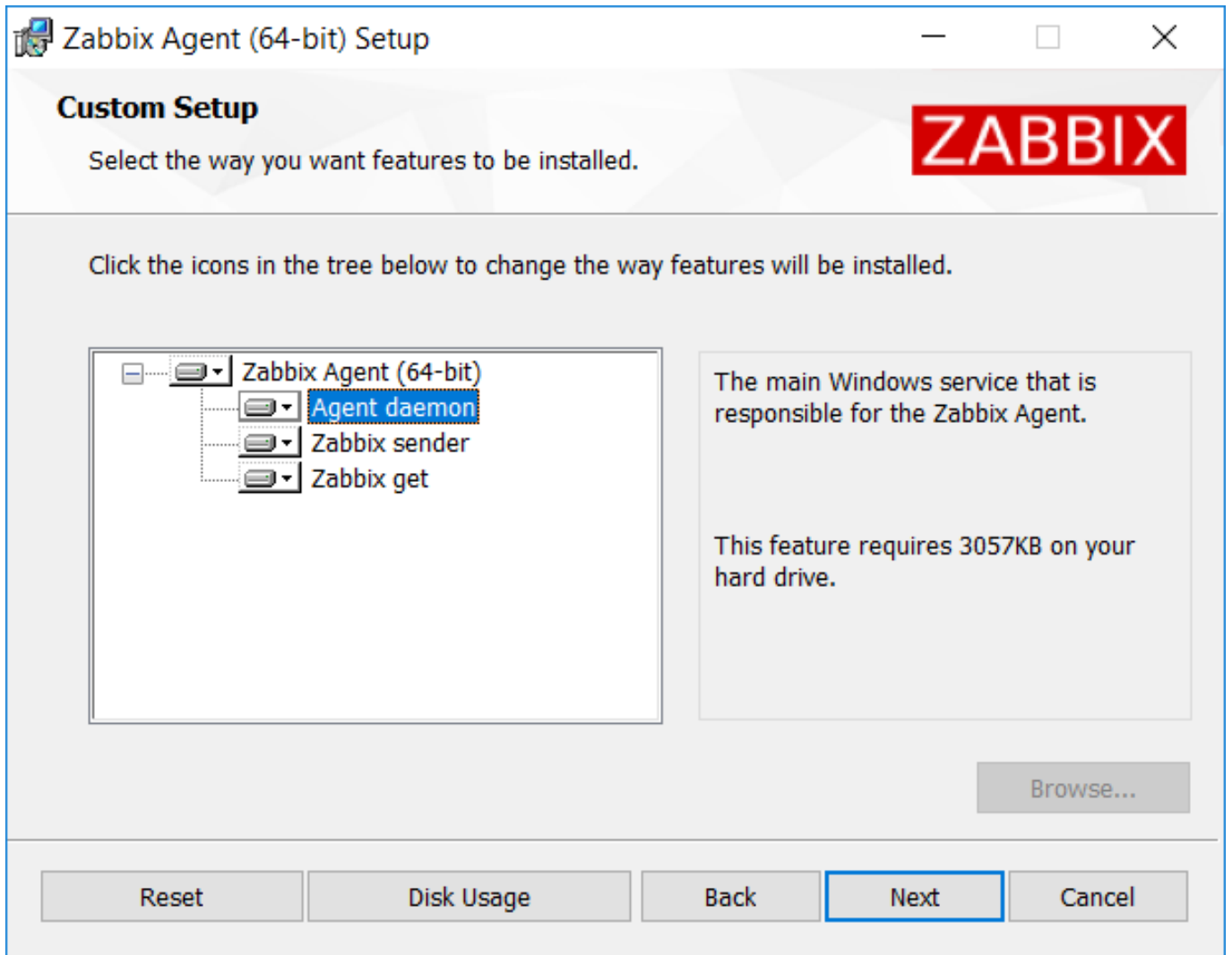
Pre-shared key identity:

Pre-shared key value:

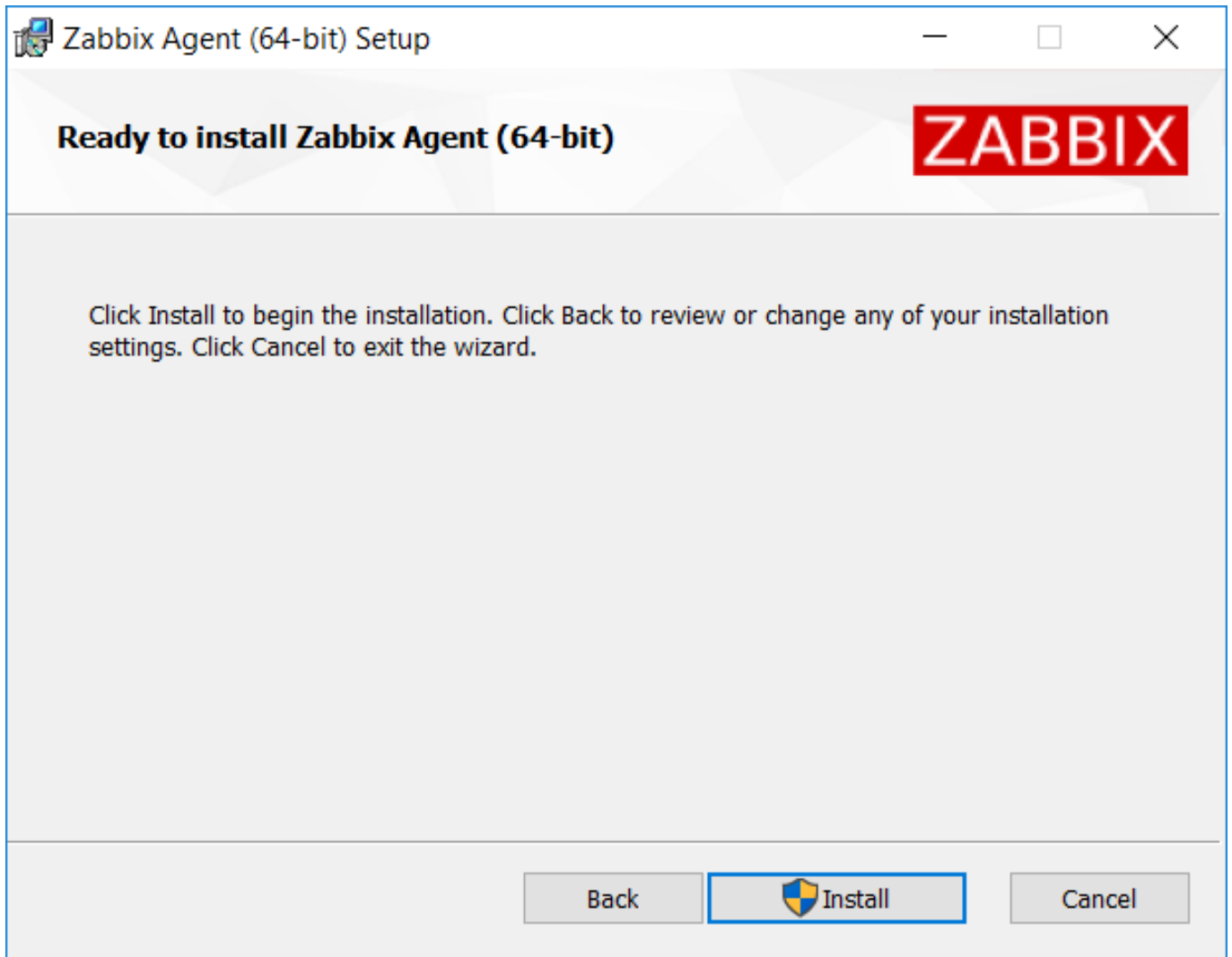
Please, set minimum required permission to access the psk.key file

Back Next Cancel

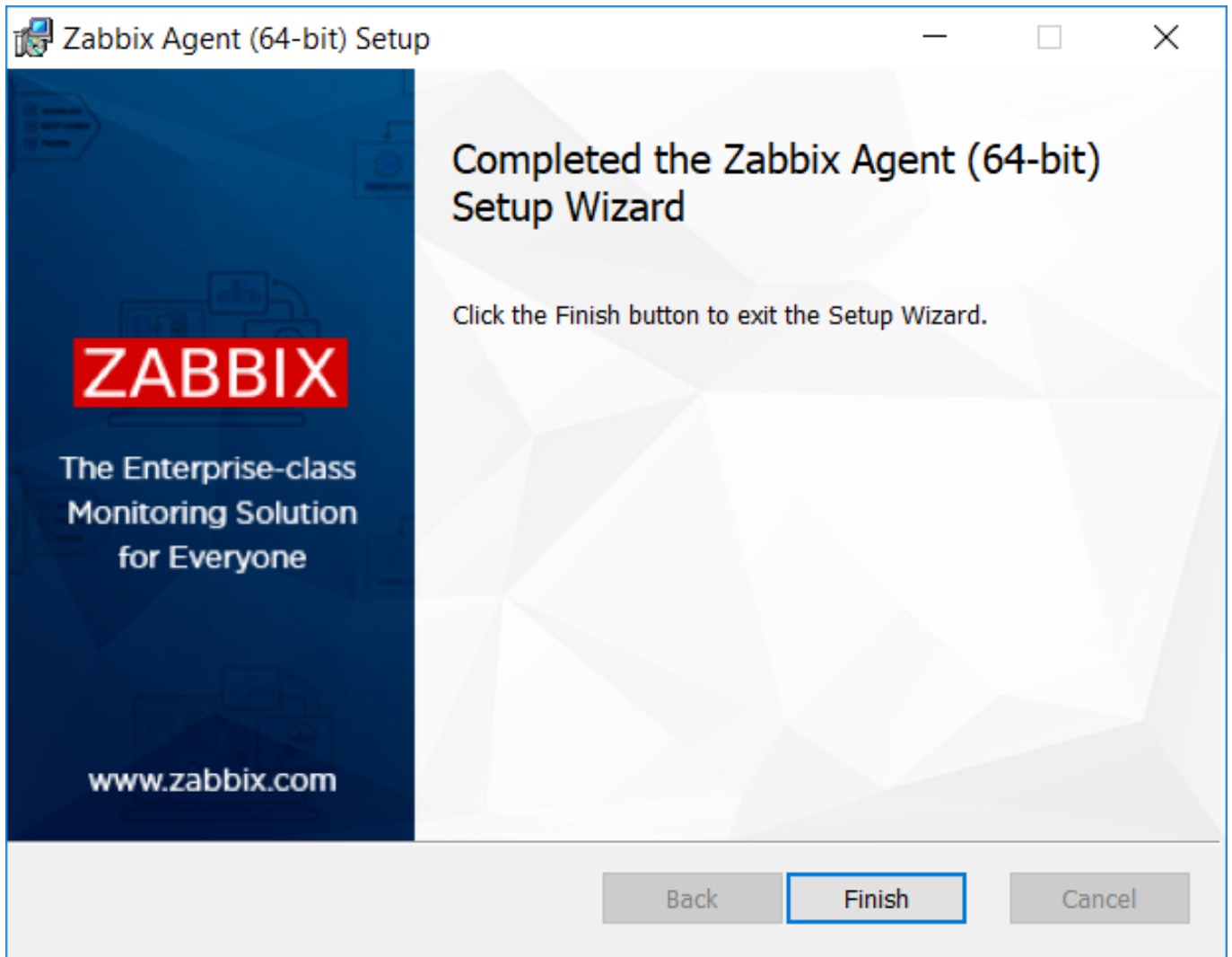
事前共有鍵の ID と値を入力します。このステップは、前のステップで Enable PSK をチェックした場合のみ利用できます。



インストールする Zabbix コンポーネントを選択します。 - Zabbix agent daemon, Zabbix sender, Zabbix get.



Zabbix コンポーネントと設定ファイルは Program Files の Zabbix Agent フォルダにインストールされます。
 zabbix_agentd.exe は自動起動の Windows サービスとして設定されます。



コマンドラインベースのインストーラ

サポートされるパラメータ

作成された MSI では、以下のパラメータがサポートされています。

番号	パラメータ	説明
1	LOGTYPE	
2	LOGFILE	
3	SERVER	
4	LISTENPORT	
5	SERVERACTIVE	
6	HOSTNAME	
7	TIMEOUT	
8	TLSCONNECT	
9	TLSACCEPT	
10	TLSPSKIDENTITY	
11	TLSPSKFILE	
12	TLSPSKVALUE	
13	TLSCAFILE	
14	TLSCRLFILE	
15	TLSSERVERCERTISSUER	
16	TLSSERVERCERTSUBJECT	
17	TLSCERTFILE	
18	TLSKEYFILE	
19	LISTENIP	
20	HOSTINTERFACE	
21	HOSTMETADATA	
22	HOSTMETADATAITEM	

番号	パラメータ	説明
23	STATUSPORT	Zabbix agent 2 only.
24	ENABLEPERSISTENTBUFFER	Zabbix agent 2 only.
25	PERSISTENTBUFFERPERIOD	Zabbix agent 2 only.
26	PERSISTENTBUFFERFILE	Zabbix agent 2 only.
27	INSTALLFOLDER	
28	ENABLEPATH	
29	SKIP	SKIP=fw - do not install firewall exception rule
30	INCLUDE	Sequence of includes separated by ;
31	ALLOWDENYKEY	Sequence of "AllowKey" and "DenyKey" parameters separated by ;. Use \\; to escape the delimiter.

インストールするには、例えば、以下のように実行します。
 SET INSTALLFOLDER=C:\Program Files\za


```
msiexec /l*v log.txt /i zabbix_agent-6.0.0-x86.msi /qn^
LOGTYPE=file^
LOGFILE="%INSTALLFOLDER%\za.log"^
SERVER=192.168.6.76^
LISTENPORT=12345^
SERVERACTIVE=:1^
HOSTNAME=myHost^
TLSCONNECT=psk^
TLSACCEPT=psk^
TLSPSKIDENTITY=MyPSKID^
TLSPSKFILE="%INSTALLFOLDER%\mykey.psk"^
TLSCAFILE="c:\temp\f.txt1"^
TLSCRLFILE="c:\temp\f.txt2"^
TLSSERVERCERTISSUER="My CA"^
TLSSERVERCERTSUBJECT="My Cert"^
TLSCERTFILE="c:\temp\f.txt5"^
TLSKEYFILE="c:\temp\f.txt6"^
ENABLEPATH=1^
INSTALLFOLDER="%INSTALLFOLDER%"^
SKIP=fw^
ALLOWDENYKEY="DenyKey=vfs.file.contents[/etc/passwd]"
```

または

```
msiexec /l*v log.txt /i zabbix_agent-6.0.0-x86.msi /qn^
SERVER=192.168.6.76^
TLSCONNECT=psk^
TLSACCEPT=psk^
TLSPSKIDENTITY=MyPSKID^
TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

6 PKG から Mac OS エージェントのインストール

概要

Zabbix Mac OS エージェントは、[ダウンロード](#) で利用可能な PKG インストーラーパッケージからインストールできます。暗号化ありまたはなしのバージョンが利用可能です。

エージェントのインストール

エージェントはグラフィカルユーザーインターフェイス、またはコマンドラインからインストールできます。次に例を示します。

```
sudo installer -pkg zabbix_agent-5.4.0-macos-amd64-openssl.pkg -target /
```

コマンドでは正しい Zabbix パッケージバージョンを指定していることを確認してください。ダウンロードしたパッケージの名前と一致する必要があります。

エージェント実行

エージェントは、インストールまたは再起動後に自動的に開始されます。

必要に応じて設定ファイル/usr/local/etc/zabbix/zabbix_agentd.conf を編集してください。

エージェントを手動で開始するには、次のコマンドを実行します。

```
sudo launchctl start com.zabbix.zabbix_agentd
```

エージェントを手動で停止するには、次のコマンドを実行します。

```
sudo launchctl stop com.zabbix.zabbix_agentd
```

アップグレード時、既存の設定ファイルは上書きされません。代わりに、新しい zabbix_agentd.conf.NEW ファイルが作成され、必要に応じて既存の設定ファイルを確認および更新するために使用します。設定ファイルを手動で変更した後は、忘れずにエージェントを再始動してください。

エージェントのトラブルシューティングと削除

このセクションでは、Zabbix エージェントのインストールのトラブルシューティングと削除に使用できる便利なコマンドをいくつか紹介します。

Zabbix エージェントが実行されているかどうかを確認する場合：

```
ps aux | grep zabbix_agentd
```

Zabbix エージェントがパッケージからインストールされているかどうかを確認する場合：

```
$ pkgutil --pkgs | grep zabbix
com.zabbix.pkg.ZabbixAgent
```

インストーラーパッケージからインストールされたファイルを確認する場合：(このビューには最初の/が表示されないことに注意してください)

```
$ pkgutil --only-files --files com.zabbix.pkg.ZabbixAgent
Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
usr/local/bin/zabbix_get
usr/local/bin/zabbix_sender
usr/local/etc/zabbix/zabbix_agentd/userparameter_examples.conf.NEW
usr/local/etc/zabbix/zabbix_agentd/userparameter_mysql.conf.NEW
usr/local/etc/zabbix/zabbix_agentd.conf.NEW
usr/local/sbin/zabbix_agentd
```

launchctl で起動された Zabbix エージェントを停止する場合：

```
sudo launchctl unload /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
```

インストーラーパッケージでインストールされたファイル (設定とログを含む) を削除する場合：

```
sudo rm -f /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
sudo rm -f /usr/local/sbin/zabbix_agentd
sudo rm -f /usr/local/bin/zabbix_get
sudo rm -f /usr/local/bin/zabbix_sender
sudo rm -rf /usr/local/etc/zabbix
sudo rm -rf /var/log/zabbix
```

パッケージから Zabbix エージェントを削除する場合：

```
sudo pkgutil --forget com.zabbix.pkg.ZabbixAgent
```

6 不安定版リリース

概要

マイナー Zabbix バージョン (つまり Zabbix 6.0.x) リリース候補のパッケージは、Zabbix 6.0.9 以降で提供されます。

以下の手順は、不安定版 Zabbix リリースリポジトリを有効にするためのものです (デフォルトでは無効)。

最初に、最新の zabbix-release パッケージをインストールまたは更新します。システムで rc パッケージを有効にするには、次の手順を実行します。

Red Hat Enterprise Linux

/etc/yum.repos.d/zabbix.repo ファイルを開き、zabbix-unstable レポジトリに対して enabled=1 を設定します。

```
[zabbix-unstable]
name=Zabbix Official Repository (unstable) - $basearch
baseurl=https://repo.zabbix.com/zabbix/5.5/rhel/8/$basearch/
```

```
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
```

Debian/Ubuntu

/etc/apt/sources.list.d/zabbix.list を開き、“Zabbix unstable repository” セクション下のコメントを外します。

```
# Zabbix unstable repository
deb https://repo.zabbix.com/zabbix/5.5/debian bullseye main
deb-src https://repo.zabbix.com/zabbix/5.5/debian bullseye main
```

SUSE

/etc/zypp/repos.d/zabbix.repo ファイルを開き、zabbix-unstable レポジトリに対して enable=1 を設定します。

```
[zabbix-unstable]
name=Zabbix Official Repository
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/5.5/sles/15/x86_64/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/5.5/sles/15/x86_64/repodata/repomd.xml.key
enabled=1
update=1
```

5 コンテナでのインストール

Docker Zabbix は各 Zabbix コンポーネントの **Docker** イメージをポータブルな自給自足のコンテナとして提供し、
 デプロイとアップデート手順の迅速化を図っています。

Zabbix コンポーネントには、MySQL、PostgreSQL データベースサポート、Apache2、Nginx Web サーバサポートが付属しています。
 これらのイメージはそれぞれ別のイメージに分かれています。

Docker のベースイメージ

Zabbix コンポーネントは Ubuntu、Alpine Linux、CentOS のベースイメージで提供されます。

イメージ	バージョン
alpine	3.12
ubuntu	20.04 (focal)
centos	8

ベースイメージが更新された場合、すべてのイメージが最新のイメージにリビルドされるように設定されています。

Docker ファイルのソース

[github.com](#) の Zabbix [official repository](#) を利用すれば、誰でも Docker ファイルの変更を追うことができます。
 プロジェクトをフォークしたり、公式 Docker ファイルをベースに独自のイメージを作成することができます。

構造

全ての Zabbix コンポーネントは以下の Docker リポジトリから入手可能です:

- Zabbix agent - [zabbix/zabbix-agent](#)
- Zabbix server
 - Zabbix server with MySQL database support - [zabbix/zabbix-server-mysql](#)
 - Zabbix server with PostgreSQL database support - [zabbix/zabbix-server-pgsql](#)
- Zabbix web-interface
 - Zabbix web-interface based on Apache2 web server with MySQL database support - [zabbix/zabbix-web-apache-mysql](#)
 - Zabbix web-interface based on Apache2 web server with PostgreSQL database support - [zabbix/zabbix-web-apache-pgsql](#)
 - Zabbix web-interface based on Nginx web server with MySQL database support - [zabbix/zabbix-web-nginx-mysql](#)
 - Zabbix web-interface based on Nginx web server with PostgreSQL database support - [zabbix/zabbix-web-nginx-pgsql](#)
- Zabbix proxy
 - Zabbix proxy with SQLite3 database support - [zabbix/zabbix-proxy-sqlite3](#)
 - Zabbix proxy with MySQL database support - [zabbix/zabbix-proxy-mysql](#)

- Zabbix Java Gateway - [zabbix/zabbix-java-gateway](#)

さらに、SNMP Trap もサポートされています。Ubuntu Trusty のみをベースにした追加リポジトリ ([zabbix/zabbix-snmptools](#)として提供されています。
 Zabbix server や Zabbix proxy と連携することが可能です。

バージョン

Zabbix コンポーネントの各リポジトリには、以下のタグが含まれています:

- latest - latest stable version of a Zabbix component based on Alpine Linux image
- alpine-latest - latest stable version of a Zabbix component based on Alpine Linux image
- ubuntu-latest - latest stable version of a Zabbix component based on Ubuntu image
- alpine-5.4-latest - latest minor version of a Zabbix 5.4 component based on Alpine Linux image
- ubuntu-5.4-latest - latest minor version of a Zabbix 5.4 component based on Ubuntu image
- alpine-5.4.* - different minor versions of a Zabbix 5.4 component based on Alpine Linux image, where * is the minor version of Zabbix component
- ubuntu-5.4.* - different minor versions of a Zabbix 5.4 component based on Ubuntu image, where * is the minor version of Zabbix component

使用方法

環境変数

すべての Zabbix コンポーネントイメージは、構成を制御するための環境変数を提供します。これらの環境変数は、各コンポーネントリポジトリにリストされています。これらの環境変数は Zabbix 構成ファイルのオプションですが、命名方法が異なります。たとえば ZBX_LOGSLOWQUERIES は、Zabbix サーバーおよび Zabbix プロキシ構成ファイルの LogSlowQueries と同じです。

Attention:

一部の設定オプションは変更できません。たとえば PIDFile や LogType

一部のコンポーネントには、公式の Zabbix 設定ファイルには存在しない特定の環境変数があります。

変数	コンポーネント	説明
DB_SERVER_HOST	Server Proxy Web interface	この変数は、MySQL または PostgreSQL サーバーの IP または DNS 名です。デフォルトの値は MySQL または PostgreSQL に対してそれぞれ mysql-server または postgres-server です。
DB_SERVER_PORT	この変数は MySQL または PostgreSQL サーバーのポートです。デフォルトの値はそれぞれ 3306 または 5432 です。	
MYSQL_USER	Server Proxy Web-interface	MySQL データベースユーザー デフォルトの値は zabbix です。
MYSQL_PASSWORD	Server Proxy Web interface	MySQL データベースパスワード デフォルトの値は 'zabbix' です。
MYSQL_DATABASE	Server Proxy Web interface	Zabbix データベース名 デフォルトの値は Zabbix サーバーの場合 zabbix、Zabbix プロキシの場合 zabbix_proxy です。
POSTGRES_USER	Server Web interface	PostgreSQL データベースユーザー デフォルトの値は 'zabbix' です
POSTGRES_PASSWORD	Server Web interface	PostgreSQL データベースパスワード デフォルトの値は 'zabbix' です
POSTGRES_DB	Server Web interface	Zabbix データベース名 デフォルトの値は Zabbix サーバーの場合 'zabbix'、Zabbix プロキシの場合 'zabbix_proxy' です
PHP_TZ	Web-interface	PHP フォーマットのタイムゾーン。サポートされているタイムゾーンは php.net で確認できます。デフォルト値は Europe/Riga です。

ZBX_SERVER_NAME	Web interface	Web インターフェイスの右上隅に表示される Zabbix インストール名 デフォルトの値は Zabbix Docker です。
ZBX_JAVAGATEWAY_ENABLE	Server Proxy	Zabbix Java ゲートウェイとの通信を有効にして、Java 関連のチェックを収集します。 デフォルトの値は "false" です。
ZBX_ENABLE_SNMP_TRAPS	Server Proxy	SNMP トラップ機能を有効にします。 zabbix-snmptraps インスタンスと共有ボリューム /var/lib/zabbix/snmptraps が Zabbix サーバーまたは Zabbix プロキシに必要です。

ボリューム

イメージでは、いくつかのマウントポイントを使用できます。これらのマウントポイントは Zabbix コンポーネントの種類によって異なります。

ボリューム	説明
Zabbix エージェント	
/etc/zabbix/zabbix_agentd.d	*.conf ファイルを含めて、UserParameter 機能を使用して Zabbix エージェントを拡張できます
/var/lib/zabbix/modules	LoadModule 機能を使用して、追加のモジュールをロードし、Zabbix エージェントを拡張できます。
/var/lib/zabbix/enc	TLS 関連のファイルを格納するために使用されます。これらのファイル名は、ZBX_TLSCAFILE,ZBX_TLSCRLFILE,ZBX_TLSKEY_FILE,ZBX_TLSPSKFILE 環境変数を使用して指定されます。
Zabbix サーバー	
/usr/lib/zabbix/alertscripts	カスタムアラートスクリプトに使用されます。 zabbix_server.conf の AlertScriptsPath パラメータです
/usr/lib/zabbix/externalscripts	外部チェック に使用されます。 zabbix_server.conf の ExternalScripts パラメータです
/var/lib/zabbix/modules	LoadModule 機能を使用して、追加のモジュールをロードし、Zabbix サーバーを拡張できます。
/var/lib/zabbix/enc	TLS 関連のファイルを格納するために使用されます。これらのファイル名は、ZBX_TLSCAFILE,ZBX_TLSCRLFILE,ZBX_TLSKEY_FILE,ZBX_TLSPSKFILE 環境変数を使用して指定されます。
/var/lib/zabbix/ssl/certs	クライアント認証用の SSL クライアント証明書ファイルの格納場所として使用されます。これは、zabbix_server.conf の SSLCertLocation パラメータです。
/var/lib/zabbix/ssl/keys	クライアント認証用の SSL 秘密鍵ファイルの格納場所として使用されます。 zabbix_server.conf の SSLKeyLocation パラメータです。
/var/lib/zabbix/ssl/ssl_ca	SSL サーバー証明書検証用の認証局 (CA) ファイルの格納場所として使用されます。 zabbix_server.conf の SSLCALocation パラメータです
/var/lib/zabbix/snmptraps	ボリュームは、snmptraps.log ファイルの場所として使用されます。これは、zabbix-snmptraps コンテナによって共有され、Zabbix サーバーの新しいインスタンスを作成するときに、volumes_from Docker オプションを使用して継承される可能性があります。共有ボリュームを使用し、ZBX_ENABLE_SNMP_TRAPS 環境変数を 'true' に切り替えることで、SNMP トラップ処理機能を有効にすることができます。新しい MIB ファイルを追加できます。サブディレクトリはサポートしていません。すべての MIB を /var/lib/zabbix/mibs に配置する必要があります。
/var/lib/zabbix/mibs	
Zabbix プロキシ	

ボリューム	説明
/usr/lib/zabbix/externalscripts	外部チェックによって使用されます。zabbix_proxy.conf の ExternalScripts パラメータです。
/var/lib/zabbix/db_data/	ボリュームにより、外部デバイスにデータベースファイルを保存できます。SQLite3 を使用する Zabbix プロキシでのみサポート
/var/lib/zabbix/modules	LoadModule 機能を使用して、追加のモジュールをロードし、Zabbix サーバーを拡張できます。
/var/lib/zabbix/enc	TLS 関連のファイルを格納するために使用されます。これらのファイル名は、ZBX_TLSCAFILE,ZBX_TLSCRLFILE,ZBX_TLSKEY_FILE,ZBX_TLSPSKFILE 環境変数を使用して指定されます。
/var/lib/zabbix/ssl/certs	クライアント認証用の SSL クライアント証明書ファイルの格納場所として使用されます。zabbix_proxy.conf の SSLCertLocation パラメータです。
/var/lib/zabbix/ssl/keys	クライアント認証用の SSL 秘密鍵ファイルの格納場所として使用されます。zabbix_proxy.conf の SSLKeyLocation パラメータです。
/var/lib/zabbix/ssl/ssl_ca	SSL サーバー証明書検証用の認証局 (CA) ファイルの格納場所として使用されます。zabbix_proxy.conf の SSLCALocation パラメータです。
/var/lib/zabbix/snmptraps	ボリュームは、snmptraps.log ファイルの場所として使用されます。これは、zabbix-snmptraps コンテナによって共有され、Zabbix サーバーの新しいインスタンスを作成するときに、volumes_from Docker オプションを使用して継承される可能性があります。共有ボリュームを使用し、ZBX_ENABLE_SNMP_TRAPS 環境変数を 'true' に切り替えることで、SNMP トラップ処理機能を有効にすることができます。新しい MIB ファイルを追加できます。サブディレクトリはサポートしていません。すべての MIB を /var/lib/zabbix/mibs に配置する必要があります。
/var/lib/zabbix/mibs	
Apache2 Web サーバーに基づく Zabbix Web インターフェイス /etc/ssl/apache2	Zabbix Web インターフェイスの HTTPS を有効にすることができます。Apache2 SSL 接続用に準備された ssl.crt ファイルと ssl.key ファイルの 2 つが格納されている必要があります。
Nginx Web サーバーに基づく Zabbix Web インターフェイス /etc/ssl/nginx	Zabbix Web インターフェイスの HTTPS を有効にすることができます。Nginx SSL 接続用に準備された ssl.crt と ssl.key および dhparam.pem が格納されている必要があります。
Zabbix snmp トラップ /var/lib/zabbix/snmptraps	受信した SNMP トラップで名前が付けられた snmptraps.log ログファイルが格納されます。
/var/lib/zabbix/mibs	新しい MIB ファイルを追加できます。サブディレクトリはサポートしていません。すべての MIB を /var/lib/zabbix/mibs に配置する必要があります。

詳細については、Docker Hub の Zabbix 公式リポジトリを参照してください。

使用例

例 1

この例は、MySQL データベースをサポートする Zabbix サーバー、Nginx Web サーバーに基づく Zabbix Web インターフェイス、および Zabbix Java ゲートウェイを実行する方法を示しています。

1. Zabbix コンポーネントコンテナ専用のネットワークを作成します。

```
# docker network create --subnet 172.20.0.0/16 --ip-range 172.20.240.0/20 zabbix-net
```

2. 空の MySQL サーバーインスタンスを開始する

```
# docker run --name mysql-server -t \
-e MYSQL_DATABASE="zabbix" \
-e MYSQL_USER="zabbix" \
-e MYSQL_PASSWORD="zabbix_pwd" \
-e MYSQL_ROOT_PASSWORD="root_pwd" \
--network=zabbix-net \
--restart unless-stopped \
-d mysql:8.0 \
--character-set-server=utf8 --collation-server=utf8_bin \
--default-authentication-plugin=mysql_native_password
```

3. Zabbix Java ゲートウェイインスタンスの起動

```
# docker run --name zabbix-java-gateway -t \
--network=zabbix-net \
--restart unless-stopped \
-d zabbix/zabbix-java-gateway:alpine-5.4-latest
```

4. Zabbix サーバーインスタンスを起動し、作成した MySQL サーバーインスタンスにインスタンスをリンクします。

```
# docker run --name zabbix-server-mysql -t \
-e DB_SERVER_HOST="mysql-server" \
-e MYSQL_DATABASE="zabbix" \
-e MYSQL_USER="zabbix" \
-e MYSQL_PASSWORD="zabbix_pwd" \
-e MYSQL_ROOT_PASSWORD="root_pwd" \
-e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
--network=zabbix-net \
-p 10051:10051 \
--restart unless-stopped \
-d zabbix/zabbix-server-mysql:alpine-5.4-latest
```

Note:

Zabbix サーバーインスタンスは、10051/TCP ポート (Zabbix トラッパー) をホストマシンに公開します。

5. Zabbix Web インターフェイスを起動し、作成した MySQL サーバーと Zabbix サーバーインスタンスにインスタンスをリンクします

```
# docker run --name zabbix-web-nginx-mysql -t \
-e ZBX_SERVER_HOST="zabbix-server-mysql" \
-e DB_SERVER_HOST="mysql-server" \
-e MYSQL_DATABASE="zabbix" \
-e MYSQL_USER="zabbix" \
-e MYSQL_PASSWORD="zabbix_pwd" \
-e MYSQL_ROOT_PASSWORD="root_pwd" \
--network=zabbix-net \
-p 80:8080 \
--restart unless-stopped \
-d zabbix/zabbix-web-nginx-mysql:alpine-5.4-latest
```

Note:

Zabbix Web インターフェイスインスタンスは、80/TCP ポート (HTTP) をホストマシンに公開します。

例 2

この例では、PostgreSQL データベースをサポートする Zabbix サーバー、Nginx Web サーバーに基づく Zabbix Web インターフェイス、および SNMP トラップ機能を実行する方法を示します。

1. Zabbix コンポーネントコンテナ専用のネットワークを作成します。

```
# docker network create --subnet 172.20.0.0/16 --ip-range 172.20.240.0/20 zabbix-net
```

2. 空の PostgreSQL サーバーインスタンスを起動する

```
# docker run --name postgres-server -t \
-e POSTGRES_USER="zabbix" \
-e POSTGRES_PASSWORD="zabbix_pwd" \
```

```
-e POSTGRES_DB="zabbix" \  
--network=zabbix-net \  
--restart unless-stopped \  
-d postgres:latest
```

3. Zabbix ANMP トラップインスタンスの開始

```
# docker run --name zabbix-smnptraps -t \  
-v /zbx_instance/snpmptraps:/var/lib/zabbix/snpmptraps:rw \  
-v /var/lib/zabbix/mibs:/usr/share/snmp/mibs:ro \  
--network=zabbix-net \  
-p 162:1162/udp \  
--restart unless-stopped \  
-d zabbix/zabbix-smnptraps:alpine-5.4-latest
```

Note:

Zabbix snmptrap インスタンスは、162/UDP ポート (SNMP トラップ) をホストマシンに公開します。

4. Zabbix サーバーインスタンスを起動し、作成した PostgreSQL サーバーインスタンスとインスタンスをリンクします。

```
# docker run --name zabbix-server-pgsql -t \  
-e DB_SERVER_HOST="postgres-server" \  
-e POSTGRES_USER="zabbix" \  
-e POSTGRES_PASSWORD="zabbix_pwd" \  
-e POSTGRES_DB="zabbix" \  
-e ZBX_ENABLE_SNMP_TRAPS="true" \  
--network=zabbix-net \  
-p 10051:10051 \  
--volumes-from zabbix-smnptraps \  
--restart unless-stopped \  
-d zabbix/zabbix-server-pgsql:alpine-5.4-latest
```

Note:

Zabbix サーバーインスタンスは、10051/TCP ポート (Zabbix トラッパー) をホストマシンに公開します。

5. Zabbix Web インターフェイスを起動し、作成した PostgreSQL サーバーと Zabbix サーバーインスタンスにインスタンスをリンクします。

```
# docker run --name zabbix-web-nginx-pgsql -t \  
-e ZBX_SERVER_HOST="zabbix-server-pgsql" \  
-e DB_SERVER_HOST="postgres-server" \  
-e POSTGRES_USER="zabbix" \  
-e POSTGRES_PASSWORD="zabbix_pwd" \  
-e POSTGRES_DB="zabbix" \  
--network=zabbix-net \  
-p 443:8443 \  
-p 80:8080 \  
-v /etc/ssl/nginx:/etc/ssl/nginx:ro \  
--restart unless-stopped \  
-d zabbix/zabbix-web-nginx-pgsql:alpine-5.4-latest
```

Note:

Zabbix Web インターフェイスインスタンスは、443/TCP ポート (HTTPS) をホストマシンに公開します。ディレクトリ /etc/ssl/nginx には、必要な名前の証明書が含まれている必要があります。

例 3

この例では、MySQL データベースをサポートする Zabbix サーバー、Nginx Web サーバーに基づく Zabbix Web インターフェイス、Red Hat 8 で podman を使用する Zabbix Java ゲートウェイを実行する方法を示します。

1. zabbix という名前と公開されたポート (Web インターフェイス、Zabbix サーバートラッパー) で新しいポッドを作成します。

```
podman pod create --name zabbix -p 80:8080 -p 10051:10051
```

2. (オプション) zabbix ポッドの場所で Zabbix エージェントコンテナを開始します。


```
podman run --name zabbix-agent \
  -eZBX_SERVER_HOST="127.0.0.1,localhost" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-agent-50:latest
```

3. ホストに `./mysql/` ディレクトリを作成し、Oracle MySQL サーバー 8.0 を起動します。

```
podman run --name mysql-server -t \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e MYSQL_ROOT_PASSWORD="root_pwd" \
  -v ./mysql/:/var/lib/mysql/:Z \
  --restart=always \
  --pod=zabbix \
  -d mysql:8.0 \
  --character-set-server=utf8 --collation-server=utf8_bin \
  --default-authentication-plugin=mysql_native_password
```

3. Zabbix サーバーコンテナを起動します。

```
podman run --name zabbix-server-mysql -t \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e MYSQL_ROOT_PASSWORD="root_pwd" \
  -e ZBX_JAVAGATEWAY="127.0.0.1" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-50
```

4. Zabbix Java ゲートウェイコンテナを起動します。

```
podman run --name zabbix-java-gateway -t \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-java-gateway-50
```

5. Zabbix Web インターフェイスコンテナを起動します。

```
podman run --name zabbix-web-mysql -t \
  -e ZBX_SERVER_HOST="127.0.0.1" \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e MYSQL_ROOT_PASSWORD="root_pwd" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-50
```

Note:

Pod zabbix は zabbix-web-mysql コンテナの 8080/TCP からホストマシンに 80/TCP ポート (HTTP) を公開します。

Docker Compose Zabbix は、Docker でマルチコンテナ Zabbix コンポーネントを定義および実行するための構成ファイルも提供します。これらの構成ファイルは、github.com の Zabbix docker 公式リポジトリ (<https://github.com/zabbix/zabbix-docker>) で入手できます。これらの構成ファイルは例として追加され、オーバーロードされています。たとえば、MySQL および SQLite3 をサポートするプロキシが含まれています。

構成ファイルにはいくつかの異なるバージョンがあります。

ファイル名	説明
<code>docker-compose_v3_alpine_mysql_latest.yml</code>	MySQL データベースをサポートする Alpine Linux で Zabbix 5.4 コンポーネントの最新バージョンを実行します。

ファイル名	説明
docker-compose_v3_alpine_mysql_latest.zabbix.yml	構成ファイルは、MySQL データベースをサポートする Alpine Linux で Zabbix コンポーネントを実行します。
docker-compose_v3_alpine_postgresql_latest.zabbix.yml	構成ファイルは、PostgreSQL データベースをサポートする Alpine Linux で Zabbix 5.4 コンポーネントの最新バージョンを実行します。
docker-compose_v3_alpine_mysql_local.zabbix.yml	構成ファイルは、MySQL データベースをサポートする Alpine Linux で Zabbix コンポーネントを実行します。
docker-compose_v3_centos_mysql_local.zabbix.yml	構成ファイルは、MySQL データベースをサポートする CentOS 8 で Zabbix 5.4 コンポーネントの最新バージョンを実行します。
docker-compose_v3_centos_mysql_latest.zabbix.yml	構成ファイルは、MySQL データベースをサポートする CentOS 8 で Zabbix コンポーネントを実行します。
docker-compose_v3_centos_postgresql_latest.zabbix.yml	構成ファイルは、PostgreSQL データベースをサポートする CentOS 8 で Zabbix 5.4 コンポーネントの最新バージョンを実行します。
docker-compose_v3_centos_postgresql_local.zabbix.yml	構成ファイルは、MySQL データベースをサポートする CentOS 8 で Zabbix コンポーネントを実行します。
docker-compose_v3_ubuntu_mysql_local.zabbix.yml	構成ファイルは、MySQL データベースをサポートする Ubuntu 20.04 で Zabbix 5.4 コンポーネントの最新バージョンを実行します。
docker-compose_v3_ubuntu_mysql_latest.zabbix.yml	構成ファイルは、MySQL データベースをサポートする Ubuntu 20.04 で Zabbix コンポーネントを実行します。
docker-compose_v3_ubuntu_postgresql_latest.zabbix.yml	構成ファイルは、PostgreSQL データベースをサポートする Ubuntu 20.04 で最新バージョンの Zabbix 5.4 コンポーネントを実行します。
docker-compose_v3_ubuntu_postgresql_local.zabbix.yml	構成ファイルは、MySQL データベースをサポートする Ubuntu 20.04 で Zabbix コンポーネントを実行します。

Attention:

Docker Compose ファイルは、Docker Compose のバージョン 3 をサポートしています。

ストレージ

Docker Compose は、compose ファイルを使って Zabbix コンポーネントを実行すると、compose ファイルのあるフォルダに `zbx_env` ディレクトリが作成されます。このディレクトリには、上記の **Volumes** セクションで説明したものと同一構造の `zbx_env` データベースを格納するためのディレクトリが作成されます。

また、`/etc/localtime` と `/etc/timezone` ファイル用の読み込み専用ボリュームも用意されます。

環境ファイル

github.com の compose ファイルと同じディレクトリに、compose ファイル内の各コンポーネントのデフォルトの環境変数が書かれた `zbx_env` ファイルがあります。これらの環境ファイルは、`.env_<component type>` のような名前になっています。

例

例 1

```
# git checkout 5.4
# docker-compose -f ./docker-compose_v3_alpine_mysql_latest.yml up -d
```

このコマンドは、Zabbix コンポーネントごとに最新の Zabbix 5.4 イメージをダウンロードし、デタッチモードで実行します。

Attention:

github.com の公式 Zabbix リポジトリから構成ファイルを含む `.env_<type of component>` ファイルをダウンロードしてください

例 2

```
# git checkout 5.4
# docker-compose -f ./docker-compose_v3_ubuntu_mysql_local.yml up -d
```

このコマンドは、ベースイメージの Ubuntu 20.04 (focal) をダウンロードし、Zabbix 5.4 コンポーネントをローカルでビルドしてデタッチモードで実行します。

6 Web インタフェースのインストール

このセクションでは、Zabbix Web インタフェースのインストール方法をステップバイステップで説明します。
 Zabbix フロントエンドは PHP で書かれているため、実行するには PHP がサポートされている Web サーバが必要です。

Note:

You can find out more about setting up SSL for Zabbix frontend by referring to these [best practices](#).

Welcome 画面

Zabbix のフロントエンドの URL をブラウザで開きます。
 Zabbix をパッケージからインストールした場合、URL は以下のようになります。

- Apache の場合: http://<server_ip_or_name>/zabbix
- Nginx の場合: http://<server_ip_or_name>

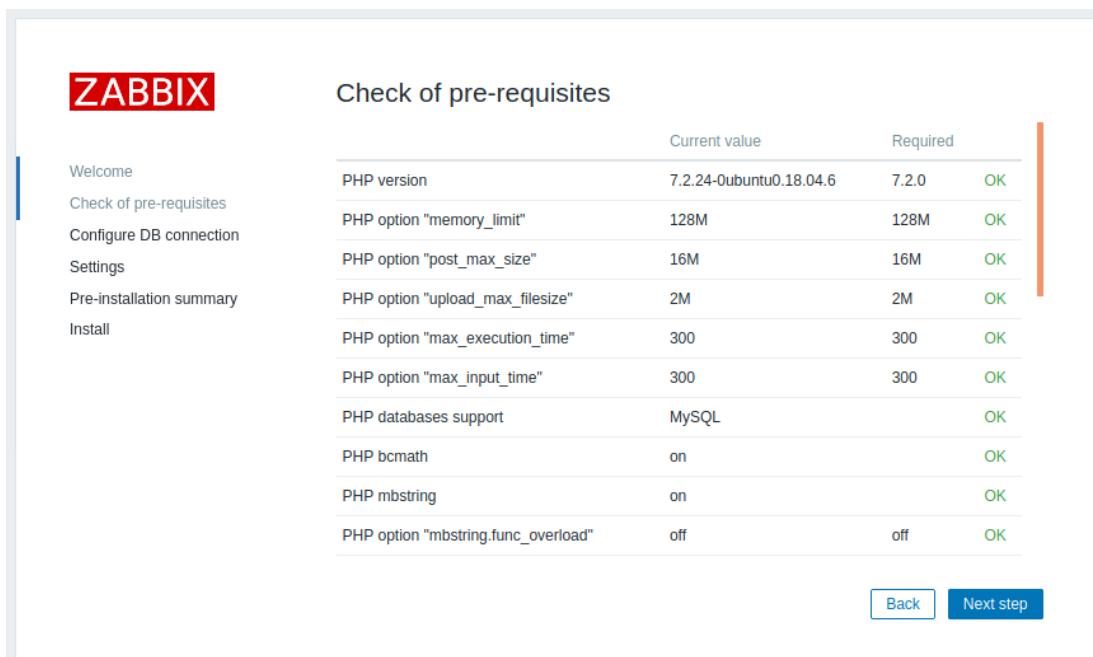
フロントエンドインストールウィザードの最初の画面が表示されます。

Default language ドロップダウンメニューを使用して、システムのデフォルト言語を変更し、
 選択した言語でのインストールプロセスを続行します (オプション)
 より詳細な情報は、[Installation of additional frontend languages](#)を参照してください。



前提条件の確認

すべてのソフトウェアの前提条件が満たされていることを確認します。



Pre-requisite	Minimum value	Description
PHP version	7.2.5	
PHP memory_limit option	128MB	In php.ini: memory_limit = 128M
PHP post_max_size option	16MB	In php.ini: post_max_size = 16M
PHP upload_max_filesize option	2MB	In php.ini: upload_max_filesize = 2M
PHP max_execution_time option	300 seconds (values 0 and -1 are allowed)	In php.ini: max_execution_time = 300
PHP max_input_time option	300 seconds (values 0 and -1 are allowed)	In php.ini: max_input_time = 300
PHP session.auto_start option	must be disabled	In php.ini: session.auto_start = 0
Database support	One of: MySQL, Oracle, PostgreSQL.	One of the following modules must be installed: mysql, oci8, pgsq
bcmath		php-bcmath
mbstring		php-mbstring
PHP mbstring.func_overload option	must be disabled	In php.ini: mbstring.func_overload = 0
sockets		php-net-socket. Required for user script support.
gd	2.0.28	php-gd. PHP GD extension must support PNG images (--with-png-dir), JPEG (--with-jpeg-dir) images and FreeType 2 (--with-freetype-dir).
libxml	2.6.15	php-xml
xmlwriter		php-xmlwriter
xmlreader		php-xmlreader
ctype		php-ctype
session		php-session
gettext		php-gettext Since Zabbix 2.2.1, the PHP gettext extension is not a mandatory requirement for installing Zabbix. If gettext is not installed, the frontend will work as usual, however, the translations will not be available.

オプションの前提条件もリストに含まれることがあります。
 失敗したオプションの前提条件はオレンジ色で表示され、ステータスは Warning です。
 オプションの前提条件が不合格でも、セットアップは続行可能です。

Attention:

Apache のユーザやユーザグループを変更する必要がある場合は、セッションフォルダへのアクセス権を確認する必要があります。
 場合によっては、Zabbix のセットアップを続行できないことがあります。

DB 接続の設定

データベースへの接続に関する詳細を入力します。
 設定時点で Zabbix データベースが作成されている必要があります。

ZABBIX

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type:

Database host:

Database port: 0 - use default port

Database name:

Store credentials in: Plain text HashiCorp Vault

User:

Password:

Database TLS encryption: Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).

Database TLS encryption オプションがチェックされている場合、**configuring the TLS connection**用の追加フィールドが表示されま
す。(MySQL または PostgreSQL のみ)

資格情報の保存に HashiCorp Vault オプションを選択した場合、
Vault API エンドポイント、シークレットパス、および認証ト
ークンを指定するためのフィールドが追加されます。

ZABBIX

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type:

Database host:

Database port: 0 - use default port

Database name:

Store credentials in: Plain text HashiCorp Vault

Vault API endpoint:

Vault secret path:

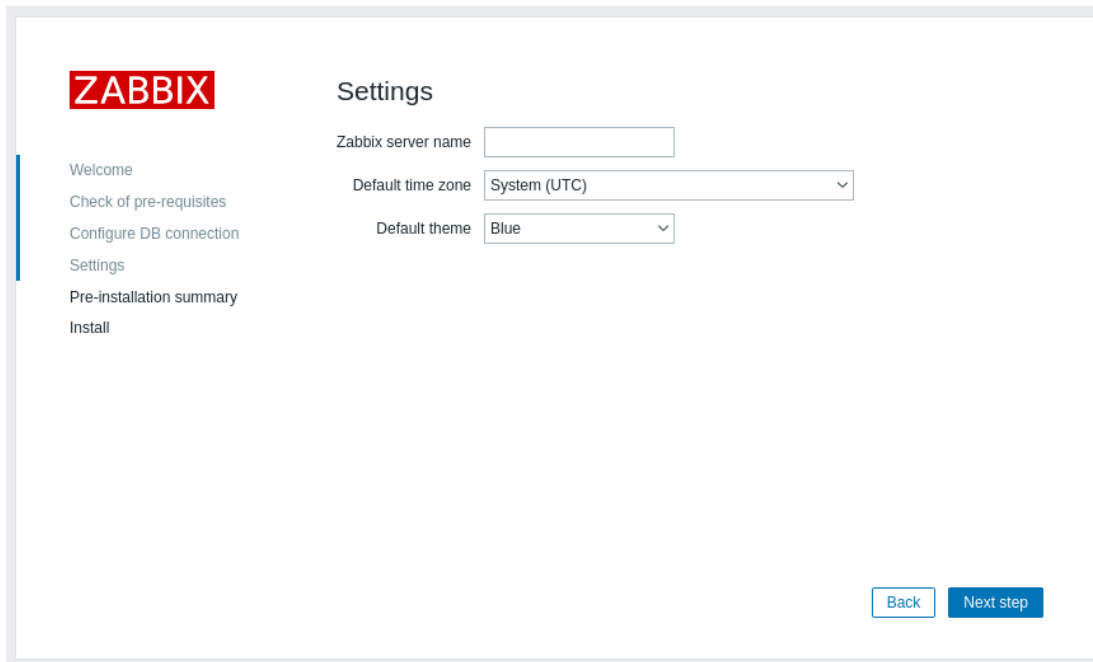
Vault authentication token:

Database TLS encryption:

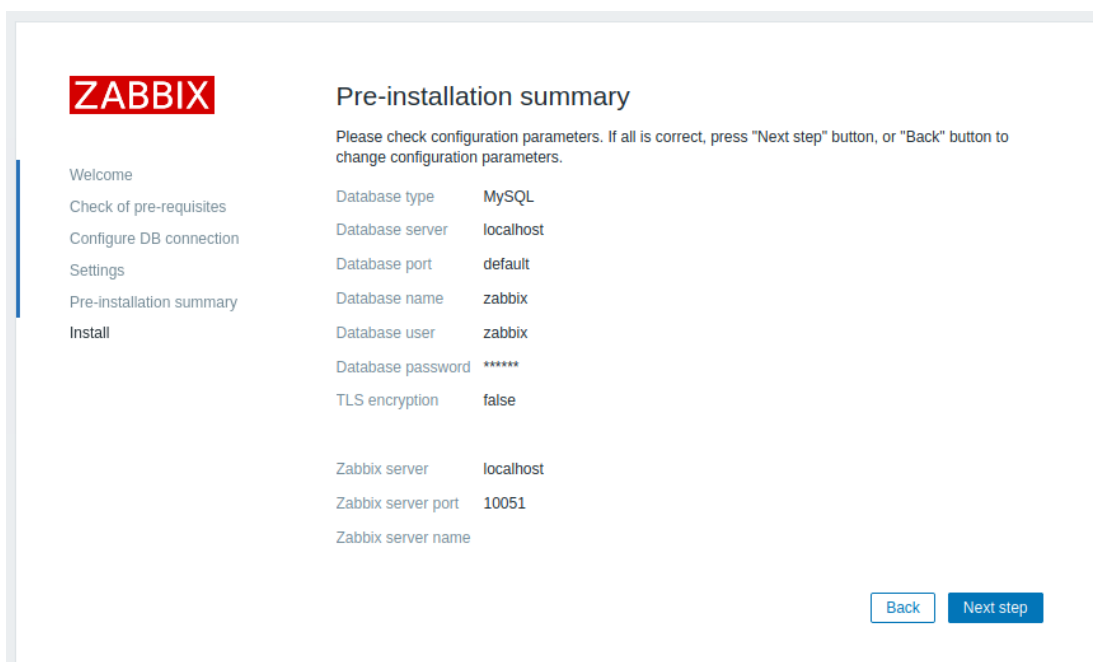
設定

Zabbix server 名の入力は任意ですが、入力した場合、メニューバーとページタイトルに表示されます。

デフォルトのtime zoneとフロントエンドのテーマを設定します。



インストール前の概要
設定の概要を確認します。



インストール

Zabbix をソースからインストールする場合、設定ファイルをダウンロードし、Zabbix の PHP ファイルを
 コピーした Web サーバの HTML ドキュメントサブディレクトリの conf/以下に配置します。

ZABBIX

Install

- Welcome
- Check of pre-requisites
- Configure DB connection
- Settings
- Pre-installation summary
- Install

Details ▲ Cannot create the configuration file.
Unable to create the configuration file.

Alternatively, you can install it manually:

1. Download the configuration file
2. Save it as "/var/www/html/zabbix/conf/zabbix.conf.php"

[Back](#) [Finish](#)

Opening zabbix.conf.php

You have chosen to open:

- zabbix.conf.php**
which is: PHP script (418 bytes)
from: http://192.168.3.194

What should Firefox do with this file?

Open with

Save File

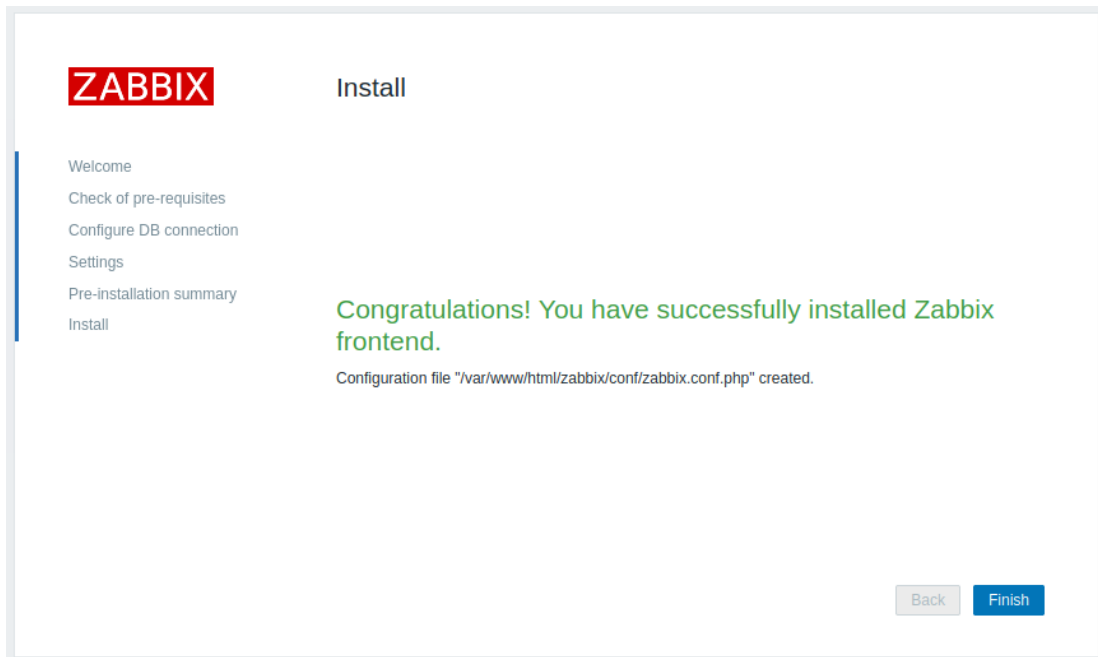
Do this automatically for files like this from now on.

[Cancel](#) [OK](#)

Note:

ウェブサーバーユーザに conf/ ディレクトリへの書き込み権限がある場合、
 設定ファイルは自動的に保存され、次の作業に進むことができます。

インストールを終了します。



ログイン

Zabbix フロントエンドの準備は完了です。
 デフォルトのユーザ名は **Admin**、パスワードは **zabbix** です。

getting started with Zabbixに進みます。

Debian/Ubuntu のフロントエンドのインストール

概要

バージョン 5.0 から、Zabbix フロントエンドは PHP バージョン 7.2 以降が必要です。
 残念ながら、Debian と Ubuntu の古いバージョンでは、PHP バージョン 7.2 以下しか提供されていません。

ディストリビューション別対応 PHP バージョン

Distribution	PHP Version
Debian 10 (buster)	7.3

Distribution	PHP Version
Debian 9 (stretch)	7.0
Debian 8 (jessie)	5.6
Ubuntu 20.04 (focal)	7.4
Ubuntu 18.04 (bionic)	7.2
Ubuntu 16.04 (xenial)	7.0
Ubuntu 14.04 (trusty)	5.5
Raspbian 10 (buster)	7.3
Raspbian 8 (stretch)	7.0

stretch, jessie, xenial および trusty* ディストリビューションでは、PHP 7.2 と依存関係がないため
 Zabbix フロントエンド 5.0 以降を簡単にインストールすることができません。
 これを考慮し、zabbix-frontend-php パッケージは、zabbix-frontend-PHP に置き換えられました。
 パッケージは zabbix-frontend-php-deprecated パッケージに置き換まりました。
 主な違いは、PHP や Web サーバパッケージへ直接依存しないことです。
 このため、ユーザはこれらのパッケージを自分で用意することができます（用意しなければなりません）。

インストールの方法ですが、
 zabbix-frontend-php-deprecated パッケージを単独でインストールしても、フロントエンドは動作しません。
 Web サーバおよび PHP 7.2 とそのモジュールは手動でインストールする必要があります。（PPA を使用する/PHP をソースからビルドする）
 我々は特定の方法を推奨するものではありません。

Note:

Debian/Ubuntu の古いバージョンで PHP 7.2 以降を入手する公式な方法は、buster/bionic にアップグレードすることです。

Zabbix フロントエンドに必要な PHP モジュールは php-gd、php-bcmath、php-mbstring、php-xml、php-ldap、php-json です。

7 アップグレード手順

概要

このセクションでは Zabbix**6.0** へのアップグレード情報を提供します。

- パッケージ使用
 - Red Hat Enterprise Linux
 - Debian/Ubuntu
- ソース

Zabbix6.0.x への直接アップグレードは Zabbix**5.4.x**、**5.2.x、5.0.x、4.4.x、4.2.x、4.0.x、3.4.x、3.2.x、3.0.x、2.4.x、2.2.x、2.0*.x から可能です。Zabbix 1.8 以前のバージョンからのアップグレードについては、各バージョンの Zabbix のドキュメントを参照してください。

Note:

外部ソフトウェアがアップグレードした Zabbix のバージョンと互換性がない場合、アップグレード後に Zabbix の一部のサーブパーティソフトウェア統合が影響を受ける可能性があることに注意してください。

パッケージでアップグレード

概要

このセクションでは、Zabbix が提供する公式 RPM および DEB パッケージを使用して**アップグレード**を成功させるために必要な手順を説明します。

- Red Hat Enterprise Linux
- Debian/Ubuntu

OS リポジトリからの Zabbix パッケージ

多くの場合、OS ディストリビューション（特に Debian ベースのディストリビューション）は独自の Zabbix パッケージを提供します。

これらのパッケージは Zabbix でサポートされていないことに注意してください。たいていの場合、これらのパッケージは古く、最新の機能とバグ修正が含まれていません。repo.zabbix.comのパッケージのみが公式にサポートされています。

OS ディストリビューションが提供するパッケージからアップグレードする場合（またはある時点でインストールした場合）、次の手順に従って公式の Zabbix パッケージに切り替えます。

1. 最初に古いパッケージをアンインストールしてください。
2. アンインストール後にファイルが残っていないか確認してください。
3. Zabbix が提供する **インストール手順**に従って公式パッケージをインストールします。

インストール済みのパッケージが破損する可能性があるため、直接更新しないでください。

1 Red Hat Enterprise Linux

概要

このセクションでは、Red Hat Enterprise Linux 用の公式 Zabbix パッケージを使用して、Zabbix 5.4.x から Zabbix 6.0.x へのアップグレードを成功させるために必要な手順を説明します。

Zabbix エージェントのアップグレードは必須ではありません（ただし推奨します）が、Zabbix サーバーとプロキシは、**同じメジャーバージョン**であることが必要です。したがって、サーバーとプロキシの設定では、Zabbix サーバーとすべてのプロキシを停止してアップグレードする必要があります。サーバーのアップグレード中にプロキシを実行し続けたとしても、プロキシのアップグレード時に古いデータが破棄され、プロキシの構成がサーバーと同期されるまで新しいデータが収集されないため、メリットがありません。

プロキシ上の SQLite データベースは、データベースのアップグレードがサポートされておらず、データベースファイルを手動で削除する必要があります。アップグレード前のプロキシからの履歴データは失われることに注意してください。プロキシが初めて起動され、SQLite データベースファイルが見つからない場合、プロキシは自動的に作成します。

データベースのサイズによっては、バージョン 6.0 へのデータベースアップグレードに時間がかかる場合があります。

Warning:

アップグレードを開始する前にアップグレードノートを必ずお読みください！

以下のアップグレードノートがあります：

アップグレード元	アップグレードノート全文	バージョン間での最も重要な変更
5.4.x	Zabbix 6.0	要求されるデータベースの最低バージョンの上昇 古いデータベースでサーバーとプロキシは起動不能 データベースの構造変更のため監査ログのレコードが消える
5.2.x	Zabbix 5.4 Zabbix 6.0	要求されるデータベースの最低バージョンが上昇 集計アイテムが個別のタイプから削除
5.0.x LTS	Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	要求される PHP の最低バージョンが 7.2.0 から 7.2.5 に変更
4.4.x	Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	IBM DB2 のサポートを削除 要求される PHP の最低バージョンを 5.4.0 から 7.2.0 に変更 要求されるデータベースの最低バージョンが上昇
4.2.x	Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	Zabbix の PHP ファイルのディレクトリ変更 jabber、Ez Texting のメディアタイプ削除
4.0.x LTS	Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	古いプロキシからアップグレードされたサーバーへのデータ通知不可 新しいエージェントは古い Zabbix サーバーとの動作不可
3.4.x	Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	'libpthread' および'zlib' ライブラリが必須 プレーンテキストのプロトコルが削除されヘッダー必須 バージョン 1.4 以前の Zabbix エージェントのサポート終了 パシブプロキシの設定に Server パラメーターが必須

アップグレード元	アップグレードノート 全文	バージョン間での最も重要な変更
3.2.x	Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	Zabbix サーバーと Web インターフェースのバックエンドデータベースとして SQLite のサポートは削除 POSIX 拡張の代わりに Perl 互換正規表現 (PCRE) をサポート Zabbix サーバーに 'libpcre' および 'libevent' ライブラリが必須 'nowait' フラグが設定されていない UserParameter、リモートコマンド、system.run[] アイテムは、Zabbix サーバーで実行されるスクリプトと同様に終了コードのチェックを追加 Zabbix Java ゲートウェイは新しい機能をサポートするためアップグレード必須
3.0.x LTS	Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	ヒストリテーブルのサイズによっては、データベースのアップグレード処理に時間がかかる
2.4.x	Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	要求される PHP の最小バージョンが 5.3.0 から 5.4.0 に変更 エージェントの LogFile パラメータが必須
2.2.x LTS	Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	ノードベースの分散監視削除
2.0.x	Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	要求される最小の PHP バージョンが 5.1.6 から 5.3.0 に変更 Zabbix サーバーが MySQL データベースと正しく連携するには、大文字と小文字を区別する文字コードセット utf8 と照合順序 utf8_bin が必須。 データベース作成スクリプト参照 PHP 拡張機能の 'mysql' の代わりに 'mysqli' を使用するよう変更

6.0 用のシステム要件でチェックしてください。

Note:

アップグレード中に並行して SSH のセッションを実行し、一方でアップグレード手順を実行し、もう一方でサーバーやプロキシのログを監視すると便利な場合があります。例えば、`tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログとエラー発生をリアルタイムで把握することができます。これは、本番環境のインスタンスにとって重要です。

アップグレード手順

1 Zabbix プロセスの停止

Zabbix サーバーを停止して、データベースに新しいデータが挿入されていないことを確認します。

```
# systemctl stop zabbix-server
```

プロキシもアップグレードする場合は、プロキシも停止します。

```
# systemctl stop zabbix-proxy
```

Attention:

アップグレードされたサーバーにアップグレードされていないプロキシがデータを報告することはできなくなりました。このアプローチは Zabbix によって推奨もサポートもされていませんでしたが、現在は正式に無効になっています。

2 既存 Zabbix データベースのバックアップ

これは非常に重要なステップです。データベースのバックアップがあることを確認してください。アップグレード手順が失敗（ディスク容量の不足、電源オフ、予期しない問題）した場合に役立ちます。

3 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

Zabbix バイナリ、設定ファイル、PHP ファイルディレクトリのバックアップコピーを作成します。

設定ファイル:

```
# mkdir /opt/zabbix-backup/  
# cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/  
# cp /etc/httpd/conf.d/zabbix.conf /opt/zabbix-backup/
```

PHP ファイルと Zabbix バイナリ:

```
# cp -R /usr/share/zabbix/ /opt/zabbix-backup/  
# cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 リポジトリ構成パッケージ更新

アップグレードを続行するには、現在のリポジトリパッケージを更新する必要があります。

```
# rpm -Uvh https://repo.zabbix.com/zabbix/6.0/rhel/8/x86_64/zabbix-release-6.0-1.el8.noarch.rpm
```

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

```
# yum upgrade zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

PostgreSQL を使用している場合は、コマンドの `mysql` を `pgsql` に置き換えます。プロキシをアップグレードする場合は、コマンドの `server` を `proxy` に置き換えてください。エージェント 2 をアップグレードする場合は、コマンドの `zabbix-agent` を `zabbix-agent2` に置き換えてください。

Apache を使用している Web インターフェースを **RHEL8** で正しくアップグレードするには、次を実行し、

```
# yum install zabbix-apache-conf
```

このファイルに必要な変更を加えます。

RHEL 7 で Web インターフェースをアップグレードするには **ディストリビューション固有の手順**に従ってください。(PHP 7.2 以降をインストールするには追加の手順が必要です)

6 コンポーネント構成パラメーターの確認

必須の変更の詳細については、アップグレードノートを参照してください。

7 Zabbix プロセスの起動

更新された Zabbix コンポーネントを起動します。

```
# systemctl start zabbix-server  
# systemctl start zabbix-proxy  
# systemctl start zabbix-agent  
# systemctl start zabbix-agent2
```

8 Web ブラウザのクッキーとキャッシュのクリア

アップグレード後、Zabbix の Web インターフェースが正しく機能するために、Web ブラウザのクッキーとキャッシュのクリアが必要になる場合があります。

マイナーバージョンのアップグレード

6.0.x はマイナーバージョンのアップグレードが可能です。(たとえば 6.0.1 から 6.0.3) マイナーバージョンのアップグレードは簡単です。

Zabbix 全体のマイナーバージョンをアップグレードするには、次のコマンドを実行します。

```
$ sudo yum upgrade 'zabbix-*'
```

Zabbix サーバーのマイナーバージョンをアップグレードするには、次のコマンドを実行します。

```
$ sudo yum upgrade 'zabbix-server-*'
```

Zabbix エージェントのマイナーバージョンをアップグレードするには、次のコマンドを実行します。

```
$ sudo yum upgrade 'zabbix-agent-*'
```

エージェント 2 の場合は

```
$ sudo yum upgrade 'zabbix-agent2-*'
```

これらのコマンドでは「upgrade」の代わりに「update」を使用可能です。「upgrade」は不要になったパッケージの削除を自動で行いますが「update」はそれらを保持します。

2 Debian/Ubuntu

概要

このセクションでは Debian/Ubuntu 用の公式 Zabbix パッケージを使用して、Zabbix 5.4.x から Zabbix 6.0.x へのアップグレードを成功させるために必要な手順を説明します。

Zabbix エージェントのアップグレードは必須ではありませんが推奨します。Zabbix サーバーとプロキシは同じメジャーバージョンである必要があります。したがってサーバープロキシ設定では、Zabbix サーバーとすべてのプロキシを停止してアップグレードする必要があります。サーバーのアップグレード中にプロキシを実行し続けても、プロキシのアップグレード中に古いデータが破棄され、プロキシ構成がサーバーと同期されるまで新しいデータが収集されないため、メリットがありません。

プロキシ上の SQLite データベースは、データベースのアップグレードがサポートされておらず、データベースファイルを手動で削除する必要があるため、アップグレード前のプロキシからの履歴データは失われることに注意してください。プロキシが初めて起動され、SQLite データベースファイルが見つからない場合、プロキシはそれを自動的に作成します。

データベースのサイズによっては、バージョン 6.0 へのデータベースアップグレードに時間がかかる場合があります。

Warning:

アップグレードする前にアップグレードノートを必ずお読みください。

次のアップグレードノートが利用可能です。

アップグレード元	アップグレードノート全文	バージョン間の重要な変更
5.4.x	For: Zabbix 6.0	要求データベースバージョン上昇 古いデータベースの場合、サーバー/プロキシは起動不能に データベース構造の変更により監査ログレコードが失効
5.2.x	For: Zabbix 5.4 Zabbix 6.0	要求データベースバージョン上昇 集計アイテムが個別のタイプから削除
5.0.x LTS	For: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	要求 PHP バージョンが 7.2.0 から 7.2.5 に
4.4.x	For: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	IBMDB2 のサポート終了 要求 PHP バージョンが 5.4.0 から 7.2.0 に 要求データベースバージョン上昇 ZabbixPHP ファイルディレクトリ変更
4.2.x	For: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	Jabber、EZ Texting メディアタイプ削除

アップグレード元	アップグレードノート全文	バージョン間の重要な変更
4.0.x LTS	For: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	古いプロキシからアップグレードされたサーバーへののにデータ報告が不能に 新しいエージェントは古い Zabbix サーバーと連携不能に
3.4.x	For: Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	「libpthread」および「zlib」ライブラリが必須に プレーンテキストプロトコルのサポートが削除されヘッダーが必須に バージョン 1.4 以前の Zabbix エージェントのサポート終了 パッシブプロキシ構成のサーバーパラメータが必須に
3.2.x	For: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	Zabbix サーバー/フロントエンドのバックエンドデータベースが削除されたため、SQLite がサポートされます。 POSIX 拡張の代わりに、Perl 互換正規表現 (PCRE) がサポートされます。 「libpcre」および「libevent」ライブラリが必須に 「nowait」フラグのないユーザーパラメータ、リモートコマンド、system.run[] アイテム、および Zabbix サーバーで実行されるスクリプトの終了コードチェックが追加 Zabbix Java ゲートウェイは、新しい機能をサポートするためにアップグレードする必要があります。
3.0.x LTS	For: Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	履歴テーブルのサイズによっては、データベースのアップグレードに時間がかかる場合があります。
2.4.x	For: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	必須 PHP バージョンが 5.3.0 から 5.4.0 に LogFile エージェントパラメータの指定が必須に
2.2.x LTS	For: Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	ノードベースの分散監視削除

アップグレード元	アップグレードノート全文	バージョン間の重要な変更
2.0.x	For: Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	必須 PHP バージョンが 5.1.6 から 5.3.0 に Zabbix サーバーが MySQL データベースと正しく連携するには、 大文字と小文字を区別するデータベースおよび文字セット utf8 と utf8_bin の照合が必要に。データベース作成スクリプトを参 照してください。 PHP 拡張機能'mysql' の代わりに'mysqli' が必須に

6.0 の要件を確認することもできます。

Note:

アップグレード中に並列で 2 つの SSH セッションを実行し、一方でアップグレード手順を実行し、もう一方でサーバー/プロキシのログを監視すると便利な場合があります。たとえば、2 番目の SSH セッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログファイルエントリと発生する可能性のあるエラーがリアルタイムで表示されます。これは、実稼働インスタンスにとって重要な場合があります。

アップグレード手順

1 Zabbix プロセスの停止

Zabbix サーバーを停止して、データベースに新しいデータが挿入されていないことを確認します。

```
# service zabbix-server stop
```

プロキシもアップグレードする場合は、プロキシも停止します。

```
# service zabbix-proxy stop
```

2 既存 Zabbix データベースのバックアップ

これは非常に重要なステップです。データベースのバックアップがあることを確認してください。アップグレード手順が失敗（ディスク容量の不足、電源オフ、予期しない問題）した場合に役立ちます。

3 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

Zabbix バイナリ、設定ファイル、PHP ファイルディレクトリのバックアップコピーを作成します。

設定ファイル:

```
# mkdir /opt/zabbix-backup/
# cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
# cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
```

PHP ファイルと Zabbix バイナリ:

```
# cp -R /usr/share/zabbix/ /opt/zabbix-backup/
# cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 リポジトリ構成パッケージ更新

アップグレードを続行するには、現在のリポジトリパッケージをアンインストールする必要があります。

```
# rm -Rf /etc/apt/sources.list.d/zabbix.list
```

新しいリポジトリパッケージをインストールします。

Debian 11 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release_6.0-1+debian11_
# dpkg -i zabbix-release_6.0-1+debian11_all.deb
```

Debian 10 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release_6.0-1+debian10_
# dpkg -i zabbix-release_6.0-1+debian10_all.deb
```

Debian 9 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release_6.0-1+debian9_a
# dpkg -i zabbix-release_6.0-1+debian9_all.deb
```

Ubuntu 20.04 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-1+ubuntu20.
# dpkg -i zabbix-release_6.0-1+ubuntu20.04_all.deb
```

Ubuntu 18.04 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-1+ubuntu18.
# dpkg -i zabbix-release_6.0-1+ubuntu18.04_all.deb
```

Ubuntu 16.04 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-1+ubuntu16.
# dpkg -i zabbix-release_6.0-1+ubuntu16.04_all.deb
```

Ubuntu 14.04 では次の通りです:

```
# wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-1+ubuntu14.
# dpkg -i zabbix-release_6.0-1+ubuntu14.04_all.deb
```

リポジトリ情報を更新します。

```
# apt-get update
```

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

```
# apt-get install --only-upgrade zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

PostgreSQL を使用している場合は、コマンドの `mysql` を `pgsql` に置き換えます。プロキシをアップグレードする場合は、コマンドの `server` を `proxy` に置き換えてください。エージェント 2 をアップグレードする場合は、コマンドの `zabbix-agent` を `zabbix-agent2` に置き換えてください。

Apache を使用している Web フロントエンドを正しくアップグレードするには、次を実行します

```
# apt-get install zabbix-apache-conf
```

ディストリビューション **Debian10 (buster)** / **Ubuntu 18.04 (bionic)** / **Raspbian 10 (buster)** より前のバージョンは、Zabbix フロントエンド 5.0 に必要な PHP7.2 以降を提供していません。古いディストリビューションへの Zabbix フロントエンドのインストールについては、[情報](#)を参照してください。

6 コンポーネント設定パラメーター確認

必須の変更の詳細については (あれば) アップグレードノートを参照してください。

新しいオプションのパラメーターについては、[What's new](#)セクションを参照してください。

7 Zabbix プロセスの開始

更新された Zabbix コンポーネントを起動します。

```
# service zabbix-server start
# service zabbix-proxy start
# service zabbix-agent start
# service zabbix-agent2 start
```

8 ウェブブラウザのクッキーとキャッシュのクリア

アップグレード後、Zabbix ウェブインターフェイスが正しく機能するために、ウェブブラウザのクッキーとキャッシュのクリアが必要になる場合があります。

マイナーバージョンアップグレード

6.0.x はマイナーバージョンアップグレード可能です。(たとえば 6.0.1 から 6.0.3) マイナーバージョンアップグレードは簡単です。

Zabbix 全体をマイナーバージョンアップグレードするには、次のコマンドを実行します。

```
$ sudo apt install --only-upgrade 'zabbix.*'
```

Zabbix サーバーをマイナーバージョンアップグレードするには、次のコマンドを実行します。

```
$ sudo apt install --only-upgrade 'zabbix-server.*'
```


Zabbix エージェントをマイナーバージョンアップグレードするには、次のコマンドを実行します。

```
$ sudo apt install --only-upgrade 'zabbix-agent.*'
```

エージェント 2 の場合は

```
$ sudo apt install --only-upgrade 'zabbix-agent2.*'
```

ソースからのアップグレード

概要

このセクションでは Zabbix の公式ソースを使用して、Zabbix5.4.x から Zabbix6.0.x へのアップグレードを成功させるために必要な手順を説明します。

Zabbix エージェントのアップグレードは必須ではありませんが推奨します。Zabbix サーバーとプロキシは同じメジャーバージョンである必要があります。したがってサーバープロキシ設定では、Zabbix サーバーとすべてのプロキシを停止してアップグレードする必要があります。サーバーのアップグレード中にプロキシを実行し続けても、プロキシのアップグレード中に古いデータが破棄され、プロキシ構成がサーバーと同期されるまで新しいデータが収集されないため、メリットがありません。

Attention:

アップグレードされたサーバーにアップグレードされていないプロキシがデータを報告することはできなくなりました。このアプローチは Zabbix によって推奨もサポートもされていませんでしたが、現在は正式に無効になっています。

プロキシ上の SQLite データベースは、データベースのアップグレードがサポートされておらず、データベースファイルを手動で削除する必要があるため、アップグレード前のプロキシからの履歴データは失われることに注意してください。プロキシが初めて起動され、SQLite データベースファイルが見つからない場合、プロキシはそれを自動的に作成します。

データベースのサイズによっては、バージョン 6.0 へのデータベースアップグレードに時間がかかる場合があります。

Warning:

アップグレードする前にアップグレードノートを必ずお読みください。

次のアップグレードノートが利用可能です。

アップグレード元	アップグレードノート全文	バージョン間の重要な変更
5.4.x	For: Zabbix 6.0	要求データベースバージョン上昇 古いデータベースの場合、サーバー/プロキシは起動不能に データベース構造の変更により監査ログレコードが失効
5.2.x	For: Zabbix 5.4 Zabbix 6.0	要求データベースバージョン上昇 累計アイテムが個別のタイプから削除
5.0.x LTS	For: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	要求 PHP バージョンが 7.2.0 から 7.2.5 に
4.4.x	For: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	IBMDB2 のサポート終了 要求 PHP バージョンが 5.4.0 から 7.2.0 に 要求データベースバージョン上昇 ZabbixPHP ファイルディレクトリ変更
4.2.x	For: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	Jabber、EZ Texting メディアタイプ削除
4.0.x LTS	For: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	古いプロキシからアップグレードされたサーバーへのデータ報告が不能に 新しいエージェントは古い Zabbix サーバーと連携不能に

アップグレード元	アップグレードノート全文	バージョン間の重要な変更
3.4.x	For: Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	「libpthread」および「zlib」ライブラリが必須に プレーンテキストプロトコルのサポートが削除されヘッダーが 必須に バージョン 1.4 以前の Zabbix エージェントのサポート終了 パシブプロキシ構成のサーバーパラメータが必須に
3.2.x	For: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	Zabbix サーバー/フロントエンドのバックエンドデータベース が削除されたため、SQLite がサポートされます。 POSIX 拡張の代わりに、Perl 互換正規表現 (PCRE) がサポート されます。 「libpcre」および「libevent」ライブラリが必須に 「nowait」フラグのないユーザーパラメータ、リモートコマン ド、system.run[] アイテム、および Zabbix サーバーで実行さ れるスクリプトの終了コードチェックが追加 Zabbix Java ゲートウェイは、新しい機能をサポートするた めにアップグレードする必要があります。 履歴テーブルのサイズによっては、データベースのアップグレ ードに時間がかかる場合があります。
3.0.x LTS	For: Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	
2.4.x	For: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	必須 PHP バージョンが 5.3.0 から 5.4.0 に LogFile エージェントパラメータの指定が必須に
2.2.x LTS	For: Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	ノードベースの分散監視削除

アップグレード元	アップグレードノート全文	バージョン間の重要な変更
2.0.x	For: Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0	必須 PHP バージョンが 5.1.6 から 5.3.0 に Zabbix サーバーが MySQL データベースと正しく連携するには、 大文字と小文字を区別するデータベースおよび文字セット utf8 と utf8_bin の照合が必要に。データベース作成スクリプトを参 照してください。 PHP 拡張機能「mysql」の代わりに「mysqli」が必須に

6.0 の要件を確認することもできます。

Note:

アップグレード中に並列で 2 つの SSH セッションを実行し、一方でアップグレード手順を実行し、もう一方でサーバー/プロキシのログを監視すると便利な場合があります。たとえば、2 番目の SSH セッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログファイルエントリと発生する可能性のあるエラーがリアルタイムで表示されます。これは、実稼働インスタンスにとって重要な場合があります。

サーバーのアップグレードプロセス

1 Zabbix プロセスの停止

Zabbix サーバーを停止して、データベースに新しいデータが挿入されていないことを確認します。

2 既存 Zabbix データベースのバックアップ

これは非常に重要なステップです。データベースのバックアップがあることを確認してください。アップグレード手順が失敗（ディスク容量の不足、電源オフ、予期しない問題）した場合に役立ちます。

3 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

Zabbix バイナリ、設定ファイル、PHP ファイルディレクトリのバックアップコピーを作成します。

4 新しいサーバーバイナリのインストール

手順を使用して、ソースから Zabbix サーバーをコンパイルします。

5 サーバー設定パラメーターの確認

必須の変更の詳細については、アップグレードノートを参照してください。

新しいオプションのパラメーターについては **What's new** セクションを参照してください。

6 新しい Zabbix バイナリの開始

新しいバイナリを開始します。ログファイルをチェックして、バイナリが正常に開始されたかどうかを確認します。

Zabbix サーバーはデータベースを自動的にアップグレードします。起動時に、Zabbix サーバーは現在の（必須およびオプションの）データベースバージョンと必要なデータベースバージョンを確認します。現在の必須バージョンが必要なバージョンよりも古い場合、Zabbix サーバーは必要なデータベースアップグレードパッチを自動的に実行します。データベースアップグレードの開始レベルと進行レベル（パーセンテージ）は、Zabbix サーバーのログファイルに書き込まれます。アップグレードが完了すると、「データベースのアップグレードが完全に完了しました」というメッセージがログファイルに書き込まれます。アップグレードパッチのいずれかが失敗した場合、Zabbix サーバーは起動しません。現在のデータベースバージョンが必要なバージョンよりも新しい場合も、Zabbix サーバーは起動しません。Zabbix サーバーは、現在のデータベースバージョンがサーバーバージョンに対応している場合にのみ起動します。

8673:20161117:104750.259 current database version (mandatory/optional): 03040000/03040000

8673:20161117:104750.259 required mandatory version: 03040000

サーバーを起動する前に：

- データベースユーザーに十分な権限があることを確認してください（テーブルの作成、テーブルの削除、インデックスの作成、インデックスの削除）
- 十分な空きディスク容量があることを確認してください。

7 新しいウェブインターフェースのインストール

要求 PHP バージョンは 7.2.5 です。必要に応じて更新し **インストール手順** に従ってください。

8 ウェブブラウザのクッキーとキャッシュのクリア

アップグレード後、Zabbix ウェブインターフェイスが正しく機能するために、ウェブブラウザのクッキーとキャッシュのクリアが必要になる場合があります。

プロキシのアップグレードプロセス

1 プロキシのを停止

Zabbix プロキシを停止します。

2 設定ファイルと Zabbix プロキシバイナリのバックアップ

Zabbix プロキシバイナリと設定ファイルのバックアップコピーを作成します。

3 新しいプロキシバイナリのインストール

手順を参照して、ソースから Zabbix プロキシをコンパイルします。

4 プロキシ構成パラメーターの確認

このバージョンでは、プロキシ**パラメータ**に必須の変更はありません。

5 新しい Zabbix プロキシの起動

新しい Zabbix プロキシを起動し、ログファイルをチェックしてプロキシが正常に開始されたかどうかを確認してください。

Zabbix プロキシはデータベースを自動的にアップグレードします。データベースのアップグレードは、**Zabbix サーバー**を起動するときと同様に行われます。

エージェントのアップグレードプロセス

Attention:

エージェントのアップグレードは必須ではありません。新しい機能にアクセスする必要がある場合にのみ、エージェントをアップグレードする必要があります。

このセクションで説明するアップグレード手順は、Zabbix エージェントと Zabbix エージェント 2 の両方をアップグレードするために使用できます。

1 エージェントの停止

Zabbix エージェントを停止します。

2 設定ファイルと Zabbix エージェントバイナリのバックアップ

Zabbix エージェントバイナリと設定ファイルのバックアップコピーを作成します。

3 新しいエージェントバイナリのインストール

手順を参照して、ソースから Zabbix エージェントをコンパイルします。

または **Zabbix ダウンロードページ** からコンパイル済みの Zabbix エージェントをダウンロードすることもできます。

4 エージェント設定パラメーターの確認

このバージョンでは **agent** または **agent 2** のパラメーターに必須の変更はありません。

5 新しい Zabbix エージェントの起動

新しい Zabbix エージェントを起動し、ログファイルをチェックして、エージェントが正常に起動したかどうかを確認します。

マイナーバージョンアップグレード

6.0.x をマイナーバージョンアップグレード (たとえば、6.0.1 から 6.0.3) する場合は、メジャーバージョンアップグレード時と同じアクションをサーバー/プロキシ/エージェントに対して実行する必要があります。唯一の違いは、マイナーバージョンアップグレードする場合、データベースに変更が加えられないことです。

8 既知の問題点

MySQL 8.0.0-8.0.17 でのプロキシの起動

MySQL のバージョン 8.0.0-8.0.17 上で `zabbix_proxy` を実行する際、以下のような "access denied" エラーが発生します：

[Z3001] connection to database 'zabbix' failed: [1227] Access denied; you need (at least one of) the SUPER

これは、MySQL 8.0.0 がセッション変数を設定するための特別な権限を強制するようになったためです。ただし、8.0.18 では、この動作は削除されました：[As of MySQL 8.0.18, setting the session value of this system variable is no longer a restricted operation.](#)

回避策としては、zabbix ユーザーに権限を与えます：

MySQL バージョン 8.0.14 - 8.0.17 の場合：

```
grant SESSION_VARIABLES_ADMIN on *.* to 'zabbix'@'localhost';
```

MySQL バージョン 8.0.0 - 8.0.13 の場合：

```
grant SYSTEM_VARIABLES_ADMIN on *.* to 'zabbix'@'localhost';
```

Timescale DB: 多数のパーティションでメモリ使用量が高くなる

PostgreSQL バージョン 9.6 ~12 は、多数のパーティションを含むテーブルを更新するときに大量のメモリを使用します ([問題レポートを参照](#))。この問題は、トレンドが比較的小さな (1 日など) チャンクに分割されている場合に、Zabbix が TimescaleDB を備えたシステムでトレンドを更新するときに発生します。これにより、デフォルトのハウスキーピング設定でトレンドテーブルに何百ものチャンクが存在することになり、PostgreSQL がメモリ不足になる可能性があります。

この問題は Zabbix 5.0.1 以降の TimescaleDB を使用した新規インストールでは解決されていますが、それ以前に TimescaleDB を Zabbix でセットアップした場合は、移行に関する注意事項[ZBX-16347](#) を参照してください。##### Timescale DB 2.5.0: 整数を含むテーブルで圧縮ポリシーが失敗する可能性がある

この問題は、TimescaleDB 2.5.0 が使用されている場合に発生します。TimescaleDB 2.5.1 以降では解決されています。

詳細については、[TimescaleDB Issue #3773](#) を参照してください。

Upgrade

SQL mode setting for successful upgrade

The `sql_mode` setting in MySQL/MariaDB must have the "STRICT_TRANS_TABLES" mode set. If it is absent, the Zabbix database upgrade will fail (see also [ZBX-19435](#)).

MariaDB 10.2.1 以前のバージョンでアップグレードする場合

MariaDB 10.2.1 以前のバージョンでは、デフォルトの行フォーマットがコンパクトであるため、
これらのバージョンでデータベーステーブルを作成した場合、Zabbix のアップグレードに失敗することがあります。
この問題は、行の形式をダイナミックに変更することで解決できます。
[ZBX-17690](#)を参照してください。

Accidental installation of EPEL Zabbix packages

With EPEL repository installed and enabled, installing Zabbix from packages will lead to EPEL Zabbix packages being installed rather than official Zabbix packages.

In this case uninstall Zabbix packages from EPEL, i.e.:

```
dnf remove zabbix-server-mysql
```

Block Zabbix packages from EPEL. Add the following line in the `/etc/yum.conf` file:

```
exclude=zabbix6.0*
```

Install Zabbix server again:

```
dnf install zabbix-server-mysql
```

Notice that official Zabbix packages have the word `release` in their version string:

```
6.0.25-release1.e18
```

MariaDB でのデータベース TLS 接続について

DBTLSConnect `parameter`の`verify_ca` オプションでは、
 MariaDB を使用したデータベース TLS 接続はサポートされません。

MySQL/MariaDB でのデッドロックの可能性

負荷が高く、複数の LLD 処理が実行されている場合、行ロック処理に関する InnoDB エラーが原因でデッドロックが発生する可能性があります。 ([upstream bug](#)参照) このエラーは、MySQL 8.0.29 で修正されましたが、MariaDB では修正されていません。詳細は、[ZBX-21506](#)を参照してください。

グローバルイベント相関

最初のイベントと 2 番目のイベントの時間間隔が非常に小さい場合 (0.5 秒以下)、イベントは正しく相関しないことがあります。

PostgreSQL 11 以前のバージョンにおける数値 (float) データ型の範囲

PostgreSQL 11 およびそれ以前のバージョンでは、浮動小数点数データ型をサポートしています。
 1.34E-154 から 1.34E+154 までの範囲になります。

NetBSD 8.0 とそれ以降

NetBSD 8.0 以降では、起動時に Zabbix の様々なプロセスがランダムにクラッシュすることがあります。
 バージョン 8.X および 9.X では、デフォルトのスタックサイズ (4MB) が小さすぎることが原因です。この値を増やす必要があります。


```
ulimit -s 10240 <br>
```

詳しくは、関連する問題報告[ZBX-18275](#)を参照してください。

Regular expression limitations in Zabbix agent 2

Zabbix agent 2 does not support lookaheads and lookbehinds in regular expressions due to the standard Go regexp library limitations.

IPMI チェック

9 (stretch) 以前の Debian および 16.04 (xenial) 以前の Ubuntu では、標準の OpenIPMI ライブラリパッケージでは IPMI チェックが動作しません。
 修正するには[ZBX-6139](#) で説明されているように OpenSSL を有効にして
 OpenIPMI ライブラリを再コンパイルしてください。

SSH チェック

- libssh2 ライブラリがパッケージからインストールされている場合、Debian、Ubuntu などの一部の Linux ディストリビューションは、暗号化された秘密鍵 (パスフレーズ付き) をサポートしません。詳細については[ZBX-4850](#) を参照してください。
- OpenSSH 8 を使用して CentOS 8 で libssh 0.9.x を使用すると、SSH チェックで「SSH サーバーからデータを読み取れません」と報告されることがあります。これは、libssh の [問題 \(詳細レポート\)](#) が原因です。このエラーは安定版の libssh 0.9.5 リリースで修正される予定です。詳細については[ZBX-17756](#) も参照してください。
- パイプ'|'を使用した SSH スクリプトで「SSH サーバーからデータを読み取れません」というエラーが発生する可能性があります。この場合 libssh ライブラリのバージョンをアップグレードすることをお勧めします。詳細については、[ZBX-21337](#) を参照してください。

ODBC チェック

- MySQL unixODBC ドライバーは、MariaDB コネクタライブラリに対してコンパイルされた Zabbix サーバーまたは Zabbix プロキシと共に使用しないでください。[アップストリームのバグ](#) のため、可能であればドライバーと同じコネクタを使用しないこともお勧めします。推奨セットアップ:

PostgreSQL, SQLite または Oracle connector → MariaDB または MySQL unixODBC driver MariaDB connector → MariaDB unixODBC driver MySQL connector → MySQL unixODBC driver

詳細および利用可能な回避策については[ZBX-7665](#) を参照してください。

- Microsoft SQL Server からクエリされた XML データは、Linux および UNIX システムでさまざまな方法で切り捨てられる場合があります。
- さまざまなバージョンの Linux 用 Oracle Instant Client を使用して Oracle データベースを監視するために ODBC チェックを使用すると、Zabbix サーバーがクラッシュすることが確認されています。[[ZBX-18402](#)](<https://support.zabbix.com/browse/ZBX-18402>)[ZBX-20803](#)も参照してください。
- FreeTDS UnixODBC ドライバーを使用している場合は、'SET NOCOUNT ON' ステートメントを SQL クエリの先頭に追加する必要があります (例:'SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ... '). そうしないと、Zabbix のデータベースモニタアイテムは「SQL クエリが空の結果を返しました」というエラーで情報の取得に失敗します。詳細については[ZBX-19917](#) を参照してください。

アイテム内の request method parameter が正しくない

HTTP チェックでのみ使用される request method parameter が、4.0 以前の Zabbix からアップグレードした結果、
 すべてのアイテムにデフォルト値ではない '1' が設定されることがあります。
 この問題の解決方法の詳細については、[ZBX-19308](#)を参照してください。

Web 監視と HTTP エージェント

いくつかの Linux ディストリビューション上で、Web シナリオや HTTP エージェント内で「SSL ピア検証」を有効にしていた時に[upstream bug](#)によってメモリーリークが発生します。[ZBX-10486](#)を参照してより詳細な情報と回避策を確認してください。

シンプルチェック

v3.10 より前のバージョンの **fping** には、重複したエコーリプライパケットを誤って処理するバグがあります。
 これは `icmping`, `icmpingloss`, `icmpingsec` アイテムに予期せぬ結果をもたらすかもしれません。
 このため、最新バージョンの **fping** を使用することをお勧めします。
 詳細は[ZBX-11726](#)を参照してください。

Errors with fping execution in rootless containers

When containers are running in rootless mode or in a specific-restrictions environment, you may face errors related to fping execution when performing ICMP checks, such as `fping: Operation not permitted` or `all packets to all resources lost`.

To fix this problem add `--cap-add=net_raw` to "docker run" or "podman run" commands.

Additionally fping execution in non-root environments may require `sysctl` modification, i.e.:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

where "1995" is the zabbix GID. For more details, see [ZBX-22833](#).

SNMP チェック

OpenBSD オペレーティングシステムを使用している場合、バージョン 5.7.3 までの Net-SNMP ライブラリの
 `use-after-free` バグにより、Zabbix server に `SourceIP` パラメータが設定されている場合
 Zabbix server がクラッシュする可能性があります。回避策として、`SourceIP` パラメータを設定しないようお願いします。
 Linux でも同様の問題が発生しますが、Zabbix server の動作が停止することはありません。
 OpenBSD の `net-snmp` パッケージに対するローカルパッチが適用され、OpenBSD 6.3 でリリースされる予定です。

SNMP データスパイク

SNMP データのスパイクが確認されています。
 詳細は[ZBX-14318](#)を参照してください。

SNMP トラップ

SNMP トラップの処理に必要な `net-snmp-perl` パッケージは、RHEL 8.0-8.2 で削除され、RHEL 8.3 で追加されました。

RHEL 8.0-8.2 を使用している場合、RHEL 8.3 にアップグレードするのが最適な解決策です。

[ZBX-17192](#)を参照してください。

RHEL 7 での alerter プロセスのクラッシュ

RHEL 7 上で Zabbix サーバーの alerter プロセスがクラッシュすることが確認されています。詳細は[ZBX-10461](#)を参照してください。

Upgrading Zabbix agent 2 (6.0.5 or older)

When upgrading Zabbix agent 2 (version 6.0.5 or older) from packages, a plugin-related file conflict error may occur. To fix the error, back up your agent 2 configuration (if necessary), uninstall agent 2 and install it anew.

On RHEL-based systems, run:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

On Debian-based systems, run:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

For more information, see [ZBX-23250](#).

Web インターフェースの言語設定のフリッピング

Web インターフェースの言語設定が、明らかにロジックなしでフリッピングすることが確認されています。
 あるページ (またはページの一部) はある言語で表示され、別のページ (またはページの一部) は別の言語で表示される。
 一般的に、この問題は複数のユーザーがいて、そのうちの何人かがあるロケールを使い、それ以外に別のロケールを使う
 ユーザーがいる場合に発生することがあります。

この問題を回避する方法として知られているのは、PHP と Apache のマルチスレッドを無効にすることです。

この問題は、ロケールの設定がどのように行われるかに関連しています。
 [in PHP](#)
 ロケールの情報はスレッド単位ではなくプロセス単位で管理されます。
 そのため、マルチスレッド環境において、同じ Apache プロセスで実行される複数のプロジェクトがある場合、
 別のスレッドでロケールが変更される可能性があります。
 その結果、Zabbix スレッドで処理できるデータが変更される可能性があります。

詳細については、関連する問題報告を参照してください。

- [ZBX-10911](#) (Web インターフェースの言語設定入れ替わりの問題)
- [ZBX-16297](#) (BC の `bcdiv` 関数を使ったグラフの数値処理に関する問題)

PHP 7.3 の opcache の設定

PHP 7.3 の設定で“opcache”が有効な場合、Zabbix の Web インターフェースを初めて読み込んだときに、
 空白の画面が表示されることがあります。これは登録された [PHP bug](#) です。
 この問題を回避するには、php.ini の opcache.optimization_level パラメータを 0x7FFFBFDF に設定してください。

グラフ

夏時間

夏時間 (DST) への変更により、X 軸ラベルの表示が不規則になります。(日付の重複、日付の欠落など)

合計集計

1 時間未満の期間のグラフで **合計集計** を使用すると、データがトレンドから取得されたときに、グラフに誤った (乗算された) 値が表示されます。

ログファイルの監視

log[] と logrt [] は、ファイルシステムが 100% 一杯の状態ではログファイルが追加される場合、ログファイルを最初から読み直します。
 (詳しくは [ZBX-10884](#) を参照してください)

MySQL のスロークエリ

Zabbix サーバは、アイテムが存在しない場合、スロークエリ (SELECT) を生成します。
 これは、以下の既知の問題によるものです。
 [MySQL 5.6/5.7 バージョンにおける既知の問題](#)
 この問題の回避策は、index_condition_pushdown optimizer を無効にします。
 詳細については [ZBX-10652](#) を参照してください。

Slow configuration sync with Oracle

Configuration sync might be slow in Zabbix 6.0 installations with Oracle DB that have high number of items and item preprocessing steps. This is caused by the Oracle database engine speed processing nlob type fields.

To improve performance, you can convert the field types from nlob to nvarchar2 by manually applying the database patch [items_nvarchar_prepare.sql](#). Note that this conversion will reduce the maximum field size limit from 65535 bytes to 4000 bytes for item preprocessing parameters and item parameters such as Description, Script item's field Script, HTTP agent item's fields Request body and Headers, Database monitor item's field SQL query. Queries to determine template names that need to be deleted before applying the patch are provided in the patch as a comment. Alternatively, if MAX_STRING_SIZE is set you can change nvarchar2(4000) to nvarchar2(32767) in the patch queries to set the 32767 bytes field size limit.

For an extended discussion, see [ZBX-22363](#).

API ログイン

user.login method を使用し、user.logout が無い
 カスタムスクリプトを実行した場合、多数のオープンユーザーセッションが作成される可能性があります。

SNMPv3 トラップにおける IPv6 アドレスの不具合について

net-snmp のバグにより、SNMP トラップで SNMPv3 を使用する場合、IPv6 アドレスが正しく表示されないことがあります。
 詳細および回避策については、[ZBX-14541](#) を参照してください。

ログイン失敗時の情報において、長い IPv6 IP アドレスが切り取られている

ログインに失敗したメッセージには、保存されている IP アドレスの最初の 39 文字だけが表示されます。
 これは、データベースフィールドの文字数制限のためです。
 つまり、39 文字より長い IPv6 IP アドレスは、不完全に表示されます。

Windows での Zabbix agent チェック

Zabbix agent 設定ファイル (zabbix_agentd.conf) の Server パラメータに既存の DNS エントリがない場合、
 Zabbix エージェントの応答時間が長くなる可能性があります。
 これは、Windows の DNS キャッシュデーモンが IPv4 アドレスのネガティブレスポンスをキャッシュしないためです。
 しかし、IPv6 アドレスではネガティブレスポンスはキャッシュされます。
 この問題を回避するには、ホスト上で IPv4 を無効にする必要があります。

YAML エクスポート/インポート

YAML には、いくつかの既知の問題があります。 [export/import](#)

- エラーメッセージが翻訳できない。
- .yaml ファイルの拡張子を持つ有効な JSON がインポートできないことがある。
- 引用符で囲まれていない日付は、自動的に Unix タイムスタンプに変換される。

SUSE で NGINX と php-fpm を使用する場合のセットアップウィザード

SUSE で NGINX + php-fpm を使用している場合、フロントエンドのセットアップウィザードで設定ファイルを保存できません。
これは、/usr/lib/systemd/system/php-fpm.service unit の設定により、Zabbix が/etc に書き込むことができないためです。
(PHP7.4で導入されました)。

2つの回避策があります。

- php-fpm systemd unit で [ProtectSystem](#) オプションを 'full' ではなく 'true' に設定する。

- /etc/zabbix/web/zabbix.conf.php ファイルを手動で保存する。

Ubuntu 20 で Zabbix Web サービスを Chromium で実行

ほとんどの場合、Zabbix Web サービスは Chromium で動作しますが、Ubuntu 20.04 で Chromium を使用すると、以下のエラーが発生します。

```
Cannot fetch data: chrome failed to start:cmd_run.go:994: <br>
WARNING: cannot create user data directory: cannot create <br>
"/var/lib/zabbix/snap/chromium/1564": mkdir /var/lib/zabbix: permission denied <br>
Sorry, home directories outside of /home are not currently supported. See https://forum.snapcraft.io/t/112
```

このエラーは、ユーザ zabbix のホームディレクトリとして /var/lib/zabbix が使用されているため発生します。

MySQL カスタムエラーコード

Azure 上の MySQL インストールで Zabbix を使用する場合、不明確なエラーメッセージ
 [9002] Some errors occurred が Zabbix ログに表示されることがあります。
 この一般的なエラーテキストは、データベースから Zabbix server または proxy に送信されます。
 エラーの原因に関する詳細な情報を得るには、Azure のログを確認してください。

PCRE2 への移行後の無効な正規表現

Zabbix 6.0 では、PCRE2 のサポートが追加されました。
 PCRE は現在もサポートされていますが、RHEL/CentOS 7 以降、SLES(全バージョン)、Debian 9 以降、Ubuntu 16.04 以降の
 Zabbix インストールパッケージは PCRE2 を使用するように更新されています。
 PCRE2 への移行は多くの利点をもたらしますが、既存の PCRE 正規表現パターンの一部が無効となったり、異なる動作をする可能性があります。
 特に、パターン `^[w-\.]` が影響を受けます。この正規表現をセマンティクスに影響を与えずに有効にするには、
 式を `^[w-\.]` に変更します。
 これは、PCRE2 がダッシュ記号を区切り記号として扱い、文字クラス内に範囲を作成するために起こります。
 以下の Zabbix インストールパッケージがアップデートされ、PCRE2 を使用するようになりました。
 RHEL/CentOS 7 以降、SLES(全バージョン)、Debian 9 以降、Ubuntu 16.04 以降

Zabbix 6.0.0-6.0.2 でのサービスの間違った変換

Zabbix 6.0 では、より柔軟で新しいサービスステータスの計算アルゴリズムが導入されました。

Zabbix 6.0 より前のバージョンから Zabbix 6.0.0、6.0.1、6.0.2 にアップグレードすると、サービスステータスの計算ルール「すべての子が障害であったときに最も重要」と「子サービスで最も重要」が入れ替わります。Zabbix 6.0.0 以降で作成されたサービスには、正しいステータス計算ルールがあります。

バージョン 6.0 より前のバージョンから Zabbix 6.0.3 以降にアップグレードすると、Zabbix はサービスステータスの計算ルールを正しく更新します。6.0.x から 6.0.3 にアップグレードしても、サービスステータスの計算ルールには影響しません。

Geomap ウィジェットエラー

NGINX を使用して古いバージョンの Zabbix からアップグレードし、アップグレード中に新しい NGINX 設定ファイルに切り替えなかった場合、
 Geomap ウィジェットの地図が正しく読み込まれないことがあります。

この問題を解決するには、古い設定ファイルを破棄し、6.0 パッケージの設定ファイルを使用し、
 e. Zabbix フロントエンドの PHP の設定セクションの [download instructions](#) に
 記載されている方法で再設定することができます。

または、既存の NGINX 設定ファイル (通常、/etc/zabbix/nginx.conf) を手動で編集することができます。
 これを行うには、ファイルを開き、以下のブロックを探します:


```
location ~ /(api\|/conf[^\.]|include|locale|vendor) {
    deny          all;
    return        404;
}
```

そして、このブロックを次のように置き換えます:

```
location ~ /(api\|/conf[^\.]|include|locale) {
    deny          all;
    return        404;
}
```

```
location /vendor {
    deny          all;
```

```
        return          404;
    }
```

Logrotate for Zabbix server and proxy

The logrotate utility is only included into packages for zabbix-agent, zabbix-agent2 and zabbix-web-service, but needs to be installed separately for Zabbix server and proxy. The logrotate dependency has been added to the server and proxy packages for RHEL and SUSE starting from Zabbix 6.4.4rc1.

Zabbix 6.0.11 での JSONPath 解析エラー

JSONPath 解析エラーは、先頭の空白と空の配列/オブジェクトの場合に発生します。Zabbix 6.0.12 で修正されました。

Missing files in Windows agent archive

The Windows Zabbix agent download ZIP file is missing zabbix_sender.h and zabbix_sender.lib files in versions 6.0.0-6.0.27, required for zabbix_sender.dll.

1 Compilation issues

These are the known issues regarding Zabbix compilation from sources. For all other cases, see the [Known issues](#) page.

Compiling Zabbix agent on HP-UX

If you install the PCRE library from a popular HP-UX package site <http://hpux.connect.org.uk>, for example from file pcre-8.42-ia64_64-11.3 you get only the 64-bit version of the library installed in the /usr/local/lib/hpux64 directory.

In this case, for successful agent compilation customized options need to be used for the "configure" script, e.g.:

```
CFLAGS="+DD64" ./configure --enable-agent --with-libpcre-include=/usr/local/include --with-libpcre-lib=/usr
```

Library in a non-standard location

Zabbix allows you to specify a library located in a non-standard location. In the example below, Zabbix will run curl-config from the specified non-standard location and use its output to determine the correct libcurl to use.

```
$ ./configure --enable-server --with-mysql --with-libcurl=/usr/local/bin/curl-config
```

This will work if it is the only libcurl installed in the system, but might not if there is another libcurl installed in a standard location (by the package manager, for example). Such is the case when you need a newer version of the library for Zabbix and the older one for other applications.

Therefore, specifying a component in a non-standard location will not always work when the same component also exists in a standard location.

For example, if you use a newer libcurl installed in /usr/local with the libcurl package still installed, Zabbix might pick up the wrong one and compilation will fail:

```
usr/bin/ld: ../../src/libs/zbxhttp/libzbxhttp.a(http.o): in function 'zbx_http_convert_to_utf8':
/tmp/zabbix-master/src/libs/zbxhttp/http.c:957: undefined reference to 'curl_easy_header'
collect2: error: ld returned 1 exit status
```

Here, the function curl_easy_header() is not available in the older /usr/lib/x86_64-linux-gnu/libcurl.so, but is available in the newer /usr/local/lib/libcurl.so.

The problem lies with the order of linker flags, and one solution is to specify the full path to the library in an LDFLAGS variable:

```
$ LDFLAGS="-Wl,--no-as-needed /usr/local/lib/libcurl.so" ./configure --enable-server --with-mysql --with-l
```

Note the -Wl,--no-as-needed option which might be needed on some systems (see also: default linking options on [Debian-based](#) systems).

9 テンプレート変更点

このページでは、Zabbix に同梱されているテンプレートの変更点を一覧で紹介いたします。

Zabbix を最新バージョンにアップグレードしても、使用するテンプレートは自動的にアップグレードされないことに注意してください。
 既存のインストール環境では、以下の方法でテンプレートを変更することをお勧めします。

- [Zabbix Git リポジトリ](#)から最新のテンプレートをダウンロードします。

- その後、Configuration→Templates でテンプレートを Zabbix に手動でインポートすることができます。
 同じ名前のテンプレートが既に存在する場合、インポートする際に Delete missing オプションをチェックすると、
 きれいにインポートすることができます。
 この方法では、更新されたテンプレートに含まれない古いアイテムは削除されます。
 (この場合、これらの古いアイテムの履歴が失われることに注意してください)

バージョン 6.0.0 での変更点

新しいテンプレート

Zabbix 6.0.0 の **new templates** のリストを参照してください。

テンプレートの変更点

- Windows by Zabbix agent と Windows by Zabbix agent active の対応する項目名および説明に {#FSLABEL} マクロが追加されました。
- vfs.file.cksum [/etc/passwd] エージェントアイテムは、以下のように変更されました。
 vfs.file.cksum[/etc/passwd,sha256] に変更されました。

- Zabbix server、Zabbix proxy、リモート Zabbix server、リモート Zabbix proxy のテンプレートに
 新しいチェック zabbix[process,odbc poller,avg,busy] が追加されました。この指標は、直近 1 分間で ODBC プロセスが
 ビジー状態であった平均時間 (単位はパーセント) を監視するために使用されます。

バージョン 6.0.2 での変更点

テンプレート Generic Java JMX に 2 つの検知ルールが含まれるようになりました。 - ガベージコレクターの検知 - メモリプールの検知

バージョン 6.0.3 での変更点

新しいテンプレート OpenWeatherMap by HTTP が利用可能になりました。

既存のテンプレートにおいて、以下の変更が行われました。

- Windows services by Zabbix agent, Windows services by Zabbix agent active, Windows by Zabbix agent, Windows by Zabbix agent active の
 各テンプレートの {\$SERVICE.NAME.NOT_MATCHES} マクロ値が更新され、拡張サービスリストのフィルタリングが可能になりました。
- テンプレート PostgreSQL by Zabbix agent 2 は、スロークエリの数をチェックし、その数が閾値を超えた場合に問題を生成するようにになりました。

バージョン 6.0.4 での変更点

新しいテンプレートが利用可能になりました。

- TrueNAS SNMP - SNMP による TrueNAS ストレージ OS の監視
- Proxmox VE by HTTP -**HTTP templates** のセットアップ手順を参照してください。

SMART by Zabbix agent 2 と SMART by Zabbix agent 2 (active) が更新されました。
 - LLD ルールの Attribute discovery は削除され、LLD ルールの Disk discovery は事前に定義されたベンダー固有の
 属性セットに基づいてディスクを検出するようになりました。 - **smart.disk.get** 項目は、すべてのディスクの代わりに、特定のディスクに関する情報のみを戻せるようになりました。

仮想ファイルシステム監視のために、ファイルシステム使用率の警告および重要な閾値を定義するための
 新しいマクロがテンプレートに追加されました。
 HOST-RESOURCES-MIB storage SNMP,
 Linux by Prom,
 Linux filesystems SNMP,
 Linux filesystems by Zabbix agent active,
 Linux filesystems by Zabbix agent,
 Mellanox SNMP,
 PFSense SNMP,
 Windows filesystems by Zabbix agent active,
 Windows filesystems by Zabbix agent.
 ファイルシステム利用状況トリガーが更新され、これらのマクロを使用できるようになりました。

バージョン 6.0.5 での変更点

新しいテンプレートが利用可能になりました。

- CockroachDB by HTTP
- Envoy Proxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP

HTTP templates のセットアップ手順を参照してください。

バージョン 6.0.6 での変更点

新しいテンプレートが利用可能になりました。

- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP

[HTTP templates](#)のセットアップ手順を参照してください。

[PostgreSQL Agent 2 template](#) has been updated:

A trigger for detecting checksum failures has been added to the Dbstat item of the PostgreSQL Agent 2 template. According to [PostgreSQL documentation](#), you can use checksums on data pages to help detect corruption by the I/O system that would otherwise be silent.

バージョン 6.0.7 での変更点

新しいテンプレート [HPE Synergy by HTTP](#) が利用可能です。

HashiCorp Consul Node by HTTP および HashiCorp Consul Cluster by HTTP のテンプレートは、Consul 名前空間をサポートするようになりました。

6.0.8 での変更点

新しいテンプレート [OPNsense by SNMP](#) が使用可能です。

バージョン 6.0.13 での変更点

新しいテンプレートが利用可能になりました：

- [AWS EC2 by HTTP](#)
- [AWS by HTTP](#)
- [AWS RDS instance by HTTP](#)
- [AWS S3 bucket by HTTP](#)

[HTTP テンプレート](#)のセットアップ手順を参照してください。

The template [Oracle by Zabbix agent 2](#) has been updated:

- The following static items, that requested data for all existing relevant DB objects in a single query, have been removed:
 - "Oracle: Get archive log info"
 - "Oracle: Get ASM stats"
 - "Oracle: Get CDB and No-CDB info"
 - "Oracle: Get PDB info"
 - "Oracle: Get tablespaces stats"
- The following agent item prototypes have been added to the corresponding discovery rules:
 - Archive log discovery rule: "Archivelog '#{DEST_NAME}': Get archive log info"
 - ASM disk groups discovery: "ASM '#{DGNAME}': Get ASM stats"
 - Database discovery: "Oracle Database '#{DBNAME}': Get CDB and No-CDB info"
 - PDB discovery: "Oracle Database '#{DBNAME}': Get PDB info"
 - Tablespace discovery: "Oracle TBS '#{TABLESPACE}': Get tablespace stats"

CHANGES IN 6.0.18

The template [Azure by HTTP](#) now also works with Azure Cosmos DB for MongoDB.

CHANGES IN 6.0.20

A new [template](#) [AWS ECS Cluster by HTTP](#) (along with its [Serverless Cluster version](#)) is available.

CHANGES IN 6.0.21

New template is available:

- [AWS Cost Explorer by HTTP](#)

CHANGES IN 6.0.22

New template is available:

- [MantisBT by HTTP](#)

CHANGES IN 6.0.23

New template is available:

- [Nextcloud by HTTP](#)

Updated templates

[PostgreSQL by ODBC](#) and [PostgreSQL by Zabbix agent 2](#) templates now include the item and trigger for monitoring PostgreSQL version.

CHANGES IN 6.0.24

Updated templates

Integration with OpenShift has been added to [Kubernetes cluster state by HTTP](#) template.

CHANGES IN 6.0.26

Updated templates

[MSSQL by ODBC](#) template:

- new item has been added - MSSQL DB '#{DBNAME}': Recovery model, which returns the database recovery model under the database discovery;
- new macros, namely, {\$MSSQL.BACKUP_FULL.USED}, {\$MSSQL.BACKUP_DIFF.USED}, {\$MSSQL.BACKUP_LOG.USED}, have been added - those can be used for disabling backup age triggers for a certain database;
- LLD rules for quorum and quorum members discovery have been added;
- the type of the LLD rules has been changed from "Database monitor" to "Dependent item";
- items with `db.odbc.discovery` key have been turned into items dependent on `db.odbc.get` item.

CHANGES IN 6.0.27

New templates

A new template is available:

- [YugabyteDB by HTTP](#), which includes the YugabyteDB Cluster by HTTP template for monitoring each YugabyteDB cluster.

CHANGES IN 6.0.28

New templates

A new template is available:

- [MSSQL by Zabbix agent 2](#)

CHANGES IN 6.0.29

Updated templates

- [FortiGate by SNMP](#) template has been supplemented with metrics regarding VPN, high availability (HA), wireless termination points (WTPs), SD-WAN health checks, and HW sensors.

CHANGES IN 6.0.30

New templates

The AWS ELB template set has been supplemented with the template [AWS ELB Network Load Balancer by HTTP](#).

10 6.0.0 へのアップグレードノート

このノートは Zabbix 5.4.x から Zabbix 6.0.0 へのアップグレードのためのものです。
すべてのノートは以下のように分類されています。

- 重要 (Critical) - アップグレードプロセスと Zabbix の機能変更に関連する最も重要な情報です。
- 情報提供 (Informational) - Zabbix の機能の変更に関する残りのすべての情報です。

Zabbix 6.0.0 は、Zabbix 5.4.0 以前のバージョンからアップグレードすることができます。
[upgrade procedure](#) セクションをご覧ください。

重要 (Critical) データベース

最適なユーザエクスペリエンスを実現し、最高の Zabbix パフォーマンスを保証するために
 様々な実運用環境において、いくつかのデータベースがサポートされています。
 パフォーマンスを確保するため、一部の旧リリースのデータベースはサポートされなくなりました。
 これは主にサービスライフポイントの終わりに近いデータベースバージョン、および
 通常のパフォーマンスに支障をきたす可能性のある未解決の問題があるバージョンに適用されます。

Zabbix 6.0 で公式にサポートしているバージョンの `database` は以下の通りです。

- MySQL/Percona 8.0.X
- MariaDB 10.5.X - 10.6.X
- PostgreSQL 13.X - 14.X
- Oracle 19c - 21c
- TimescaleDB 2.0.1-2.3
- SQLite 3.3.5-3.34.X

デフォルトでは、サポートされていないデータベースのバージョンが検出された場合、Zabbix server と proxy は起動しません。ただし、推奨はしませんが、[server](#) や [proxy](#) (/manual/appendix/config/zabbix_proxy) の AllowUnsupportedDBVersions 設定パラメータを変更することで、DB バージョンをチェックしないようにすることができます。

プライマリーキー 新規インストール時に、履歴テーブルを含むすべてのテーブルでプライマリーキーを使用するようになりました。既存環境への主キーの自動アップグレードはありません。
 履歴テーブルを主キーに手動でアップグレードする方法については以下のリンクを参照してください。

[MySQL/MariaDB](#)
 [PostgreSQL](#)
 [TimescaleDB v1 and v2](#)
 [Oracle](#)

PCRE2 のサポート

PCRE2 のサポートが追加されました。PCRE は引き続きサポートされていますが、Zabbix は、PCRE または PCRE2 のどちらかのライブラリを使ってコンパイルすることができ、両方を同時に使用することはできません。

以下の Zabbix のインストールパッケージが更新され、PCRE2 が使用できるようになりました：- RHEL 7 以降 - SLES (すべてのバージョン) - Debian 9 以降 - Ubuntu 16.04 以降

PCRE2 に移行後、いくつかの正規表現を更新する必要があることに注意してください。特にパターン `^[\w-\.]` は `^[-\\w\\.]` に変更しないと正常に機能しません - 詳細は[既知の問題点](#)を確認してください。

ODBC チェックの分離処理 ODBC チェックの処理は、通常のポーラプロセスから、独立した server / proxy プロセス ODBC ポーラに移動しました。
 この変更により、ポーラプロセスで作成されるデータベースへの接続数を制限することができます。
 以前は、ODBC チェックは Zabbix agent アイテムや SSH チェックなどでも動作する通常のポーラによって実行されていました。
 新しい設定パラメータ StartODBCPollers が Zabbix [server](#) と [proxy](#) 設定ファイルに追加されました。

内部項目 zabbix[process,<type>] を使用すると、ODBC ポーラの負荷を監視できます。

監査ログ

Zabbix の監査ログを改善し、監査ログを完全に信頼性の高いものにするために、従来のデータベース構造を変更する必要がありました。
 アップグレードにより、DB テーブル auditlog と auditlog_details は新しいフォーマットテーブルに置き換えられ、
 古い監査記録は保存されません。

新しい [section Administration](#) → [General](#) メニューに Audit log が追加され、監査ログの有効化 (デフォルト) または無効化ができるようになりました。
 監査に関するハウスキーピング設定は、これまで Housekeeper メニューセクションにありましたが、
 新しい Audit_log メニューに移動されました。既存のハウスキーピングの設定は保存されます。

API 変更点

Zabbix 6.0.0 の [API changes](#) の一覧を参照してください。

シンプルマクロを式マクロに置き換える

シンプルマクロの機能を式マクロに移行 [transferred](#) しました。既存のシンプルマクロは、バージョンアップ時に式マクロに変換されます。
 長さの制限により変換できないマクロは変換されず、警告がログファイルに警告が出力されます。

マクロ

位置指定マクロはサポートされなくなりました

アイテム名の位置指定マクロ (\$1, \$2...\$9) のサポートは Zabbix 4.0 から非推奨となり、完全に削除されました。

アイテム名のユーザーマクロはサポートされなくなりました。

Zabbix 4.0 から非推奨となっていたアイテム名 (ディスカバリールール名を含む) のユーザーマクロのサポートは、完全に廃止されました。

Monitoring → Overview を削除

Monitoring メニューの Overview セクションが完全に削除されました。
 同じ機能は、Data overview と Trigger overview のダッシュボードを使用してアクセスすることができます。
 [widgets](#) を参照してください。

継承されたトリガーの依存関係を変更することはできません

テンプレートから継承されたトリガーの依存関係を変更することはできなくなりました。
 これは、テンプレートトリガーの依存関係を更新すると、継承されたトリガーの依存関係も上書きされるためです。
 したがって、トリガーの依存関係は常にルートテンプレートレベルでのみ設定する方がより確実です。

情報提供 (Informational) 履歴/トレンドの非推奨内部 item

以下の内部 item は非推奨となり、将来のメジャーリリースで削除される予定です。

- zabbix[history]
- zabbix[history_log]
- zabbix[history_str]
- zabbix[history_text]
- zabbix[history_uint]
- zabbix[trends]
- zabbix[trends_uint]

Zabbix agent 2 プラグイン

Zabbix agent 2 プラグインに、個別の **configuration file** が追加されました。
 デフォルトでは、これらのファイルは `./zabbix_agent2.d/plugins.d/` ディレクトリに配置されます。
 このパスは、agent 2 設定ファイルの `Include` パラメータで指定し、`zabbix_agent2.conf` または `zabbix_agent2.win.conf` ファイルの場所を指定します。

ユーザーパスワード

従来は、ユーザー設定フォームとログインフォームの両方で、ユーザーパスワードのスペースが自動的に切り取られていました。
 設定可能な **password complexity requirements** が導入されたため、パスワードのスペースが削除されなくなりました。
 パスワードにスペースが含まれていると考えていたユーザーは、通常通りログインすることができなくなり
 スペースを除いた「古い」パスワードを入力しなければならなくなります。
 引き続きパスワードの再作成が必要です。

Prometheus メトリクスの一括処理

Prometheus メトリクスの前処理キューに、依存項目の一括処理が導入されたため、依存項目が並列処理されなくなり
 処理速度に影響が出る可能性があります。

ランタイムコマンドの転送

Zabbix サーバとプロキシのランタイムコマンドは、Unix シグナルではなく、ソケット経由で発行されるようになりました。
 この変更により、ランタイムコントロールオプションの操作性が向上しました。
 - コマンド実行の結果がコンソールに表示されるようになりました。- ノード番号の代わりに HA ノード名など、より長い入力パラメータを送信できるようになりました。

お気に入りカスタムグラフに対応しなくなりました

Monitoring -> Hosts -> Graphs でカスタムグラフをお気に入りに追加することができなくなります。アップグレード後、既存のカスタムグラフはすべてお気に入りから削除されます。

サービス監視

サービス監視機能に関連するいくつかの **メジャーアップデート** が行われました。
 既存のサービスツリー構成は、アップグレード中に以下のように変更されます。

- 問題とサービスの間のトリガーベースの依存関係は、サービスと問題のタグベースのマッピングに置き換えられます。
 サービスにリンクされたトリガーには、新しいタグ `ServiceLink: <trigger ID>:<trigger name>` (タグ値は 32 文字に切り詰められます) が設定されます。
 リンクされたサービスには、同じ **problem tag** が付けられます。

- ハード依存とソフト依存は、もはや存在しません。その代わりに、1 つのサービスは複数の親サービスを持つことになります。

- 'Status calculation algorithm' が、以下のルールでバージョンアップされます。

 - 計算しない → ステータスを OK にする
 - 子ノードのうち 1 つでも問題があれば問題あり → 子ノードのうち最も重要なもの
 - すべての子ノードに問題がある場合、問題あり → すべての子ノードに問題がある場合、最もクリティカルになる
- SLA はもはやサービス属性ではなく、複数のサービスに割り当てることができる別個のエンティティです。
 アップグレードの際、同一の SLA がグループ化され、各グループに 1 つの SLA が作成されます。
 サービスには、マッチング用に新しい **service tag SLA:<ID>** が付与されます。

こちらをご参照ください。

- **service monitoring upgrade** の詳細
- **services** の設定

11 6.0.1 へのアップグレードノート

Configuration Syncer

Configuration Syncer のパフォーマンスが向上しました。
 テンプレートが大量にある場合は、server / proxy のキャッシュサイズを大きくすることを推奨します。
 また、未使用のテンプレートは削除することをお勧めします。

server / proxy のデフォルトのキャッシュサイズが 32M に増加しましたので、ご注意ください。

item の変更

Zabbix agent 2 に **item net.dns** と **net.dns.record** の
 ネイティブサポートが追加されました。Windows 版 Zabbix agent 2 では、これらのアイテムは ip パラメータで
 カスタム DNS IP アドレスを使用できるようになり、timeout と count パラメータを無視しないようになりました。

12 6.0.2 へのアップグレードノート

このマイナーバージョンには、アップグレードノートはありません。

13 6.0.3 へのアップグレードノート

このマイナーバージョンには、アップグレードノートはありません。

14 6.0.4 へのアップグレードノート

このマイナーバージョンには、アップグレードノートはありません。

15 6.0.5 へのアップグレードノート

このマイナーバージョンには、アップグレードノートはありません。

16 6.0.6 へのアップグレードノート

外部 **MongoDB** プラグイン ローダブル MongoDB プラグイン

MongoDB **プラグイン** は、Zabbix エージェント 2 の一部ではなくなり、ロード可能なプラグインとして利用できるようになりました。サポートされている MongoDB バージョンのリストが 2.6-5.3 に拡張されました。

MongoDB プラグインは、将来追加される可能性のある他のロード可能なプラグインとともに、新しいリポジトリ **zabbix-agent2-plugins** に保存されます。パッケージ zabbix-release は、このリポジトリをユーザーのシステムに追加します。

公式の Zabbix パッケージからのインストールで MongoDB の監視を継続するには、zabbix-release パッケージを更新し、zabbix-agent2-plugin-mongodb パッケージをインストールします。

ソースは [CDN](#) で入手できます。

17 6.0.7 へのアップグレードノート

シンボリックリンク名の展開 シンボリックリンク名とシンボリックリンクのフルパスは、シンボリックリンクターゲットに解決される代わりに、`vfs.dir.get []` および `vfs.file.get []` アイテムで返されるようになりました。

18 6.0.8 へのアップグレードノート

このマイナーバージョンには、アップグレードノートはありません。

19 6.0.9 へのアップグレードノート

このマイナーバージョンには、アップグレードノートはありません。

20 6.0.10 へのアップグレードノート

重大な変更

PostgreSQL プラグインはロード可能なプラグインに移動しました PostgreSQL **プラグイン** は、Zabbix エージェント 2 に組み込まれなくなり、ロード可能なプラグインになりました。

この変更により、プラグインリポジトリを直接プルすることができなくなったため、Ansible、Chef などの自動化が機能しなくなる可能性があります。

[PostgreSQL loadable plugin](#) リポジトリを参照してください。

21 6.0.11 へのアップグレードノート

重大な変更

DB スキーマの更新 新しいカラム `name_upper` がデータベースのテーブル **items** に追加されました。カラムには、アイテム名または LLD ルール名の大文字バージョンが設定されます。この変更は、API クエリ**最適化**の大文字と小文字を区別しない検索でインデックスを使用できるようにするために実装されました。

API クエリの最適化 `hosts` テーブルと `items` テーブルで名前を検索するときに使用される API のデータベースクエリが最適化され、より効率的に処理されるようになりました。この変更の結果、アップグレード中に決定論的トリガーを作成する必要があります。

MySQL と MariaDB では、バイナリロギングが有効で、スーパーユーザー権限がなく、MySQL 構成ファイルに `log_bin_trust_function_creators = 1` が設定されていない場合、GLOBAL `log_bin_trust_function_creators = 1` を設定する必要があります。MySQL のコンソールを使用して変数を設定するには、以下を実行します：

```
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
```

アップグレードが正常に完了したら、`log_bin_trust_function_creators` を無効にできます：

```
mysql> SET GLOBAL log_bin_trust_function_creators = 0;
```

トリガーは、PostgreSQL および Oracle データベース用にも作成されます。

JSONPath 解析エラー JSONPath の解析エラーは、空白で始まる場合と空の配列/オブジェクトの場合にこのバージョンで発生します。Zabbix 6.0.12 で修正されました。

LLD フィルター内の AND/OR の評価 このバージョンでは、ローレベルディスカバリーフィルター/オーバーライドでの AND/OR の評価が失敗する可能性があります。Zabbix 6.0.12 で修正されました。

22 6.0.12 へのアップグレードノート

history syncers のパフォーマンス改善 history syncer のパフォーマンスは、新しい read-write ロックを導入することで改善されました。これにより、構成キャッシュにアクセスしながら共有読み取りロックを使用して、history syncer、trapper、proxy poller 間のロックが削減されます。新しいロックは、構成キャッシュリロードを実行する configuration syncer によってのみロックされた書き込みができます。

23 6.0.13 へのアップグレードノート

重要な変更 ロード可能なプラグインのバージョン管理

Zabbix エージェント 2 の **ロード可能なプラグイン** は、Zabbix 自体と同じバージョン管理システムを使用するようになり、次のバージョンに変更されました。

- MongoDB 1.2.0 -> MongoDB 6.0.13
- PostgreSQL 1.2.1 -> PostgreSQL 6.0.13

これらのプラグインは、Zabbix 6.0 のマイナーバージョンでサポートされています。各プラグインのソースコードリポジトリには、専用の release/6.0 ブランチが含まれるようになりました (以前は master ブランチのみがありました)。

構成のインポート **構成のインポート** プロセスの変更を参照してください。

Zabbix エージェント 2 を使用して **Oracle** データベース内の個別のテーブルスペースでクエリを実行する Oracle プラグインでサポートされている次の **Zabbix エージェント 2 アイテム** に、次のオプションパラメーターが追加されました。

```
-oracle.diskgroups.stats[< 既存のパラメータ >,< ディスクグループ >] -oracle.archive.info[< 既存のパラメータ >,< 宛先 >] -oracle.cdb.info[< 既存のパラメータ >,< データベース >] -oracle.pdb.info[< 既存のパラメータ >,< データベース >] -oracle.ts.stats[< 既存のパラメータ >,< テーブルスペース >,< タイプ >]
```

これらのパラメーターを使用すると、すべてのデータではなく個別のインスタンスでクエリを実行できるため、パフォーマンスが向上します。

Zabbix エージェント 2 のオープンファイルディスクリプタ制限の緩和 Zabbix エージェント 2 パッケージに同梱されている systemd サービスファイルは、オープンファイル記述子の制限を 8196 と宣言するようになりました。以前は、システムのデフォルト制限である 1024 が使用されていました。新しい制限は、デフォルトの Zabbix エージェント 2 を構成するには十分です。非標準のエージェント 2 構成を使用している場合 (たとえば追加のプラグインや拡張機能を使用している場合)、この制限を手動でさらに増やす必要がある場合があります。その場合 systemd ユニットファイルの LimitNOFILE パラメーターを調整します。

24 6.0.14 へのアップグレードノート

JavaScript のプリプロセッシングにおけるオブジェクトの制限 プリプロセッシングにおける **JavaScript オブジェクト** に対して次の制限が導入されました。

- log() メソッドでログに記録できるすべてのメッセージの合計サイズは、スクリプト実行ごとに 8MB に制限されています。
- 複数の「HttpRequest」オブジェクトの初期化は、スクリプト実行ごとに 10 に制限されています。
- addHeader() メソッドを使用して単一の HttpRequest オブジェクトに追加できるヘッダーフィールドの合計長は 128K バイトに制限されています (特殊文字とヘッダー名を含む)。

25 6.0.15 へのアップグレードノート

このマイナーバージョンにアップグレードノートはありません。

26 6.0.16 へのアップグレードノート

このマイナーバージョンにアップグレードノートはありません。

27 6.0.17 へのアップグレードノート

Geomap アトリビューションでの **HTML** サポートが終了 **Geomap ダッシュボードウィジェット** の帰属テキストから HTML サポートが廃止され、プレーンテキストのみを含めることができるようになりました。このフィールドに既に HTML が含まれている場合、アップグレード後にはプレーンテキストとして表示されます。

地理マップ の設定 [管理] → [全般] セクションで、[属性] フィールドは、[タイルプロバイダー] が **その他** に設定されている場合にのみ表示されるようになりました。

28 6.0.18 へのアップグレードノート

プロキシ履歴のハウスキーピング プロキシ履歴のハウスキーピングの際に、プロキシのデータベースから削除されるサイクルあたりのデータ量の制限が削除されました。

変更前のhousekeeper はHousekeepingFrequency 4 倍時間分までしか、古い情報の削除をしませんでした。例えばHousekeepingFrequency が"1" に設定されている場合、最大 4 時間分の情報が(古いエントリーから順番に)削除されました。プロキシがProxyOfflineBuffer で設定されたデータよりも古いデータを絶えず受信する場合、過剰なデータ蓄積が発生する可能性があります。

この制限が取り除かれたことで、より効果的なプロキシ履歴ハウスキーピングが行えるようになりました。

29 6.0.19 へのアップグレードノート

集計関数 count_foreach 関数は、項目のデータが存在しない場合、またはデータがフィルターに一致しない場合、配列内の一致する項目に対して「0」を返すようになりました。以前はそのような項目は無視されていました(集計にデータは追加されませんでした)。

30 Upgrade notes for 6.0.20

This minor version doesn't have any upgrade notes.

31 Upgrade notes for 6.0.21

Maximum JSON Depth

The maximum allowed JSON depth has been set to 64 to improve security and performance during JSON parsing.

Macro functions

The range of the fmtnum macro function is now limited to 0-20.

32 Upgrade notes for 6.0.22

Autoregistration table cleared from orphaned records

Previously the autoreg_host table was never cleared. Now orphaned records that reference neither an autoregistration event nor an existing host will be removed by the Housekeeper.

PostgreSQL plugin parameters**

The following PostgreSQL plugin parameters are no longer mandatory if Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect is set to verify_ca or verify_full:

- Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile
- Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

33 Upgrade notes for 6.0.23

Breaking changes Agent 2 plugins

The correct master host is now returned if specified in the `mysql.replication.get_slave_status[connString,<username>,<password>]` item "masterHost" parameter. Previously the "masterHost" parameter was ignored and always the first master host was returned.

If this parameter is not specified, all hosts are returned.

MySQL plugin parameters

The following MySQL plugin parameters are no longer mandatory if `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` is set to `verify_ca` or `verify_full`:

- `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile`
- `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile`

PostgreSQL-TimescaleDB combination no longer checked

Zabbix no longer checks for the supported PostgreSQL-TimescaleDB combination. As before, Zabbix does check for the supported PostgreSQL version or TimescaleDB version separately.

34 Upgrade notes for 6.0.24

This minor version doesn't have any upgrade notes.

MongoDB plugin parameters

The following MongoDB plugin parameters are no longer mandatory if `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` is set to `verify_ca` or `verify_full`:

- `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile`
- `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile`

35 Upgrade notes for 6.0.25

This minor version doesn't have any upgrade notes.

Consistency introduced in sha256 checksums of `vfs.file.cksum` item

In earlier Zabbix versions, the Zabbix agent item `vfs.file.cksum` produced different sha256 sums for the same file depending on the platform (processor architecture) – sha256 sums on AIX, HP-UX (Itanium) and Solaris (SPARC) differed from those produced on Intel x86-64 platforms.

The issue has now been fixed; however, after upgrading, monitoring sha256 sums of files on AIX, HP-UX, or Solaris SPARC may result in false positives of files having been modified.

36 Upgrade notes for 6.0.26

This minor version doesn't have any upgrade notes.

37 Upgrade notes for 6.0.27

This minor version doesn't have any upgrade notes.

Zabbix agent 2 support on Windows

To prevent critical security vulnerabilities, the minimum Windows version for Zabbix agent 2 has been raised to Windows 10/Server 2016. See note under [Supported platforms](#) for more information.

38 Upgrade notes for 6.0.28

Guest user authorization

Automatic login for the guest user has been removed. After this change, the guest user will need to log in like any other user. Previously, a guest could immediately get to almost any monitoring or reporting page without going through authorization.

Invalid regular expression in proc.* items

proc.* agent items will now become 'not supported' if an invalid regular expression is supplied.

39 Upgrade notes for 6.0.29

This minor version doesn't have any upgrade notes.

5. クイックスタート

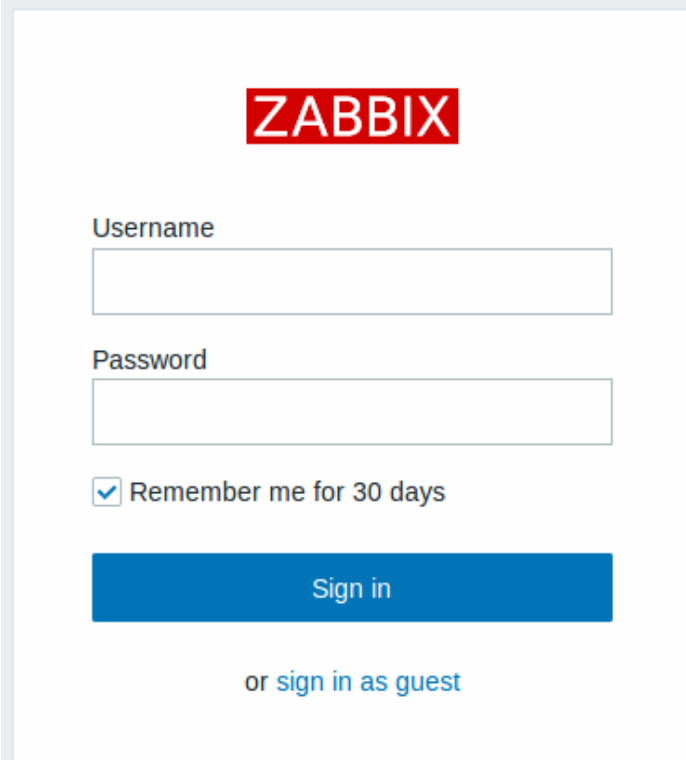
サイドバーからクイックスタートセクションのコンテンツにアクセスしてください。

1 ログインとユーザー設定

概要

このセクションでは、Zabbix にログインしてシステムユーザーを設定する方法を学習します。

ログイン



The screenshot shows the Zabbix login interface. At the top center is the Zabbix logo in white text on a red rectangular background. Below the logo are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Underneath the password field is a checkbox with a checkmark inside, followed by the text 'Remember me for 30 days'. Below this is a solid blue button with the white text 'Sign in'. At the bottom of the form area, there is a link that reads 'or sign in as guest' in blue text.

これは Zabbix のウェルカム画面です。ユーザー名に **Admin** パスワードに **zabbix** と入力して Zabbix の **スーパーユーザー** としてログインします。構成および管理メニューへのアクセスが可能となります。

ブルートフォース攻撃からの保護

ログイン試行が 5 回連続して失敗した場合、Zabbix インターフェースは総当たり攻撃と辞書攻撃を防ぐために 30 秒間一時停止します。

失敗したログイン試行の IP アドレスは、ログインに成功時表示されます。

ユーザーの追加

ユーザーに関する情報を表示するには、[管理]→**ユーザー**に移動します。

<input type="checkbox"/>	Alias	Name	Surname	User role	Groups	Is online?	Login	Frontend access	API access	Debug mode	Status
<input type="checkbox"/>	Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2020-10-28 11:38:16)	OK	System default	Enabled	Enabled	Enabled
<input type="checkbox"/>	guest	John	Snow	User role	Guests	No (2020-07-16 11:06:52)	OK	System default	Enabled	Disabled	Disabled

Displaying 2 of 2 found

新しいユーザーを追加するには、[ユーザーの作成] をクリックします。

新しいユーザーフォームで、必ず既存の**ユーザーグループ**の 1 つにユーザーを追加してください。例:Zabbix administrators

User Media Permissions

* Alias

Name

Surname

* Groups X
type here to search

* Password

* Password (once again)

すべての必須入力フィールドは、赤いアスタリスクでマークされています。

デフォルトでは、新規ユーザーにはメディア（通知配信方法）がありません。作成するには、[メディア] タブに移動し、追加をクリックします。

Type

* Send to [Remove](#)

[Add](#)

* When active

Use if severity Not classified
 Information
 Warning
 Average
 High
 Disaster

Enabled

このポップアップで、ユーザーの電子メールアドレスを入力します。

メディアがアクティブになる期間を指定できます。(入力フォーマットについては[期間の指定](#)ページを参照してください。デフォルトでは、メディアは常にアクティブです。メディアがアクティブになる[トリガーの重大度レベル](#)をカスタマイズすることもできますが、現時点ではすべてを有効のままにしておきます。

[追加] をクリックしてメディアを保存し、[権限] タブに移動します。

[権限] タブには、必須フィールドロールがあります。このロールは、ユーザーが表示できるフロントエンド要素と、ユーザーが実行できるアクションを決定します。[選択] を押してリストからロールの 1 つを選択します。たとえば Admin role を選択して、すべての Zabbix フロントエンドセクションへのアクセスを許可します。後で権限を変更したり、より多くのユーザーロールを設定することも可能です。ロールを選択すると、権限が同じタブ内に表示されます。

User Media **Permissions**

Role: Admin role

User type: Admin

Permissions: Host group: All groups, Permissions: None

Permissions can be assigned for user groups only.

Access to UI elements

Monitoring: Dashboard, Problems, Hosts, Overview, Latest data, Maps, Discovery, Services

Inventory: Overview, Hosts

Reports: Availability report, Triggers top 100, Notifications, Scheduled reports

Configuration: Host groups, Templates, Hosts, Maintenance, Actions, Discovery, Services

Access to modules
No enabled modules found.

Access to API
Enabled

Access to actions

Create and edit dashboards, Create and edit maps, Create and edit maintenance

Add problem comments, Change severity, Acknowledge problems, Close problems, Execute scripts

Manage API tokens, Manage scheduled reports

ユーザープロパティフォームの [追加] をクリックして、ユーザーを保存します。新しいユーザーがユーザーリストに表示されます。

<input type="checkbox"/>	Alias ↕	Name	Surname	User role	Groups	Is online?	Login	Frontend access	API access	Debug mode	Status
<input type="checkbox"/>	Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2020-10-28 11:42:05)	OK	System default	Enabled	Enabled	Enabled
<input type="checkbox"/>	guest	John	Snow	User role	Guests	No (2020-07-16 11:06:52)	OK	System default	Enabled	Disabled	Disabled
<input type="checkbox"/>	user			Admin role	Zabbix administrators	No	OK	System default	Enabled	Enabled	Enabled

Displaying 3 of 3 found

権限の追加

デフォルトでは、新しいユーザーにはホストにアクセスするための権限がありません。権限を付与するには [グループ] 列のグループをクリックします。(この場合-'Zabbix administrators') グループプロパティフォームで [権限] タブに移動します。

≡ User groups

User group Permissions Tag filter

Permissions	Host group	Permissions
	All groups	None

type here to search Select Read-wr

Include subgroups

[Add](#)

Update Delete Cancel

このユーザーは Linux servers グループへの読み取り専用アクセス権を持っているため、ユーザーグループ選択フィールドの横にある選択をクリックします。

Host groups

- Name
- Discovered hosts
- Hypervisors
- Linux servers
- Templates
- Templates/Applications
- Virtual machines
- Zabbix servers

Select

このポップアップで、[Linux servers] の横にあるチェックボックスをオンにして [選択] をクリックすると Linux servers が選択フィールドに表示されます。[読み取り] ボタンをクリックしてアクセス許可レベルを設定し、[追加] をクリックしてグループをアクセス許可のリストに追加します。ユーザーグループのプロパティフォームで、[更新] をクリックします。

Zabbix では、ホストへのアクセス権は個々のユーザーではなく、**ユーザーグループ**に割り当てられます。

以上で新しいユーザーの資格情報を使用してログインを試みることができます。

2 新規ホスト

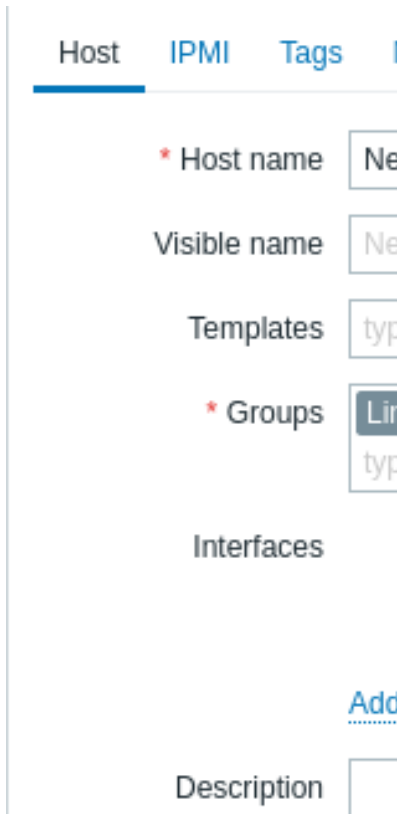
概要

このセクションでは、新しいホストを設定する方法を学習します。

Zabbix の「ホスト」とは、監視するネットワークエンティティ（物理、仮想）です。Zabbix で「ホスト」とすることができるものは非常に柔軟です。物理サーバー、ネットワークスイッチ、仮想マシン、または一部のアプリケーションもホストとすることができます。

ホストの追加

Zabbix で設定されたホストに関する情報は、設定 → ホストとモニタリング → ホストで利用できます。「Zabbix server」という事前定義されたホストがすでに 1 つありますが、別のホストを追加します。



新しいホストを追加するには、**ホストの作成**をクリックします。するとホスト構成フォームが表示されます。

すべての必須入力フィールドは、赤いアスタリスクでマークされています。

最低限入力が必要な項目は次のとおりです。

ホスト名 -ホスト名を入力します。英数字、スペース、ドット、ダッシュ、アンダースコアを使用できます。

グループ -[選択] ボタンをクリックして 1 つまたは複数の既存のグループを選択するか、存在しないグループ名を入力して、新しいグループを作成します。

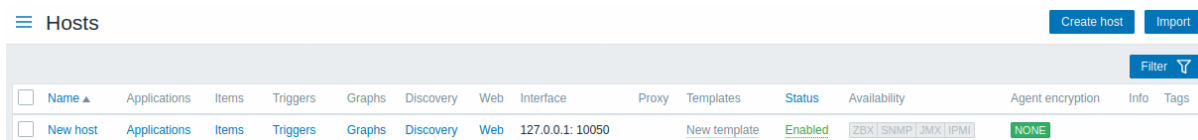
すべてのアクセス許可は、個々のホストではなく、ホストグループに割り当てられます。そのため、ホストは少なくとも 1 つのグループに属している必要があります。

インターフェース : **IP** アドレス




技術的には必須フィールドではありませんが、ホストの IP アドレスを入力することをお勧めします。これが Zabbix サーバーの IP アドレスである場合は、Zabbix エージェント設定ファイルの「サーバー」ディレクティブで指定する必要があることに注意してください。

その他のオプションは今のところデフォルトとします。

完了したら、[追加] をクリックします。新しいホストがホストリストに表示されます。



[Availability] 列には、各インターフェイスごとのホストの可用性のインジケータが含まれています。Zabbix エージェントインターフェイスを定義したので、エージェントの可用性アイコン（「ZBX」が付いている）から、ホストの可用性を確認できます。

-  - ホストステータスが確立されていません。メトリックチェックはまだ行われていません
-  - ホストが利用可能で、メトリックチェックが成功しました
-  - ホストが利用できないため、メトリックチェックに失敗しました（マウスカーソルをアイコンの上に移動すると、エラーメッセージが表示されます）。インターフェースのクレデンシャルが正しくないことが原因で、通信にエラーが発生している可能性があります。Zabbix サーバーが実行されていることを確認し、後でページを更新してみてください。

3 新規アイテム

概要

このセクションでは、どうやってアイテムを設定するかを説明します。

アイテムとは、Zabbix でデータを収集する基礎となるものです。アイテムなくしてデータは存在しえません。アイテムこそがメトリックの指標となり、ホストから収集するデータの種類を定義します。

アイテムの追加

全てのアイテムはホスト周辺でグループ化されてます。なので、サンプルアイテムを設定するには、設定からホストへ行き、作成した“新しいホスト”を見つけます。

新しいホストからアイテムのリンクをクリックし、アイテムの作成をクリックします。これによってアイテムを定義するフォームが表示されます。

The screenshot shows the Zabbix Item configuration form with the following fields and values:

- Name:** CPU load
- Type:** Zabbix agent
- Key:** system.cpu.load
- Type of information:** Numeric (float)
- Host interface:** 127.0.0.1:10050
- Units:** (empty)
- Update interval:** 1m
- Custom intervals:**

Type	Interval	Period
Flexible Scheduling	50s	1-7,00:00-24:00

[Add](#)
- History storage period:** Storage period, 90d
- Trend storage period:** Storage period, 365d

入力が必須の項目には、赤いアスタリスクが表記されます。

上記のサンプル図では、必須項目はこのようになります。

名称

CPU load を入力します。これが表示されるアイテムの名称となります。

キー手動で system.cpu.load を数値として入力します。これはアイテムの技術的な名称となり、収集されるデータの種類の識別に使われます。このキーは Zabbix Agent によって前もって定義されているキーの一つです。

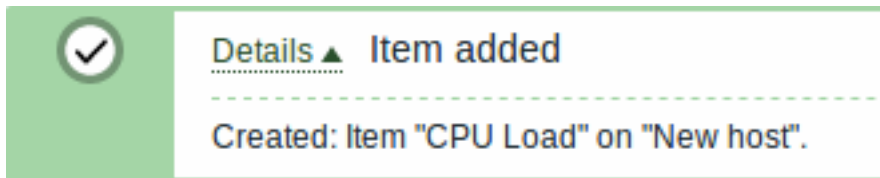
情報の種類この項目は予想されるデータの形式を定義します。system.cpu.load キーの場合、ここは自動的に Numeric (float) として設定されます。

Note:

アイテム履歴の保存期間を 7 日か 14 日に短縮することも出来ます。これはデータベースを大量の履歴から解放する良い手段となります。

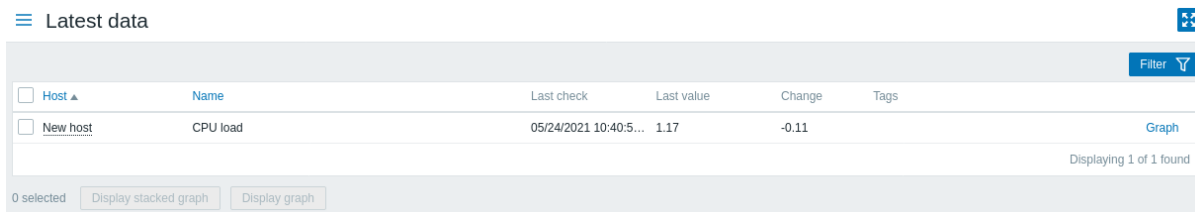
他のオプションについては、現状はデフォルトのままで良いと思われます。

完了したら追加をクリックします。新しいアイテムはアイテムリスト上に表示されるはずですが、リストの上の詳細をクリックすると、何が起きたのかを確認する事が出来ます。



データの閲覧

アイテムが設定された後、データがきちんと収集されたか気になる方は、モニタリングから最新のデータへ移動し、フィルター内の新規ホストで適用をクリックしてください。



とはいえ、最初のデータが反映されるまで 60 秒程かかることがあります。これはデフォルトの設定として、サーバーが設定の変更を読み取るのと、新しいアイテムの実行にかかるようになっています。

もし変更列に何も数値が表示されない場合、それはその時点で数値が一つしか受け取れてないのかもしれませんが。他の数値が届くまで 30 秒お待ちください。

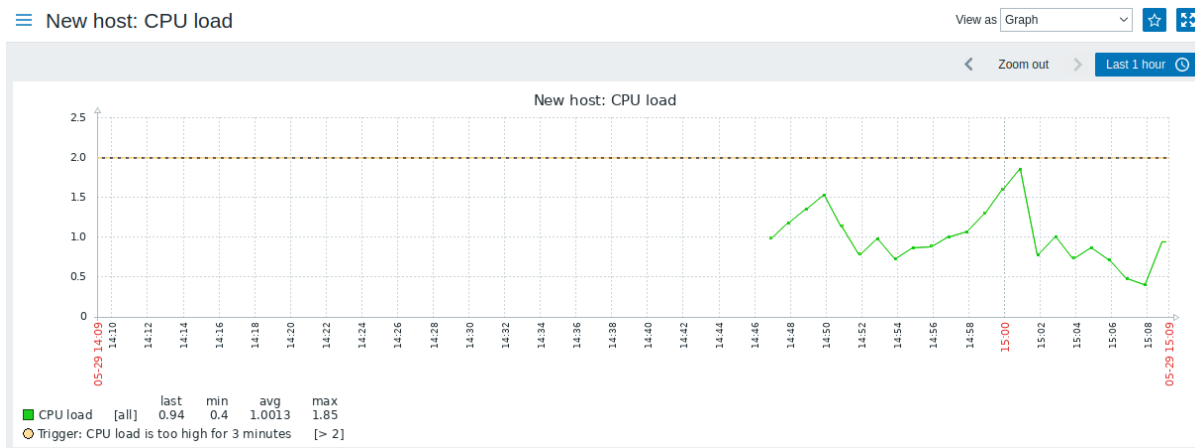
もしアイテムに関する情報をスクリーンショット上に表示されていない場合、下記の項目をチェックしてみてください。

- スクリーンショットと同様の数値をキー、情報の種類の二つのアイテム上で入力してある。
- エージェントとサーバーが両方とも起動している。
- ホストのステータスが「監視中」になっていて、その利用ステータスアイコンが緑になっている。
- ホストドロップダウンでホストが選択されており、その項目がアクティブになっている。

グラフ

アイテムがしばらくの間実行されてるのなら、その様子を視覚的に確認することが出来るかもしれません。シンプルグラフは監視してる全ての数値アイテムに対し、設定の変更不要で利用できます。グラフは実行時に作成されます。

グラフを見るには、モニタリング → 最新のデータに行き、アイテムの横にあるグラフのリンクをクリックします。



4 新規トリガー

概要

このセクションでは、トリガーを設定する方法を学習します。

アイテムはデータのみを収集します。受信データを自動的に評価するには、トリガーを定義する必要があります。トリガーには、データの許容レベルのしきい値を定義する式を含めます。

受信データがしきい値を超えるとトリガーが「起動」するか、ステータスが「問題」になり、注意が必要なことが発生したことを通知します。データがしきい値を下回るとトリガーはステータスを「OK」に戻します。

トリガーの追加

アイテムのトリガーを構成するには Configuration → Hosts に移動し、'New host' を見つけて、その横にある Triggers をクリックし Create trigger をクリックします。これによりトリガー定義フォームが表示されます。

Trigger Tags Dependencies

* Name CPU load too high on 'New host' for 3 minutes

Event name CPU load too high on 'New host' for 3 minutes

Operational data

Severity Not classified Information Warning Average High Dis

* Expression avg(/New host/system.cpu.load, 3m)>2 Add

[Expression constructor](#)

OK event generation Expression Recovery expression None

PROBLEM event generation mode Single Multiple

OK event closes All problems All problems if tag values match

Allow manual close

URL

Description

Enabled

Add Cancel

入力必須項目は次のとおりです。

Name

- 今回は CPU load too high on 'New host' for 3 minutes と設定しています。これは、リストなどに表示されるトリガー名になります。

Expression

- 今回は次のように入力します : avg(/New host/system.cpu.load,3m)>2

トリガー式が最後の記号まで正しく入力されていることを確認してください。ここでのアイテムキー (system.cpu.load) は、アイテムを参照するために使用されます。この式は、3 分間の CPU 負荷平均値が 2 を超過することをしきい値としています。[トリガー式の構文解説](#)

完了したら Add をクリックします。新しいトリガーがトリガーリストに表示されます。

トリガーステータスの表示

トリガーを定義すると、ステータスを確認したい場合があります。

CPU 負荷がトリガーで定義したしきい値を超えた場合、Monitoring → Problems に表示されます。

Time	<input type="checkbox"/> Severity	Recovery time	Status	Info	Host ▲	Problem	Operational data	Duration
16:23:06	<input type="checkbox"/> Not classified		PROBLEM		New host	CPU load too high on "New host" for 3 minutes	6.6	56s

ステータス列の点滅は、過去 30 分間に発生したトリガーステータスの最近の変更を示します。

5 障害発生のお知らせ

概要

このセクションでは、Zabbix で通知形式のアラートを設定する方法について説明します。

データを収集するアイテムや、障害発生時に「fire」と通知するトリガーがあれば、Zabbix フロントエンドを直接触っていない時でも、重要なイベント等を警告してくれる機能として便利です。

これが通知機能の事です。障害発生時に、Eメールは最もよく使われる通知手段となります。ここではどうやって Eメール通知を設定するか学びます。

Eメール設定

インストール直後の Zabbix には、既定の配信方法がいくつかあります。Eメールもその一つです。

Eメールの設定には、管理 → メディアタイプと進み、設定済みの一覧の中から Email を選んでクリックします。

Media types

<input type="checkbox"/>	Name ▲	Type	Status	Used in actions	Details
<input type="checkbox"/>	Email	Email	Enabled		SMTP server: "mail.zabbix.com",
<input type="checkbox"/>	Mattermost	Webhook	Enabled		
<input type="checkbox"/>	Opsgenie	Webhook	Enabled		

Eメール設定フォームが表示されます。

Media types

Media type	Message templates	Options
		<p>* Name <input type="text" value="Email"/></p> <p>Type <input type="text" value="Email"/></p> <p>* SMTP server <input type="text" value="mail.zabbix.com"/></p> <p>SMTP server port <input type="text" value="25"/></p> <p>* SMTP helo <input type="text" value="zabbix.com"/></p> <p>* SMTP email <input type="text" value="zabbix-info@zabbix.com"/></p> <p>Connection security <input type="text" value="None"/> <input type="text" value="STARTTLS"/> <input type="text" value="SSL/TLS"/></p> <p>Authentication <input type="text" value="None"/> <input type="text" value="Username and password"/></p> <p>Message format <input type="text" value="HTML"/> <input type="text" value="Plain text"/></p> <p>Description <input type="text"/></p> <p>Enabled <input checked="" type="checkbox"/></p> <p><input type="button" value="Add"/> <input type="button" value="Cancel"/></p>

入力必須項目には赤いアスタリスクが表示されます。

[メディアタイプ] タブで、SMTP サーバー、SMTP helo、送信元メールアドレスの値を、お使いの環境に合わせて設定してください。

Note:

'送信元メールアドレス' は、Zabbix から送信される通知の 'From' アドレスとして使用されます。

次に、障害メッセージの内容を定義します。内容はメッセージテンプレートタブで設定されたメッセージテンプレートによって定義されます。

追加をクリックしてメッセージテンプレートを作成し、メッセージタイプとして 障害を選択します。

Message template ✕

Message type

Subject

Message

```
Problem started at {EVENT.TIME} on {EVENT.DATE}
Problem name: {EVENT.NAME}
Host: {HOST.NAME}
Severity: {EVENT.SEVERITY}
Operational data: {EVENT.OPDATA}
Original problem ID: {EVENT.ID}
{TRIGGER.URL}
```

入力完了後、追加をクリックして、フォームを保存します。

これで機能するメディアタイプとして'Email' が設定されました。メディアタイプは(新しいユーザーの設定の時に行ったように)、特定の配信アドレスを設定してユーザーとリンクさせる必要があります。そうしないと使用されません。

新規アクション

Zabbixのアクションの動作の一つに、通知の配信があります。通知を設定するには、設定→アクションと進み、アクションの作成をクリ

☰ Actions

Action
Operations

*** Name**

Conditions	Label	Name
	Add	

Enabled

*** At least one operation must exist.**

ックします。

入力が必須な項目には赤いアスタリスクが表示されます。

この入力フォームでは、アクションの名前を入力します。

最も単純なケースでは、特定の条件を設定しない限り、トリガーが OK から Problem に変化した際にアクションが実行されます。

「アクション」は何をすべきかの定義が必要で、オペレーションタブで行われます。操作ブロック内の追加をクリックすると、新しい操作フォームが表示されます。

Operation details ✕

Operation type

Steps - (0 - infinitely)

Step duration (0 - use action default)

*** At least one user or user group must be selected.**

Send to User groups	User group	Action
	Add	

Send to Users	User	Action
	user (New User)	Remove
	Add	

Send only to

Custom message

Conditions	Label	Name	Action
	Add		

入力が必要な項目には赤いアスタリスクが表示されます。

ユーザーへの送信内の追加をクリックし、定義済みのユーザー (user) を選択します。Send only to の値に E メールを設定します。選択後、追加をクリックすると、動作が追加されます。

≡ Actions

Action Operations

*** Default operation step duration**

Pause operations for suppressed problems

Operations	Steps	Details	Start in	Duration
	1	Send message to users: user (New User) via Email	Immediately	Default
	Add			

以上が単純なアクションの設定となります。アクションフォームの追加をクリックしてください。

通知受信

通知の配信の設定が出来たら、実際に通知を受け取ってみましょう。

トリガーが「fires」になり、障害通知を受け取れるように、わざとホストの CPU 負荷を上げます。

ホストのコンソールを開き、以下を実行してください。

cat /dev/urandom | md5sum

md5sumは一度または複数回でも実行します。

次にモニタリング → 最新データへ行き、CPU Load の値がどの程度増加したかを確認します。トリガーが fire になるには、3 分間の動作で「CPU 負荷」の値が「2」を超える必要があることを覚えておきましょう。一度でもそうなれば... - モニタリング → 障害で、障害が点滅してる筈です。 - あなたの E メールアドレスに障害発生 of E メールが届いてる筈です。

Attention:

もし通知が上手く動作してない場合 - もう一度 E メール of 設定とアクション of 設定が正常かを確認してみてください。 - **ユーザーの追加**にて説明したように、作成したユーザーが、そのイベントを生成したホストに関して、少なくとも読込のパーミッションを持っていることを確認してください。Zabbix Administrators のユーザーグループにこのユーザーは所属しており、ホストが所属している Linux servers ホストグループに対して、少なくとも読み取り権限を得ている必要があります。 - レポート → アクション履歴で更にアクション履歴を確認できます。

6 新規テンプレート

概要

このセクションでは、テンプレートの設定について学びます。

前回はアイテム、トリガー、問題発生時のメールによる通知の設定方法を学びました。

これらの手順は柔軟性に富んではいますが、例えば千台のホストに問題が発生した場合には多くの手順を踏む必要があるため、自動化することが出来れば便利と思われれます。

そこで役に立つのがテンプレートです。これは役に立つアイテムやトリガー等をグループ化して、一度の手順で何度も再使用できるようにする為のものです。

テンプレートがホストにリンクされると、ホストはそのテンプレートの全容を学習します。ですので、基本的には前もって準備されたチェックの数々を迅速に適用することができます。

テンプレートの追加方法

テンプレートを使用するには、まずテンプレートを作成する必要があります。設定 → テンプレートと進み、テンプレートの作成をクリ

The screenshot shows the 'Add Template' dialog box in Zabbix. It has four tabs: 'Template', 'Tags', 'Macros', and 'Value mapping'. The 'Template' tab is active. The form contains the following fields:

- * Template name: New template
- Visible name: New template
- Templates: type here to search
- * Groups: Templates (selected), type here to search
- Description: (empty text area)

At the bottom right, there are two buttons: 'Add' and 'Cancel'.

ックしてください。すると、テンプレート設定フォームが開きます。

記入が必須の項目には、赤いアスタリスクが表記されます。

入力必須なパラメーターは以下の通りです。

テンプレート名 - テンプレート名を入力します。アルファベット、数字、スペース、下線を使用可能です。

グループ - セレクトボタンをクリックして、1つ、または複数のグループを選択します。テンプレートはグループに所属している必要があります。

選択が終わったら、「追加」をクリックします。これで新規作成したテンプレートがテンプレートリストに表示されます。

≡ Templates

<input type="checkbox"/>	Name ▲	Hosts	Items	Triggers	Graphs	Dashboards	Discovery	Web
<input type="checkbox"/>	New template	Hosts	Items	Triggers	Graphs	Dashboards	Discovery	Web

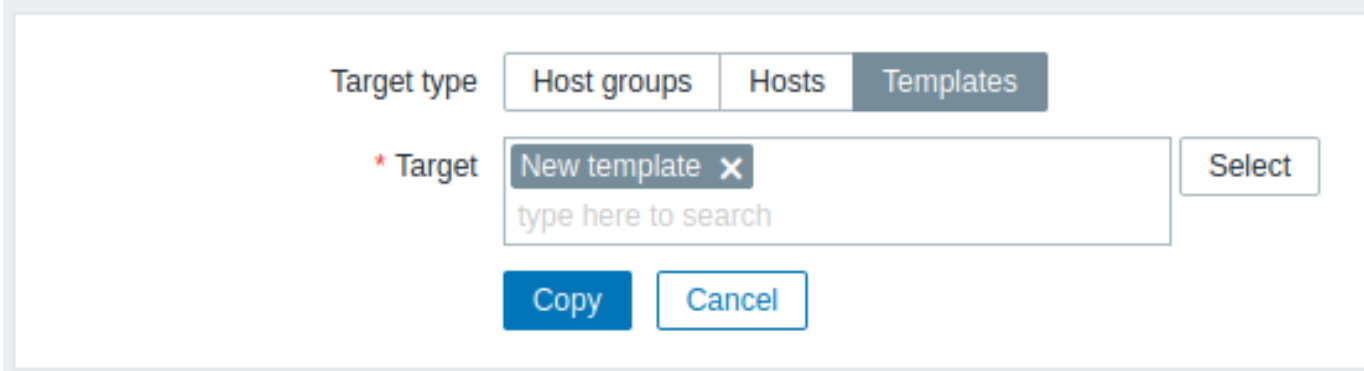
御覧の通り、テンプレートはそこにはありますが、まだその中にはアイテムやトリガー等の実体は入っていません。

テンプレートにアイテムを追加する

アイテムをテンプレートに挿入するには、新規ホストのアイテムリストに移動します。設定 → ホストでアイテム新規ホストの横にあるアイテムをクリックします。

そして

- CPU Load のチェックボックスをチェックを入れます。
- リストの下にあるコピーをクリックします。
- アイテムをコピーするテンプレートを選びます。



The screenshot shows a configuration window for a Zabbix template. At the top, there are three tabs: 'Host groups', 'Hosts', and 'Templates', with 'Templates' selected. Below the tabs is a search field labeled '* Target' containing the text 'New template' with a close button (X) and a 'Select' button. Below the search field are two buttons: 'Copy' and 'Cancel'.

記入が必須の項目には、赤いアスタリスクが表記されます。

- コピーのクリックこの段階で設定 → テンプレートに行くと、新規テンプレートに新しいアイテムが追加されているはずですが。

この段階では 1 つの項目だけにしておきますが、同様にほかのアイテムやトリガー等を、希望の目的 (OS の監視、単一のアプリケーションのモニタリング等) に対する動作の完全なセットを完成させるまでテンプレートに追加する事が出来ます。

テンプレートをホストにリンクさせる

準備が出来たテンプレートがあれば、後はホストに追加するだけです。設定 → ホストに移動し、新規ホストをクリックしてプロパティフォームを開き、テンプレートのフィールドがある事を確認します。

テンプレートのフィールドに新規テンプレートを入力します。作成したテンプレートの名前がドロップダウンリストに表示されているはずなので、スクロールして選択します。テンプレートフィールドに表示されている事を確認します。

フォーム内にある更新をクリックし、変更を保存します。これで全ての動作を保有するテンプレートがホストに追加されます。

想像の通り、この手順で他のどのホストにも適用する事が出来ます。アイテムやトリガー等の動作の変更は、テンプレートがリンクされているホストに伝搬されます。

事前に定義されたテンプレートをホストにリンクする方法

お気づきの様に、Zabbix には様々な OS、デバイス、アプリケーションに対応した定義済みのテンプレートが準備されています。すぐにモニタリングを開始する場合、これらの用意されたうち適切なテンプレートをホストにリンクする事が出来ますが、これらのテンプレートは状況に応じて微調整の必要がある事に気を付けてください。チェックのうちいくつかは不要かもしれませんし、機能状態確認の間隔が頻発すぎるかもしれません。

テンプレート に関する更なる情報はこちら。

6. Zabbix アプライアンス

概要 手動でセットアップしたり、Zabbix の既存のサーバーを再利用したりする代わりに、ユーザーは Zabbix アプライアンスまたは Zabbix アプライアンスのインストール CD イメージをダウンロードできます。

Zabbix アプライアンスとインストール CD のバージョンは AlmaLinux 8 (x86_64) が基となっています。

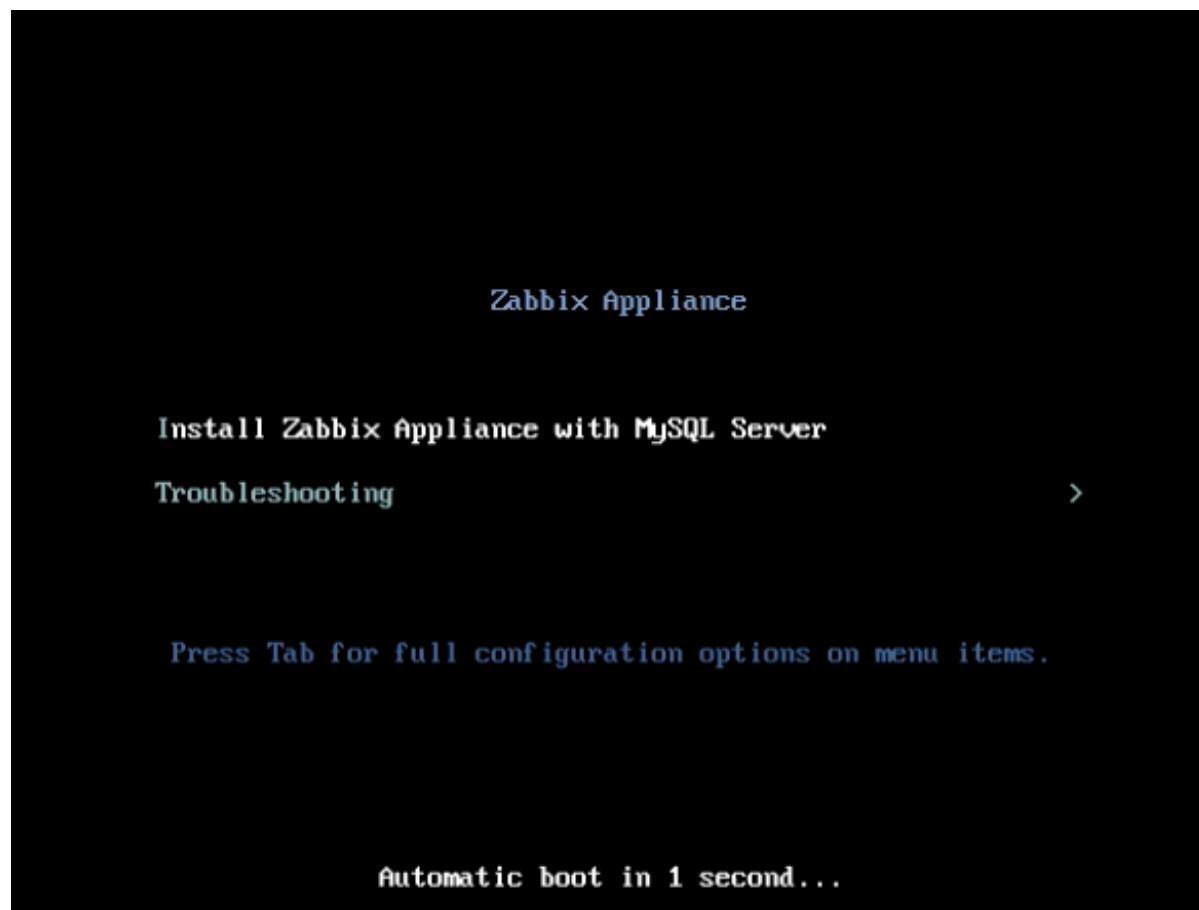
Zabbix アプライアンスのインストール CD は、Zabbix サーバー (MySQL) の即時展開に使用できます。

このアプライアンスを使用して、Zabbix を評価できます。アプライアンスは、本番環境での使用を目的としたものではありません。

システム要件：

- RAM : 1.5 GB
- ディスク容量: 仮想マシンに少なくとも 8GB を割り当てる必要があります

Zabbix インストール CD/DVD のブートメニュー：



Zabbix アプライアンスには、(MySQL で構成および実行されている) Zabbix サーバーと Web インターフェースが含まれています。

Zabbix 仮想アプライアンスは次の形式で利用できます。

- VMWare (.vmx)
- オープン仮想化フォーマット (.ovf)
- Microsoft Hyper-V 2012 (.vhdx)
- Microsoft Hyper-V 2008 (.vhd)
- KVM、Parallels、QEMU、USB スティック、VirtualBox、Xen (.raw)
- KVM、QEMU (.qcow2)

開始するには、アプライアンスを起動し、アプライアンスが DHCP で設定された IP にブラウザで接続します。

ホストで DHCP を有効にする必要があります。

仮想マシン内から IP アドレスを取得するには、次のコマンドを実行します。

```
ip addr show
```

Zabbix の Web インターフェースにアクセスするには、**http://<host_ip>** にアクセスします (ホストのブラウザからのアクセスについては、VM ネットワーク設定でブリッジモードを有効にする必要があります)。

アプライアンスが Hyper-V で起動しない場合は、Ctrl + Alt+F2 を押して tty セッションを切り替えることができます。

1 AlmaLinux 8 の設定変更 アプライアンスは AlmaLinux 8 が基となっています。AlmaLinux 8 の基本的な設定からいくつか変更されています。

1.1 リポジトリ

公式 Zabbix[リポジトリ](/manual/installation/install_from_packages/rhel) が/etc/yum.repos.d に追加されています。

```
[zabbix]
name=Zabbix Official Repository - $basearch
baseurl=http://repo.zabbix.com/zabbix/6.0/rhel/8/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/// etc / pki / rpm-gpg / RPM-GPG-KEY-ZABBIX-A14FE591
```

1.2 ファイアウォールの構成

アプライアンスは、事前定義された iptables のファイアウォールルールを使用します。

- SSH (22 TCP) ポートを開放;
- Zabbix エージェント (10050 TCP) と Zabbix トラッパー (10051 TCP) ポートを開放;
- HTTP (80 TCP) および HTTPS (443 TCP) ポートを開放;
- SNMP trap (162 UDP) ポートを開放;
- NTP (53 UDP) ポートの発信接続を開放;
- 1 秒あたり 5 パケットに制限された ICMP パケット;
- 他のすべての着信接続はドロップ;

1.3 静的 IP アドレスを使用する

デフォルトでは、アプライアンスは DHCP を使用して IP アドレスを取得します。静的 IP アドレスを指定するには：

- root ユーザーとしてログインします。
- /etc/sysconfig/network-scripts/ifcfg-eth0 ファイルを開きます。
- BOOTPROTO=dhcp を BOOTPROTO=none に書き換えます。
- 次の行を追加します。
 - IPADDR=< アプライアンスの IP アドレス >
 - PREFIX=<CIDR プレフィックス >
 - GATEWAY=< ゲートウェイの IP アドレス >
 - DNS1=<DNS サーバーの IP アドレス >
- **systemctl restartnetwork** コマンドを実行します。

必要に応じて RedHat の公式[ドキュメント](#)を参照してください。

1.4 タイムゾーンの変更

デフォルトではアプライアンスはシステムクロックに UTC を使用します。タイムゾーンを変更するには適切なファイルを /usr/share/zoneinfo から/etc/localtime にコピーします。例：

```
cp /usr/share/zoneinfo /Europe/Riga/etc/localtime
```

2 Zabbix の構成 Zabbix アプライアンスのセットアップには次のパスワードと構成の変更があります。

2.1 ユーザー認証情報 (ログイン: パスワード)

システム：

- root:zabbix

Zabbix フロントエンド：

- Admin:zabbix

データベース：

- root:< ランダム >
- zabbix:< ランダム >

Note:

データベースパスワードは、インストールプロセス中にランダムに生成されます。ルートパスワードは/root/.my.cnf ファイル内に保存されます。「root」アカウントでパスワードを入力する必要はありません。

データベースユーザーのパスワードを変更するには、次のファイル等で変更を行う必要があります。

- MySQL;

- /etc/zabbix/zabbix_server.conf;
- /etc/zabbix/web/zabbix.conf.php;

Note:

サーバーとフロントエンドには、それぞれ別々のユーザー zabbix_srv と zabbix_web が定義されています。

2.2 ファイルロケーション

- コンフィグファイルは/etc/zabbix にあります。
- Zabbix サーバー、プロキシ、エージェントのログファイルは/ var / log /zabbix にあります。
- Zabbix フロントエンドは/usr/share/zabbix にあります。
- ユーザー **zabbix** のホームディレクトリは/var/lib/zabbix です。

2.3 Zabbix の設定変更

- フロントエンドのタイムゾーンはヨーロッパのリガに設定されています。(これは/etc/php-fpm.d/zabbix.conf で変更できます)

3 フロントエンドのアクセス デフォルトでは、フロントエンドへのアクセスはどこからでも許可されています。

フロントエンドには http://<host> からアクセスできます。

これは/etc/nginx/conf.d/zabbix.conf でカスタマイズできます。このファイルを変更した後、Nginx を再起動する必要があります。SSH を使用してログインし、root ユーザーとして以下を実行します。

```
systemctl restart nginx
```

4 ファイアウォール デフォルトでは上記の構成変更 にリストされているポートのみが開いています。追加のポートを開くには"/etc/sysconfig/iptables" ファイルを変更し、ファイアウォールルールを再読み込みします。

```
systemctl reload iptables
```

5 アップグレード Zabbix アプライアンスパッケージはアップグレードされる場合があります。これを行うには、次を実行します。

```
dnf update zabbix\*
```

6 システムサービス Systemd サービスが利用可能です :

```
systemctl list-units zabbix\*
```

7 フォーマット固有のメモ 7.1 VMware

vmdk 形式のイメージは、VMware Player、Server、および Workstation 製品で直接使用できます。ESX、ESXi、および vSphere で使用するには、[VMware コンバーター](#)を使用して変換する必要があります。

VMware コンバーターを使用する場合、ハイブリッドネットワークアダプタで問題が発生する可能性があります。その場合は、変換プロセス中に E1000 アダプターを指定してみてください。変換の完了後、既存のアダプターを削除して E1000 アダプターを追加することもできます。

7.2 HDD/フラッシュイメージ (raw)

```
dd if=./zabbix_appliance_5.2.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

/dev/sdc をフラッシュ/HDD ディスクデバイスに置き換えます。

7. 設定

設定セクションのコンテンツにアクセスするには、サイドバーをご利用ください。

1 テンプレートの設定

概要

テンプレートを設定するには、最初に一般的なパラメーターを定義してテンプレートを作成し、次にエンティティ (アイテム、トリガー、グラフなど) をテンプレートに追加する必要があります。

テンプレートの作成

次の手順でテンプレートを作成します。

- 設定 → テンプレートに移動
- テンプレートの作成をクリック
- テンプレートの属性を編集

テンプレートタブには、一般的なテンプレートの属性が含まれています。

The screenshot shows a configuration form for a template. At the top, there are tabs: 'Template', 'Tags 1', 'Macros 9', and 'Value mapping'. The 'Template' tab is selected. The form contains the following fields:

- * Template name:** A text input field containing 'Elasticsearch'.
- Visible name:** A text input field containing 'Elasticsearch'.
- Templates:** A text input field with the placeholder 'type here to search' and a 'Select' button to the right.
- * Groups:** A text input field with 'Templates/Applications' selected and a 'Select' button to the right.
- Description:** A large empty text area.

At the bottom of the form, there are two buttons: 'Add' (in blue) and 'Cancel' (in white with a blue border).

必須入力フィールドには赤いアスタリスクが付いています。

テンプレート属性:

パラメータ	説明
テンプレート名	一意なテンプレート名。英数字、スペース、ドット、ダッシュ、アンダースコアを使用可能。先頭と末尾のスペースは使用不可。
表示名	リストやマップなどでの表示名
テンプレート	1つ以上の”ネストされた”テンプレートをこのテンプレートにリンクします。すべてのエンティティ(アイテム、トリガー、グラフ等)は、リンクされたテンプレートから継承されます。新しいテンプレートをリンクするにはテンプレートフィールドにテンプレート名を入力し始めると、一致するテンプレートのリストが表示されるのでリストから選択します。またはフィールドの横にある【選択】をクリックして、ポップアップウィンドウのリストからテンプレートを選択することもできます。テンプレートフィールドで選択されたテンプレートは、テンプレート設定フォームが保存または更新されたときにテンプレートにリンクされます。テンプレートのリンクを解除するには、リンクされたテンプレートブロックの2つのオプションのいずれかを使用します。 リンクを削除 - テンプレートのリンクを解除しますが、そのアイテム、トリガー、グラフは保持します リンクと保存データを削除 - リンクを解除しますテンプレートを作成し、そのアイテム、トリガー、およびグラフをすべて削除します
グループ	テンプレートが属するホスト/テンプレートグループ
説明	テンプレートの説明

タグタブでは、テンプレートレベルのタグを定義できます。このテンプレートにリンクされたホストのすべての障害は、ここに入力された値でタグ付けされます。

Template Tags 1 Macros 9 Value mapping

Name	Value
App	MySQL
tag	value

[Add](#)

ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT} および {HOST.ID} マクロはタグでサポートされています。

Macros タブでは、テンプレートレベルのユーザーマクロを名前と値のペアとして定義できます。マクロ値は、プレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できます。説明の追加もサポートされています。

Template Tags 1 Macros 9 Value mapping

Template macros Inherited and template macros

Macro	Value		Description
{\$TEMPLATE_THRESHOLD1}	10M	T	description
{\$TEMPLATE_THRESHOLD2}	20M	T	description
{\$TEMPLATE_THRESHOLD3}	30M	T	description
{\$TEMPLATE_THRESHOLD4}	40M	T	description
{\$TEMPLATE_THRESHOLD5}	50M	T	description

継承されたマクロとテンプレートマクロオプションを選択すると、リンクされたテンプレートのマクロとグローバルマクロが表示されます。ここには、テンプレートに対して定義されたすべてのユーザーマクロが、それらが解決する値と元の値とともに表示されます。

Template Tags 1 Macros 9 Value mapping

Template macros Inherited and template macros

Macro	Effective value	
{\$AGENT.TIMEOUT}	3m	T
Timeout after which agent is considered unavailable. Works only for agents reachable from Zabbix server/proxy (passive mode).		
{\$CPU.UTIL.CRIT}	90	T
description		
{\$IFERRORS.WARN}	2	T
description		
{\$IFCONTROL}	1	T

便宜上、それぞれのテンプレートとグローバルマクロ設定へのリンクが提供されていますが、テンプレートレベルでネストされたテンプレート/グローバルマクロを編集して、テンプレート上にマクロのコピーを作成することもできます。

値のマッピングタブでは、[値のマッピング/manual/config/items/mapping](#)) でアイテムデータのわかりやすい表現を設定できます。

ボタン:

Add

テンプレートを追加します。追加されたテンプレートがリストに表示されます。

Update

既存のテンプレートのプロパティを更新します

Clone	リンクされたテンプレートから継承されたエンティティ (アイテム、トリガーな祖) を含む、現在のテンプレートのプロパティに基づいて別のテンプレートを作成します。
Full clone	リンクされたテンプレートから継承され、現在のテンプレートに直接添付されたエンティティ (アイテム、トリガー等) を含む、現在のテンプレートのプロパティに基づいて別のテンプレートを作成します。
Delete	テンプレートを削除します。テンプレートのエンティティ (アイテム、トリガー等) は、リンクされたホストに残ります。
Delete and clear	リンクされたホストからテンプレートとそのすべてのエンティティを削除します。
Cancel	テンプレートプロパティの編集をキャンセルします。

テンプレートを作成したら、それにいくつかのエンティティを追加します。

Attention:

最初にアイテムをテンプレートに追加する必要があります。トリガーとグラフは、対応するアイテムがないと追加できません。

アイテム、トリガー、グラフの追加

次の手順でテンプレートに項目を追加します。

- 設定 → ホスト (または テンプレート) に移動
- 追加したいホスト/テンプレートの行にある アイテムをクリック
- テンプレートに追加したい項目のチェックボックスをマーク
- アイテムリストの下にあるコピーをクリック
- アイテムのコピー先のテンプレート (またはテンプレートのグループ) を選択してコピーをクリック

選択したすべてのアイテムをテンプレートにコピーする必要があります。

トリガーとグラフの追加は、先に必要な項目が追加されている場合にのみ追加できることに注意してください。

ダッシュボードの追加

ダッシュボードをテンプレートに追加するには、設定 → テンプレートで次の手順を実行します。

- テンプレートの行にある ダッシュボードをクリック
- **ダッシュボードの設定**のガイドラインに従ってダッシュボードを設定

Attention:

テンプレートダッシュボードに含めることができるウィジェットは、クラシックグラフ、グラフプロトタイプ、時計、プレーンテキスト、URL です

Note:

テンプレートダッシュボードから作成されたホストダッシュボードへのアクセスの詳細については、**host dashboard**セクションを参照してください

ローレベルディスカバリの設定

マニュアルの**ローレベルディスカバリ**セクションを参照

Web シナリオの追加

Web シナリオをテンプレートに追加するには、**設定** → **テンプレート**で次の手順を実行します。

- テンプレートの行で **Web** をクリック
- **configuring web scenarios**に従って、Web シナリオを設定します。

2 リンク作成/リンク削除

概要

リンクは、テンプレートをホストに適用するプロセスです。リンクを解除すると、テンプレートとの関連付けがホストから削除されます。

Attention:

テンプレートは、ホストグループではなく、個々のホストに直接リンクされます。テンプレートをホストグループに追加するだけではリンクされません。ホストグループは、ホストとテンプレートの論理的なグループ化にのみ使用されます。

テンプレートのリンク

テンプレートをホストにリンクするには、次の手順を実行します。

- 設定 → ホストに移動
- 必要なホストをクリックします
- テンプレートフィールドにテンプレート名の入力を開始します。一致するテンプレートのリストが表示されますのでスクロールして選択
- またはフィールドの横にある選択をクリックして、ポップアップウィンドウのリストから 1 つまたは複数のテンプレートを選択
- ホスト属性フォームで追加または更新をクリックします

ホストには、テンプレートのすべてのエンティティ (アイテム、トリガー、グラフなど) が含まれます。

Attention:

複数のテンプレートを同じホストにリンクすると、それらのテンプレートに同じ項目キーを持つ項目がある場合、失敗します。また、トリガーとグラフはアイテムを使用するため、同じアイテムキーを使用する場合、複数のテンプレートから単一のホストにリンクすることもできません。

エンティティ (アイテム、トリガー、グラフなど) がテンプレートから追加される場合:

- ホスト上に以前存在していた同一のエンティティは、テンプレートのエンティティとして更新されます
- テンプレートからのエンティティが追加されます
- テンプレートのリンクの前に、ホスト上にのみ存在していた、直接リンクされたエンティティは変更されません。

リストでは、テンプレートのすべてのエンティティにテンプレート名がプレフィックスとして付けられ、これらが特定のテンプレートに属していることを示します。テンプレート名自体 (灰色のテキスト) は、テンプレートレベルでこれらのエンティティのリストにアクセスできるリンクです。

一部のエンティティ (アイテム、トリガー、グラフなど) にテンプレート名のプレフィックスが付いていない場合は、それが以前にホストに存在し、テンプレートによって追加されなかったことを意味します。

エンティティの一意性の基準

テンプレートからエンティティ (アイテム、トリガー、グラフなど) を追加する場合、それらのエンティティのどれがホストに既に存在し、更新する必要があるか、およびどのエンティティが異なるかを知ることが重要です。同一性/相違点を決定するための一意性の基準は次のとおりです。

- アイテムの場合 - アイテムキー
- トリガーの場合 - トリガー名と式
- カスタムグラフの場合 - グラフ名とその項目

テンプレートを複数のホストにリンクする

複数ホストのテンプレートリンクを更新するには、設定 → ホストでチェックボックスをオンにしていくつかのホストを選択し、リストの下にある ****一括更新**** をクリックして、テンプレートをリンクを選択します。

Mass update

Host IPMI Tags Macros Inventory Encryption Value mapping

Link templates

Link

Replace

Unlink

type here to search

Clear when unlinking

追加のテンプレートをリンクするには、一致するテンプレートを提供するドロップダウンが表示されるまで、オートコンプリートフィールドにテンプレート名を入力し、スクロールしてリンクするテンプレートを選択します。

置換オプションを使用すると、以前にホストにリンクされていたテンプレートのリンクを解除しながら、新しいテンプレートをリンクできます。リンクを削除オプションを使用すると、リンクを解除するテンプレートを指定できます。リンク解除時にクリアオプションを使用すると、以前にリンクされたテンプレートのリンクを解除できるだけでなく、それらから継承されたすべての要素（アイテム、トリガーなど）を削除することもできます。

Note:

Zabbix には、定義済みのテンプレートセットがかなり多く用意されています。これらは参照用に使用できますが、アイテムが多すぎたりデータのポーリングが頻繁に行われたりする可能性があるため、本番環境ではそのまま使用せず、実際のニーズに合わせて微調整してください。

リンクされたエンティティの編集

テンプレートからリンクされたアイテムまたはトリガーを編集しようとする、多くの主要なオプションが編集できなくなっています。テンプレートの考え方としては、テンプレートレベルでワンタッチで編集するというものであるため、これは理にかなっています。ただし、個々のホストでアイテムを有効/無効にしたり、更新間隔、履歴の長さ、その他のパラメーターを設定したりすることはできます。

エンティティを完全に編集する場合は、テンプレートレベルで編集する必要があります（テンプレートレベルのショートカットがフォーム名に表示されます）。これらの変更は、このテンプレートがリンクされているすべてのホストに影響することに注意してください。

テンプレートのリンク削除

テンプレートをホストからリンク削除するには、次の手順を実行します。

- 設定 → ホストに移動
- 必要なホストをクリックしてテンプレートフィールドを見つけます
- テンプレートの横にある リンク削除または リンクと保存データを削除をクリックして、リンクを解除します。
- ホスト属性フォームで更新をクリック

リンクを削除オプションを選択すると、テンプレートとの関連付けが削除され、そのすべてのエンティティ（アイテム、トリガー、グラフなど）はホストに残ります。

リンクと保存データを削除オプションを選択すると、テンプレートとそのすべてのエンティティ（アイテム、トリガー、グラフなど）との関連付けが削除されます。

3 ネストする

概要

ネストは、1つのテンプレートが1つ以上の他のテンプレートを包含する方法です。

さまざまなサービス、アプリケーションなどの個々のテンプレートでエンティティを分離することは理にかなっていますが、ホストにリンクする必要のあるテンプレートが非常に多くなる可能性があります。構造を単純化するために、いくつかのテンプレートを1つのテンプレートにリンクすることができます。

ネストの利点は、1つのテンプレート（“ネスト元”、親テンプレート）のみをホストにリンクする必要があり、ホストはリンクされたテンプレート（“ネストされた”、子テンプレート）のすべてのエンティティを自動的に継承することです。たとえば、テンプレート

T1 と T2 をテンプレート T3 にリンクする場合、T3 に T1 と T2 のエンティティを追加しますが、その逆は行いません。テンプレート A をテンプレート B および C にリンクすると、B および C に A のエンティティが追加されます。

ネストされたテンプレートの構成

テンプレートをリンクするには、既存のテンプレートまたは新しいテンプレートを取得する必要があります。次に:

- **テンプレート設定フォーム**を開く
- テンプレートフィールドを見つける
- 選択をクリックして、ポップアップウィンドウでテンプレートを選択
- ポップアップウィンドウで必要なテンプレートを選択し、選択をクリックしてテンプレートをリストに追加します。
- テンプレート設定フォームで追加または更新をクリックします。

テンプレート自体のすべてのエンティティおよびすべてのリンクされたテンプレートのエンティティ (アイテム、トリガー、グラフなど) は、ホストによって継承されるリンクされたテンプレートダッシュボードを除いて、テンプレート設定に表示されます。

リンクされたテンプレートのリンクを解除するには、同じフォームでリンクを削除またはリンクと保存データを削除ボタンを使用し、更新をクリックします。

リンクを削除オプションを選択すると、リンクされたテンプレートとの関連付けが削除されますが、すべてのエンティティ (アイテム、トリガー、グラフなど) は削除されません。

リンクと保存データを削除オプションを選択すると、リンクされたテンプレートとそのすべてのエンティティ (アイテム、トリガー、グラフなど) との関連付けが削除されます。

4 一括更新

概要

いくつかのテンプレートの属性を一度に変更したい場合があります。個々のテンプレートを開いて編集する代わりに、一括更新機能を使用できます。

一括更新手順

次の手順で一部のテンプレートを一括更新します。

- **テンプレート一覧**で更新したいテンプレートの前にあるチェックボックスにチェックを入れる
- 一括更新をクリック
- 更新が必要な属性 (テンプレート, タグ, マクロまたは値のマッピング) を含むタブに移動します。
- 更新する属性のチェックボックスをマークして、新しい値を入力します

Mass update

Template Tags Macros Value mapping

Link templates Link Replace Unlink

type here to search Select

Clear when unlinking

Host groups Add Replace Remove

type here to search Select

Description Original

Update Cancel

テンプレートリンケージ更新の各ボタンを選択すると、次のオプションが利用可能になります。

- リンク - リンクする追加のテンプレートを指定
- 置換 - テンプレートにリンクされていたリンクを解除しながら、新しくリンクするテンプレートを指定
- リンクを削除 - リンクを解除するテンプレートを指定

リンク/リンク解除するテンプレートを指定するには、一致するテンプレートを提供するドロップダウンが表示されるまで、オートコンプリートフィールドにテンプレート名を入力し始めます。下にスクロールして、必要なテンプレートを選択します。

リンク削除時にクリアオプションを使用すると、以前にリンクされたテンプレートのリンクを解除できるだけでなく、それらから継承されたすべての要素 (アイテム、トリガーなど) を削除することもできます。

ホストグループのチェックボックスを選択すると、次のオプションが利用可能になります。

- 追加 - 既存のホストグループから追加のホストグループを指定したり、テンプレートに完全に新しいホストグループを入力したりできます
- 置換 - 既存のホストグループからテンプレートを削除し、このフィールドで指定されたもの (既存または新しいホストグループ) に置き換え
- 削除 - テンプレートから特定のホストグループを削除

これらのフィールドはオートコンプリートです。入力を開始すると、一致するホストグループのドロップダウンが表示されます。ホストグループが新しい場合は、ドロップダウンの文字列の後に (new) で示されます。下にスクロールして選択します。

Mass update

Template
Tags
Macros
Value mapping

Tags

Add
Replace
Remove

Name	Value
tag	value

[Add](#)

ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、および {HOST.ID} マクロはタグでサポートされています。名前が同じで値が異なるタグは重複とは見なされず、同じテンプレートに追加できることに注意してください。

Mass update

Template
Tags
Macros
Value mapping

Macros

Add
Update
Remove
Remove all

Macro	Value		Description
{SMACRO}	value	T v	description

[Add](#)

Update existing

マクロのチェックボックスを選択すると、次のオプションを持つボタンを使用できます。

- 追加 - テンプレートに追加のユーザーマクロを指定できます。既存の設定を上書きチェックボックスがチェックされている場合、指定されたマクロ名の値、タイプ、および説明が更新されます。オフにすると、その名前のマクロがテンプレートに既に存在する場合、更新されません。

- 更新 - このリストで指定されたマクロの値、タイプ、および説明を置き換えます。未設定時に追加チェックボックスがチェックされている場合、テンプレートに存在しなかったマクロは新しいマクロとして追加されます。オフにするとテンプレートに既に存在するマクロのみが更新されます。
- 削除 - 指定したマクロをテンプレートから削除します。選択したものを除くボックスがチェックされている場合、リストで選択されている以外のすべてのマクロが削除されます。チェックを外すと、リストで選択されたマクロのみが削除されます。
- すべて削除 - テンプレートからすべてのユーザーマクロを削除します。すべてのマクロを削除することを確認チェックボックスがチェックされていない場合、すべてのマクロの削除を確認するポップアップウィンドウが開きます。

Mass update

Template
Tags
Macros
Value mapping

Value mapping

Add
Update
Rename
Remove
Remove all

Name
Value

[Add](#)
[Add from](#)

Update existing

値のマッピングのチェックボックスを選択すると、次のオプションを持つボタンを使用できます。

- 追加 - テンプレートに値マップを追加します。既存の設定を上書にチェックすると、この名前の値マップのすべてのプロパティが更新されます。オフの場合、その名前の値マップが既に存在する場合、更新されません。
- 更新 - 既存の値マップを更新します。未設定時に追加をチェックすると、テンプレートに存在しなかった値マップは新しい値マップとして追加されます。オフにするとテンプレートに既に存在する値マップのみが更新されます。
- 名前変更 - 既存の値マップに新しい名前を付けます
- 削除 - 指定された値マップをテンプレートから削除します。選択したものを除くをチェックすると、指定されたもの以外のすべての値マップが削除されます。
- すべて削除 - テンプレートからすべての値マップを削除します。すべての値のマッピングを削除することを確認チェックボックスがマークされていない場合、削除の確認を求めるポップアップウィンドウが開きます。

必要な変更がすべて完了したら更新をクリックすると、選択したすべてのテンプレートの属性が更新されます。

1 ホストとホストグループ

「ホスト」とは？

Zabbix のホストは、監視したいデバイス（サーバ、ワークステーション、スイッチなど）のことです。

ホストの作成は Zabbix の初期設定の 1 つです。例えば、サーバ「x」のいくつかのパラメータを監視したい場合、はじめに「Server X」などの名前でホストを作成する必要があります。その後、そのホストに監視アイテムを追加できます。

ホストは、ホストグループにまとめられています。

引き続き [ホストの設定](#) を参照してください。

1 ホストの設定

概要

Zabbix の Web インターフェースでホストの設定を行うには、以下のようにします。

- 設定 → ホストまたは 監視データ → ホストに遷移します
- 右側の ホストの作成 (または、既に登録済みのホスト名) をクリックします。
- ホストのパラメータをフォームに入力します。

また、複製およびすべて複製ボタンは、既存のホストのフォームで使用できます。複製をクリックすると、すべてのホストのパラメータとテンプレートのリンクは保持されます。(これらのテンプレートからすべてのエンティティを保持します) すべて複製をクリックすると、さらに、直接接続されているエンティティ (アプリケーション、アイテム、トリガー、グラフ、ローレベルディスクカバリールール、Web シナリオ) を保持します。

注意: ホストをクローンした場合、テンプレート上のすべてのエンティティはそのまま保持されます。
 これらのエンティティに加えられたすべての変更 (アイテムの監視間隔の変更、正規表現の変更、ローレベルディスカバリルールへのプロトタイプ追加など) は、新しいホストには複製されず、テンプレート上にそのまま残ります。

構成

ホストタブには、一般的なホストの属性が表示されます。

Host

Host **IPMI** Tags Macros 2 Inventory ● Encryption Value mapping 1

* Host name

Visible name

Templates	Name	Action
	Linux OS agent	Unlink Unlink and clear
	App Zabbix Server	Unlink Unlink and clear

* Groups

Interfaces	Type	IP address	DNS name
	Agent	<input type="text" value="127.0.0.1"/>	<input type="text"/>
▼	SNMP	<input type="text" value="127.0.0.1"/>	<input type="text"/>

[Add](#)

Description

Monitored by proxy

Enabled

すべての必須入力フィールドには赤いアスタリスクが表示されます。

パラメータ	説明
ホスト名	Enter a unique host name. Alphanumerics, spaces, dots, dashes and underscores are allowed. However, leading and trailing spaces are disallowed. Note: With Zabbix agent running on the host you are configuring, the agent configuration file parameter Hostname must have the same value as the host name entered here. The name in the parameter is needed in the processing of active checks .

パラメータ	説明
表示名	Enter a unique visible name for the host. If you set this name, it will be the one visible in lists, maps, etc instead of the technical host name. This attribute has UTF-8 support.
テンプレート	Link templates to the host. All entities (items, triggers, graphs, etc) will be inherited from the template. To link a new template, start typing the template name in the Link new templates field. A list of matching templates will appear; scroll down to select. Alternatively, you may click on Select next to the field and select templates from the list in a popup window. The templates that are selected in the Link new templates field will be linked to the host when the host configuration form is saved or updated. To unlink a template, use one of the two options in the Linked templates block: Unlink - unlink the template, but preserve its items, triggers and graphs Unlink and clear - unlink the template and remove all its items, triggers and graphs Listed template names are clickable links leading to the template configuration form.
ホストグループ	Select host groups the host belongs to. A host must belong to at least one host group. A new group can be created and linked to the host group by adding a non-existing group name.
インターフェース	Several host interface types are supported for a host: Agent, SNMP, JMX and IPMI. No interfaces are defined by default. To add a new interface, click on Add in the Interfaces block, select the interface type and enter IP/DNS, Connect to and Port info. Note: Interfaces that are used in any items cannot be removed and link Remove is grayed out for them. See Configuring SNMP monitoring for additional details on configuring an SNMP interface (v1, v2 and v3).
	IP アドレス Host IP address (optional).
	DNS 名 Host DNS name (optional).
	接続方法 Clicking the respective button will tell Zabbix server what to use to retrieve data from agents: IP - Connect to the host IP address (recommended) DNS - Connect to the host DNS name
	ポート標準 TCP/UDP port number. Default values are: 10050 for Zabbix agent, 161 for SNMP agent, 12345 for JMX and 623 for IPMI. Check the radio button to set the default interface.
説明	Enter the host description.
プロキシによる監視	The host can be monitored either by Zabbix server or one of Zabbix proxies: (no proxy) - host is monitored by Zabbix server Proxy name - host is monitored by Zabbix proxy "Proxy name"
有効	Mark the checkbox to make the host active, ready to be monitored. If unchecked, the host is not active, thus not monitored.

IPMI タブには、IPMI の管理属性が表示されます。

パラメーター	説明
認証アルゴリズム	Select the authentication algorithm.
特権レベル	Select the privilege level.
ユーザー名	User name for authentication. User macros may be used.

パラメーター	説明
パスワード	Password for authentication. User macros may be used.

タグタブでは、ホストレベルの**タグ**を定義します。このホストのすべての障害は、ここで入力された値でタグ付けされます。

Name	Value
Service	JIRA

[Add](#)

ユーザーマクロ、{INVENTORY.*}、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID}などのマクロは、タグでサポートされています。

マクロタブでは、ホストレベルの**ユーザーマクロ**を名前と値のペアで定義することができます。マクロの値は、プレーン・テキスト、シークレット・テキスト、または Vault シークレットとして保持することに注意してください。
 説明文を追加することもできます。

Macro	Value	D
{\$HOST_MACRO}	1	T
{\$SNMP_COMMUNITY}	public	T

[Add](#)

継承したマクロとホストマクロオプションを選択すると、テンプレートレベルおよびグローバルなユーザーマクロを表示することができます。ここには、ホスト用に定義されたすべてのユーザーマクロが、それらが解決する値と元の値とともに表示されます。

Macro	Effective value	Templa
{\$AGENT.TIMEOUT}	3m	T Change ← Templa
Timeout after which agent is considered unavailable. Works only for agents reachable from Zabbix server/proxy (passive mode).		
{\$CPU_UTIL_CRIT}	90	T Change ← Templa
description		
{\$HOST_MACRO}	1	T Remove

便宜上、各テンプレートとグローバルマクロの設定へのリンクを設けています。また、ホストレベルでテンプレート/グローバルマクロを編集し、ホスト上にマクロのコピーを効果的に作成することも可能です。

インベントリタブでは、ホストの**インベントリ**情報を手動で入力することができます。また、このホストの自動でのインベントリ作成を有効にするか、インベントリ作成を無効にするかを選択できます。

Disabled Manual **Automatic**

Type Zabbix server

Type (Full details)

インベントリが有効な場合（手動または自動）、タブ名とともに緑色の点が表示されます。

暗号化

暗号化タブでは、ホストとの暗号化された接続を要求するようにできます。

パラメータ	説明
ホストへの接続	How Zabbix server or proxy connects to Zabbix agent on a host: no encryption (default), using PSK (pre-shared key) or certificate.
ホストからの接続	Select what type of connections are allowed from the host (i.e. from Zabbix agent and Zabbix sender). Several connection types can be selected at the same time (useful for testing and switching to other connection type). Default is "No encryption".
発行者	Allowed issuer of certificate. Certificate is first validated with CA (certificate authority). If it is valid, signed by the CA, then the Issuer field can be used to further restrict allowed CA. This field is intended to be used if your Zabbix installation uses certificates from multiple CAs. If this field is empty then any CA is accepted.
サブジェクト	Allowed subject of certificate. Certificate is first validated with CA. If it is valid, signed by the CA, then the Subject field can be used to allow only one value of Subject string. If this field is empty then any valid certificate signed by the configured CA is accepted.
PSK アイデンティティ	Pre-shared key identity string. Do not put sensitive information in the PSK identity, it is transmitted unencrypted over the network to inform a receiver which PSK to use.
PSK	Pre-shared key (hex-string). Maximum length: 512 hex-digits (256-byte PSK) if Zabbix uses GnuTLS or OpenSSL library, 64 hex-digits (32-byte PSK) if Zabbix uses mbed TLS (PolarSSL) library. Example: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

値のマッピング

値のマッピングタブでは、値のマッピングを使用してアイテムの値を分かりやすい表現で表示できるよう設定できます。

ホストグループの作成

Attention:

特権管理者だけが、ホストグループを作成可能です。

Zabbix の Web インターフェースでホストグループを作成する場合は、以下の手順で行います：

- 設定 → ホストグループへ遷移します
- 画面右上隅のホストグループの作成をクリックします。
- ホストグループのパラメータをフォームに入力する

≡ Host groups

* Group name Europe/Latvia/Riga/Zabbix servers

必須入力項目には、赤いアスタリスクが表示されています。

パラメーター	説明
グループ名	固有のホストグループ名を入力します。

階層的なホストグループを作成するには、Europe/Latvia/Riga/Zabbix servers のようにスラッシュ区切り記号「/」を使用します。
3つの親ホストグループ (Europe/Latvia/Riga) が存在しない場合でも、このグループを作成することができます。
この場合、これらの親ホストグループの作成はユーザが行うことができ、自動的に作成されることはありません。
スラッシュの先頭と末尾、および複数のスラッシュの連続は許可されません。また、「/」のエスケープにも対応していません。
Zabbix 3.2.0 からホストグループの階層表現に対応しました。||すべてのサブグループに権限とタグフィルターを適用|このチェックボックスは、特権管理者のみが利用でき、既存のホストグループを編集するときのみ利用できます。
このチェックボックスをマークして更新をクリックすると、同じレベルの権限/タグフィルターがすべての階層化されたホストグループに適用されます。階層化されたホストグループに異なる**権限**が割り当てられたユーザーグループについては、親ホストグループのアクセス権レベルが階層化されたグループに対して適用されます。
これは1回限りのオプションで、データベースには保存されません。
このオプションは Zabbix 3.4.0 からサポートされています。|

階層化されたホストグループへの権限

- 既存の親ホストグループに対して子ホストグループを作成する場合
 ユーザーグループのパーミッションは親から継承されます
 (例: Riga が既に存在する場合、Riga/Zabbix servers を作成する場合)

- 既存の子ホストグループに対して親ホストグループを作成する場合
 親に対するパーミッションは設定されません
 (例: Riga/Zabbix servers が既に存在する場合に Riga を作成する場合など)

2 インベントリ

概要

Zabbix でネットワークデバイスのインベントリを管理することができます。

Zabbix フロントエンドには特別な Inventory メニューがあります。
しかし、初期状態ではそこにデータは表示されず、データを入力する場所でもありません。
インベントリデータの作成は、ホストの設定時に手動で行います。
または、いくつかの自動配置オプションを使用することで自動的に行われます。

インベントリの構築

手動モード

configuring a host時に、Host inventory タブでデバイスの種類、シリアル番号、場所、責任者などの詳細を入力します。
インベントリ情報に入力されるデータです。

ホストのインベントリ情報に URL が含まれ、それが'http' または'https' で始まる場合
 Inventory セクションをクリック可能なリンクが表示されます。

自動モード

ホストインベントリは、自動的に入力することもできます。この機能を使用するには
 ホストを構成するときに、Host inventory タブのインベントリモードが Automatic に設定されている必要があります。

それから、**configure host items**で、それぞれの属性で宛先フィールド指定することで
 (Item will populate host inventory field と呼ばれる)、任意のホストインベントリフィールドに値を入力することができます。

インベントリデータの自動収集に特に有効なアイテムを収集することができます。

- system.hw.chassis[full|type|vendor|model|serial] - default is [full], root permissions needed
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] -default is [all,full]
- system.hw.devices[pci|usb] - default is [pci]
- system.hw.macaddr[interface,short|full] - default is [all,full], interface is regexp
- system.sw.arch
- system.sw.os[name|short|full] - default is [name]

- system.sw.packages[package,manager,short|full] - default is [all,all,full], package is regexp

インベントリモードの選択

インベントリモードは、ホスト設定フォームで選択することができます。

新しいホストのデフォルトのインベントリモードは、Default host inventory mode の Administration → General → Other の設定に基づいて選択されます。

ネットワーク検出または自動登録によって追加されたホストについては、以下のように定義することができます。Set host inventory mode で、手動または自動モードを選択することができます。この操作は、Default host inventory mode の設定を上書きします。

インベントリ概要

既存のすべてのインベントリデータの詳細は、Inventory メニューに表示されます。

Inventory → Overview では、インベントリの様々なフィールドによるホストのカウントを取得できます。

Inventory → Hosts では、インベントリ情報を持つすべてのホストを見ることができます。ホスト名をクリックすると、インベントリの詳細が表示されます。

≡ Host inventory

The screenshot shows the 'Overview' tab of the Zabbix host inventory page for a host named 'Zabbix server'. The page is divided into several sections:

- Host name:** Zabbix server
- Agent interfaces:** A table with columns for IP address (127.0.0.1), DNS name, Connect to (IP, DNS), and Port (10050).
- SNMP interfaces:** A table with columns for IP address (127.0.0.1), DNS name, Connect to (IP, DNS), and Port (161).
- OS:** Linux version 5.3.0-46-generic (buildd@lcy01-amd64-013) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)) #38~18.04.1-Ubuntu SMP
- Monitoring:** Links to Web, Latest data, Problems, Graphs, and Dashboards.
- Configuration:** Links to Host, Items (148), Triggers (67), Graphs (28), Discovery (4), and Web (1).
- A **Cancel** button is located at the bottom.

The **Overview** タブは以下を表示します。

パラメータ	説明
Host name	Name of the host. Clicking on the name opens a menu with the scripts defined for the host. Host name is displayed with an orange icon, if the host is in maintenance.
Visible name	Visible name of the host (if defined).
Host (Agent, SNMP, JMX, IPMI) interfaces	This block provides details of the interfaces configured for the host.
OS	Operating system inventory field of the host (if defined).
Hardware	Host hardware inventory field (if defined).
Software	Host software inventory field (if defined).
Description	Host description.
Monitoring	Links to monitoring sections with data for this host: Web, Latest data, Problems, Graphs, Dashboards.

パラメータ	説明
Configuration	Links to configuration sections for this host: Host, Applications, Items, Triggers, Graphs, Discovery, Web. The amount of configured entities is listed in parenthesis after each link.

Details タブには、入力されている (空ではない) すべてのインベントリ・フィールドが表示されます。

インベントリマクロ

通知用にホストインベントリマクロ {INVENTORY.} が用意されています。

例えば、”{INVENTORY.LOCATION1} のサーバーに問題が発生しました。
 {INVENTORY.CONTACT1}、電話番号 {INVENTORY.POC.PRIMARY.PHONE.A1}”

詳しくは、[supported macro](#) ページをご覧ください。

3 一括アップデート

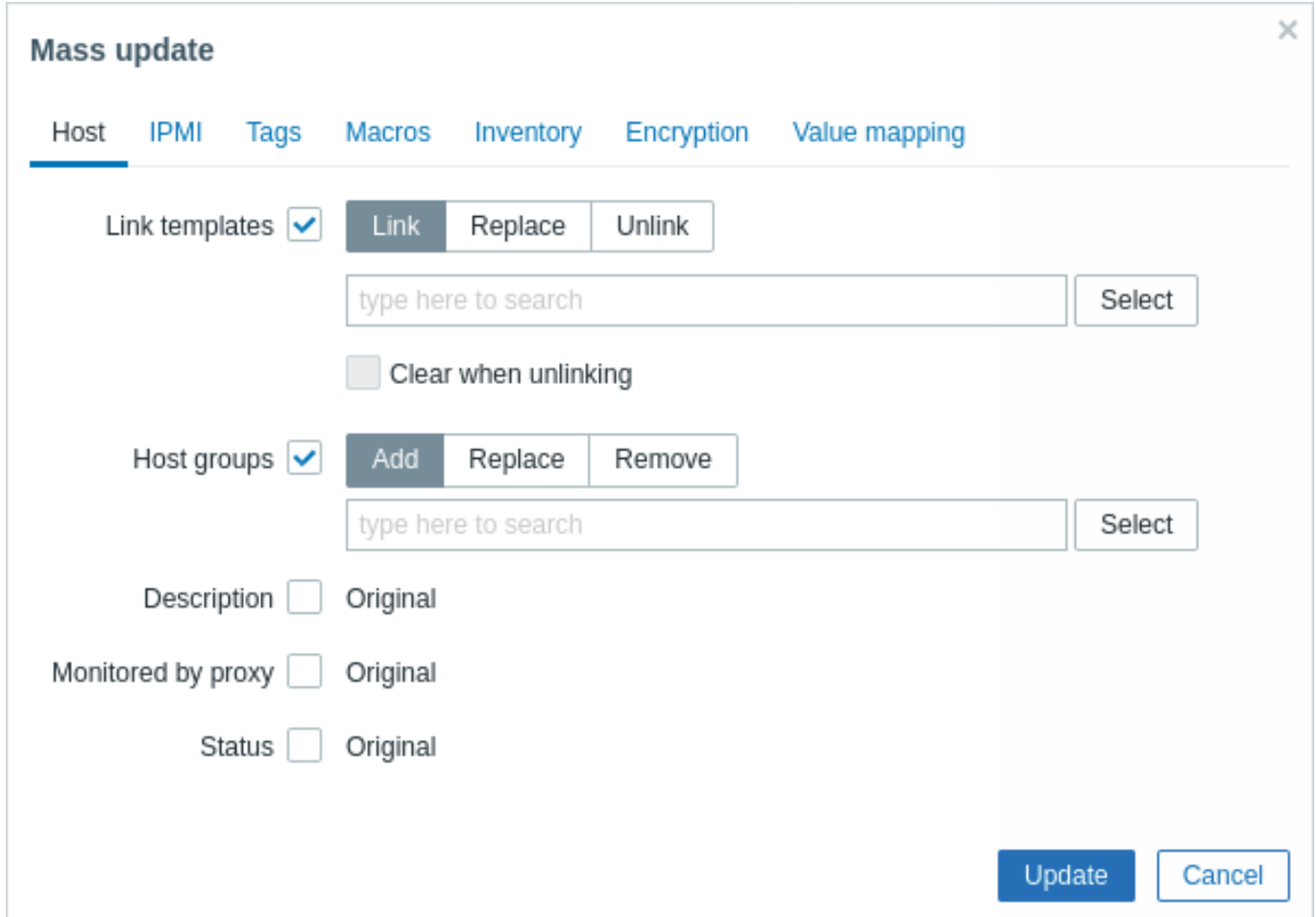
概要

一度に多くのホストの属性を変更したい場合があります。
 このような場合、個々のホストを開いて編集する代わりに一括アップデート機能を使います。

一括アップデート機能の使用

いくつかのホストを一括でアップデートするには、以下のようにします。

- **host list** の更新したいホストの前にあるチェックボックスをオンにします。
- リストの下にある **Mass update** をクリックします。
- 必要な属性のタブ (ホスト、IPMI、Tags、Macros、Inventory、Encryption、Value mapping) に移動します。
- 更新する属性のチェックボックスをマークし、新しい値を入力します。



各ボタンを選択すると、以下のオプションが利用できます。

The following options are available when selecting the respective button for **template** linkage update:

- Link - 追加でリンクするテンプレートを指定します。
- Replace - リンクするテンプレートを指定し、それ以前にホストとリンクしていたテンプレートはリンク解除します。
- Unlink - リンクを解除するテンプレートを指定します。

Link/Unlink するテンプレートを指定するには、テンプレート名をオートコンプリートフィールドに入力します。
 該当するテンプレートがドロップダウンで表示されるので、スクロールダウンして、必要なテンプレートを選択します。

Clear when unlinking オプションを使用すると、以前にリンクされたテンプレートを解除するだけでなく、
 テンプレートから継承されたすべての要素（アイテム、トリガーなど）も削除されます。

各ボタンを選択すると、**host group** の更新のための、次のオプションが利用できます。

- Add - 既存のホストグループから追加のホストグループを指定したり、全く新しいホストグループを入力することができます。
- Replace - 既存のホストグループからホストを削除し、このフィールドで指定されたもの（既存または新規のホストグループ）と置き換えます。
- Remove - 特定のホストグループをホストから削除します。

これらのフィールドはオートコンプリートで、入力を開始すると、一致するホストグループのドロップダウンが表示されます。
 ホストグループが新しい場合、それはドロップダウンに表示され、文字列の後に (new) が表示されます。
 選択するには下にスクロールするだけです。

Mass update

Host **IPMI** Tags Macros Inventory Encryption Value mapping

Authentication algorithm Original

Privilege level Operator

Username Original

Password Original

Mass update

Host IPMI **Tags** Macros Inventory Encryption Value mapping

Tags **Add** Replace Remove

Name	Value
<input type="text" value="tag"/>	<input type="text" value="value"/>

[Add](#)

ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、
 {HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロは、タグでサポートされています。
 同じ名前異なる値を持つタグは、“重複”とはみなされず、同じホストに追加されることに注意してください。

Mass update

Host IPMI Tags **Macros** Inventory Encryption Value mapping

Macros

Add

Update

Remove

Remove all

Macro

Value

Description

{\$MACRO}

value

T

description

[Add](#)

Update existing

各ボタンを選択すると、マクロ更新用に以下のオプションが利用できます。

- Add - ホストに対して追加のユーザマクロを指定することができます。 Update existing チェックボックスをオンにすると、指定したマクロの値、タイプ、および説明が更新されます。 チェックをはずすと、その名前のマクロがすでにホスト上に存在する場合は、更新されません。
- Update - このリストで指定されたマクロの値、型、説明を置き換えます。 Add missing チェックボックスをオンにすると、ホスト上に存在しなかったマクロを新しいマクロとして追加します。 チェックを外すと、既にホスト上に存在するマクロのみが更新されます。
- Remove - 指定されたマクロをホストから削除します。 Except selected にチェックを入れると、 リストで指定された以外のすべてのマクロが削除されます。 チェックを外すと、 リストで指定されたマクロだけが削除されます。
- Remove all - ホストからすべてのユーザマクロを削除します。 I confirm to remove all macros チェックボックスが チェックされていない場合、新しいポップアップウィンドウが開き、すべてのマクロの削除を確認します。

Mass update

Host IPMI Tags Macros **Inventory** Encryption Value mapping

Inventory mode

Disabled

Manual

Automatic

Type Original

Type (Full details) Original

Name Original

Alias Original

インベントリフィールドを一括更新するには、Inventory mode を '手動' または '自動' に設定する必要があります。

Mass update

Host IPMI Tags Macros Inventory Encryption Value mapping

Connections Connections to host No encryption PSK Certificate

Connections from host No encryption
 PSK
 Certificate

* PSK identity

* PSK

Mass update

Host IPMI Tags Macros Inventory Encryption Value mapping

Value mapping

Name	Value
Add Add from	
<input type="checkbox"/> Update existing	

バリュemapの更新には、以下のオプションを持つボタンが用意されています。

- Add - クリックすると、ホストにバリュemapを追加します。Update existing をマークした場合、
 この名前のバリュemapのすべてのプロパティが更新されます。
 それ以外の場合、その名前のバリュemapがすでに存在する場合は、更新されません。

- Update - 既存のバリュemapを更新します。Add missing をマークすると、ホスト上に以前は存在しなかった
 バリュemapが新しいバリュemapとして追加されます。
 それ以外の場合、ホスト上に既に存在するバリュemapのみが更新されます。

- Rename - 既存のバリュemapに新しい名前を付けます。

- Remove - 指定された値mapをホストから削除します。もし Except selected を選択した場合、
 指定したものを除くすべてのバリュemapが削除されます。

- Remove all - ホストからすべてのバリュemapを削除します。もし、I confirm to remove all value maps
 チェックボックスがマークされていない場合、新しいポップアップウィンドウが開き、削除を確認します。

必要な変更をすべて終えたら、Update をクリックします。選択したすべてのホストの属性は、それに応じて更新されます。

2 アイテム

概要

アイテムは、ホストからデータを収集するためのものです。

ホストを設定したら、実際のデータを取得するために、いくつかの監視項目を追加する必要があります。

1つのアイテムが1つのチェック項目です。多くの項目を素早く追加する方法の1つは、定義済みの1つのテンプレートをホストに紐づけすることです。しかし、システムのパフォーマンスを最適化するためには、テンプレートを微調整して、本当に必要な数のアイテムと間隔の監視を行うようにする必要があります。

単一のアイテムで、ホストから収集するデータの種類を指定します。

そのために、**アイテムキー**を使用します。具体的には、キー名が **system.cpu.load** というアイテムは、プロセッサの負荷のデータを収集しますが、キー名が **net.if.in** というアイテムは、受信トラフィックの情報を収集します。

さらにキーでパラメータを指定するには、キー名の後に角括弧で囲みます。system.cpu.load[avg5] は、過去 5 分間のプロセッサ負荷の平均を返し、net.if.in[eth0] は、インターフェース eth0 での受信トラフィックを表示します。

Note:

サポートされるすべてのアイテムタイプとアイテムキーについては、**アイテムタイプ**の各セクションを参照してください。

アイテムの作成 に進んでください。

1 アイテムの作成

概要

Zabbix の Web インターフェースでアイテムを作成するには、以下の手順で行います：

- 設定 → ホストを選択
- ホストの行の アイテムをクリック
- 画面右上の アイテムの作成をクリック
- フォーム内のアイテムのパラメータを入力

また、既存のアイテムを開き、複製ボタンを押し、別の名前で作成することもアイテムを作成することができます。

設定

アイテムタブには、一般的なアイテムの属性が表示されます。

Item Tags Preprocessing

* Name

Type

* Key

Type of information

* Host interface

Units

* Update interval

Custom intervals

Type	Interval	Period
<input checked="" type="checkbox"/> Flexible <input type="checkbox"/> Scheduling	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>

[Add](#)

* History storage period

* Trend storage period

Value mapping

Populates host inventory field



Description

Enabled

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
名前	アイテムの名前を設定します。
タイプ	アイテムのタイプです。それぞれのアイテムタイプセクションを参照してください。
キー	アイテムキーです (2048 文字まで)。 サポートされているアイテムキーは、それぞれのセクションに記載されています。 キーは、1つのホスト内で一意である必要があります。 キータイプが「Zabbix エージェント」、「Zabbix エージェント (アクティブ)」、「シンプルチェック」の場合、Zabbix エージェントか Zabbix サーバーがキーの値をサポートしている必要があります。 正しいキーのフォーマットも参照してください。

パラメータ	説明
データ型	<p>Type of data as stored in the database after performing conversions, if any.</p> <p>Numeric (unsigned) - 64bit unsigned integer</p> <p>Numeric (float) - 64bit floating point number</p> <p>This type will allow precision of approximately 15 digits and range from approximately -1.79E+308 to 1.79E+308 (with exception of PostgreSQL 11 and earlier versions).</p> <p>Receiving values in scientific notation is also supported. E.g. 1.23E+7, 1e308, 1.1E-4.</p> <p>Character - short text data</p> <p>Log - long text data with optional log related properties (timestamp, source, severity, logeventid)</p> <p>Text - long text data. See also text data limits.</p> <p>For item keys that return data only in one specific format, matching type of information is selected automatically.</p>
ホストインターフェース	<p>Select the host interface. This field is available when editing an item on the host level.</p>
単位	<p>If a unit symbol is set, Zabbix will add post processing to the received value and display it with the set unit postfix.</p> <p>By default, if the raw value exceeds 1000, it is divided by 1000 and displayed accordingly. For example, if you set bps and receive a value of 881764, it will be displayed as 881.76 Kbps.</p> <p>The JEDEC memory standard is used for processing B (byte), Bps (bytes per second) units, which are divided by 1024. Thus, if units are set to B or Bps Zabbix will display:</p> <p>1 as 1B/1Bps 1024 as 1KB/1KBps 1536 as 1.5KB/1.5KBps</p> <p>Special processing is used if the following time-related units are used:</p> <p>unixtime - translated to "yyyy.mm.dd hh:mm:ss". To translate correctly, the received value must be a Numeric (unsigned) type of information.</p> <p>uptime - translated to "hh:mm:ss" or "N days, hh:mm:ss" For example, if you receive the value as 881764 (seconds), it will be displayed as "10 days, 04:56:04"</p> <p>s - translated to "yyy mmm ddd hhh mmm sss ms"; parameter is treated as number of seconds. For example, if you receive the value as 881764 (seconds), it will be displayed as "10d 4h 56m"</p> <p>Only 3 upper major units are shown, like "1m 15d 5h" or "2h 4m 46s". If there are no days to display, only two levels are displayed - "1m 5h" (no minutes, seconds or milliseconds are shown). Will be translated to "< 1 ms" if the value is less than 0.001.</p> <p>Note that if a unit is prefixed with !, then no unit prefixes/processing is applied to item values. See unit conversion.</p>
監視間隔	<p>Retrieve a new value for this item every N seconds. Maximum allowed update interval is 86400 seconds (1 day).</p> <p>Time suffixes are supported, e.g. 30s, 1m, 2h, 1d.</p> <p>User macros are supported.</p> <p>A single macro has to fill the whole field. Multiple macros in a field or macros mixed with text are not supported.</p> <p>Note: The update interval can only be set to '0' if custom intervals exist with a non-zero value. If set to '0', and a custom interval (flexible or scheduled) exists with a non-zero value, the item will be polled during the custom interval duration.</p> <p>Note that the first item poll after the item became active or after update interval change might occur earlier than the configured value.</p> <p>An existing passive item can be polled for value immediately by pushing the Execute now button.</p>

パラメータ	説明
監視間隔のカスタマイズ	<p>You can create custom rules for checking the item:</p> <p>Flexible - create an exception to the Update interval (interval with different frequency)</p> <p>Scheduling - create a custom polling schedule. For detailed information see Custom intervals.</p> <p>Time suffixes are supported in the Interval field, e.g. 30s, 1m, 2h, 1d.</p> <p>User macros are supported.</p> <p>A single macro has to fill the whole field. Multiple macros in a field or macros mixed with text are not supported.</p> <p>Scheduling is supported since Zabbix 3.0.0.</p> <p>Note: Not available for Zabbix agent active items.</p>
履歴の保存期間	<p>Select either:</p> <p>Do not keep history - item history is not stored. Useful for master items if only dependent items need to keep history. This setting cannot be overridden by global housekeeper settings.</p> <p>Storage period - specify the duration of keeping detailed history in the database (1 hour to 25 years). Older data will be removed by the housekeeper. Stored in seconds.</p> <p>Time suffixes are supported, e.g. 2h, 1d. User macros are supported.</p> <p>The Storage period value can be overridden globally in Administration → General → Housekeeper.</p> <p>If a global overriding setting exists, a green  info icon is displayed. If you position your mouse on it, a warning message is displayed, e.g. Overridden by global housekeeper settings (1d). It is recommended to keep the recorded values for the smallest possible time to reduce the size of value history in the database. Instead of keeping a long history of values, you can keep longer data of trends.</p>
トレンドの保存期間	<p>See also History and trends.</p> <p>Select either:</p> <p>Do not keep trends - trends are not stored. This setting cannot be overridden by global housekeeper settings.</p> <p>Storage period - specify the duration of keeping aggregated (hourly min, max, avg, count) history in the database (1 day to 25 years). Older data will be removed by the housekeeper. Stored in seconds.</p> <p>Time suffixes are supported, e.g. 24h, 1d. User macros are supported.</p> <p>The Storage period value can be overridden globally in Administration → General → Housekeeper.</p> <p>If a global overriding setting exists, a green  info icon is displayed. If you position your mouse on it, a warning message is displayed, e.g. Overridden by global housekeeper settings (7d). Note: Keeping trends is not available for non-numeric data - character, log and text.</p>
値のマッピング	<p>See also History and trends.</p> <p>Apply value mapping to this item. Value mapping does not change received values, it is for displaying data only. It works with Numeric(unsigned), Numeric(float) and Character items. For example, "Windows service states".</p>

パラメータ	説明
ログの時間の形式	<p>Available for items of type Log only. Supported placeholders:</p> <ul style="list-style-type: none"> * y: Year (1970-2038) * M: Month (01-12) * d: Day (01-31) * h: Hour (00-23) * m: Minute (00-59) * s: Second (00-59) <p>If left blank the timestamp will not be parsed. For example, consider the following line from the Zabbix agent log file: " 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)." It begins with six character positions for PID, followed by date, time, and the rest of the line. Log time format for this line would be "pppppp:yyyyMMdd:hhmmss". Note that "p" and ":" chars are just placeholders and can be anything but "yMdhms".</p>
ホストインベントリフィールドの自動設定	<p>You can select a host inventory field that the value of item will populate. This will work if automatic inventory population is enabled for the host. This field is not available if Type of information is set to 'Log'. Enter an item description. Mark the checkbox to enable the item so it will be processed. Click on the link to view the latest data for the item. This link is only available when editing an already existing item.</p>
説明 有効 最新データ	

Note:

アイテムタイプ固有のフィールドは、[アイテムタイプのページ](#)に記述されています。

Note:

ホストレベルで既存の**テンプレート**レベルの項目を編集する場合、多くのフィールドが読み取り専用になります。フォームのヘッダーにあるリンクを使って、テンプレートレベルに移動し、そこで編集することができます。テンプレートレベルでの変更は、そのテンプレートがリンクされているすべてのホストに対するアイテムが変更されることに注意してください。

タグタブでは、アイテムレベルの**タグ**を定義することができます。

The screenshot shows the 'Item Tags' configuration page. At the top, there are tabs for 'Item', 'Tags 1', and 'Preprocessing'. The 'Item tags' tab is selected, and it shows a sub-tab for 'Inherited and item tags'. Below this, there is a table with two columns: 'Name' and 'Value'. The first row contains 'Application' in the 'Name' column and 'CPU' in the 'Value' column. Below the table, there is a blue 'Add' button with a plus sign.

アイテムの値の保存前処理

保存前処理タブでは、受け取った値に対して**変換するルール**を定義することができます。

テスト

ある項目をテストし、正しく設定されれば、実際の値を得ることができます。テストは、アイテムが保存される前でも行うことができます。

テストは、ホストとテンプレートのアイテム、アイテムのプロトタイプ、およびローレベルディスカバリールールで利用できます。アクティブなアイテムでは、テストは利用できません。

アイテムのテストは、以下のパッシブアイテムのタイプに対応しています。

- Zabbix エージェント
- SNMP エージェント (v1, v2, v3)
- IPMI エージェント
- SSH エージェント
- Telnet エージェント
- JMX エージェント
- シンプルチェック (icmping* や vmware.* のアイテムは除く)
- Zabbix インターナル
- 計算アイテム
- 外部チェック
- データベースモニタ
- HTTP エージェント
- スクリプト

アイテムをテストするには、アイテム設定画面の下部にある テストボタンをクリックします。テストボタンは、テストできない項目 (アクティブチェックや除外されたシンプルチェックなど) については、無効になります。

The screenshot shows a configuration form for an item. The 'Description' field contains the text 'Space utilization in % for /'. Below the description, there is an 'Enabled' checkbox which is checked. At the bottom of the form, there are three buttons: 'Add', 'Test', and 'Cancel'. A green arrow points to the 'Test' button, indicating it is the focus of the instruction.

アイテムテストフォームには、必要なホストパラメータ (ホストアドレス、ポート、プロキシ名/プロキシなし)、およびアイテム固有の詳細 (SNMPv2 コミュニティや SNMPv3 セキュリティ認証など) のフィールドがあります。これらのフィールドはコンテキストを認識します:

- 値は、可能な限り事前に入力されています。
 エージェントが必要な項目については、ホストで選択されたエージェントインターフェースから情報を取得します。

- テンプレートアイテムの値は、手動で入力する必要があります。

- プレーンテキストのマクロ値は展開されます。

- 値 (または値の一部) がシークレットまたは Vault マクロであるフィールドは空であり
 手動で入力する必要があります。アイテムパラメータにシークレットマクロ値が含まれている場合、
 "アイテムにシークレット値を持つユーザー定義マクロが含まれています。
 これらのマクロの値は手動で入力してください。" という警告メッセージが表示されます。

- フィールドは、アイテムの種類によって必要ない場合は無効になります。
 (たとえば、ホストアドレスのフィールドとプロキシのフィールドは、計算項目で無効になります)

アイテムをテストするには、値の取得をクリックします。値の取得に成功すると、値フィールドに入力され、現在の値 (存在する場合) が前の値フィールドに移動し、同時に前の時間フィールド、すなわち 2 つの値の時間差 (クリック数) を計算します。そして EOL シーケンスを検出し、"\n\r" を検出した場合、CRLF に置き換えようとしています。

Test item ✕

Get value from host

Host address Port

Proxy

Value Time

Previous value Prev. time

End of line sequence

設定が正しくない場合は、考えられる原因を説明するエラーメッセージが表示されます。

Test item

! Invalid second parameter.

Get value from host

Host address

Proxy

Value

ホストから正常に取得された値は、[保存前処理](#) のテストに使用することも可能です。

フォームのボタン

フォームの下部にあるボタンで、いくつかの操作を行うことができます。

<div style="border: 1px solid #007bff; background-color: #007bff; color: white; padding: 5px; margin-bottom: 10px; width: 60px; text-align: center;">Add</div> <div style="border: 1px solid #007bff; background-color: #007bff; color: white; padding: 5px; margin-bottom: 10px; width: 60px; text-align: center;">Update</div> <div style="border: 1px solid #007bff; padding: 5px; margin-bottom: 10px; width: 60px; text-align: center;">Clone</div> <div style="border: 1px solid #007bff; padding: 5px; width: 60px; text-align: center;">Execute now</div>	<p>アイテムの追加。このボタンは新規のアイテムの時のみ使用できます。</p> <p>アイテムのプロパティの更新。</p> <p>現在のアイテムプロパティをベースにして別のアイテムを作成。</p> <p>アイテムの値を取得。アイテムのタイプがパッシブなものだけ対応しています。(詳細参照) 注意値の取得時には設定キャッシュは更新されません。つまり直近のアイテムの設定変更が反映されません。</p>
---	--

Test	値を取得できるかのテスト実行。
Delete and clear	アイテムのヒストリとトレンドを削除。
Delete	アイテムの削除。
Cancel	アイテムのプロパティ変更をキャンセル。

テキストデータの制限

テキストデータの制限は、データベースのバックエンドに依存します。
 テキストデータをデータベースに格納する前に、データベースの定義と一致するように切り捨てられます。

DBMS	情報の種類		
	文字列	ログ	テキスト
MySQL	255 文字	65536 バイト	65536 バイト
PostgreSQL	255 文字	65536 文字	65536 文字
Oracle	255 文字	65536 文字	65536 文字

単位変換

デフォルトでは、項目の単位を指定すると、乗数の接頭辞が追加されます。
 例えば、入力値'2048'に単位'B'を指定した場合、'2KB'と表示されます。

単位を変換しないようにするには、!という接頭語を使用します。例えば!Bのように。
 感嘆符の有無にかかわらず、変換がどのように行われるかを理解するために、次の値と単位の例を参照してください。


```
1024 !B → 1024 B
1024 B → 1 KB
61 !s → 61 s
61 s → 1m 1s
0 !uptime → 0 uptime
0 uptime → 00:00:00
0 !! → 0 !
0 ! → 0
```

Note:

Zabbix4.0 以前は、ハードコードされたユニットストップリストがありました。
 ms,rpm,RPM,% から構成されていました。このストップリストは廃止されました。
 このような単位の変換を防ぐ正しい表記方法は ms,!rpm,!RPM,!%です。

カスタムスクリプトの制限

使用可能なカスタムスクリプトの長さは、使用するデータベースによって異なります。

DBMS	文字数の上限	バイト数の上限
MySQL	65535	65535
Oracle Database	2048	4000
PostgreSQL	65535	制限なし
SQLite (Zabbix プロキシ専用)	65535	制限なし

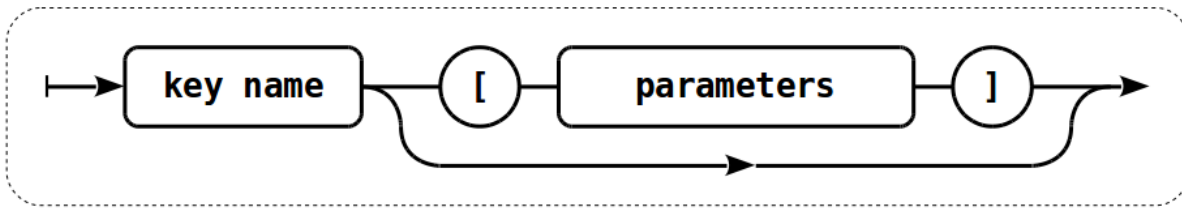
サポート対象外の項目

何らかの理由で値が取得できない項目は、サポート対象外になることがあります。このような項目は、標準的な **監視間隔**で再チェックされます。

サポート対象外のアイテムは、取得不可のステータスで報告されます。

1 アイテムキーのフォーマット

キーパラメータを含むアイテムキーのフォーマットは、シンタックスルールに従わなければなりません。次の図は、サポートされる構文を示しています。各ポイントで許可される要素や文字は、矢印をたどって決定することができます。矢印の先にあるブロックに到達できる場合は許可され、到達できない場合は許可されません。

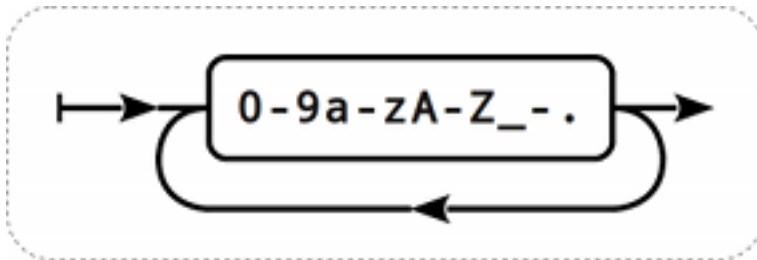


有効なアイテムキーを作成するには、まず、キー名を指定します。次に、パラメータを指定するかしないかを選択します。
キー名

キー名自体には、使用できる文字の範囲が限られており、その文字が相互に続くだけです。許可される文字は：
0-9a-zA-Z_-. です。

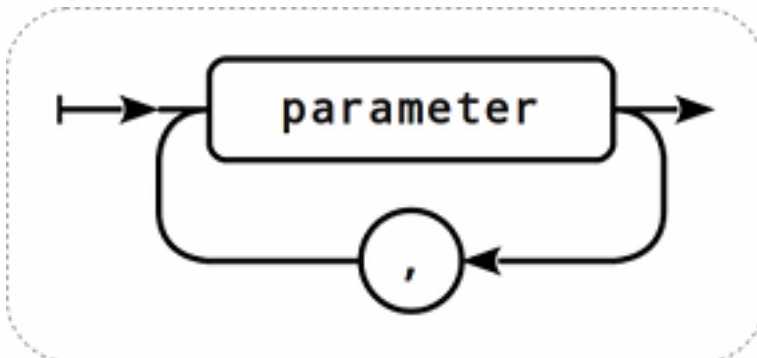
つまり、以下の文字種を指します。

- すべての数字
- すべての小文字
- すべての大文字
- アンダースコア
- ダッシュ
- ドット

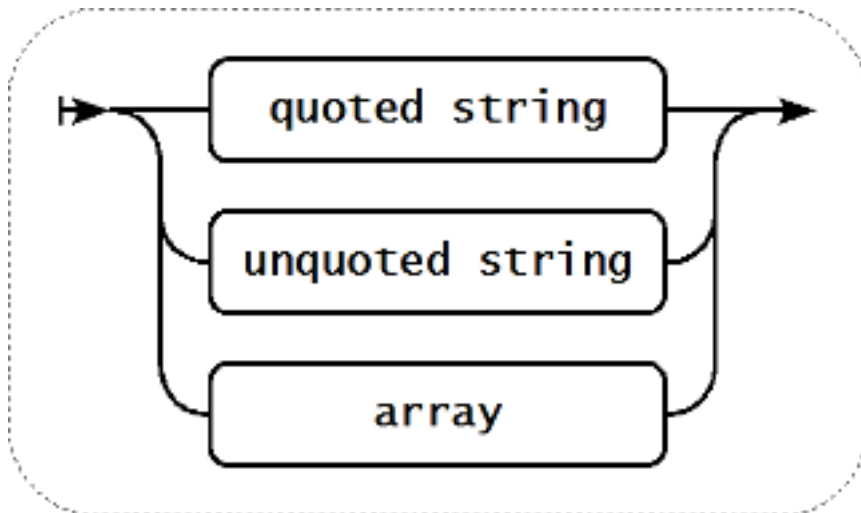


キーパラメーター

アイテムキーは、カンマ区切りで複数のパラメータを持つことができます。



各キーパラメータは、引用符で囲まれた文字列、引用符で囲まれていない文字列、または配列のいずれかになります。



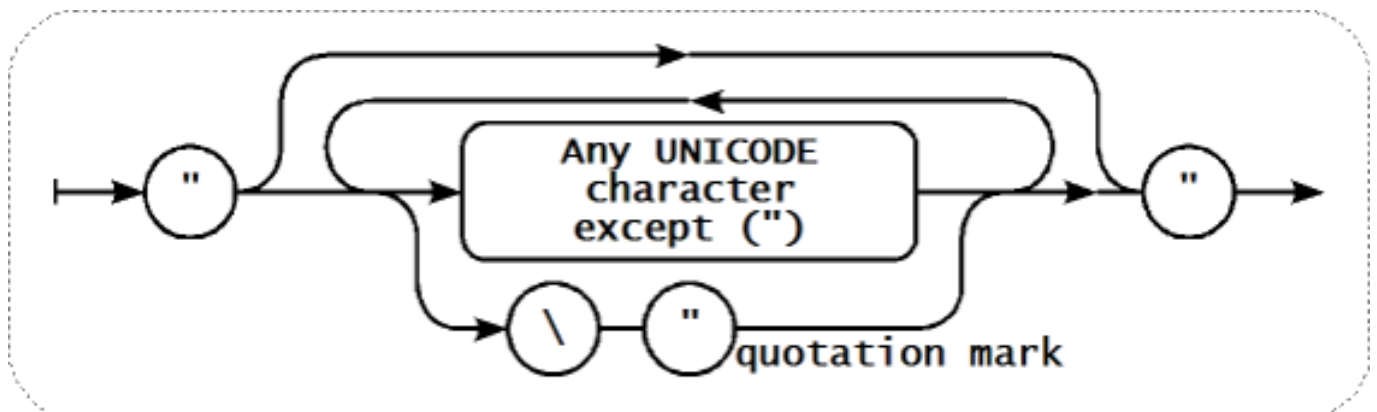
また、パラメータを空にして、デフォルト値を使用することも可能です。
 その場合、さらにパラメータを指定する場合は、適切な数のカンマを追加する必要があります。
 例えば、アイテムキー `icmpping[,200,500]` は、個々の ping の間隔が 200 ミリ秒、ping の間隔が
 500 ミリ秒、その他のパラメータはデフォルトのままです。

パラメータ - 引用符で囲まれた文字列

key パラメータが引用符で囲まれた文字列の場合、任意の Unicode 文字が使用可能です。

key パラメータがカンマを含む場合、このパラメータは引用されなければなりません。

キーパラメータが引用符を含む場合、このパラメータは引用符で囲む必要があります。
 また、パラメータ文字列の一部である各引用符は、バックスラッシュでエスケープする必要があります。

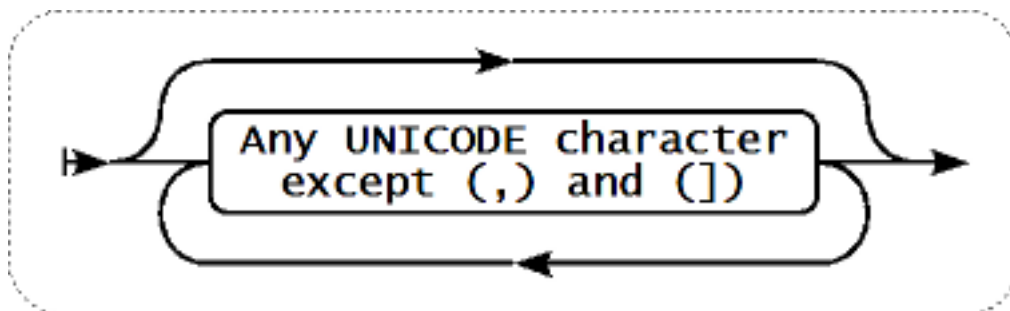


Warning:

アイテムのキーパラメータを引用する場合は、ダブルクォートのみを使用します。
 シングルクォーテーションはサポートされていません。

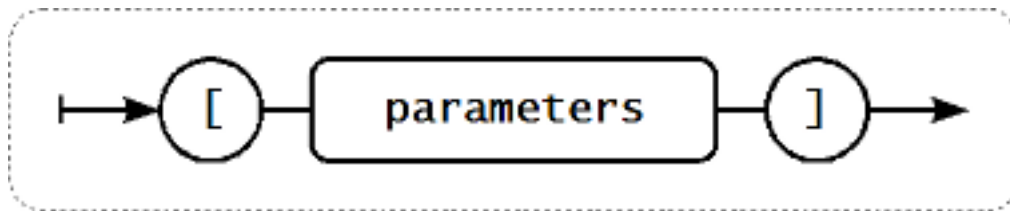
パラメータ - ダブルコーテーションがない文字列

キーパラメータにダブルコーテーションがない文字列の場合、カンマと右大括弧 (]) を除く全ての Unicode 文字が使用可能です。
 ダブルコーテーションで囲まれていないパラメータで、左大括弧 ([) で始まる文字列は使用できません。



パラメータ - 配列

キーパラメータが配列の場合は、再び角括弧で囲みます。
 個々のパラメータは、複数のパラメータを指定する規則と構文に沿うようになります。



Attention:

多段階のパラメータ配列、例えば `
 [a, [b, [c, d]], e]` のような複数レベルのパラメータ配列は許可されません。

2 監視間隔のカスタマイズ

概要

項目がチェックされる時間帯について、カスタムルールを作成することができます。
 デフォルトの更新間隔を再定義できる 例外設定と、特定の時間または一連の時間にアイテムチェックを実行することができる 定期設定の 2 つの方法があります。

例外設定

例外設定は、特定の時間帯のデフォルトの更新間隔を再定義することができます。
 例外設定は、監視間隔と期間で定義されます。

- 監視間隔 - 指定された期間の更新間隔です。
- 期間 - 例外設定が有効な期間 (詳細については、[期間指定フォーマット](#)を参照してください。)

例外設定は最大 7 つまで定義できます。
 複数の例外設定の監視間隔が重なった場合、最も小さい監視間隔の値が重なる期間に使用されます。
 重なる設定の監視間隔の最小値が '0' である場合、ポーリングは行われません。
 例外設定の期間外では、デフォルトの監視間隔が使用されます。

例外設定の監視間隔が期間の長さと同じ場合、アイテムは一度だけチェックされることに注意してください。例外設定の監視間隔が期間より大きい場合、アイテムは一度だけチェックされるか、全くチェックされないかもしれません。(したがって、このような構成は望ましくありません)。もし 監視間隔が 期間より短い場合、そのアイテムは少なくとも 1 回チェックされます。

例外設定の監視間隔が '0' に設定されている場合、そのアイテムは例外設定の期間中はポーリングされません。期間が終了するとデフォルトの監視間隔に従ってポーリングが再開されます。

例:

監視間隔	期間	説明
10	1-5,09:00-18:00	アイテムは勤務時間中に 10 秒ごとにチェックする。
0	1-7,00:00-7:00	アイテムは夜にチェックを行わない。
0	7-7,00:00-24:00	アイテムは日曜日にはチェックを行わない。
60	1-7,12:00-12:01	アイテムは毎日 12:00 にチェックする。これは、スケジュールされたチェック用に回避策として用意されたものであることに注意してください。Zabbix 3.0 以降では、このようなチェックには定期設定を使用してください。

定期設定

定期設定は、特定の時間にアイテムのチェックを実行するために使用します。
 例外設定は、デフォルトのアイテムの監視間隔を再定義するよう設計されていますが、定期設定は、並行して実行される独立したチェックスケジュールを指定するために使用されます。

定期設定の監視間隔は次のように定義されます :
 `md<filter>wd<filter>h<filter>m<filter>s<filter>` 意味は :

- **md** - 日付
- **wd** - 曜日
- **h** - 時
- **m** - 分
- **s** - 秒

<filter> は、その接頭辞の値 (日付、時、分、秒) を指定するために用いられ、次のように定義されます。
 `[<from>[<-to>]] [</step>][,<filter>]`

- `<from>` と `<to>` はマッチする値の範囲 (含む) を定義します。
 もし `<to>` が省略された場合、フィルターは `<from>` - `<from>` の範囲にマッチします。
 もし `<from>` が省略された場合、フィルターはすべての可能な値にマッチします。

- `<step>` は、範囲内の数値をどのようにスキップするかを指定します。
 デフォルトでは `<step>` の値は 1 で、これは定義された範囲のすべての値にマッチすることを意味します。

フィルタの定義は任意であるが、少なくとも 1 つのフィルタが使用されなければなりません。
 フィルタは範囲か `<step>` 値を定義しなければなりません。

空のフィルタは、下位レベルのフィルタが定義されていない場合は '0' に、それ以外の場合は可能なすべての値にマッチします。
 例えば、もし時間フィルタが省略された場合は、'0' の時間だけがマッチします。
 時間フィルタが空の場合は、すべての時間の値にマッチします。

`<from>` と `<to>` それぞれのフィルタプレフィックスに有効な値は:

接頭辞	説明	<code><from></code>	<code><to></code>
md	Month days	1-31	1-31
wd	Week days	1-7	1-7
h	Hours	0-23	0-23
m	Minutes	0-59	0-59
s	Seconds	0-59	0-59

`<from>` の値は `<to>` の値より小さいか等しくなければなりません。
 `<step>` の値は、1 以上かつ `<to>` - `<from>` 以下の値でなければなりません。

一桁の月日、時間、分、秒の値には、先頭に 0 を付けることができます。
 例えば、`md01-31` と `h/02` は有効な間隔です。
 しかし、`md01-031` と `wd01-07` は有効ではありません。

Zabbix の Web インターフェースでは、複数のスケジューリング間隔を別々の行に入力することができます。
 Zabbix API では、セミコロン「;」をセパレーターとして 1 つの文字列に連結されます。

ある時刻が複数の区間に一致する場合、一度だけ実行されます。
 例えば、`wd1h9;h9` は月曜日の午前 9 時に一度だけ実行されます。

監視間隔	実行される日時
<code>m0-59</code>	毎分
<code>h9-17/2</code>	9:00 から 2 時間ごと (9:00, 11:00 ...)
<code>m0,30 or m/30</code>	各時の 0 分と 30 分 (hh:00 と hh:30)
<code>m0,5,10,15,20,25,30,35,40,45,50,55 or m/5</code>	0 分から 5 分ごと
<code>wd1-5h9</code>	月曜から金曜までの 9:00
<code>wd1-5h9-18</code>	月曜から金曜までの 9:00,10:00,...,18:00
<code>h9,10,11 or h9-11</code>	毎日の 9:00、10:00、11:00
<code>md1h9m30</code>	毎月 1 日の 9:30
<code>md1wd1h9m30</code>	毎月 1 日かつ月曜日の 9:30
<code>h9m/30</code>	毎日の 9:00、9:30
<code>h9m0-59/30</code>	毎日の 9:00、9:30
<code>h9,10m/30</code>	毎日の 9:00、9:30、10:00、10:30
<code>h9-10m30</code>	毎日の 9:30、10:30
<code>h9m10-40/30</code>	毎日の 9:10、9:40
<code>h9,10m10-40/30</code>	毎日の 9:10、9:40、10:10、10:40
<code>h9-10m10-40/30</code>	毎日の 9:10、9:40、10:10、10:40
<code>h9m10-40</code>	毎日の 9:10、9:11、9:12、... 9:40
<code>h9m10-40/1</code>	毎日の 9:10、9:11、9:12、... 9:40
<code>h9-12,15</code>	毎日の 9:00、10:00、11:00、12:00、15:00
<code>h9-12,15m0</code>	毎日の 9:00、10:00、11:00、12:00、15:00
<code>h9-12,15m0s30</code>	毎日の 9:00:30、10:00:30、11:00:30、12:00:30、15:00:30
<code>h9-12s30</code>	毎日の 9:00:30、9:01:30、9:02:30、.. 12:58:30、12:59:30
<code>h9m/30;h10 (API 用の文法)</code>	毎日の 9:00、9:30、10:00
<code>h9m/30</code>	毎日の 9:00、9:30、10:00
<code>h10 (Web インターフェース内での別の行として追加)</code>	

Aligning time zones for proxies and agent 2

Note that Zabbix proxies and agent 2 use their local time zones when processing scheduling intervals.

For this reason, when scheduling intervals are applied to items monitored by Zabbix proxy or agent 2 active items, it is recommended to set the time zone of the respective proxies or agent 2 the same as Zabbix server, otherwise the **queue** may report item delays incorrectly.

The time zone for Zabbix proxy or agent 2 can be set using the environment variable TZ in the systemd unit file:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

2 アイテムの値の保存前処理

概要

保存前処理を行うことで、受け取った項目の値の変換ルールを定義することができます。
 データベースに保存する前に、1 つまたは複数の変換を行うことができます。

変換は定義された順番に実行されます。
 保存前処理は Zabbix サーバまたはプロキシ (プロキシでアイテムを監視している場合) で実行されます。

保存前処理に渡される値はすべて文字列型であり、(アイテムの設定で定義された) 目的の値の型への変換は、
 保存前処理パイプラインの最後に実行されることに注意してください。
 ただし、変換は対応する保存前処理工程で必要とされる場合に行われることもあります。
 詳細は[保存前処理の詳細](#)をご参照ください。

使用例も併せてご参照ください。

設定

保存前処理のルールは、アイテム **設定** の保存前処理タブで定義します。

Preprocessing steps	Name	Parameters	Custom on fail
1:	Change per second		<input type="checkbox"/>
2:	Custom multiplier	0.01	<input type="checkbox"/>

Type of information: Numeric (float)

Buttons: Add, Test, Cancel

Attention:

保存前処理のいずれかに失敗すると、アイテムは**取得不可**となります。ただし、サポートされている変換の失敗時のカスタマイズオプションを使用して、カスタムのエラー処理を指定した場合はこの限りではありません。ログアイテムの場合、ログメタデータ (値なし) は常にアイテムの取得不可状態をリセットし、再びサポートされるようになります。

ユーザーマクロおよびユーザーマクロのコンテキストは、アイテム保存前処理のパラメータや JavaScript のコード内でサポートされています。

Note:

マクロをその値で置き換える場合、コンテキストは無視されます。マクロの値はそのままコードに挿入されるため、JavaScript のコードに値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があります。

Type

Transformation

Description

Text

Regular expression	Match the value to the <pattern> regular expression and replace value with <output>. The regular expression supports extraction of maximum 10 captured groups with the \N sequence. Failure to match the input value will make the item unsupported. Parameters: pattern - regular expression output - output formatting template. An \N (where N=1...9) escape sequence is replaced with the Nth matched group. A \0 escape sequence is replaced with the matched text. Please refer to regular expressions section for some existing examples. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value or set a specified error message.
--------------------	---

Replace	Find the search string and replace it with another (or nothing). All occurrences of the search string will be replaced. Parameters: search string - the string to find and replace, case-sensitive (required) replacement - the string to replace the search string with. The replacement string may also be empty effectively allowing to delete the search string when found. It is possible to use escape sequences to search for or replace line breaks, carriage return, tabs and spaces "\n \r \t \s"; backslash can be escaped as "\\" and escape sequences can be escaped as "\\n". Escaping of line breaks, carriage return, tabs is automatically done during low-level discovery.
Trim	Remove specified characters from the beginning and end of the value.
Right trim	Remove specified characters from the end of the value.
Left trim	Remove specified characters from the beginning of the value.

Structured data

XML XPath

Extract value or fragment from XML data using XPath functionality. For this option to work, Zabbix server must be compiled with libxml support.

Examples:

`number(/document/item/value)`
will extract 10 from

```
<document><item><value>10
```

`number(/document/item/@attribute="10")`

will extract 10 from

```
<document><item  
attribute="10"></item></d
```

```
/document/item
```

will extract

```
<item><value>10</value></
```

```
from
```

```
<document><item><value>10
```

Note that

namespaces are not supported.

If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error-handling options: either to discard the value, set a specified value or set a specified error message.

JSON Path

Extract value or fragment from JSON data using [JSONPath functionality](#).

If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error-handling options: either to discard the value, set a specified value or set a specified error message.

Type

CSV to JSON

Convert CSV file data into JSON format.

For more information, see: [CSV to JSON preprocessing](#).

XML to JSON

Convert data in XML format to JSON.

For more information, see: [Serialization rules](#).

If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error-handling options: either to discard the value, set a specified value or set a specified error message.

Arithmetic

Custom multiplier	<p>Multiply the value by the specified integer or floating-point value.</p> <p>Use this option to convert values received in KB, MBps, etc into B, Bps. Otherwise Zabbix cannot correctly set prefixes (K, M, G etc).</p> <p>Note that if the item type of information is Numeric (unsigned), incoming values with a fractional part will be trimmed (i.e. '0.9' will become '0') before the custom multiplier is applied.</p> <p>Supported: scientific notation, for example, 1e+70 (since version 2.2); user macros and LLD macros (since version 4.0); strings that include macros, for example, <code>{#MACRO}e+10</code>, <code>{\$MACRO1}e+{\$MACRO2}</code> (since version 5.2.3)</p> <p>The macros must resolve to an integer or a floating-point number.</p> <p>If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.</p>
-------------------	---

Type

Change

Simple change

Calculate the difference between the current and previous value. Evaluated as **value - prev_value**, where $\text{value} - \text{current value}$; $\text{prev_value} - \text{previously received value}$

This setting can be useful to measure a constantly growing value. If the current value is smaller than the previous value, Zabbix discards that difference (stores nothing) and waits for another value. Only one change operation per item is allowed.

If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

Change per second Calculate the value change (difference between the current and previous value) speed per second. Evaluated as **(value-
prev_value)/(time-
prev_time)**, where value - current value; prev_value - previously received value; time - current timestamp; prev_time - timestamp of previous value. This setting is extremely useful to get speed per second for a constantly growing value. If the current value is smaller than the previous value, Zabbix discards that difference (stores nothing) and waits for another value. This helps to work correctly with, for instance, a wrapping (overflow) of 32-bit SNMP counters. Note: As this calculation may produce floating-point numbers, it is recommended to set the 'Type of information' to Numeric (float), even if the incoming raw values are integers. This is especially relevant for small numbers where the decimal part matters. If the floating-point values are large and may exceed the 'float' field length in which case the entire value may be lost, it is actually suggested to use Numeric

Boolean to decimal

Convert the value from boolean format to decimal. The textual representation is translated into either 0 or 1. Thus, 'TRUE' is stored as 1 and 'FALSE' is stored as 0. All values are matched in a case-insensitive way. Currently recognized values are, for:

TRUE - true, t, yes, y, on, up, running, enabled, available, ok, master

FALSE - false, f, no, n, off, down, unused, disabled, unavailable, err, slave

Additionally, any non-zero numeric value is considered to be TRUE and zero is considered to be FALSE.

If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

	Octal to decimal	Convert the value from octal format to decimal. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.
	Hexadecimal to decimal	Convert the value from hexadecimal format to decimal. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.
Custom scripts	JavaScript	Enter JavaScript code in the block that appears when clicking in the parameter field or on a pencil icon. Note that available JavaScript length depends on the database used . For more information, see: JavaScript preprocessing .
Validation		

In range	Define a range that a value should be in by specifying minimum/maximum values (inclusive). Numeric values are accepted (including any number of digits, optional decimal part and optional exponential part, negative values). User macros and low-level discovery macros can be used. The minimum value should be less than the maximum. At least one value must exist. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.
Matches regular expression	Specify a regular expression that a value must match. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

Does not match
regular expression

Specify a regular expression that a value must not match. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

Check for error in
JSON

Check for an application-level error message located at JSONpath. Stop processing if succeeded and the message is not empty; otherwise, continue processing with the value that was before this preprocessing step. Note that these external service errors are reported to the user as is, without adding preprocessing step information. No error will be reported in case of failing to parse invalid JSON. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

Check for error in XML

Check for an application-level error message located at XPath. Stop processing if succeeded and the message is not empty; otherwise, continue processing with the value that was before this preprocessing step. Note that these external service errors are reported to the user as is, without adding preprocessing step information. No error will be reported in case of failing to parse invalid XML. If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

Check for error using a regular expression

Check for an application-level error message using a regular expression. Stop processing if succeeded and the message is not empty; otherwise, continue processing with the value that was before this preprocessing step. Note that these external service errors are reported to the user as is, without adding preprocessing step information.
Parameters:
pattern - regular expression
output - output formatting template. An \N (where N=1...9) escape sequence is replaced with the Nth matched group. A \0 escape sequence is replaced with the matched text.
If you mark the Custom on fail checkbox, the item will not become unsupported in case of failed preprocessing step and it is possible to specify custom error handling options: either to discard the value, set a specified value, or set a specified error message.

Check for not supported value

Check if there was an error in retrieving item value. Normally that would lead to the item turning unsupported, but you may modify that behavior by specifying the Custom on fail error-handling options: to discard the value, to set a specified value (in this case the item will stay supported and the value can be used in triggers) or set a specified error message. Note that for this preprocessing step, the Custom on fail checkbox is grayed out and always marked. This step is always executed as the first preprocessing step and is placed above all others after saving changes to the item. It can be used only once. Supported since 5.2.0.

Throttling

Discard unchanged

Discard a value if it has not changed. If a value is discarded, it is not saved in the database and Zabbix server has no knowledge that this value was received. No trigger expressions will be evaluated, as a result, no problems for related triggers will be created/resolved. Functions will work only based on data that is actually saved in the database. As trends are built based on data in the database, if there is no value saved for an hour then there will also be no trends data for that hour. Only one throttling option can be specified for an item. Note that it is possible for items monitored by Zabbix proxy that very small value differences (less than 0.000001) are correctly not discarded by proxy, but are stored in the history as the same value if the Zabbix server database [has not been upgraded](#).

Discard unchanged with heartbeat

Discard a value if it has not changed within the defined time period (in seconds). Positive integer values are supported to specify the seconds (minimum - 1 second). Time suffixes can be used in this field (e.g. 30s, 1m, 2h, 1d). User macros and low-level discovery macros can be used in this field.

If a value is discarded, it is not saved in the database and Zabbix server has no knowledge that this value was received. No trigger expressions will be evaluated, as a result, no problems for related triggers will be created/resolved. Functions will work only based on data that is actually saved in the database. As trends are built based on data in the database, if there is no value saved for an hour then there will also be no trends data for that hour. Only one throttling option can be specified for an item.

Note that it is possible for items monitored by Zabbix proxy that very small value differences (less than 0.000001) are correctly not discarded by proxy, but are stored in the history as the same value if the Zabbix server

Type

Prometheus

Prometheus
pattern

Use the following query to extract required data from Prometheus metrics.
See [Prometheus checks](#) for more details.

Prometheus to
JSON

Convert required Prometheus metrics to JSON.
See [Prometheus checks](#) for more details.

Attention:

保存前処理のステップの変更と調整のために、Zabbix は必要に応じて新しい値を計算/比較するために、最後の値を覚えておく必要があります。これらの以前の値は保存前処理マネージャによって処理されます。Zabbix サーバまたはプロキシを再起動した場合、または保存前処理ステップを変更した場合、Zabbix は最後の値を記憶し、必要に応じて新しい値を計算/比較します。

- 差分、1 秒あたりの差分ステップの場合、次の値は無視されます。
 次の値は、変化量を計算する前の値がないため、無視されます。
- 変更がなければ破棄、指定病内に変化がなければ破棄ステップの場合、
 次の値は、たとえ破棄される必要があったとしても、破棄されることはありません。

アイテムのデータ型パラメータは、少なくとも 1 つの保存前処理工程が定義されている場合、タブの下部に表示されます。
 必要であれば、保存前処理タブを開いたまま情報の種類を変更することができます。
 [アイテムの作成](#) を参照してください。

Note:

カスタム倍率を使用する場合、または値を 1 秒あたりの差分として保存する場合、数値 (整数) に設定されている項目で、計算結果の値が実際には浮動小数点数であっても、小数点以下を切り捨てて整数値として格納することで、正しい値として扱われます。

保存前処理のテスト

保存前処理ステップのテストは、アイテムの値の受信と保存前処理を待たずに、複雑なプリプロセスパイプラインが期待される結果をもたらすことを確認するのに有効です。

Item	Tags	Preprocessing 3																
Preprocessing steps																		
		<table border="1"><thead><tr><th>Name</th><th>Parameter</th><th>Custom on fail</th><th>Actions</th></tr></thead><tbody><tr><td>1: Regular expression</td><td>[(0-9)+]</td><td><input type="checkbox"/></td><td>Test Re</td></tr><tr><td>2: Regular expression</td><td>[(0-9+)]</td><td><input type="checkbox"/></td><td>Test Re</td></tr><tr><td>3: Regular expression</td><td>[(0-9+)]</td><td><input type="checkbox"/></td><td>Test Re</td></tr></tbody></table>	Name	Parameter	Custom on fail	Actions	1: Regular expression	[(0-9)+]	<input type="checkbox"/>	Test Re	2: Regular expression	[(0-9+)]	<input type="checkbox"/>	Test Re	3: Regular expression	[(0-9+)]	<input type="checkbox"/>	Test Re
Name	Parameter	Custom on fail	Actions															
1: Regular expression	[(0-9)+]	<input type="checkbox"/>	Test Re															
2: Regular expression	[(0-9+)]	<input type="checkbox"/>	Test Re															
3: Regular expression	[(0-9+)]	<input type="checkbox"/>	Test Re															
		Add																
Type of information	<input type="text" value="Text"/>																	
	<input type="button" value="Add"/>	<input type="button" value="Test"/> <input type="button" value="Cancel"/>																

- 仮想的な値に対してテストすることが可能です。
- ホストからの実際の値に対してテストすることが可能です。

保存前処理の各ステップを個別にテストすることも、すべてのステップを一緒にテストすることも可能です。アクションブロックのテストまたはすべてのテストボタンをクリックすると、テストウィンドウが表示されます。

仮想的な値のテスト

✕
Test item

⚠ cannot perform regular expression "[0-9+]" match for value of type "string": invalid regular expression: missing terminating] for character class ✕

Get value from host

Value Time

Not supported

Previous value Prev. time

End of line sequence LF CRLF

Preprocessing steps	Name	Result
	1: Regular expression	15
	2: Regular expression	1
	3: Regular expression	!

Test
Cancel

パラメータ	説明
Get value from host	If you want to test a hypothetical value, leave this checkbox unmarked.
Value	See also: Testing real value . Enter the input value to test. Clicking in the parameter field or on the view/edit button will open a text area window for entering the value or code block.
Not supported	Mark this checkbox to test an unsupported value. This option is useful to test the Check for not supported value preprocessing step.
Time	Time of the input value is displayed: <code>now</code> (read-only).
Previous value	Enter a previous input value to compare to.
Previous time	Only for Change and Throttling preprocessing steps. Enter the previous input value time to compare to. Only for Change and Throttling preprocessing steps. The default value is based on the 'Update interval' field value of the item (if '1m', then this field is filled with <code>now-1m</code>). If nothing is specified or the user has no access to the host, the default is <code>now-30s</code> .
Macros	If any macros are used, they are listed along with their values. The values are editable for testing purposes, but the changes will only be saved within the testing context.
End of line sequence	Select the end of line sequence for multiline input values: LF - LF (line feed) sequence CRLF - CRLF (carriage-return line-feed) sequence.
Preprocessing steps	Preprocessing steps are listed; the testing result is displayed for each step after the Test button is clicked. If the step failed in testing, an error icon is displayed. The error description is displayed on mouseover. In case "Custom on fail" is specified for the step and that action is performed, a new line appears right after the preprocessing test step row, showing what action was done and what outcome it produced (error or value).
Result	The final result of testing preprocessing steps is displayed in all cases when all steps are tested together (when you click on the Test all steps button). The type of conversion to the value type of the item is also displayed, for example Result converted to Numeric (unsigned).

各保存前処理後の結果を見るには、テストをクリックしてください。

テストセッションの間に、個々のステップまたは全ステップについてテスト値を保存することが出来るので、保存前処理やアイテムの設定を変更しても、情報を再入力することなくテストウィンドウに戻ることができます。ただし、ページの更新時に値は失われます。

テストは Zabbix サーバーで行われます。フロントエンドは対応するリクエストをサーバに送信し結果を待ちます。リクエストには入力値と保存前処理のステップ (拡張ユーザマクロを含む) が含まれます。変化および絞り込みのステップでは、オプションで前値と時間を指定することができます。サーバーは、各保存前処理ステップの結果を応答します。

すべての技術的エラーまたは入力検証エラーは、テストウィンドウ上部のエラーボックスに表示されます。

実測値のテスト

保存前処理を実測値に対してテストするため、

- ホストから値を取得チェックボックスをマークします。

- ホストパラメータ (ホストアドレス、ポート、プロキシ名 / プロキシなし) および項目固有の詳細
 (SNMPv2 コミュニティや SNMPv3 セキュリティ認証など) を入力または確認します。これらのフィールドは、コンテキストを認識します。

 - エージェントを必要とする項目については、ホストの選択されたエージェントインターフェースから情報を取得することで、
 可能な限り値は事前に入力されます。

 - テンプレートアイテムの場合は、値を手動で入力する必要があります。

 - プレーンテキストのマクロ値は解決されます。

 - 値 (または値の一部) がシークレットまたは Vault マクロであるフィールドは空であるため、手動で入力しなければなりません。
 アイテムのパラメータにシークレットマクロ値が含まれている場合、次のような警告メッセージが表示されます。
 " アイテムは秘密な値を持つユーザー定義マクロを含んでいます。これらのマクロの値は手動で入力する必要があります。 "

 - アイテムタイプのコンテキストで必要ない場合、フィールドは無効化されます。
 フィールドは、アイテムの種類によって必要ない場合は無効になります
 (たとえば、ホストアドレスとプロキシフィールドは、計算されたアイテムでは無効になります)。

- 保存前処理をテストするには、値の取得とテストをクリックします。

アイテム設定画面で値のマッピングを指定している場合 (' 値のマッピング' フィールド)、アイテムテストダイアログは最終結果の後にもう 1 行表示します。' 値のマッピングを適用した結果' という名前の別の行が表示されます。

ホストから実測値を取得する際に特有のパラメータ:

パラメータ	説明
Get value from host	Mark this checkbox to get a real value from the host.
Host address	Enter the host address. This field is automatically filled by the address of the item host interface.
Port	Enter the host port. This field is automatically filled by the port of item host interface.

パラメータ	説明
Additional fields for SNMP interfaces (SNMP version, SNMP community, Context name, etc)	See Configuring SNMP monitoring for additional details on configuring an SNMP interface (v1, v2 and v3). These fields are automatically filled from the item host interface.
Proxy	Specify the proxy if the host is monitored by a proxy. This field is automatically filled by the proxy of the host (if any).

残りのパラメータについては、上記の[仮説値のテスト](#)を参照してください。

1 使用例

概要

このセクションでは、プリプロセスステップを使用していくつかの実用的なタスクを実行する例を紹介します。

VMware のイベントログをフィルタリングする

正規表現のプリプロセスを使用して、VMWare イベントログの不要なイベントをフィルタリングします。

- 動作中の VMWare Hypervisor ホストで、イベントログの項目 `vmware.eventlog[<url>,<mode>]`が存在し、
 正しく動作していることを確認します。
 Template VM VMWare テンプレートがホストの作成時にリンクされている場合、イベントログ item は
 ハイパーバイザー上に既に存在する可能性があることに注意してください。

- VMWare Hypervisor ホストで、**dependent item**に'Log' タイプを作成してイベントログをマスターに設定します。

従属 item の"Preprocessing" タブで、"Matches regular expression" という検証オプションを選択し、パターンを記入します。

`.* logged in .*` - は、イベントログのすべてのログインイベントをフィルタリングします。

`\\bUser\s+\K\S+` - イベントログからユーザー名を含む行のみをフィルターします。

Attention:

注意点：正規表現にマッチしない場合、従属 item がサポートされなくなり、対応するエラーメッセージが表示されます。

 これを避けるために、"Custom on fail" チェックボックスをマークし、マッチしない値を破棄するように選択します。

マッチンググループと出力制御を使用できる別の方法は、
 "Preprocessing " タブで"Regular expression " オプションを選択し、以下のようにパラメータを入力することです。

パターン `.*logged in.*`, output: `"\0"` イベントログのすべてのログインイベントをフィルタリングします。

パターン `"User (.*)?(?=\)"`, output: `"\1"` - イベントログからユーザー名のみをフィルタリングします。

2 プリプロセッシングの詳細

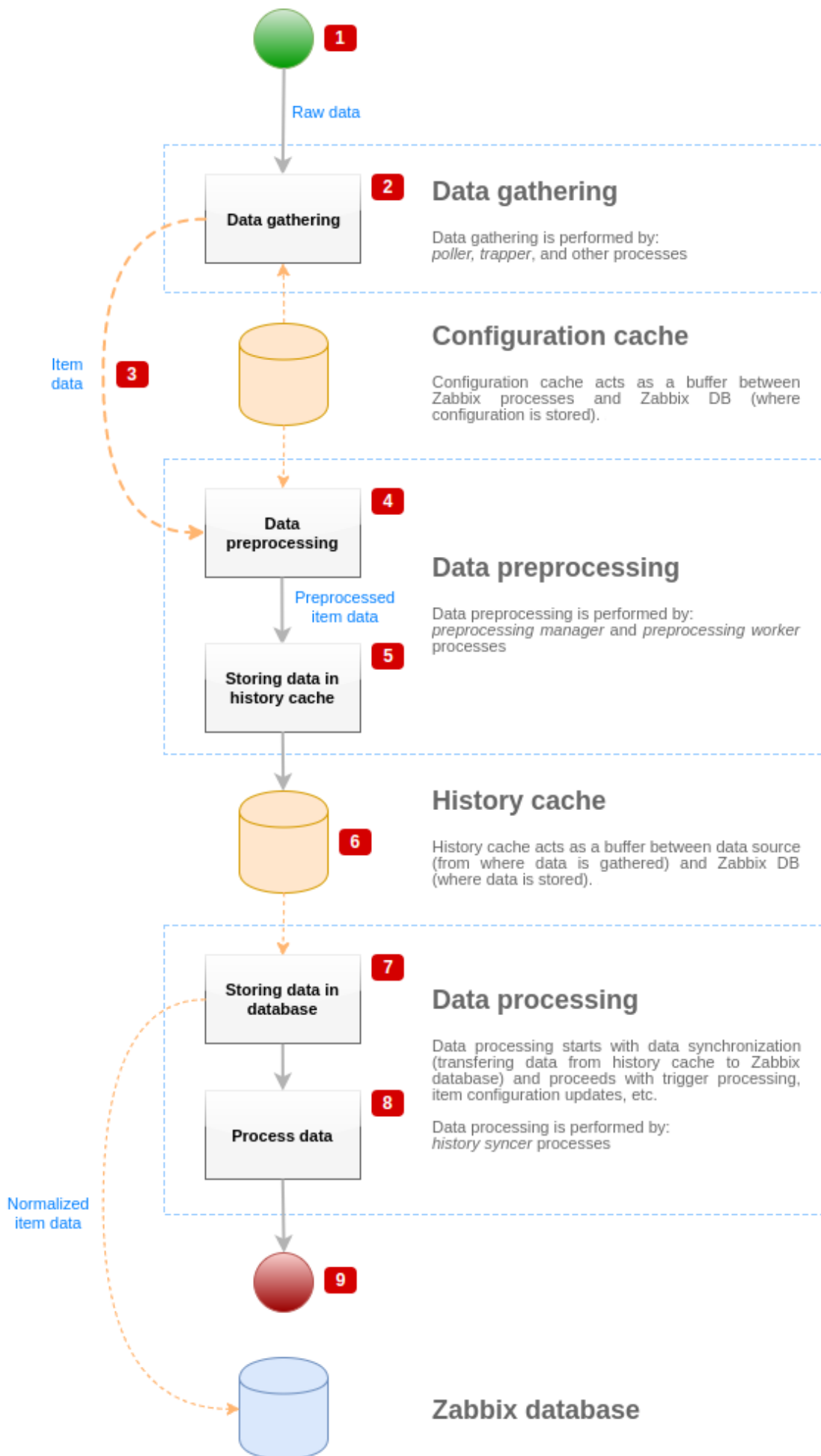
概要

ここでは、item 値のプリプロセッシングについて説明します。item 値のプリプロセッシングでは、
 受け取った item 値に対して **transformation rules**を定義し、実行することができます。

プリプロセスは、Zabbix3.4 で追加されたプリプロセスマネージャプロセスとプリプロセスワーカーによって管理され、
 プリプロセスのステップを実行することができます。
 プリプロセスのステップは、異なるデータ収集元からのすべての値 (プリプロセスあり/なし) が、
 ヒストリーキャッシュに追加される前に、プリプロセッシングマネージャを通過します。
 データ収集装置 (ポーラ、トラッパーなど) とプリプロセスの間で、ソケットベースの IPC 通信が使用されます。
 Zabbix server または Zabbix proxy(proxy が監視する item の場合) のどちらかがプリプロセスを実行します。

item 値の処理

データソースから Zabbix データベースへのデータフローを可視化するために、以下の簡略化された図を使用します。



上の図は、item 値処理に関連するプロセス、オブジェクト、アクションのみを簡略化して示しています。この図には条件分岐、エラー処理、ループなどは示していません。プリプロセスマネージャのローカルデータキャッシュも表示されていません。データフローに直接影響しないためです。この図の目的は、item 値処理に関わるプロセスと、それらが相互作用する方法を示すことです。

- データ収集は、データソースからの生データから始まる。このデータには ID、タイムスタンプ、値（複数の値も可能）のみが含まれます。
- アクティブチェック、パッシブチェック、トラッパー item など、どのような種類のデータ収集装置を使っても、考え方は同じで、データ形式と通信の開始方法を変えるだけです。（データ収集者が接続とデータを待っているか、データ収集者が通信を開始し、データを要求しているか）生データは検証され、item 設定は設定キャッシュから取得されます。（データは設定データでリッチ化されます）。
- データ収集者からプリプロセスマネージャへのデータの受け渡しには、ソケットベースの IPC メカニズムが使用されます。この時点でデータ収集装置はプリプロセスマネージャからの応答を待たずにデータ収集を続ける。
- データのプリプロセッシングが行われる。これには、プリプロセスステップの実行と依存項目処理が含まれます。

Note:

item はプリプロセッシングが行われている間にいずれかの処理が失敗した場合、状態が NOT SUPPORTED に変更されることがあります。

- プリプロセスマネージャのローカルデータキャッシュから履歴データが履歴キャッシュに流される。
- この時点でデータの流は止まり、次の履歴キャッシュの同期（履歴シンサープロセスがデータの同期を行う）までデータの流は停止します。
- 同期処理はデータの正規化から始まり、データを Zabbix データベースに格納することから始まります。データの正規化では、以下の変換を行います。目的の item タイプ (item 設定で定義されたタイプ) への変換を行います。テキストデータについては、あらかじめ定義されたサイズに基づき、切り捨てを行います。テキストデータは、それらのタイプに許容される事前定義されたサイズ (文字列は HISTORY_STR_VALUE_LEN、テキストは HISTORY_TEXT_LEN、ログ値には HISTORY_LOG_VALUE_LEN) に基づいて切り捨てられます。データは正規化された後、Zabbix データベースに送信されます。

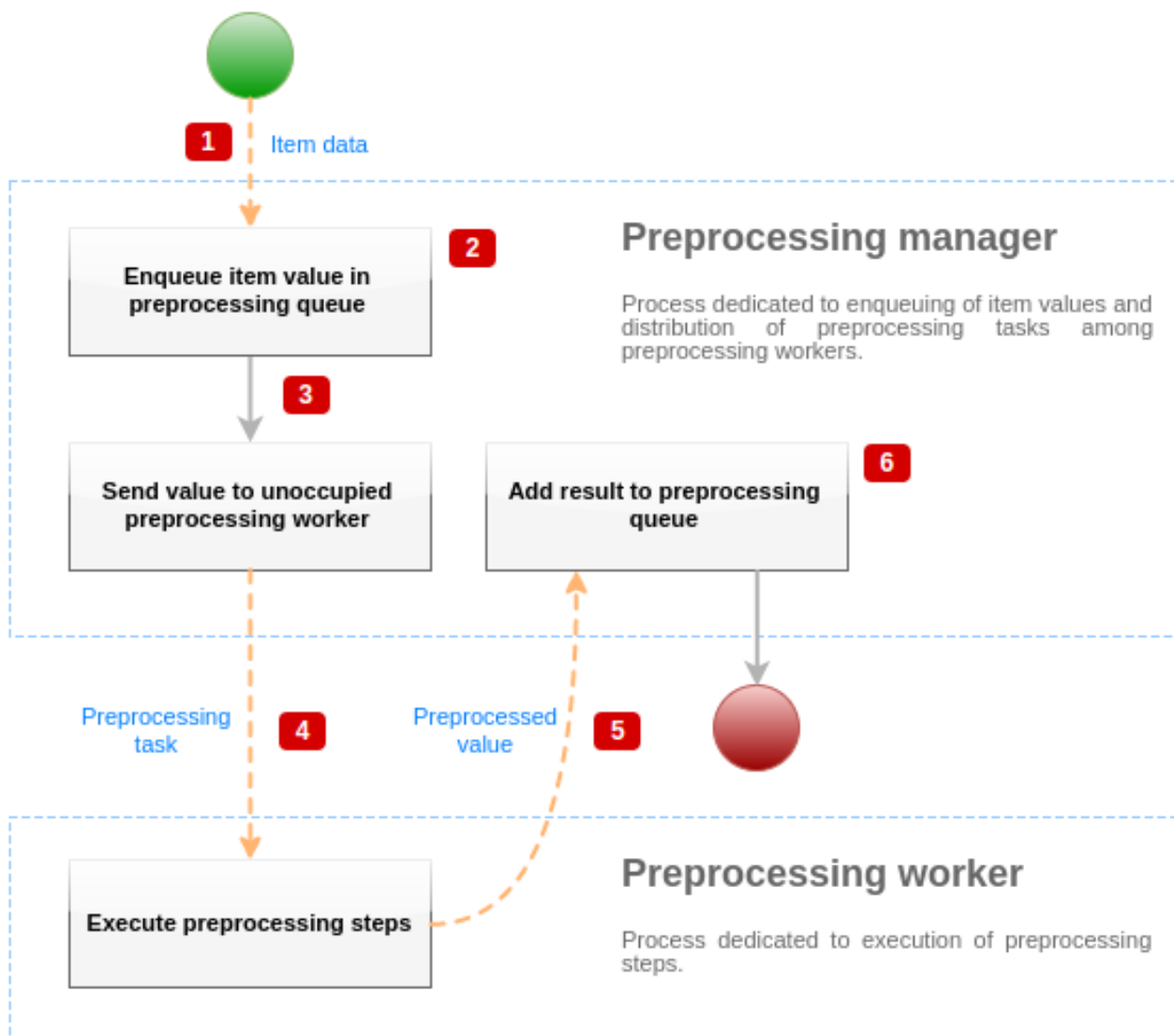
Note:

データの正規化に失敗した場合 (例: テキスト値を数値に変換できない場合)、item は NOT SUPPORTED に状態を変更されることがあります。

- 収集されたデータは処理される - トリガーがチェックされ、item が NOT SUPPORTED になった場合は設定が更新される
- item 値処理の観点からは、これでデータフローは終了したとみなされる。

item 値のプリプロセッシング

データのプリプロセッシングを可視化するために、以下のような簡略化した図を示します。



上の図は、item 値のプリプロセッシングに関連するプロセス、オブジェクト、および主なアクションのみを簡略化した形で示しています。この図では条件分岐、エラー処理、ループは示していません。ただ1つのプリプロセスワーカーだけが示されています。この項目は少なくとも1つのプリプロセス工程を実行する必要があると想定している。この図の目的は、item 値のプリプロセスパイプラインの考え方を示すことである。

- item データと item 値は、ソケットベースの IPC メカニズムを使用してプリプロセスマネージャに渡される。
- item はプリプロセスキューに入れられる。

Note:

item はプリプロセスキューの最後または先頭に配置されます。Zabbix 内部アイテムは常にプリプロセスキューの先頭に配置され、その他のアイテムタイプは最後にキューイングされます。

- この時点で、少なくとも1つの未使用の(タスクを実行していない)プリプロセスワーカーが存在するまでデータフローは停止します。
- プリプロセスワーカーが利用可能になると、プリプロセスタスクはそのワーカーに送られます。
- プリプロセスが完了すると(プリプロセスステップの実行に失敗した場合も成功した場合も)処理された値がプリプロセッシングマネージャに戻されます。
- プリプロセスマネージャは、結果を(item 値の型によって定義される)希望する形式に変換し、結果をプリプロセスキューに入れる。もし現在の item に依存する item がある場合、依存する item もプリプロセスキューに追加されます。ただし、値が設定され、NOT SUPPORTED 状態でないマスター item に限ります。

値処理のパイプライン

item 値処理は、複数のプロセスによって、複数のステップ(またはフェーズ)で実行されます。これが要因となって、

- 従属 item は値を受け取ることができるが、親 item は値を受け取ることができない。これは、次のユースケースを使用することで実現できます。

- 親 item は値型が UINT (トラッパー項目も使用できる)、従属項目は値型 TEXT を持つ。

- 親 item、従属 item とともに前処理は必要ありません。

- テキスト値 (例えば"abc") は親 item に渡す必要があります。

- 実行するプリプロセスステップがないので、プリプロセスマネージャーは、
 親 item が NOT SUPPORTED 状態でないかどうか、値が設定されているかどうかを確認し (両方とも真)、
 親 item と同じ値を持つ従属アイテムを待ちます。

- 親 item と従属アイテムの両方が履歴同期フェーズに到達すると、親 item は NOT SUPPORTED になります。
 なぜなら、値の変換エラー (テキストデータは符号なし整数に変換できない) が発生するからです。

その結果、従属 item は値を受け取り、親 item の状態が NOT SUPPORTED に変わります。

- 従属 item は親 item の履歴に存在しない値を受け取ります。
 このユースケースは、以下の点を除き、親 item の種類を除いて、前のものと非常によく似ています。
 たとえば、CHAR 型が親 item に使用されている場合、親 item の値は
 親 item の値は履歴の同期段階で切り捨てられます。
 従属 item の値は親 item の初期値 (切り捨てられない) から取得します。

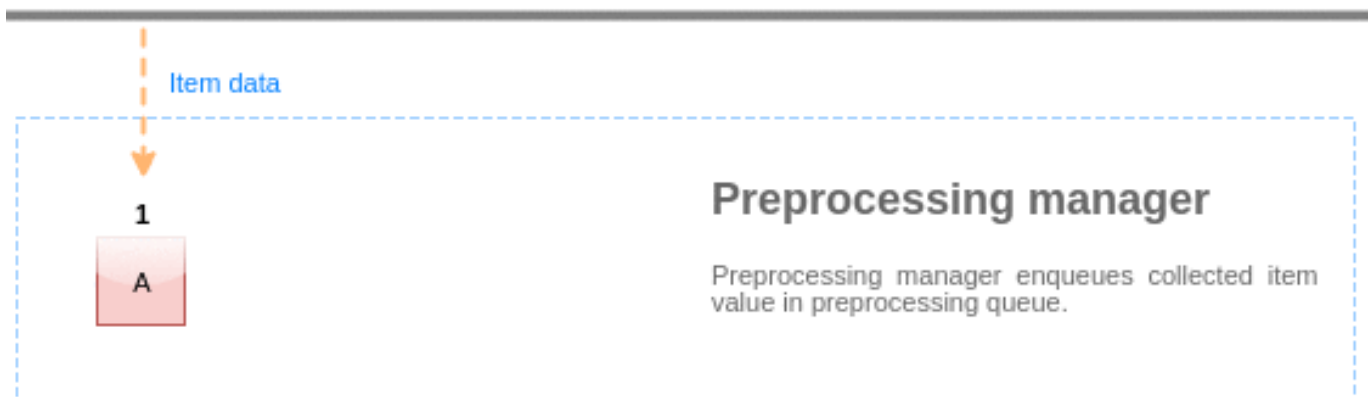
プリプロセスキュー

プリプロセスキューは、プリプロセスによって得られた値の順序を保持したまま、値を保存する FIFO データ構造です。
 FIFO ロジックにはいくつかの例外があります:

- 内部 item はキューの最初にエンキューされる。
- 従属 item は常に親 item の後にエンキューされる

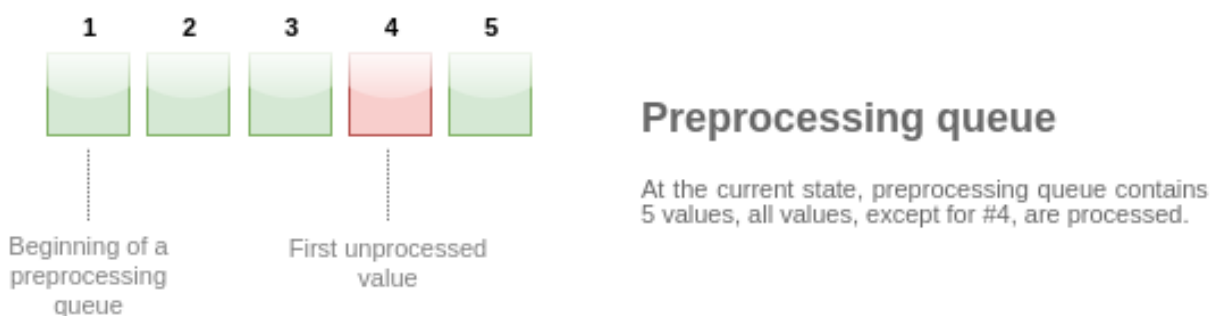
プリプロセスキューのロジックを視覚化するために、次の図を使用します。

Data gatherers



Preprocessing workers

プリプロセスキューからの値は、キューの先頭から最初の未処理キューにフラッシュされます。
 したがって、例えば、プリプロセスマネージャは、値 1、2、3 をフラッシュしますが、値 4 はまだ処理されていないため、
 値 5 はフラッシュしません。



フラッシュ後、キューには 2 つの値 (4 と 5) だけが残され、プリプロセッシングマネージャのローカルデータキャッシュに追加され、
 その後、値がローカルキャッシュからヒストリーキャッシュに転送されます。
 プリプロセッシングマネージャは、ロ

一カルデータキャッシュから、単一 item モードまたはバルクモード
 (依存する item や一括して受け取った値に使用) で値をフラッシュすることができます。

プリプロセスワーカー

Zabbix サーバ設定ファイルでは、プリプロセスワーカープロセスの数を設定することができます。
 StartPreprocessors 設定パラメータを使用して、フォークされたプリプロセスワーカーインスタンスの数を設定する必要があります。
 最適なプリプロセスワーカーの数は、"プリプロセッシング可能な" アイテムの数など、多くの要因によって決まります。
 データ収集プロセスの数、item の前処理にかかる平均ステップ数などです。

しかし、大きな XML/JSON チャンクのパースなどの重いプリプロセスがないと仮定すると、
 プリプロセスワーカーの数はデータ収集プロセスの総数と一致させることができます。この方法では、ほとんどの場合
 (ギャザラーからのデータが大量に来る場合を除く)、少なくとも 1 つの空いたプリプロセスワーカーが存在することになります。

Warning:

データ収集プロセスが多すぎる (ポーラ、到達不能ポーラ、ODBC ポーラ、HTTP ポーラ、Java ポーラ、pingers、trappers、
 プロキシポーラ) ケースで、IPMI マネージャ、SNMP トラップ、プリプロセスワーカーと一緒に使用すると、
 プリプロセスマネージャのプロセスごとのファイルディスクリプタを使い果たす可能性があります。
 この場合、Zabbix サーバは停止します。(通常は起動後すぐに停止しますが、もっと時間がかかる場合もあります)
 この状況を回避するために、設定ファイルを修正するか、制限値を上げる必要があります。

3 JSONPath 機能

概要

このセクションでは、item 値のプリプロセスステップでサポートされる JSONPath 機能の詳細について説明します。

JSONPath は、ドットで区切られたセグメントで構成されます。
 セグメントは、JSON 値の名前 * のような単純な単語と、あるいは角括弧 [] で囲まれたより複雑な構造体です。
 セグメントを区切るドットはオプションで、省略することもできます。例えば

Path	説明
\$.object.name	Return the object.name contents.
\$.object['name']	Return the object.name contents.
\$.object.['name']	Return the object.name contents.
\$["object"]['name']	Return the object.name contents.
\$.['object'].["name"]	Return the object.name contents.
\$.object.history.length()	Return the number of object.history array elements.
[\$?(@.name == 'Object')].price.first()	Return the price field of the first object with name 'Object'.
[\$?(@.name == 'Object')].history.first().length()	Return the number of history array elements of the first object with name 'Object'.
[\$?(@.price > 10)].length()	Return the number of objects with price being greater than 10.

Escaping special characters from LLD macro values in JSONPath も参照してください。

対応セグメント

セグメント	説明
<name>	オブジェクトプロパティを名前で照合します
*	すべてのオブジェクトプロパティに一致します
['<name>']	オブジェクトプロパティを名前で照合します
['<name>', '<name>', ...]	リストされた名前のいずれかでオブジェクトプロパティを照合します
[<index>]	配列要素をインデックスで照合します
[<number>, <number>, ...]	リストされたインデックスのいずれかで配列要素を照合します
[*]	すべてのオブジェクトプロパティまたは配列要素に一致します

セグメント	説明
[<start>:<end>]	定義された範囲で配列要素を一致させます: <start> - 一致する最初のインデックス (含む)。指定しない場合は、最初からすべての配列要素に一致します。負の場合は、配列の末尾からの開始オフセットを指定します。 <end> - 一致する最後のインデックス (除外)。指定されていない場合は、すべての配列要素を最後まで一致させます。負の場合、配列の末尾からの開始オフセットを指定します。
[?(<expression>)]	フィルター式を適用してオブジェクト/配列要素を一致させます

前例を無視して一致するセグメント (切り離されたセグメント) を見つけるには、接頭辞'..' を付ける必要があります。たとえば、\$.name または \$..['name'] はすべての'name' プロパティの値を返します。

一致した要素名は、JSONPath に ~ サフィックスを追加することで抽出できます。一致したオブジェクトの名前または一致した配列項目の文字列形式のインデックスを返します。出力形式は、他の JSONPath クエリと同じ規則に従います。明確なパスの結果は'そのまま'返され、不明確なパスの結果は配列で返されます。ただし、明確なパスに一致する要素の名前を抽出する意味はあまりありません - それはすでに知られているからです。

フィルター式

フィルター式は、中置表記の算術式です。

サポートされているオペラント:

オペラント	説明	例
"<text>"	テキスト定数	'value: \'1\''
'<text>'		"value: '1'"
<number>	科学表記法をサポートする数値定数	123
<jsonpath starting with \$>	入力ドキュメントのルートノードから JSONPath によって参照される値。明確なパスのみがサポートされています。	\$.object.name
<jsonpath starting with @>	現在のオブジェクト/要素から JSONPath によって参照される値。明確なパスのみがサポートされています。	@.name

サポートされている演算子:

演算子	タイプ	説明	結果
-	binary	減算	Number.
+	binary	加算	Number.
/	binary	除算	Number.
*	binary	乗算	Number.
==	binary	等しい	Boolean (1 or 0).
!=	binary	等しくない	Boolean (1 or 0).
<	binary	未満	Boolean (1 or 0).
<=	binary	以下	Boolean (1 or 0).
>	binary	超過	Boolean (1 or 0).
>=	binary	以下	Boolean (1 or 0).
==~	binary	正規表現に一致	Boolean (1 or 0).
!	binary	ブール値の not	Boolean (1 or 0).
	binary	ブール値の or	Boolean (1 or 0).
&&	binary	ブール値の and	Boolean (1 or 0).

関数

関数は JSONPath の最後で使用できます。前の関数が次の関数によって受け入れられる値を返す場合、複数の関数をチェーンできます。

サポートされている関数:

関数	説明	入力	出力
avg	入力配列の数値の平均値	数値の配列	数値
min	入力配列の数値の最小値	数値の配列	数値

関数	説明	入力	出力
max	入力配列の数値の最大値	数値の配列	数値
sum	入力配列の数値の合計	数値の配列	数値
length	入力配列の要素数	配列	数値
first	最初の配列要素	配列	入力配列の内容に応じた JSON コンストラクト (オブジェクト、配列、値)

引用符で囲まれた数値は、JSONPath 集計関数によって受け入れられます。これは、集計が必要な場合、値が文字列型から数値に変換されることを意味します。

入りに互換性がない場合、関数はエラーを生成します。

出力値

JSONPaths は、定型パスと不定型パスに分けることができます。
 定型パスは、NULL または単一のマッチのみを返すことができます。不定型パスは、
 複数の名前/インデックスリスト、配列スライス、式セグメントを含む JSONPath を返します。
 しかし関数が使用されると、JSONPath は定型的なものになります。関数は常に単一の値を出力するからです。

定型パスは、参照するオブジェクト/配列/値を返します。
 不定形のパスは、マッチしたオブジェクト/配列/値の配列を返します。

空白文字

空白文字 (スペース、タブ文字) は、ブラケット記法のセグメントで自由に使用することができます。
 例: `$('a') [0] [? ($.b == 'c')] [: -1].first()`.

文字列

文字列は、シングルクォート (') またはダブルクォート (") で囲みます。
 文字列の内部では、シングルクォートまたはダブルクォート (どちらで囲むかによって異なる) と
 バックスラッシュ \ は、バックスラッシュ \ 文字でエスケープされます。

例

入力データ

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
```



```

"delivery": {
  "servicegroup": 1000,
  "description": "Next day delivery in local town",
  "active": true,
  "price": 5
},
"bookbinding": {
  "servicegroup": 1001,
  "description": "Printing and assembling book in A5 format",
  "active": true,
  "price": 154.99
},
"restoration": {
  "servicegroup": 1002,
  "description": "Various restoration methods",
  "active": false,
  "methods": [
    {
      "description": "Chemical cleaning",
      "price": 46
    },
    {
      "description": "Pressing pages damaged by moisture",
      "price": 24.5
    },
    {
      "description": "Rebinding torn book",
      "price": 99.49
    }
  ]
}
]
}
},
"filters": {
  "price": 10,
  "category": "fiction",
  "no filters": "no \"filters\""
},
"closed message": "Store is closed",
"tags": [
  "a",
  "b",
  "c",
  "d",
  "e"
]
}
}

```

JSONPath	タイプ	結果	コメント
\$.filters.price	definite	10	
\$.filters.category	definite	fiction	
\$.filters['no filters']	definite	no "filters"	
\$.filters	definite	{ "price": 10, "category": "fiction", "no filters": "no \"filters\"" }	
\$.books[1].title	definite	Sword of Honour	
\$.books[-1].author	definite	J. R. R. Tolkien	
\$.books.length	definite	4	
\$.tags[:]	indefinite	["a", "b", "c", "d", "e"]	
\$.tags[2:]	indefinite	["c", "d", "e"]	

JSONPath	タイプ	結果	コメント
<code>\$.tags[:3]</code>	indefinite	<code>["a", "b", "c"]</code>	
<code>\$.tags[1:4]</code>	indefinite	<code>["b", "c", "d"]</code>	
<code>\$.tags[-2:]</code>	indefinite	<code>["d", "e"]</code>	
<code>\$.tags[: -3]</code>	indefinite	<code>["a", "b"]</code>	
<code>\$.tags[: -3].length()</code>	indefinite	<code>2</code>	
<code>\$.books[0, 2].title</code>	indefinite	<code>["Sayings of the Century", "Moby Dick"]</code>	
<code>\$.books[1]['author', 'title']</code>	indefinite	<code>["Evelyn Waugh", "Sword of Honour"]</code>	
<code>\$.id</code>	indefinite	<code>[1, 2, 3, 4]</code>	
<code>\$.services.price</code>	indefinite	<code>[5, 154.99, 46, 24.5, 99.49]</code>	
<code>\$.books[?(@.id == 4 - 0.4 * 5)].title</code>	indefinite	<code>["Sword of Honour"]</code>	このクエリは、クエリで算術演算を使用できることを示しています。もちろん、このクエリは <code>\$.books[?(@.id == 2)].title</code> に簡略化できます。
<code>\$.books[?(@.id == 2 \\ \\ @.id == 4)].title</code>	indefinite	<code>["Sword of Honour", "The Lord of the Rings"]</code>	
<code>\$.books[?(!(@.id == 2))].title</code>	indefinite	<code>["Sayings of the Century", "Moby Dick", "The Lord of the Rings"]</code>	
<code>\$.books[?(@.id != 2)].title</code>	indefinite	<code>["Sayings of the Century", "Moby Dick", "The Lord of the Rings"]</code>	
<code>\$.books[?(@.title =~ " of ")].title</code>	indefinite	<code>["Sayings of the Century", "Sword of Honour", "The Lord of the Rings"]</code>	
<code>\$.books[?(@.price > 12.99)].title</code>	indefinite	<code>["The Lord of the Rings"]</code>	
<code>\$.books[?(@.author > "Herman Melville")].title</code>	indefinite	<code>["Sayings of the Century", "The Lord of the Rings"]</code>	
<code>\$.books[?(@.price > \$.filters.price)].title</code>	indefinite	<code>["Sword of Honour", "The Lord of the Rings"]</code>	
<code>\$.books[?(@.category == "Sword of Honour", "Moby Dick", "The Lord of the Rings")].title</code>	indefinite	<code>["Sword of Honour", "Moby Dick", "The Lord of the Rings"]</code>	

JSONPath	タイプ	結果	コメント
<code>\$..[?(@.id)]</code>	indefinite	[{ "category": "reference", "author": "Nigel Rees", "title": "Sayings of the Century", "price": 8.95, "id": 1 }, { "category": "fiction", "author": "Evelyn Waugh", "title": "Sword of Honour", "price": 12.99, "id": 2 }, { "category": "fiction", "author": "Herman Melville", "title": "Moby Dick", "isbn": "0-553-21311-3", "price": 8.99, "id": 3 }, { "category": "fiction", "author": "J. R. R. Tolkien", "title": "The Lord of the Rings", "isbn": "0-395-19395-8", "price": 22.99, "id": 4 }]	
<code>\$.services..[?(@.id > 50)].description</code>	indefinite	["Printing and assembling book in A5 format", "Rebinding torn book"]	
<code>\$..id.length()</code>	definite	4	
<code>\$.books[?(@.id == 2)].title.first()</code>	indefinite	Sword of Honour	
<code>\$.tags.first().length()</code>	definite	5	<code>\$.tags</code> は不定パスなので、一致した要素の配列を返します - ["a", "b", "c", "d", "e"], <code>first()</code> は最初の要素 - ["a", "b", "c", "d", "e"] そして最後に <code>length()</code> はその長さ - 5 を計算します。
<code>\$.books[*].price.min()</code>	definite	8.95	
<code>\$.price.max()</code>	definite	154.99	
<code>\$.books[?(@.category == "fiction")].price.avg()</code>	definite	14.99	
<code>\$.books[?(@.category == "fiction")].title</code>	indefinite		一致のないクエリは、明確なパスと不明確なパスに対して NULL を返します。
<code>\$.services[?(@.servicegroup == "true")]</code>	definite	[{"id": 1001, "servicegroup": "true"}]	ブール値の比較では、テキスト定数を使用する必要があります。
<code>\$.services[?(@.servicegroup == "false")]</code>	definite	[{"id": 1002, "servicegroup": "false"}]	ブール値の比較では、テキスト定数を使用する必要があります。
<code>\$.services[?(@.servicegroup == "true")]~.first()</code>	definite	[{"id": 1001, "servicegroup": "true"}]	

JSONPath における LLD マクロ値からの特殊文字のエスケープ

JSONPath のプリプロセスでローレベルディスカバリーマクロが使用され、
 その値が解決されるとき、以下の特殊文字のエスケープのルールが適用されます。

- バックスラッシュ (\) とダブルクォート (") 文字のみがエスケープ対象です。

- 解決されたマクロ値がこれらの文字を含む場合、各文字はバックスラッシュでエスケープされます。

- これらの文字がすでにバックスラッシュでエスケープされている場合は、エスケープとはみなされず、
 バックスラッシュと次の特殊文字の両方が再度エスケープされます。

例:

JSONPath	LLD マクロの値	置換後
<code>\$.[?(@.value == "{#MACRO}")]</code>	special "value"	<code>\$.[?(@.value == "special \"value\"")]</code>
	c:\temp	<code>\$.[?(@.value == "c:\\temp")]</code>
	a\\b	<code>\$.[?(@.value == "a\\\\"b")]</code>

特殊文字を含むマクロを式で使用する場合は、二重引用符で囲む必要があります:

JSONPath	LLD マクロの値	置換後	結果
<code>\$.[?(@.value == "{#MACRO}")]</code>	special "value"	<code>\$.[?(@.value == "special \"value\"")]</code>	OK
<code>\$.[?(@.value == {#MACRO})]</code>		<code>\$.[?(@.value == special \"value\"")]</code>	Bad JSONPath expression

特殊文字を含むマクロをパスで使用する場合は、角括弧 と二重引用符で囲む必要があります:

JSONPath	LLD マクロの値	置換後	結果
<code>\$.["{#MACRO}"].value</code>	c:\temp	<code>\$.["c:\\temp"].value</code>	OK
<code>\$.{#MACRO}.value</code>		<code>\$.c:\\temp.value</code>	Bad JSONPath expression

4 JavaScript のプリプロセッシング

概要

ここでは、JavaScript によるプリプロセッシングについて詳しく説明します。

JavaScript のプリプロセッシング

JavaScript のプリプロセッシングは、JavaScript の関数を 1 つのパラメータ 'value' とユーザー提供の関数本体で呼び出すことで行われます。プリプロセッシングの結果は、この関数から返される値です。例えば、華氏から摂氏への変換を行う場合、ユーザーは次のように入力する必要があります。

```
return (value - 32) * 5 / 9
```

JavaScript のプリプロセスパラメータは、server で JavaScript 関数にラップされます。

```
function (value)
{
  return (value - 32) * 5 / 9
}
```

入力パラメータ 'value' は常に文字列として渡されます。戻り値はいくつかの例外とともに
 自動的に ToString() メソッドで文字列に変換されます (失敗した場合は、エラーが文字列として返されます)

- 未定義の値を返すとエラーになります。
- null 値を返すと、入力値が破棄されます。
 Custom on fail' アクションの 'Discard value' プリプロセスのようなものです。

エラーは、値/オブジェクト (通常、文字列または Error オブジェクト) を投げることで返されます。

例:

```
if (value == 0)
  throw "Zero input value"
return 1/value
```

各スクリプトには 10 秒間の実行タイムアウトが設定されており、超過するとエラーを返します。
 (タイムアウトが発生するまでには、もっと時間がかかる場合があります)
 64 メガバイトのヒープ制限があります。

JavaScript のプリプロセスステップのバイトコードはキャッシュされ、次のステップ適用時に再利用されます。
 アイテムのプリプロセスステップを変更すると、キャッシュされたスクリプトはリセットされ、後で再コンパイルされます。

連続したランタイムエラー (3 回連続) は、あるスクリプトが次のスクリプトの実行環境を破壊する可能性を
 軽減するために、エンジンが再初期化されます。この動作は DebugLevel4 以上でログに記録されます。

JavaScript の前処理は、Duktape(<https://duktape.org/>) JavaScript エンジンで実装されています。

こちらをご覧ください:
 [Additional JavaScript objects and globalfunctions](#)

スクリプトでマクロを使用する

JavaScript のコードの中で、ユーザーマクロを使用することができます。
 スクリプトにユーザーマクロが含まれている場合、これらのマクロは、特定のプリプロセスを実行する前に、
 server / proxy によって解決されます。注意点としてはフロントエンドで前処理ステップをテストする場合、
 マクロの値は引き出されないので、手動で入力する必要があります。

Note:

マクロがその値で置き換えられるとき、コンテキストは無視されます。
 マクロの値はそのままコードに挿入されるため、マクロの値を挿入する前にエスケープを追加することはできません。
 このため、場合によっては JavaScript のエラーを引き起こす可能性があることをご承知おきください。

以下の例では、受信した値が `{${THRESHOLD}}` マクロの値を超えた場合、代わりに閾値 (存在する場合) が返されます。

```
var threshold = '{${THRESHOLD}}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

追加の JavaScript オブジェクト

概要

本セクションでは、Duktape で実装された JavaScript の Zabbix への追加と、サポートされるグローバル JavaScript 関数について説明します。

組込オブジェクト

Zabbix

Zabbix オブジェクトは Zabbix の内部機能とのインタラクションを提供します。

メソッド	説明
<code>log(loglevel, message)</code>	Writes <message> into Zabbix log using <loglevel> log level (see configuration file DebugLevel parameter).

例:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

以下のエイリアスを使用することができます:

エイリアス	~へのエイリアス
<code>console.log(object)</code>	<code>Zabbix.log(4, JSON.stringify(object))</code>
<code>console.warn(object)</code>	<code>Zabbix.log(3, JSON.stringify(object))</code>
<code>console.error(object)</code>	<code>Zabbix.log(2, JSON.stringify(object))</code>

メソッド	説明
<code>sleep(delay)</code>	Delay JavaScript execution by delay milliseconds.

例 (実行を 15 秒遅らせる) :

```
Zabbix.sleep(15000)
```

HttpRequest

このオブジェクトは、単純な HttpRequest を行うための cURL ハンドルをカプセル化したものです。エラーは例外として投げられません

Attention:

HttpRequest は Zabbix 5.4 から採用された新しい名称です。以前は CurlHttpRequest と呼ばれていました。Zabbix 5.4 からメソッド名も変更されました。旧オブジェクト/メソッド名は非推奨となり、Zabbix 6.0 以降サポートが終了します。

メソッド	説明
<code>addHeader(name, value)</code>	Adds HTTP header field. This field is used for all following requests until cleared with the <code>clearHeader()</code> method.
<code>clearHeader()</code>	Clears HTTP header. If no header fields are set, HttpRequest will set Content-Type to application/json if the data being posted is JSON-formatted; text/plain otherwise.
<code>connect(url)</code>	Sends HTTP CONNECT request to the URL and returns the response.
<code>customRequest(method, url, data)</code>	Allows to specify any HTTP method in the first parameter. Sends the method request to the URL with optional data payload and returns the response.
<code>delete(url, data)</code>	Sends HTTP DELETE request to the URL with optional data payload and returns the response.
<code>getHeaders(<asArray>)</code>	Returns the object of received HTTP header fields. The <code>asArray</code> parameter may be set to "true" (e.g. <code>getHeaders(true)</code>), "false" or be undefined. If set to "true" the received HTTP header field values will be returned as arrays; this should be used to retrieve the field values of multiple same-name headers. If not set or set to "false", the received HTTP header field values will be returned as strings.
<code>get(url, data)</code>	Sends HTTP GET request to the URL with optional data payload and returns the response.
<code>head(url)</code>	Sends HTTP HEAD request to the URL and returns the response.
<code>options(url)</code>	Sends HTTP OPTIONS request to the URL and returns the response.
<code>patch(url, data)</code>	Sends HTTP PATCH request to the URL with optional data payload and returns the response.
<code>put(url, data)</code>	Sends HTTP PUT request to the URL with optional data payload and returns the response.
<code>post(url, data)</code>	Sends HTTP POST request to the URL with optional data payload and returns the response.
<code>getStatus()</code>	Returns the status code of the last HTTP request.
<code>setProxy(proxy)</code>	Sets HTTP proxy to "proxy" value. If this parameter is empty then no proxy is used.
<code>setHttpAuth(bitmask, username, password)</code>	Sets enabled HTTP authentication methods (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) in the 'bitmask' parameter. The HTTPAUTH_NONE flag allows to disable HTTP authentication. Examples: <code>request.setHttpAuth(HTTPAUTH_NTLM HTTPAUTH_BASIC, username, password)</code> <code>request.setHttpAuth(HTTPAUTH_NONE)</code>
<code>trace(url, data)</code>	Sends HTTP TRACE request to the URL with optional data payload and returns the response.

例:

```
try {
  Zabbix.log(4, 'jira webhook script value='+value);

  var result = {
    'tags': {
      'endpoint': 'jira'
```

```

    }
  },
  params = JSON.parse(value),
  req = new HttpRequest(),
  fields = {},
  resp;

req.addHeader('Content-Type: application/json');
req.addHeader('Authorization: Basic '+params.authentication);

fields.summary = params.summary;
fields.description = params.description;
fields.project = {"key": params.project_key};
fields.issuetype = {"id": params.issue_id};
resp = req.post('https://tsupport.zabbix.lan/rest/api/2/issue/',
  JSON.stringify({"fields": fields})
);

if (req.getStatus() != 201) {
  throw 'Response code: '+req.getStatus();
}

resp = JSON.parse(resp);
result.tags.issue_id = resp.id;
result.tags.issue_key = resp.key;
} catch (error) {
  Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
  Zabbix.log(4, 'jira issue creation failed : '+error);

  result = {};
}

return JSON.stringify(result);

```

XML

XML オブジェクトは、item 内の XML データの処理と、ローレベルディスカバリープリプロセッシングおよびウェブフックを可能にします。

Attention:

XML オブジェクトを利用するためには、libxml2 がサポートされた状態で server / proxy がコンパイルされている必要があります。

メソッド	説明
XML.query(data, expression)	Retrieves node content using XPath. Returns null if node is not found. expression - an XPath expression; data - XML data as a string.
XML.toJson(data)	Converts data in XML format to JSON.
XML.fromJson(object)	Converts data in JSON format to XML.

例:

Input:

```

<menu>
  <food type = "breakfast">
    <name>Chocolate</name>
    <price>$5.95</price>
    <description></description>
    <calories>650</calories>
  </food>
</menu>

```

Output:

```
{
  "menu": {
    "food": {
      "@type": "breakfast",
      "name": "Chocolate",
      "price": "$5.95",
      "description": null,
      "calories": "650"
    }
  }
}
```

シリアライズ規則

XML から JSON への変換は、以下の規則に従って処理されます (JSON から XML への変換は、逆の規則が適用されます)。

1. XML の属性は、名前の先頭に '@' を付加したキーに変換されます。

例:

Input:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Output:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. セルフクロージングの要素 (<foo/>) は、'null' 値を持つものとして変換されます。

例:

Input:

```
<xml>
  <foo/>
</xml>
```

Output:

```
{
  "xml": {
    "foo": null
  }
}
```

3. 空の属性 ("" の値) は、空文字列 ("" の値) として変換されます。

例:

Input:

```
<xml>
  <foo bar="" />
</xml>
```

Output:

```
{
  "xml": {
```



```
    "foo": {
      "@bar": ""
    }
  }
}
```

4. 同じ要素名を持つ複数の子ノードは、値の配列を値とする 1 つのキーに変換されます。

例:

Input:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Output:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. text 要素に属性がなく、子要素もない場合は、文字列として変換されます。

例:

Input:

```
<xml>
  <foo>BAZ</foo>
</xml>
```

Output:

```
{
  "xml": {
    "foo": "BAZ"
  }
}
```

6. text 要素に子がなく、属性がある場合、text の内容はキーが 'text'、値が content の要素に変換され、
 属性はシリアライズ規則 1 のとおりに変換されます。

例:

Input:

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Output:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

グローバル JavaScript 関数

Duktape でグローバル JavaScript の機能を追加実装しました:

- btoa(string) - 文字列を base64 文字列にエンコードする
- atob(base64_string) - base64 文字列をデコードする

```
try {
  b64 = btoa("utf8 string");
  utf8 = atob(b64);
}
catch (error) {
  return {'error.name' : error.name, 'error.message' : error.message}
}
```

- md5(string) - 文字列の MD5 ハッシュを計算する
- sha256(string) - 文字列の SHA256 ハッシュを計算します。

5 CSV から JSON へのプリプロセス

概要

このプリプロセスで、CSV ファイルのデータを JSON 形式に変換することが可能です。サポートしているのは

- items (item prototypes)
- low-level discovery rules

設定

CSV から JSON へのプリプロセスステップを設定する場合:

- **item** の Preprocessing タブ/ **discovery rule** の設定にアクセスします。
- Add をクリックします。
- CSV to JSON オプションを選択します。

Preprocessing steps	Name	Parameters
1:	CSV to JSON	<input type="text" value=","/> <input type="text" value=""/> <input checked="" type="checkbox"/> With header
Custom on fail: <input type="button" value="Discard value"/> <input type="button" value="Set value to"/> <input type="button" value="Set error to"/> <input type="button" value="error messa"/>		

[Add](#)

1 番目のパラメータで、カスタムデリミタを設定することができます。CSV 入力の最初の行が "Sep=" で始まり、
 UTF-8 文字が 1 つ続く場合、最初のパラメータが設定されていない場合、その文字がデリミタとして使用されることに注意ください。
 1 番目のパラメータが設定されておらず、かつ "Sep=" 行からデリミタが取得できない場合は、カンマがセパレーターとして使用されます。

オプションの第 2 パラメータには、引用記号を設定することができます。

With header row チェックボックスがマークされている場合、ヘッダ行の値は列名として解釈されます
 (詳しくは [Header processing](#) を参照してください)。

Custom on fail チェックボックスがマークされている場合、プリプロセスが失敗しても、その item はサポートされなくなることはありません。
 さらに、カスタムエラー処理オプションを設定することができます:
 値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する。

ヘッダー処理

CSV ファイルのヘッダ行は、2 種類の方法で処理することができます。

- With header row チェックボックスがマークされている場合、ヘッダー行の値は列名として解釈されます。この場合、列名は一意でなければならず、データ行はヘッダー行より多くの列名を含んではなりません。
- With header row チェックボックスがマークされていない場合 - ヘッダーラインはデータとして解釈されます。列名は自動的に生成されます。(1,2,3,4...)

CSV ファイルの例:

```
Nr,Item name,Key,Qty
1,active agent item,agent.hostname,33
"2","passive agent item","agent.version","44"
3,"active,passive agent items",agent.ping,55
```

Note:

入力の引用フィールド内の引用文字は、その前に別の引用文字を置くことでエスケープしなければならない。

ヘッダー行の処理

ヘッダ行が検出される場合の JSON 出力:

```
[
  {
    "Nr": "1",
    "Item name": "active agent item",
    "Key": "agent.hostname",
    "Qty": "33"
  },
  {
    "Nr": "2",
    "Item name": "passive agent item",
    "Key": "agent.version",
    "Qty": "44"
  },
  {
    "Nr": "3",
    "Item name": "active,passive agent items",
    "Key": "agent.ping",
    "Qty": "55"
  }
]
```

ヘッダーライン処理なし

ヘッダ行が検出できない場合の JSON 出力:

```
[
  {
    "1": "Nr",
    "2": "Item name",
    "3": "Key",
    "4": "Qty"
  },
  {
    "1": "1",
    "2": "active agent item",
    "3": "agent.hostname",
    "4": "33"
  },
  {
    "1": "2",
    "2": "passive agent item",
    "3": "agent.version",
    "4": "44"
  },
  {
    "1": "3",
    "2": "active,passive agent items",
    "3": "agent.ping",
    "4": "55"
  }
]
```

3 アイテムのタイプ

概要

アイテムのタイプでは、システムからデータを取得するためのさまざまな方法をカバーします。
各アイテムのタイプには、サ

ポートされるアイテムキーと必須パラメータのセットがあります。

現在、Zabbix が提供しているアイテムタイプは以下の通りです。

- Zabbix エージェント
- SNMP エージェント
- SNMP トラップ
- IPMI
- シンプルチェック
 - VMware 監視
- ログ監視
- 計算アイテム
 - アグリゲート
- Zabbix インターナル
- SSH
- Telnet
- 外部チェック
- Zabbix トラッパー
- JMX
- データベースモニタ
- 依存アイテム
- HTTP
- Prometheus
- スクリプト

すべてのアイテムのタイプの詳細は、このセクションのサブページに含まれています。
アイテムのタイプはデータ収集のための多くのオプションを提供しますが、**ユーザーパラメーター** または **ローダブルモジュール**を使用することもできます。

一部のチェックは Zabbix サーバーだけで実行されます (エージェントレス監視)
 Zabbix エージェント、または Zabbix Java gateway (JMX 監視) が必要なものもあります。

Attention:

特定のアイテムのタイプで特定のインターフェイスが必要な場合 (IPMI チェックにはホスト上の IPMI インターフェイスが必要のように) そのインターフェイスはホストの定義に存在する必要があります。

ホストの定義には複数のインターフェイスを設定することができます (Zabbix エージェント、SNMP エージェント、JMX、IPMI)
アイテムが複数のインターフェイスを使用できる場合、利用可能なホストインターフェイスを検索します。
 エージェント → SNMP → JMX → IPMI の順で検索され、最初にリンクされる適切なインターフェイスが表示されます。

テキストを返すすべてのアイテム (文字、ログ、テキストタイプの情報) は空白のみを返し、(該当する場合) 戻り値を空の文字列に設定できます (2.0 以降サポート)。

1 Zabbix エージェント

概要

これらのチェックは、Zabbix agent との通信を利用してデータを収集します。

agent チェックには **passive and active** (パッシブとアクティブ) の 2 種類あります。
アイテムの設定時に、必要なタイプを選択することができます。

- Zabbix agent - パッシブチェック用
- Zabbix agent(active) - アクティブチェック用

サポートされている item キー

この表は、Zabbix agent item に使用できる item キーの詳細を示します。

こちらをご参照ください。

- プラットフォームがサポートするアイテム
- Zabbix エージェント 2 でサポートされているアイテムキー
- Windows エージェント固有の項目キー
- Windows エージェントアイテムの最小アクセス許可レベル

必須パラメータとオプションパラメータ

角括弧のないパラメータは必須です。< > でマークされたパラメータはオプションです。

コマンドラインユーティリティでの使用方法

コマンドラインから `zabbix_agentd` または `zabbix_get` を使用して `item` キーをテストまたは使用する場合、
シェルの構文も考慮する必要があります。

例えば、キーの特定のパラメータを二重引用符 (") で囲む必要がある場合、二重引用符を明示的にエスケープする必要があります。
明示的にエスケープしないと、シェルによって二重引用符が特殊文字としてトリミングされ、Zabbix ユーティリティに渡されません。

例:

```
$ zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'
```

```
$ zabbix_agentd -t 'vfs.dir.count[/var/log,,,\"file,dir\",,0]'
```

Zabbix メトリクス

アイテムキー			
説明	戻り値	パラメータ	コメント
agent.hostmetadata			
エージェント ホストのメタ データ	String		HostMetadata または HostMetadataItem パラメータの値。何も定義されていない場 合は空の文字列を返します。 Zabbix 6.0 以降でサポートされています。
agent.hostname			
エージェント ホスト名	String		戻り値: パッシブチェック - エージェント構成ファ イルの Hostname パラメータにリストされ ている最初のホストの名前 アクティブチェック - 現在のホスト名の名 前
agent.ping			
エージェント の可用性チェ ック	なし - 利用不 可 1 - 利用可能		nodata() トリガー関数を使用して、ホスト が使用可能かを確認します。
agent.variant			
Zabbix エー ジェントのバ リエーション (Zabbix エ ージェントま たは Zabbix エージェント 2)	Integer		戻り値の例: 1 - Zabbix エージェント 2 - Zabbix エージェント 2
agent.version			
Zabbix エー ジェントのバ ージョン	String		戻り値の例: 6.0.3
zabbix.stats[<ip>,<port>]			
Zabbix サー バーまたは プロキシの 内部メトリ ックのセッ トをリモ ートで返 します	JSON object	ip - リモートで照会するサーバー/プロキシ の IP/DNS/ネットワークマスクリスト (デ フォルトは 127.0.0.1) port - リモートでクエリを実行するサー バー/プロキシのポート (デフォルトは 10051)	統計リクエストは、ターゲットインスタ ンスの 'StatsAllowedIP' server/proxy パラメ ータにリストされているアドレスからのみ受 け入れられることに注意してください。 選択された一連の内部メトリックがこの アイテムによって返されます。詳しく は Zabbix 統計情報の遠隔監視 を参照して ください。
zabbix.stats[<ip>,<port>,queue,<from>,<to>]			

アイテムキー

Zabbix サーバ またはリモ ートプロキシ で遅延してい る、キュー内 の監視対象ア イテムの数を 返します	JSON object	ip - リモートで照会するサーバー/プロキシ の IP/DNS/ネットワークマスクリスト (デフ ォルトは 127.0.0.1) port - リモートでクエリを実行するサー バ/プロキシのポート (デフォルトは 10051) queue - 定数 (そのまま使用) from - 最小遅延 (デフォルト 6 秒) to - 最大遅延 (デフォルトは無量大)	統計リクエストは、ターゲットインスタ ンスの 'StatsAllowedIP' server/proxy パラメ ータにリストされているアドレスからのみ受 け入れられることに注意してください。
---	-------------	--	--

脚注

¹Linux 特有の注意事項です。Zabbix agent は、ファイルシステム/proc に読み取り専用でアクセスできる必要があります。
www.grsecurity.org のカーネルパッチにより、非特権ユーザのアクセス権が制限されます。

² `vfs.dev.read[]`、`vfs.dev.write[]`: 3 時間以上アクセスされない場合、Zabbix エージェントは「古くなった」デバイス接続を終了します。
この現象は、動的にパスが変更されるデバイスがシステムに存在する場合、またはデバイスのパスが手動で
削除された場合などに発生します。また、これらの item は、3 時間以上の更新間隔を使用する場合、常に '0' を返します。

³ `vfs.dev.read[]`、`vfs.dev.write[]`: 最初のパラメータにデフォルトの `all*` が使われた場合、キーは統計の概要を返します。
`sda`、`sdb` のようなすべてのブロックデバイスとそのパーティション (`sda1`、`sda2`、`sdb3`...) のようなすべてのブロックデバイス
と、
それらのブロックデバイス/パーティションに基づく複数のデバイス (MD raid)、および論理ボリュームを含む統計情報の概要を返します。
このような場合、返される値は絶対値としてではなく、相対値 (時間的に動的) としてのみ考慮する必要があります。

SSL (HTTPS) は、agent が cURL をサポートしてコンパイルされている場合のみサポートされます。
それ以外の場合、この項目はサポートされないこととなります。

エンコーディング設定

取得したデータが破損していないことを確認するために、チェック処理に正しいエンコードを指定することができます。
チェック処理に適したエンコーディング (例: `'vfs.file.contents'`) を `encoding` パラメータで指定します。
サポートされているエンコーディングのリスト (コードページ識別子) の一覧は、`libiconv` (GNU プロジェクト) のドキュメントや、
Microsoft Windows SDK documentation for "Code Page Identifiers" を参照してください。

もし、`encoding` パラメータでエンコーディングを指定しない場合は、以下の解決方法が適用されます。

- エンコーディングが指定されていない場合 (または空文字列の場合)、UTF-8 であるとみなされ、データは「そのまま」処理されます。

- BOM 解析 - `'vfs.file.contents'`、`'vfs.file.regex'`、`'vfs.file.regmatch'` の item に適用されます。
ファイルの先頭にあるバイトオーダーマーク (BOM) を使って、正しいエンコーディングを決定しようとします。
BOM が存在しない場合、代わりに標準的なエンコーディングが適用されます。

agent item のトラブルシューティング

- パッシブ agent で使用する場合、server 設定の Timeout 値は、agent の Timeout 値より大きい必要があります。そうしなければ、server から agent への要求がタイムアウトして、item の値を取得できないかもしれません。

カーネルデータ

アイテムキー

説明	戻り値	パラメータ	コメント
kernel.maxfiles OS がサポートするオープンファイルの最大数	Integer		
kernel.maxproc OS がサポートするプロセスの最大数	Integer		
kernel.openfiles 現在開いているファイル記述子の数	Integer		このアイテムは Zabbix 6.0 以降でサポートされています

ログデータ

ログ監視に関する追加情報も参照してください。

アイテムキー

説明	戻り値	パラメータ	コメント
----	-----	-------	------

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent_dir>]

ログファイル Log
監視

file - ログファイルのフルパスとファイル名
regexp - 正規表現に必要なパターンを記述
encoding - コードページ識別子
maxlines - エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの最大改行数。このパラメータは `zabbix_agentd.conf` の 'MaxLinesPerSecond' の値を上書きします。
mode (バージョン 2.0 以降) - 使用可能な値:
 all (デフォルト), skip - 古いデータの処理をスキップします (新しく作成されたアイテムのみに影響します)
output (バージョン 2.2 以降) - オプションの出力フォーマットテンプレート。**\O** エスケープシーケンスは、テキストの一致した部分 (一致が始まる最初の文字から一致が終了する文字まで) に置き換えられますが、**\N** (N=1... 9) エスケープシーケンスは、N 番目に一致したグループ (または N がキャプチャされたグループの数を超える場合は空の文字列) に置き換えられます。
maxdelay (バージョン 3.2 以降) - 秒単位の最大遅延。タイプ: 浮動小数点。値: 0 - (デフォルト) ログファイルの行を無視しません。> 0.0 - "maxdelay" 秒以内に分析された最新の行を取得するために古い行を無視します。使用する前に **maxdelay** の注意事項を確認してください!
options (バージョン 4.4.7 以降) - 追加オプション:
 mtime-noread - 一意でないレコード。ファイルサイズが変更された場合にのみ再読み込みされます (更新時間の変更は無視されます)。(このパラメータは 5.0.2 以降では非推奨です。現在は mtime が無視されるためです。)
persistent_dir (バージョン 5.0.18、5.4.9 以降、Unix システムの zabbix_agentd のみ。Agent2 ではサポートされていません) - 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項も参照してください。

項目は **アクティブチェック** として設定する必要があります。

ファイルが見つからないか、アクセス許可が許可されていない場合、アイテムはサポートされなくなります。

output が空のままの場合 - 一致したテキストを含む行全体が返されます。'Result is TRUE' を除くすべてのグローバル正規表現タイプは、常に一致した行全体を返し、**output** パラメータは無視されることに注意してください。

output パラメータを使用したコンテンツの抽出は、エージェントで行われます。

例:

```
=> log[/var/log/syslog]
=> log[/var/log/syslog,error]
=> log[/home/zabbix/logs/logfile,,,100]
```

ログレコードから数値を抽出するための **output** パラメータの使用:

```
=> log[/app1/app.log,"task run [0-9.]+
sec, processed ([0-9.]+) records, [0-9.]+
errors",,,1] → ログレコード "2015-11-13
10:08:26 task run 6.08 sec, processed
6080 records, 0 errors" に一致し、'6080'
のみをサーバーに送信します。数値を送信
するため、この項目の "情報の種類" を "数
値 (符号なし)" に設定し、その値をグラフ
やトリガーなどで利用することができます。
```

サーバーに送信する前にログレコードを書き換えるための **output** パラメータの使用:

```
=> log[/app1/app.log,"([0-9 :-]+) task run
([0-9.]+) sec, processed ([0-9.]+) records,
([0-9.]+) errors",,,1 RECORDS: \3,
ERRORS: \4, DURATION: \2"] → ログレコー
ド "2015-11-13 10:08:26 task run 6.08 sec,
processed 6080 records, 0 errors" に一致
し、そこから変更されたレコー
ド "2015-11-13 10:08:26 RECORDS: 6080,
ERRORS: 0, DURATION: 6.08" をサーバー
に送信します。
```

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent_dir>]

アイテムキー

監視対象のログファイルで一致した行数	Integer	<p>file - ログファイルのフルパスとファイル名</p> <p>regexp - 正規表現に必要なパターンを記述</p> <p>encoding - コードページ識別子</p> <p>maxproclines - エージェントが分析する1秒あたりの新しい行の最大数 (10000 を超えることはできません)。</p> <p>zabbix_agentd.conf のデフォルト値は <code>10*MaxLinesPerSecond</code> です。</p> <p>mode - 使用可能な値: all (デフォルト), skip - 古いデータの処理をスキップします (新しく作成されたアイテムのみに影響します)</p> <p>maxdelay - 秒単位の最大遅延。タイプ: 浮動小数点値: 0 - (デフォルト) ログファイルの行を無視しません。 > 0.0 - "maxdelay" 秒以内に分析された最新の行を取得するために、古い行を無視します。使用前に maxdelay の注意事項をお読みください!</p> <p>options (バージョン 4.4.7 以降) - 追加オプション: mtime-noread - 一意でないレコード。ファイルサイズが変更された場合にのみ再読み込みされます (更新時間の変更は無視されます)。(このパラメーターは 5.0.2 以降では非推奨です。現在は mtime が無視されるためです。)</p> <p>persistent_dir (バージョン 5.0.18、5.4.9 以降、Unix システムの zabbix_agentd のみ。Agent2 ではサポートされていません) - 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項も参照してください。</p>	<p>項目は アクティブチェック として設定する必要があります</p> <p>一致する行は、エージェントによる最後のログチェック以降の新しい行でカウントされるため、アイテムの更新間隔に依存しません。</p> <p>ファイルが見つからないか、アクセスが許可されていない場合、アイテムはサポートされなくなります。</p> <p>この項目は、Windows イベントログではサポートされていません。</p> <p>Zabbix 3.2.0 以降でサポートされています。</p>
--------------------	---------	--	--

logrt[file_regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent_dir>]

ローテーションされるログファイルの監視

Log

file_regexp - ファイルの絶対パスと**正規表現**で記述したファイル名。注ファイル名だけが正規表現であること。

regexp - **正規表現**に必要なパターンを記述
encoding - コードページ識別子

maxlines - エージェントが Zabbix サーバまたはプロキシに送信する 1 秒あたりの最大改行数。このパラメータは、**zabbix_agentd.conf**の'MaxLinesPerSecond' の値を上書きします。

mode (バージョン 2.0 以降) - 使用可能な値:

all (デフォルト), skip - 古いデータの処理をスキップします (新しく作成されたアイテムのみに影響します)

output (バージョン 2.2 以降) - オプションの出力フォーマットテンプレート。**\0** エスケープシーケンスは、テキストの一致した部分 (一致が始まる最初の文字から一致が終了する文字まで) に置き換えられますが、**\N** (N=1... 9) エスケープシーケンスは、N 番目に一致したグループ (または N がキャプチャされたグループの数を超える場合は空の文字列) に置き換えられます。

maxdelay (バージョン 3.2 以降) - 秒単位の最大遅延。タイプ: 浮動小数点。値: 0 - (デフォルト) ログファイルの行を無視しません。> 0.0 - "maxdelay" 秒以内に分析された最新の行を取得するために、古い行を無視します。使用する前に**maxdelay** の注意事項をお読みください!

options (バージョン 4.0 以降。

mtime-reread, mtime-noread オプション 4.4.7 以降) - ログファイルのローテーションの種類とその他のオプション。使用可能な値:

rotate (デフォルト),

copytruncate - copytruncate は maxdelay と一緒に使用できないことに注意してください。この場合、maxdelay は 0 にするか指定しないでください。**copytruncate** メモを参照してください。

mtime-reread - 一意でないレコード、変更時間またはサイズが変更された場合に再読み込み (デフォルト),

mtime-noread - 一意でないレコード。サイズが変更された場合にのみ再読み取りされます (更新時間の変更は無視されます)

persistent_dir (バージョン 5.0.18、5.4.9 以降、Unix システムの zabbix_agentd のみ。Agent2 ではサポートされていません) - 永続ファイルを保存するディレクトリの絶対パス名。**永続ファイル**に関する追加の注意事項も参照してください。

アイテムは**アクティブチェック**として構成する必要があります。

ログローテーションは、ファイルの最終更新時刻に基づいています。

logrt は設計されていることに注意してください。1 つの現在アクティブなログファイルを処理し、他のいくつかの一致する非アクティブなファイルをローテーションします。たとえば、ディレクトリに多数のアクティブなログファイルがある場合は、各ログファイルに対して個別の logrt 項目を作成する必要があります。そうしないと、1 つの logrt アイテムがあまりにも多くのファイルを取得すると、メモリが枯渇し、監視がクラッシュする可能性があります。

output が空のままの場合 - 一致したテキストを含む行全体が返されます。'Result is TRUE' を除くすべてのグローバル正規表現タイプは、常に一致した行全体を返し、output パラメータは無視されることに注意してください。

output パラメータを使用したコンテンツ抽出はエージェントで行われます。
例:

```
=> logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",",,100] -> "logfile1" のようなファイルに一致します ("logfile1" には一致しません)
=> logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] -> "logfile_abc_1" や "logfile_001" などのファイルからデータを収集します。
```

ログレコードから数値を抽出するための **output** パラメータの使用:

```
=> logrt[/app1/^test.*log$, "task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",",,\1] -> ログレコード "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" に一致し、'6080' のみをサーバーに送信します。数値を送信するため、この項目の "情報の種類" を "数値 (符号なし)" に設定し、その値をグラフやトリガーなどで利用することができます。
```

サーバーに送信する前にログレコードを書き換えるための **output** パラメータの使用:

```
=> logrt[/app1/^test.*log$, "([0-9.:-]+) task run ([0-9.:-]+) sec, processed ([0-9.:-]+) records, ([0-9.:-]+) errors",",,\1 RECORDS: \3, ERRORS: \4, DURATION: \2"] -> ログレコード "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" に一致し、変更されたレコード "2015-11-13 10:08:26 RECORDS: 6080, ERRORS: 0, DURATION: 6.08" をサーバーに送信します。
```

アイテムキー

logrt.count[file_regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent_dir>]

ローテーションされる監視対象ログファイル内の一致する行数

Integer

file_regexp - ファイルへの絶対パスと**正規表現** ファイル名のパターンを記述

regexp - **正規表現**に必要なパターンを記述

encoding - コードページ**識別子**

maxproclines - エージェントが分析する1秒あたりの新しい行の最大数 (10000 を超えることはできません)。

zabbix_agentd.conf のデフォルト値は 10*‘MaxLinesPerSecond’ です。

mode - 使用可能な値:

all (デフォルト), skip - 古いデータの処理をスキップします (新しく作成されたアイテムのみに影響します)

maxdelay - 秒単位の最大遅延。タイプ: フロート。値: 0 - (デフォルト) ログファイルの行を無視しません。> 0.0 - 「maxdelay」秒以内に分析された最新の行を取得するために、古い行を無視します。使用前に**maxdelay** の注意事項をお読みください!

options (バージョン 4.0 以降; 4.4.7 以降の mtime-reread、mtime-noread オプション) - ログファイルのローテーションのタイプとその他のオプション。使用可能な値: rotate (デフォルト),

copytruncate - copytruncate は maxdelay と一緒に使用できないことに注意してください。この場合、maxdelay は 0 にするか指定しないでください。copytruncate メモを参照,

mtime-reread - 一意でないレコード、更新時刻またはサイズが変更された場合に再読み込み (デフォルト),

mtime-noread - 一意でないレコード。サイズが変更された場合にのみ再読み取りされず (更新時間の変更は無視されます)

persistent_dir (バージョン 5.0.18、5.4.9 以降、Unix システムの zabbix_agentd のみ。Agent2 ではサポートされていません) - 永続ファイルを保存するディレクトリの絶対パス名。**永続ファイル**に関する追加の注意事項も参照してください。

項目は**アクティブチェック**として設定する必要があります

一致する行は、エージェントによる最後のログチェック以降の新しい行でカウントされるため、アイテムの更新間隔に依存しません。

ログのローテーションは、ファイルの最終変更時刻に基づいています。

この項目は、Windows イベントログではサポートされていません。

Zabbix 3.2.0 以降でサポートされています。

Modbus データ

アイテムキー

説明	戻り値	パラメータ	コメント
modbus.get [endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]			

アイテムキー

Modbus データを読み取ります	JSON object	endpoint - protocol://connection_string として定義されたエンドポイント slave id - スレーブ ID function - Modbus 機能 address - 最初のレジストリ、コイル、または入力アドレス count - 読み取るレコード数 type - データタイプ endianness - エンディアンの設定 offset - 'address' から始まるレジスタの数。結果は破棄されます。	Zabbix 5.2.0 以降でサポートされています。
-------------------	-------------	---	-----------------------------

パラメータの[詳細説明](#)を参照してください。

ネットワークデータ

アイテムキー

説明	戻り値	パラメータ	コメント
net.dns [<ip>,name,<type>,<timeout>,<count>,<protocol>]			
DNS サービスが稼働しているかどうかを確認します	0 - DNS がダウンしていません (サーバーが応答しなかったか、DNS の解決に失敗しました) 1 - DNS がアップしていません	ip - DNS サーバーの IP アドレス (デフォルトの DNS サーバーの場合は空のままにします。Windows では Zabbix エージェント 2 でサポートされています。Zabbix エージェントでは無視されます) name - クエリする DNS 名 * type - クエリされるレコードタイプ (デフォルトは SOA) timeout (Zabbix エージェント 2 バージョン 6.0.1 以降を使用しない限り、 Windows では無視されます) - 秒単位のリクエストのタイムアウト (デフォルトは 1 秒) count (Zabbix エージェント 2 バージョン 6.0.1 以降を使用しない限り、 Windows では無視されます) - リクエストの試行回数 (デフォルトは 2) protocol ** (バージョン 3.0 以降) - DNS クエリの実行に使用されるプロトコル: udp (デフォルト) または tcp	例: => net.dns[8.8.8.8,example.com,MX,2,1] type の可能な値は次のとおりです。 ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA、NULL、WKS (Windows の Zabbix エージェント、すべての OS の Zabbix エージェント 2 ではサポートされていません)、HINFO、MINFO、TXT、AAAA、SRV 国際化されたドメイン名はサポートされていません。代わりに IDNA でエンコードされた名前を使用してください。 SRV レコードタイプは、Zabbix 1.8.6 (Unix) および 2.0.0 (Windows) 以降でサポートされています。 Zabbix 2.0 以前の命名 (引き続きサポートされています): net.tcp.dns
net.dns.record [<ip>,name,<type>,<timeout>,<count>,<protocol>]			

アイテムキー

DNS クエリを実行します	必要な種類の情報を含む文字列	ip - DNS サーバーの IP アドレス (デフォルトの DNS サーバーの場合は空のままにします。Zabbix エージェント 2 バージョン 6.0.1 以降を使用しない限り、Windows では無視されます) name - クエリする DNS 名 type - クエリするレコードタイプ (デフォルトは SOA) timeout (Zabbix エージェント 2 バージョン 6.0.1 以降を使用しない限り、Windows では無視されます) - リクエストのタイムアウト秒単位 (デフォルトは 1 秒) count (Zabbix エージェント 2 バージョン 6.0.1 以降を使用しない限り、Windows では無視されます) - リクエストの試行回数 (デフォルトは 2) ** protocol** (バージョン 3.0 以降) - DNS クエリの実行に使用されるプロトコル: udp (デフォルト) または tcp	例: => net.dns.record[8.8.8.8,example.com,MX,2,1] type の可能な値は次のとおりです: ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, * WKS* (Windows の Zabbix エージェント、すべての OS の Zabbix エージェント 2 ではサポートされていません)、HINFO、MINFO、TXT、AAAA、SRV 国際化されたドメイン名はサポートされていません代わりに IDNA でエンコードされた名前を使用してください。 SRV レコードタイプは、Zabbix 1.8.6 (Unix) および 2.0.0 (Windows) 以降でサポートされています。 Zabbix 2.0 より前の命名 (引き続きサポートされています): net.tcp.dns. クエリ
net.if.collisions [if]	ウィンドウ外 Integer コリジョン数	if - ネットワークインターフェース名	
net.if.discovery	ネットワーク JSON オブジェクト インターフェースのリスト。 ローレベルデイスカバリーに使用されます。		Zabbix 2.0 以降でサポートされています。 FreeBSD、OpenBSD、NetBSD は Zabbix 2.2 以降で対応。 一部の Windows バージョン (Server 2008 など) では、インターフェース名で非 ASCII 文字をサポートするために、最新の更新プログラムをインストールする必要がある場合があります。
net.if.in [if,<mode>]	ネットワーク Integer インターフェースの受信トラフィック統計	if - ネットワークインターフェース名 (Unix); ネットワークインターフェースの完全な説明または IPv4 アドレス。または、中かっこで囲まれている場合は、ネットワークインターフェース GUID (Windows) mode - 使用可能な値: bytes - バイト数 (デフォルト) packets - パケット数 errors - エラーの数 dropped - ドロップされたパケットの数 overruns (fifo) - FIFO バッファエラーの数 frame - パケットフレーミングエラーの数 compressed - デバイスドライバーによって送信または受信された圧縮パケットの数 multicast - デバイスドライバーによって受信されたマルチキャストフレームの数	Windows では、アイテムは 64 ビットカウンターから値を取得します (利用可能な場合)。64 ビットインターフェース統計カウンターは、Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが使用できない場合、エージェントは 32 ビットカウンターを使用します。 Zabbix 1.8.6 以降、Windows でのマルチバイトインターフェース名がサポートされています。 例: => net.if.in[eth0,errors] => net.if.in[eth0] net.if.discovery または net.if.list アイテムを使用して、Windows でネットワークインターフェースの説明を取得できます。 このキーを Change per second 前処理ステップで使用して、1 秒あたりのバイト数の統計を取得できます。
net.if.out [if,<mode>]			

アイテムキー

ネットワーク インターフェ ースの発信ト ラフィック統 計	Integer	if - ネットワークインターフェイス名 (Unix); ネットワークインターフェイスの完全な説明または IPv4 アドレス。または、中かっこで囲まれている場合は、ネットワークインターフェイス GUID (Windows) mode - 使用可能な値: bytes - バイト数 (デフォルト) packets - パケット数 errors - エラーの数 dropped - ドロップされたパケットの数 overruns (fifo) - FIFO バッファエラーの数 collisions (colls) - 検出された衝突の数インターフェイス carrier - デバイスドライバによって検出されたキャリア損失の数 compressed - デバイスドライバによって送信された圧縮パケットの数	Windows では、アイテムは 64 ビットカウンターから値を取得します (利用可能な場合)。64 ビットインターフェイス統計カウンターは、Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが使用できない場合、エージェントは 32 ビットカウンターを使用します。 Zabbix エージェント 1.8.6 バージョン以降、Windows でのマルチバイトインターフェイス名がサポートされています。 例: => net.if.out[eth0,errors] => net.if.out[eth0] net.if.discovery または net.if.list アイテムを使用して、Windows でネットワークインターフェイスの説明を取得できます。 このキーを Change per second 前処理ステップで使用して、1 秒あたりのバイト数の統計を取得できます。
net.if.total [if,<mode>] ネットワーク インターフェ ースの着信お よび発信トラ フィック統計 の合計	Integer	if - ネットワークインターフェイス名 (Unix); ネットワークインターフェイスの完全な説明または IPv4 アドレス。または、中かっこで囲まれている場合は、ネットワークインターフェイス GUID (Windows) mode - 使用可能な値: bytes - バイト数 (デフォルト) packets - パケット数 errors - エラーの数 dropped - ドロップされたパケットの数 overruns (fifo) - FIFO バッファエラーの数 compressed - デバイスドライバーによって送受信された圧縮パケットの数	Windows では、アイテムは 64 ビットカウンターから値を取得します (利用可能な場合)。64 ビットインターフェイス統計カウンターは、Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが使用できない場合、エージェントは 32 ビットカウンターを使用します。 例: => net.if.total[eth0,errors] => net.if.total[eth0] net.if.discovery または net.if.list アイテムを使用して、Windows でネットワークインターフェイスの説明を取得できます。 このキーを Change per second 前処理ステップで使用して、1 秒あたりのバイト数の統計を取得できます。 ドロップされたパケットは、net.if.in と net.if.out の両方がプラットフォーム上でドロップされたパケットに対して機能する場合にのみサポートされることに注意してください。
net.tcp.listen [port] この TCP ポー トが LISTEN 状態かどうか を確認します	0 - LISTEN 状態ではありません 1 - LISTEN 状態です	port - TCP ポート番号	例: => net.tcp.listen[80]

Zabbix 1.8.4
以降でサポー
トされている
Linux の場合

Zabbix 3.0.0
以降、Linux カーネル 2.6.14
以降では、
TCP ソケット
のリスニング
に関する情報
は、可能であ
ればカーネル
の NETLINK イ
ンターフェイ
スから取得さ
れます。それ
以外の場合、
情報は
/proc/net/tcp
および
/proc/net/tcp6
ファイルから
取得されます。

net.tcp.port[<ip>,<port>]

指定したポートに TCP 接続できるかどうかをチェックします	0 - 接続できません 1 - 接続できます	ip - IP または DNS 名 (デフォルトは 127.0.0.1) ** ポート ** - ポート番号
--------------------------------	---------------------------	---

例:
=> net.tcp.port[,80] → ポート 80 で実行されている Web サーバーの可用性をテストするために使用できます。

単純な TCP パフォーマンステストには、net.tcp.service.perf[tcp,<ip>,<port>] を使用します。

これらのチェックにより、システムデーモンのログファイルに追加のメッセージが記録される場合があることに注意してください (通常、SMTP および SSH セッションがログに記録されます)

net.tcp.service[service,<ip>,<port>]

アイテムキー

サービスが実行中であり、TCP 接続を受け入れているかどうかを確認します。

0 - サービスがダウンしています
1 - サービスが実行されています

service - 次のいずれか:
ssh、ldap、smtp、ftp、http、pop、nntp、imap、tcp、https、telnet (詳細を参照)
ip - IP アドレス (デフォルトは 127.0.0.1)
port - ポート番号 (デフォルトでは、標準のサービスポート番号が使用されます)

例:
=> net.tcp.service[ftp,,45] → TCP ポート 45 で FTP サーバーの可用性をテストするために使用できます。

これらのチェックにより、システムデーモンのログファイルに追加のメッセージが記録される場合があることに注意してください (通常、SMTP および SSH セッションがログに記録されます)。

暗号化されたプロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには net.tcp.port を使用してください。

Windows での LDAP と HTTPS のチェックは、Zabbix エージェント 2 でのみサポートされています。

telnet チェックでは、ログインプロンプト (末尾の「:」) が検索されることに注意してください。

HTTPS サービスの確認については、[既知の問題](#)も参照してください。

https および telnet サービスは、Zabbix 2.0 以降でサポートされています。

net.tcp.service.perf[service,<ip>,<port>]

TCP サービスのパフォーマンスをチェックします。

0 - サービスがダウンしています
seconds - サービスへの接続にかかった秒数

サービス - 次のいずれか:
ssh、ldap、smtp、ftp、http、pop、nntp、imap、tcp、https、telnet (詳細を参照)
ip - IP アドレス (デフォルトは 127.0.0.1)
port - ポート番号 (デフォルトでは、標準のサービスポート番号が使用されます)

例:
=> net.tcp.service.perf[ssh] → SSH サーバーからの初期応答速度をテストするために使用できます。

暗号化されたプロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには net.tcp.service.perf[tcp,<ip>,<port>] を使用してください。

Windows での LDAP と HTTPS のチェックは、Zabbix エージェント 2 でのみサポートされています。

telnet チェックでは、ログインプロンプト (末尾の「:」) が検索されることに注意してください。

HTTPS サービスの確認については、[既知の問題](#)も参照してください。

https および telnet サービスは、Zabbix 2.0 以降でサポートされています。

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

アイテムキー

パラメータに一致する TCP ソケットの数を返します。	Integer	<p>laddr - ローカル IPv4/6 アドレスまたは CIDR サブネット</p> <p>lport - ローカルポート番号またはサービス名</p> <p>raddr - リモート IPv4/6 アドレスまたは CIDR サブネット</p> <p>rport - リモートポート番号またはサービス名</p> <p>state - 接続状態 (established, syn_sent, syn_recv, fin_wait1, fin_wait2, time_wait, close, close_wait, last_ack, listen, closing)</p>	<p>この項目は、Zabbix エージェント/エージェント 2 の両方の Linux でのみサポートされています。Zabbix エージェント 2 では、64 ビット Windows でもサポートされています。</p> <p>例: => net.tcp.socket.count[,80,,,established] → ローカル TCP ポート 80 が「確立」状態かどうかを確認する</p> <p>この項目は Zabbix 6.0 以降でサポートされています。</p>
net.udp.listen [port]			
この UDP ポートが LISTEN 状態かどうかを確認します。	0 - LISTEN 状態ではありません 1 - LISTEN 状態です	port - UDP ポート番号	<p>例: => net.udp.listen[68]</p> <p>Zabbix 1.8.4 以降でサポートされている Linux の場合</p>
net.udp.service [service,<ip>,<port>]			
サービスが実行中で、UDP 要求に応答しているかどうかを確認します。	0 - サービスがダウンしています 1 - サービスは実行中です	<p>サービス - ntp (詳細を参照)</p> <p>ip - IP アドレス (デフォルトは 127.0.0.1)</p> <p>port - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)</p>	<p>例: => net.udp.service[ntp,,45] → UDP ポート 45 で NTP サービスの可用性をテストするために使用できます。</p> <p>このアイテムは Zabbix 3.0.0 以降でサポートされていますが、以前のバージョンでは net.tcp.service[] アイテムに対して ntp サービスが利用可能でした。</p>
net.udp.service.perf [service,<ip>,<port>]			
UDP サービスのパフォーマンスをチェックします。	0 - サービスがダウンしています seconds - サービスからの応答を待機するのに費やされた秒数	<p>service - ntp (詳細を参照)</p> <p>ip - IP アドレス (デフォルトは 127.0.0.1)</p> <p>port - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)</p>	<p>例: => net.udp.service.perf[ntp] → NTP サービスからの応答時間をテストするために使用できます。</p> <p>このアイテムは Zabbix 3.0.0 以降でサポートされていますが、ntp サービスは以前のバージョンの net.tcp.service[] アイテムで利用できました。</p>
net.udp.socket.count [<laddr>,<lport>,<raddr>,<rport>,<state>]			
パラメータに一致する TCP ソケットの数を返します	Integer	<p>laddr - ローカル IPv4/6 アドレスまたは CIDR サブネット</p> <p>lport - ローカルポート番号またはサービス名</p> <p>raddr - リモート IPv4/6 アドレスまたは CIDR サブネット</p> <p>**rport - リモートポート番号またはサービス名</p> <p>
state
 - 接続状態 (established, unconn)</p>	<p>この項目は、Zabbix エージェント/エージェント 2 の両方の Linux でのみサポートされています。Zabbix エージェント 2 では、64 ビット Windows でもサポートされています。</p> <p>例: => net.udp.socket.count [,,,listening] → UDP ソケットが“listening” 状態にあるかどうかを確認します</p> <p>この項目は Zabbix 6.0 以降でサポートされています。</p>

プロセスデータ

アイテムキー

説明	戻り値	パラメータ	コメント
proc.cpu.util [<name>,<user>,<type>,<cmdline>,<mode>,<zone>]			

アイテムキー

プロセスの
CPU 使用率

Float

name - プロセス名 (デフォルトはすべてのプロセス)
user - ユーザー名 (デフォルトはすべてのユーザー)
type - CPU 使用率のタイプ:
total (デフォルト), user, system
cmdline - コマンドラインでフィルター
(**正規表現**です)
mode - データ収集モード: avg1 (デフォルト), avg5, avg15
zone - ターゲットゾーン: current (デフォルト), all. このパラメータは、Solaris でのみサポートされています

例:

```
=> proc.cpu.util[,root] → "root" ユーザーの下で実行されているすべてのプロセスの CPU 使用率  
=> proc.cpu.util[zabbix_server,zabbix] → zabbix ユーザーの下で実行されているすべての zabbix_server プロセスの CPU 使用率
```

返される値は、単一の CPU コアの使用率に基づいています。たとえば、2つのコアを完全に使用するプロセスの CPU 使用率は 200% です。

プロセスの CPU 使用率データは、最大 1024 個の一意の (名前、ユーザー、およびコマンドラインによる) クエリをサポートするコレクターによって収集されます。過去 24 時間にアクセスされなかったクエリは、コレクターから削除されます。

注意エージェントがゾーンをサポートしていない Solaris でコンパイルされているが、ゾーンがサポートされている新しい Solaris で実行されている場合、zone パラメータを current (またはデフォルト) に設定すると、エージェントは NOTSUPPORTED を返します。(エージェントは結果を現在のゾーンのみに制限することはできません)。ただし、この場合は all がサポートされます。

このキーは Zabbix 3.0.0 以降でサポートされており、複数のプラットフォームで利用できます (**プラットフォームでサポートされている項目を参照**)

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

アイテムキー

プロセスが使用するメモリ
(バイト単位)

Integer -
mode が max,
min, sum の時

Float - mode
が avg の時

name - プロセス名 (デフォルトはすべてのプロセス)
user - ユーザー名 (デフォルトはすべてのユーザー)
mode - 使用可能な値:
avg, max, min, sum (デフォルト)
cmdline - コマンドラインでフィルター
(正規表現です)
memtype - **メモリの種類** プロセスで使用

例:

```
=> proc.mem[,root] → "root" ユーザーの下で実行されているすべてのプロセスで使用されるメモリ  
=> proc.mem[zabbix_server,zabbix] → zabbix ユーザーの下で実行されているすべての zabbix_server プロセスによって使用されるメモリ  
=> proc.mem[,oracle,max,oracleZABBIX] → コマンドラインに oracleZABBIX が含まれているオラクルの下で実行されている、最もメモリを消費するプロセスによって使用されるメモリ
```

注意: 複数のプロセスが共有メモリを使用する場合、プロセスが使用するメモリの合計は、非現実的な大きな値になる可能性があります。

注意事項を参照して **name** および **cmdline** パラメータ (Linux 固有) でプロセスを選択します。

このアイテムがコマンドラインから呼び出され、コマンドラインパラメーターが含まれている場合 (たとえばエージェントテストモードで: `zabbix_agentd -t proc.mem[,,,apache2]`)、1 つ余分なプロセスが実行されているように見えます。これはエージェントが自分自身をカウントするためです。

memtype パラメータは、Zabbix 3.0.0 以降のいくつかの**プラットフォーム**でサポートされています。

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

アイテムキー

プロセス数	Integer	name - プロセス名 (デフォルトはすべてのプロセス) user - ユーザー名 (デフォルトはすべてのユーザー) state (disk および trace オプションはバージョン 3.4.0 以降) - 使用可能な値: all (デフォルト), disk - 中断不能な sleep, run - 実行中, sleep - 中断可能な sleep, trace - 停止, zomb - ゾンビプロセス cmdline - コマンドラインでフィルター (正規表現です) zone - ターゲットゾーン: current (デフォルト), all. このパラメータは Solaris でのみサポートされています。	例: => proc.num[mysql] → mysql ユーザーの下で実行されているプロセスの数 => proc.num[apache2,www-data] → www-data ユーザーの下で実行されている apache2 プロセスの数 => proc.num[oracle,sleep,oracleZABBIX] → コマンドラインに oracleZABBIX を含む Oracle で実行されているスリープ状態のプロセスの数 注意事項 を参照して name および cmdline パラメータ (Linux 固有) でプロセスを選択します。 Windows では、name および user パラメータのみがサポートされています。 このアイテムがコマンドラインから呼び出され、コマンドラインパラメーターが含まれている場合 (たとえばエージェントテストモードで: zabbix_agentd -t proc.mem[, , ,apache2]), 1 つ余分なプロセスが実行されているように見えます。これはエージェントが自分自身をカウントするためです。 注意エージェントがゾーンをサポートしていない Solaris でコンパイルされているが、ゾーンがサポートされている新しい Solaris で実行されている場合、zone パラメータを current (またはデフォルト) に設定すると、エージェントは NOTSUPPORTED を返します。(エージェントは結果を現在のゾーンのみに制限することはできません)。ただし、この場合は all がサポートされます。
-------	---------	--	--

センサーデータ

アイテムキー

説明	戻り値	パラメータ	コメント
sensor [device,sensor,<mode>] ハードウェア センサー読取	Float	device - デバイス名 sensor - センサー名 mode - 使用可能な値: avg, max, min (このパラメーターを省略すると、デバイスとセンサーは逐語的に扱われます)	コメント Linux 2.4 では /proc/sys/dev/sensors を読み取ります。 例: => sensor[w83781d-i2c-0-2d,temp1] Zabbix 1.8.4 より前では、sensor[temp1] 形式が使用されていました。 Linux 2.6 以降では /sys/class/hwmon を読み取ります。 Linux の センサー アイテムの詳細な説明を参照してください。

アイテムキー

OpenBSD で hw.sensors MIB を読み取りま
す。

例:

=> sensor[cpu0,temp0] → 1 つの CPU の
温度

=> sensor["cpu[0-2]\$,temp,avg] → 最初
の 3 つの CPU の平均温度

Zabbix 1.8.4 以降の OpenBSD でサポート
されています。

システムデータ

アイテムキー

説明	戻り値	パラメータ	コメント
system.boottime			
システムの起 動時間	Integer (Unix タイムスタン プ)		
system.cpu.discovery			
検出された CPU/CPU コア のリスト。ロ ーレベルディ スカバリに使 用されます。	JSON オブジ ェクト		2.4.0 以降のすべてのプラットフォームで サポートされています
system.cpu.intr			
デバイスの割 り込み	Integer		
system.cpu.load[<cpu>,<mode>]			
CPU 負荷.	Float	cpu - 使用可能な値: all (デフォルト)、percpu (バージョン 2.0 以降; 総負荷をオンライン CPU 数で割った もの) mode - 使用可能な値: avg1 (1 分間の平均、デフォルト)、avg5、 avg15	例: => system.cpu.load[,avg5].
system.cpu.num[<type>]			
CPU の数。	Integer	type - 使用可能な値: online (デフォルト)、max	例: => system.cpu.num
system.cpu.switches			
コンテキスト スイッチの数	Integer		
system.cpu.util[<cpu>,<type>,<mode>,<logical_or_physical>]			

アイテムキー

CPU 使用率	Float	<p>cpu - <CPU 番号 > または all (デフォルト)</p> <p>type - 使用可能な値: user(デフォルト)、idle、nice、system (Windows のデフォルト)、iowait、interrupt、softirq、steal、guest (Linux カーネル 2.6.24 以降)、guest_nice (Linux カーネル 2.6.33 以降)</p> <p>このパラメータの詳細については、プラットフォーム固有 も参照してください。</p> <p>モード - 使用可能な値: avg1 (1 分間の平均、デフォルト)、avg5、avg15</p> <p>logical_or_physical (バージョン 5.0.3 以降、AIX のみ) - 使用可能な値: logical (デフォルト)、physical。このパラメータは AIX でのみサポートされます。</p>	<p>Windows では、値は Processor Time パフォーマンスカウンタを使用して取得されます。Windows 8 以降、タスクマネージャ (Processor Utility パフォーマンスカウンタに基づいて CPU 使用率を表示しますが、以前のバージョンでは Processor Time カウンタでした。</p> <p>例: => system.cpu.util[0,user,avg5]</p>
旧名称: sys-tem.cpu.idleX, sys-tem.cpu.niceX, sys-tem.cpu.systemX, sys-tem.cpu.userX			
system.hostname [<type>,<transform>]			
システムのホスト名	String	<p>type (Windows でのみサポートされるバージョン 5.4.7 より前) - 使用可能な値:netbios (Windows のデフォルト),host (Linux のデフォルト),shorthost (バージョン 5.4.7 以降; 最初のドットの前のホスト名の一部、ドットのない名前の完全な文字列を返します)</p> <p>transform (バージョン 5.4.7 以降) - 使用可能な値: none (デフォルト)、lower (小文字に変換)</p>	<p>この値は、Windows では GetComputerName() (netbios の場合) または gethostname() (host の場合) 関数によって取得され、他のシステムでは uname() システム API 出力から nodename を取得することによって取得されます。</p> <p>戻り値の例: Linux の場合: => system.hostname → linux-w7x1 => system.hostname → example.com => system.hostname[shorthost] → 例</p> <p>Windows の場合: => system.hostname → WIN-SERV2008-I6 => system.hostname[host] → Win-Serv2008-I6LonG => system.hostname[host,lower] → win-serv2008-i6long</p> <p>より詳細な説明も参照してください。</p>
system.hw.chassis [<info>]			

アイテムキー

シャーン情報	String	情報 - フル (デフォルト)、モデル、シリアル、タイプ、ベンダーのいずれか	例: system.hw.chassis[full] Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
			このキーは、 SMBIOS テーブルの可用性に依存します。 sysfs から DMI テーブルを読み込もうとします。sysfs アクセスが失敗した場合は、メモリから直接読み込もうとします。
			sysfs やメモリから読み込んで値を取得するため、 root パーミッションが必要です。
			Zabbix 2.0 以降でサポートされています。
system.hw.cpu [<cpu>,<info>]			
CPU 情報	String または integer	cpu - <CPU 番号 > または all (デフォルト) info - 使用可能な値: full (デフォルト),curfreq, maxfreq, model, vendor	例: => system.hw.cpu[0,vendor] → AuthenticAMD
			/proc/cpuinfo および /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_ から情報を収集します。
			CPU 番号と curfreq の場合 または maxfreq が指定されている場合は、数値 (Hz) が返されます。
			Zabbix 2.0 以降でサポートされています
system.hw.devices [<type>]			
PCI または USB デバイスのリスト	Text	type (バージョン 2.0 以降) - pci (デフォルト) または usb	例: => system.hw.devices[pci] → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge [..]
			lspci または lsusb ユーティリティ (パラメータなしで実行) の出力を返します。
system.hw.macaddr [<interface>,<format>]			
MAC アドレスのリスト	String	インターフェース - all (デフォルト) または 正規表現 format - full (デフォルト) または short	名前が指定された interface 正規表現 に一致するインターフェイスの MAC アドレスを一覧表示します (all すべてのインターフェイスのリスト)。
			例: => system .hw.macaddr["eth0\$",full] → [eth0] 00:11:22:33:44:55
			format が short と指定されている場合、インターフェース名と同一の MAC アドレスはリストされません。
			Zabbix 2.0 以降でサポートされています。
system.localtime [<type>]			
システム時刻	Integer - type を utc として使用 String - type を local として使用	type (バージョン 2.0 以降) - 使用可能な値: utc - (デフォルト) エポック (00:00:00 UTC、1970 年 1 月 1 日) からの時間 (秒単位) local - "yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm" 形式の時刻	パッシブチェック としてのみ使用する必要があります。
			例: => system.localtime[local] → このキーを使用してアイテムを作成し、それを使用して Clock ダッシュボードウィジェット にホスト時間を表示します

system.run[command,<mode>]

ホスト上で指
定されたコマ
ンドを実行し
ます
コマンドの結
果テキスト
1 - mode を
nowait として
使用 (コマン
ドの結果に関
係なく)

command - 実行のためのコマンド
mode - 使用可能な値:
wait - 実行の終了を待つ (デフォルト),
nowait - 終了を待たない

切り捨てられた末尾の空白を含め、最大
512KB のデータを返すことができます。
正しく処理するには、コマンドの出力がテ
キストである必要があります。

例:

=> system.run[ls -l /] → ルートディレクト
リの詳細ファイル一覧。

注: system.run 項目はデフォルトで無効に
なっています。有効にする方法。

項目の戻り値は、コマンドが生成する標準
エラーとともに標準出力となります。終了
コードはチェックされません。

Zabbix 2.4.0 以降、空の結果が許可されま
す。

コマンド実行も参照してください。

system.stat[resource,<type>]

アイテムキー

システム統計 Integer または float

ent - このパーティションが受け取る資格のあるプロセッサユニットの数 (float)

kthr,<type> - カーネルスレッドの状態に関する情報:
r - 実行可能なカーネルの平均数スレッド (float)
b - 仮想メモリマネージャーの待機キューに配置されたカーネルスレッドの平均数 (float)

memory,<type> - 仮想メモリと実メモリの使用に関する情報:
avm - アクティブな仮想ページ (整数)
fre - フリーリストのサイズ (整数)

page,<type> - ページフォールトとページングアクティビティに関する情報:
fi - 1 秒あたりのファイルページイン (float)
fo - 1 秒あたりのファイルページアウト (float)
pi - ページングスペースからページインされたページ (float)
po - ページングスペースにページアウトされたページ (float)
fr - 解放されたページ (ページ置換) (float)
sr - ページ置換アルゴリズムによってスキャンされたページ (float)

* ***faults,<type>** - トラップと割り込みレート:
in - デバイス割り込み (float)
sy - システムコール (float)
cs - カーネルスレッドコンテキストスイッチ (float)
cpu,<type> - プロセッサ時間の使用率の内訳:
us - ユーザー時間 (float)
sy - システム時間 (float)
id - アイドル時間 (float)
wa - システムに未処理のディスク/NFS I/O 要求があったアイドル時間 (float)
pc - 消費された物理プロセッサの数 (float)
ec - 消費されたライセンス容量のパーセンテージ (float)
lbusy - ユーザーおよびシステムレベルで実行中に発生した論理プロセッサの使用率を示します (float)
app - 共有プールで使用可能な物理プロセッサを示します (float)
disk,<type>** - ディスク統計:
bps - ドライブに転送 (読み取りまたは書き込み) されたデータの量を 1 秒あたりのバイト数 (整数) で示します。
tps - 物理ディスク/テープに発行された 1 秒あたりの転送数を示します (float)

system.sw.arch

ソフトウェア String
アーキテクチャ情報

system.sw.os[<info>]

例:

=> system.sw.arch → i686

uname() 関数から情報を取得します。

Zabbix 2.0 以降でサポートされています。

アイテムキー

OS 情報	String	info - 使用可能な値: full (デフォルト)、short または name	例: => system.sw.os[short]→ Ubuntu 2.6.35-28.50-generic 2.6.35.11 情報は以下から取得されます (すべてのディストリビューションにすべてのファイルとオプションが存在するわけではないことに注意してください): /proc/version (full) /proc/version_signature (short) サポートしているシステムの /etc/os-release の PRETTY_NAME パラメータ、または /etc/issue.net (name) Zabbix 2.0 以降でサポートされています。
system.sw.packages [<package>,<manager>,<format>]			
インストール済みパッケージのリスト	Text	パッケージ - all (デフォルト) または 正規表現 manager - all (デフォルト) またはパッケージマネージャー ** format** - full (デフォルト) または short	指定された package 正規表現 と名前が一致するインストール済みパッケージを (アルファベット順に) リストします (all はそれらすべてをリストします)。 例: => system. sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal 対応パッケージマネージャー (実行コマンド): dpkg (dpkg --get-selections) pkgtool (ls /var/log/packages) rpm (rpm -qa) pacman (pacman -Q) format が full と指定されている場合、パッケージはパッケージマネージャーごとにグループ化されます (各マネージャーは角かっこで囲まれた名前が始まる別の行にあります)。 format が * と指定されている場合 short*、パッケージはグループ化されず、1 行にリストされます。 Zabbix 2.0 以降でサポートされています。
system.swap.in [<device>,<type>]			
(デバイスからメモリに) 統計情報をスワップします	Integer	device - スワップに使用されるデバイス (デフォルトは all) type - 使用可能な値: count (スワップインの数)、sectors (スワップインされたセクター)、pages (スワップインされたページ)。 このパラメーターの詳細については、 プラットフォーム固有 も参照してください。	例: => system.swap.in[,pages] この情報のソース: /proc/swaps、/proc/partitions、/proc/stat (Linux 2.4) /proc/swaps、/proc/diskstats、/proc/vmstat (Linux 2.6)
system.swap.out [<device>,<type>]			
(メモリからデバイスに) 統計情報をスワップアウトします	Integer	device - スワップに使用されるデバイス (デフォルトは all) type - 使用可能な値: count (スワップアウトの数)、sectors (スワップアウトされたセクター)、pages (スワップアウトされたページ)。 このパラメーターの詳細については、 プラットフォーム固有 も参照してください。	例: => system.swap.out[,pages] この情報のソース: /proc/swaps、/proc/partitions、/proc/stat (Linux 2.4) /proc/swaps、/proc/diskstats、/proc/vmstat (Linux 2.6)

system.swap.size[<device>,<type>]

スワップ領域のサイズ (バイト単位または合計に対する割合)

Integer - バイトの場合

Float - パーセントの場合

device - スワップに使用されるデバイス (デフォルトは all)

type - 使用可能な値:
 free (空きスワップ領域、デフォルト)、
 pfree (空きスワップ領域、パーセント)、
 pused (使用済みスワップ領域、パーセント)、
 total (合計スワップ領域)、
 used (使用済みスワップ領域)

スワップサイズが 0 の場合、pfree、pused は Windows ではサポートされないことに注意してください。
 このパラメータの詳細については、[プラットフォーム固有](#) も参照してください。

例:
 => system.swap.size[,pfree] → 空きスワップ領域の割合

device が指定されていない場合、Zabbix エージェントはスワップデバイス (ファイル) のみを考慮し、物理メモリは無視されます。たとえば、Solaris システムでは、swap -s コマンドには物理メモリとスワップデバイスの一部が含まれます (swap -l とは異なります)

このキーは、仮想化 (VMware ESXi、VirtualBox) Windows プラットフォームで誤ったスワップ領域のサイズ/割合を報告する可能性があることに注意してください。この場合、perf_counter [\700(_Total)\702] キーを使用して正しいスワップ領域のパーセンテージを取得できます。

system.uname

システムの ID String

戻り値の例 (Unix):
 FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: 月 11 月 i386

戻り値の例 (Windows):
 Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86

Zabbix 2.2.0 以降の Unix では、このアイテムの値は uname() システムコールで取得されます。以前は "uname -a" を呼び出すことによって取得されていました。この項目の値は、"uname -a" の出力とは異なる場合があります、他のソースに基づいて "uname -a" が出力する追加情報は含まれません。

Zabbix 3.0 以降の Windows では、この項目の値は Win32_OperatingSystem および Win32_Processor WMI クラスから取得されます。以前は、揮発性の Windows API および文書化されていないレジストリキーから取得されていました。OS 名 (エディションを含む) は、ユーザーの表示言語に翻訳される場合があります。一部のバージョンの Windows では、商標記号と余分なスペースが含まれています。

Windows では OS アーキテクチャを返しますが、Unix では CPU アーキテクチャを返します。

item configuration では、**s** または **uptime** 単位を使用して読み取り可能な値を取得します。

who コマンドは、エージェント側で値を取得するために使用されます

system.uptime

システム稼働時間 (秒単位) Integer

system.users.num

ログインしているユーザーの数 Integer

アイテムキー			
説明	戻り値	パラメータ	コメント
vfs.dev.discovery			
ブロックデバイスとそのタイプのリスト。ローレベルディスクカバリに使用されます。	JSON object		この項目は Linux プラットフォームでのみサポートされています。 Zabbix 4.4.0 以降でサポートされています。
vfs.dev.read [<device>,<type>,<mode>]			
ディスク読み取り統計	Integer - sectors、operations、bytes で type を使用 Float - sps、ops、bps で type を使用 注: 更新間隔が 3 時間以上の場合 ² 、常に '0' を返します	device - ディスクデバイス (デフォルトは all ³) type - 使用可能な値: sectors、operations、bytes、sps、ops、bps 'type' パラメータのサポートとデフォルトはプラットフォームに依存することに注意してください。プラットフォーム固有の詳細を参照してください。 sps、ops、bps は、それぞれ 1 秒あたりのセクター、オペレーション、バイト数を表します。 mode - 使用可能な値: avg1 (1 分間の平均、デフォルト)、avg5、avg15。 このパラメータは、sps、ops、bps の type でのみサポートされます。	相対デバイス名 (sda など) とオプションの /dev/ プレフィックス (/dev/sda)。 LVM 論理ボリュームがサポートされていません。 さまざまな OS の 'type' パラメータのデフォルト値: AIX - operations FreeBSD - bps Linux - sps OpenBSD - operations Solaris - bytes 例: => vfs.dev.read[,operations] サポートされているプラットフォームの sps、ops、および bps は、以前は 8 つのデバイス (個別に 7 つ、すべてに 1 つ) に制限されていました。Zabbix 2.0.1 以降、この制限は 1024 デバイスです (1023 個の個別デバイスと 1 個のすべて)
vfs.dev.write [<device>,<type>,<mode>]			
ディスク書き込み統計	Integer - sectors、operations、bytes で type を使用 Float - sps、ops、bps で type を使用 注: 更新間隔が 3 時間以上の場合 ² 、常に '0' を返します	device - ディスクデバイス (デフォルトは all ³) type - 使用可能な値: sectors、operations、bytes、sps、ops、bps 'type' パラメータのサポートとデフォルトはプラットフォームに依存することに注意してください。プラットフォーム固有の詳細を参照してください。 sps、ops、bps は、それぞれ 1 秒あたりのセクター、オペレーション、バイト数を表します。 mode - 使用可能な値: avg1 (1 分間の平均、デフォルト)、avg5、avg15。 このパラメータは、sps、ops、bps の type でのみサポートされます。	相対デバイス名 (sda など) とオプションの /dev/ プレフィックス (/dev/sda)。 LVM 論理ボリュームがサポートされていません。 さまざまな OS の 'type' パラメータのデフォルト値: AIX - operations FreeBSD - bps Linux - sps OpenBSD - operations Solaris - bytes 例: => vfs.dev.read[,operations] サポートされているプラットフォームの sps、ops、および bps は、以前は 8 つのデバイス (個別に 7 つ、すべてに 1 つ) に制限されていました。Zabbix 2.0.1 以降、この制限は 1024 デバイスです (1023 個の個別デバイスと 1 個のすべて)
vfs.dir.count [dir,<regex_incl>,<regex_excl>,<types_incl>,<types_excl>,<max_depth>,<min_size>,<max_size>,<min_age>,<max_age>]			

ディレクトリ
エン트리数 Integer

dir - ディレクトリへの絶対パス
regex_incl - 含めるエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する **正規表現**。空の場合はすべて含める (デフォルト値)
regex_excl - **正規表現** 除外するエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述します。空の場合は除外しない (デフォルト値)
types_incl - カウントするディレクトリエントリの種類、使用可能な値:
 file - 通常のファイル、dir - サブディレクトリ、sym - シンボリックリンク、sock - ソケット、bdev - ブロックデバイス、cdev - キャラクターデバイス、fifo - FIFO、* dev* - "bdev,cdev"、all - すべてのタイプ (デフォルト)、つま
 り "file,dir,sym,sock,bdev,cdev,fifo" と同義。複数のタイプはコマンドで区切り、引用符で囲む必要があります。
types_excl - カウントしないディレクトリエントリタイプ (<types_incl> を参照)。
 <types_incl> と <types_excl> の両方にエン트리タイプがある場合、このタイプのディレクトリエントリはカウントされません。
max_depth - 走査するサブディレクトリの最大深度。-1 (デフォルト) - 無制限、0 - サブディレクトリへの下降なし。
min_size - カウントするファイルの最小サイズ (バイト単位)。それより小さいファイルはカウントされません。 **メモリサフィックス** を使用できます。
max_size - カウントするファイルの最大サイズ (バイト単位)。それより大きいファイルはカウントされません。 **メモリサフィックス** を使用できます。
min_age - カウントされるディレクトリエントリの最小経過時間 (秒単位)。最近のエント리는カウントされません。 **時間のサフィックス** を使用できます。
max_age - カウントされるディレクトリエントリの最大経過時間 (秒単位)。古いエント리는カウントされません (変更時間)。 **タイムサフィックス** を使用できます。
regex_excl_dir - **正規表現** 除外するディレクトリ
 の名前パターンを記述します。ディレクトリのすべてのコンテンツが除外されます (regex_excl とは対照的)

環境変数。%APP_HOME%、\$HOME、および%TEMP% はサポートされていません。

疑似ディレクトリ"." および".." はカウントされません。

ディレクトリトラバーサルでシンボリックリンクをたどることはありません。

Windows では、ディレクトリのシンボリックリンクはスキップされます。

エントリのサイズを計算するとき、**regex_incl** と **regex_excl** の両方がファイルとディレクトリに適用されますが、トラバースするサブディレクトリを選択するときは無視されます (**regex_incl** が "(?i)^\.+\.zip\$" で **max_depth** が設定されていない場合、すべてのサブディレクトリがトラバースされますが、zip タイプのファイルのみがカウントされます)。

実行時間は、エージェントのデフォルトのタイムアウト値 **構成** (3 秒) によって制限されます。大規模なディレクトリトラバーサルにはそれよりも時間がかかる可能性があるため、データは返されず、アイテムはサポートされなくなります。部分カウントは返されません。

サイズでフィルタリングすると、通常のファイルのみが意味のあるサイズになります。Linux および BSD では、ディレクトリのサイズもゼロ以外です (通常は数 Kb)。デバイスのサイズはゼロです。/dev/sda1 のサイズは、それぞれのパーティションサイズを反映していません。したがって、<min_size> と <max_size> を使用する場合は、驚きを避けるために <types_incl> を "file" として指定することをお勧めします。

例:
 ⇒ vfs.dir.count[/dev] - /dev 内のデバイス数を監視 (Linux)
 ⇒
 vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"]
 - 一時ディレクトリ内のファイル数を監視 (Windows)

Zabbix 4.0.0 以降でサポートされています。

vfs.dir.get[dir,<regex_incl>,<regex_excl>,<types_incl>,<types_excl>,<max_depth>,<min_size>,<max_size>,<min_age>,<max_age>]

ディレクトリ
エントリリス
ト JSON

dir - ディレクトリへの絶対パス
regex_incl - 含めるエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する**正規表現**。空の場合はすべて含める (デフォルト値)
regex_excl - 除外するエンティティ (ファイル、ディレクトリ、シンボリックリンク) の名前パターンを記述する**正規表現**。空の場合は除外しない (デフォルト値)
types_incl - リストするディレクトリエントリの種類。使用可能な値:
 file - 通常のファイル、dir - サブディレクトリ、sym - シンボリックリンク、sock - ソケット、bdev - ブロックデバイス、cdev - キャラクターデバイス、fifo - FIFO、* dev* - "bdev,cdev"、all - すべてのタイプ (デフォルト)、つま
 り "file,dir,sym,sock,bdev,cdev,fifo" と同義。複数のタイプはコマンドで区切り、引用符で囲む必要があります。
types_excl - リストしないディレクトリエントリタイプ (<types_incl> を参照)。
 <types_incl> と <types_excl> の両方にエントリタイプがある場合、このタイプのディレクトリエントリはリストされません。
max_depth - 走査するサブディレクトリの最大深度。-1 (デフォルト) - 無制限、0 - サブディレクトリへの下降なし。
min_size - リストされるファイルの最小サイズ (バイト単位)。それより小さいファイルはリストされません。**メモリサフィックス**を使用できます。
max_size - リストされるファイルの最大サイズ (バイト単位)。それより大きいファイルはカウントされません。**メモリサフィックス**を使用できます。
min_age - リストされるディレクトリエントリの最小経過時間 (秒単位)。最近のエントリは表示されません。**タイムサフィックス**を使用できます。
max_age - リストされるディレクトリエントリの最大経過時間 (秒単位)。古いエントリはリストされません (変更時間) **タイムサフィックス**を使用できます。
regex_excl_dir - 除外するディレクトリ
 の名前パターンを記述する**正規表現**。ディ
 レクトリのすべてのコンテンツが除外されま
 す (regex_excl とは対照的)

環境変数。%APP_HOME%、\$HOME、およ
 び%TEMP% はサポートされていません。
 疑似ディレクトリ。”

ディレクトリトラバーサルでシンボリック
 リンクをたどることはありません。

Windows では、ディレクトリのシンボリック
 クリンクはスキップされます。

エントリのサイズを計算するとき、
regex_incl と **regex_excl** の両方がフ
 ァイルとディレクトリに適用されますが、
 トラバースするサブディレクトリを選択す
 るときは無視されます (**regex_incl**
 が "(?i)^.+ の場合) .zip\$" で **max_depth** が
 設定されていない場合、すべてのサブディ
 レクトリが走査されますが、タイプが zip
 のファイルのみが一覧表示されます)。

実行時間は、エージェントのデフォルトの
 タイムアウト値**構成** (3 秒) によって制限さ
 れます。大規模なディレクトリトラバーサ
 ルにはそれよりも時間がかかる可能性があ
 るため、データは返されず、アイテムはサ
 ポートされなくなります。部分的なリスト
 は返されません。

サイズでフィルタリングすると、通常のフ
 ァイルのみが意味のあるサイズになります。
 Linux および BSD では、ディレクトリのサ
 イズもゼロ以外です (通常は数 Kb)。デバイ
 スのサイズはゼロです。**/dev/sda1** のサイ
 ズは、それぞれのパーティションサイズを
 反映していません。したがって、
 <min_size> と <max_size> を使用する
 場合は、驚きを避けるために
 <types_incl> を "file" として指定するこ
 とをお勧めします。

例:
 ⇒ vfs.dir.get[/dev] - /dev のデバイス一覧を
 取得 (Linux)
 ⇒
 vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"]
 - 一時ディレクトリのファイル一覧を取得
 (Windows)

Zabbix 6.0.0 以降でサポートされて
 います。

vfs.dir.size[dir,<regex_incl>,<regex_excl>,<mode>,<max_depth>,<regex_excl_dir>]

アイテムキー

ディレクトリ のサイズ (バ イト単位)	Integer	dir - ディレクトリへの絶対パス regex_incl - 含めるエンティティ (ファイ ル、ディレクトリ、シンボリックリンク) の 名前パターンを記述する正規の式; 空の場 合はすべて含める (デフォルト値) regex_excl - 除外するエンティティ (ファイ ル、ディレクトリ、シンボリックリンク) の名前パターンを記述する正規の式。空の 場合は除外しない (デフォルト値) mode - 可能な値: apparent (デフォルト) - ディスク使用量で はなく見かけのファイルサイズを取得しま す (「du -sb dir」 として機能します), disk - ディスクの使用状況を取得します (「du -s -B1 dir」 として機能します)。du コマンドと は異なり、vfs.dir.size アイテムは、ディレ クトリサイズを計算するときに隠しファイ ルを考慮に入れます (dir 内で「du -sb .[^.]*」 として機能します)。 max_depth - 最大トラバースするサブディ レクトリの深さ。-1 (デフォルト) - 無制限、 0 - サブディレクトリへの下降なし。 regex_excl_dir - 正規の式除外するディレ クトリ の名前パターン。ディレクトリのす べてのコンテンツが除外されます (regex_excl とは対照的)	少なくとも zabbix ユーザーの読み取り権限 を持つディレクトリのみが計算されます。 Windows では、シンボリックリンクはスキ ップされ、ハードリンクが取り込まれます。 ディレクトリが大きい場合やドライブが遅 い場合、エージェントとサーバー/proxy 構 成ファイル。必要に応じてタイムアウト値 を増やしてください。 例: => vfs.dir.size[/tmp,log] - 'log' を含む /tmp 内のすべてのファイルのサイズを計算しま す => vfs.dir.size[/tmp,log,^.+\.old\$] - '.old' を含むファイルを除く、'log' を含む /tmp 内 のすべてのファイルのサイズを計算しま す ファイルサイズの制限は、大きなフ ァイルのサポート によって異なります。 Zabbix 3.4.0 以降でサポートされています。
vfs.file.cksum [file,<mode>] UNIX cksum アルゴリズム によって計算 されたファイ ルチェックサ ム	整数 - mode を crc32 とし て使用 文字列 - mode を md5、 sha256 とし て使用	ファイル - ファイルへのフルパス モード - crc32 (デフォルト)、md5、sha256	例: => vfs.file.cksum[/etc/passwd] 戻り値の例 (それぞれ crc32/md5/sha256): 675436101 9845acf68b73991eb7fd7ee0ded23c44 ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb
vfs.file.contents [file,<encoding>] ファイル内容 取得	Text	file - ファイルへのフルパス encoding - コードページ識別子	例: => vfs.file.contents[/etc/passwd] このアイテムは、64K バイト以下のファイ ルに限定されています。 Zabbix 2.0 以降でサポートされています。
vfs.file.exists [file,<types_incl>,<types_excl>]			

アイテムキー

ファイルが存在するか確認	0 - 見つかりません 1 - 指定したタイプのファイルが存在しません	file - ファイルへのフルパス types_incl - 含めるファイルタイプのリスト。使用可能な値: file (通常のファイル、デフォルト (types_excl が設定されていない場合))、dir (ディレクトリ)、sym (シンボリックリンク)、sock (ソケット)、bdev (ブロックデバイス)、cdev (キャラクターデバイス)、fifo (FIFO)、dev (同義語) "bdev,cdev")、all (言及されているすべてのタイプ、types_excl が設定されている場合のデフォルト)。 types_excl - 除外するファイルタイプのリスト。可能な値については types_incl を参照してください。(デフォルトでは、除外されるタイプはありません)	複数のタイプはコンマで区切り、セット全体を引用符"" で囲む必要があります。 Windows では、コマンドラインユーティリティを使用して zabbix_get.exe または agent2 を呼び出す場合、二重引用符をバックスラッシュ '\' でエスケープし、アイテムキー全体を二重引用符で囲む必要があります。 同じタイプが <types_incl> と <types_excl> の両方にある場合、このタイプのファイルは除外されます。 例: => vfs.file.exists[/tmp/application.pid] => vfs.file.exists[/tmp/application.pid,"file,dir,sym"] => vfs.file.exists[/tmp/application_dir,dir] ファイルサイズの制限は、 大きなファイルのサポート によって異なります。 ディレクトリが存在しないディレクトリ内で検索された場合、アイテムが Windows でサポートされなくなる可能性があることに注意してください。 vfs.file.exists[C:\no\dir,dir] ('no' は存在しません)。 UNIX ライクなシステムでサポートされているファイルタイプ: 通常のファイル、ディレクトリ、シンボリックリンク、ソケット、ブロックデバイス、キャラクターデバイス、FIFO Windows でサポートされているファイルタイプ: 通常のファイル、ディレクトリ、シンボリックリンク 例: => vfs.file.get[/etc/passwd] → /etc/passwd ファイルに関する情報 (タイプ、ユーザー、権限、SID、uid など) を含む JSON を返す Zabbix 6.0 以降でサポートされています。 例: => vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf] 戻り値の例: b5052dec577e0fffd622d6ddc017e82 このアイテムのファイルサイズ制限 (64 MB) は、バージョン 1.8.6 で削除されました。 ファイルサイズの制限は、 大きなファイルのサポート によって異なります。
vfs.file.get [file]	JSON object	file - ファイルへのフルパス	
ファイルに関する情報を返す			
vfs.file.md5sum [file]	string(ファイルの MD5 チェックサム)	file - ファイルへのフルパス	
ファイルの MD5 チェックサム			
vfs.file.owner [file,<ownertype>,<resulttype>]			

アイテムキー

ファイル所有者取得	Character string	file - ファイルへのフルパス ownertype - user (デフォルト) または group (Unix のみ) resulttype - name (デフォルト) または id; for id - Unix では uid/gid、Windows では SID を返します	例: => <code>vfs.file.owner[/tmp/zabbix_server.log]</code> → /tmp/zabbix_server.log のファイル所有者を返す => <code>vfs.file.owner[/tmp/zabbix_server.log,,id]</code> → /tmp/zabbix_server.log のファイル所有者 ID を返す
vfs.file.permissions [file]	Unix パーミッションを持つ 8 進数を含む 4 桁の文字列を返す	String file - ファイルへのフルパス	Zabbix 6.0 以降でサポートされています。 Windows ではサポートされていません。 例: => <code>vfs.file.permissions[/etc/passwd]</code> → /etc/passwd のパーミッションを返す (例: '0644')
vfs.file.regex [file,regexp,<encoding>,<start line>,<end line>,<output>]	ファイル内文字列検索	一致した文字列を含む行、またはオプションの output パラメータで指定された行 file - ファイルへのフルパス regexp - 必要なパターンを説明する正規の式 encoding - コードページ識別子 start line - 検索する最初の行の番号 (デフォルトではファイルの最初の行)。 end line - 検索する最後の行の番号 (デフォルトではファイルの最後の行)。 output - オプションの出力フォーマットテンプレート。\\0 エスケープシーケンスは、テキストの一致した部分 (一致が始まる最初の文字から一致が終了する文字まで) に置き換えられますが、\\N (N=1... 9) エスケープシーケンスは N 番目に一致したグループ (または、N がキャプチャされたグループの数を超える場合は空の文字列) に置き換えられます。	例: => <code>vfs.file.regex[/etc/passwd,zabbix]</code> => <code>vfs.file.regex[/path/to/some/file,"{[0-9]+}\$","3,5,\\1]</code> => <code>vfs.file.regex[/etc/passwd,"^zabbix:.*{[0-9]+}\$","\\1]</code> → ユーザー zabbix の ID を取得する
vfs.file.regmatch [file,regexp,<encoding>,<start line>,<end line>]	ファイル内文字列検索	0 - 一致なし 1 - 一致有り file - ファイルへのフルパス regexp - 必要なパターンを説明する正規の式 encoding - コードページ識別子 start line - 検索する最初の行の番号 (デフォルトではファイルの最初の行) end line - 検索する最後の行の番号 (デフォルトではファイルの最後の行)	例: => <code>vfs.file.regmatch[/var/log/app.log,error]</code>
vfs.file.size [file,<mode>]			

アイテムキー			
ファイルサイズ (bytes 単位).	Integer	file - ファイルのフルパス mode - 使用可能な値: bytes (デフォルト) または lines (空の行もカウントされます)	ファイルには、ユーザー zabbix の読み取り権限が必要です。 例: => vfs.file.size[/var/log/syslog] ファイルサイズの制限は、 大きなファイルのサポート によって異なります。 mode パラメータは、Zabbix 6.0 以降でサポートされています。
vfs.file.time [file,<mode>] ファイル時間情報	Integer (Unix timestamp)	file - ファイルのフルパス mode - 可能な値: modify (デフォルト) - ファイルの内容を最後に変更した時間 access - ファイルを最後に読み取った時間 change - ファイルを最後に変更した時間	例: => vfs.file.time[/etc/passwd,modify] ファイルサイズの制限は、 大きなファイルのサポート によって異なります。
vfs.fs.discovery マウントされたファイルシステムとそのタイプのリスト。ローレベルディスクカバリで使用。	JSON object		Zabbix 2.0 以降でサポートされています。 {#FSDRIVETYPE} マクロは、Zabbix 3.0 以降で Windows でサポートされています。 {#FSLABEL} マクロは、Zabbix 6.0 以降で Windows でサポートされています。
vfs.fs.get マウントされたファイルシステム、そのタイプ、ディスク容量、および inode 統計のリスト。ローレベルディスクカバリで使用。	JSON object		Zabbix 4.4.5 以降でサポートされています。 {#FSLABEL} マクロは、Zabbix 6.0 以降の Windows でサポートされています。
vfs.fs.inode [fs,<mode>] inode の数またはパーセンテージ	Integer - 数値用 Float - パーセント用	fs - ファイルシステム mode - 使用可能な値: total (デフォルト)、free、used、//pfree // (free、パーセンテージ)、pused (used、パーセンテージ)	例: => vfs.fs.inode[/,pfree]
vfs.fs.size [fs,<mode>] ディスク容量 (バイト単位または合計に対する割合)	Integer - bytes 用 Float - パーセント用	fs - ファイルシステム mode - 使用可能な値: total (デフォルト)、free、used、pfree (free、パーセンテージ)、pused (used、パーセンテージ)	マウントされたボリュームの場合、ローカルファイルシステムのディスク容量が返されます。 例: => vfs.fs.size[/tmp,free] free モードを使用する場合、ファイルシステムの予約済みスペースは考慮され、含まれません。

仮想メモリデータ

アイテムキー			
説明	戻り値	パラメータ	コメント
vm.memory.size [<mode>]			

アイテムキー			
メモリサイズ (バイト単位ま たは合計に対 するパーセン テージ)	Integer - バイ ト単位 Float - パーセ ンテージ	mode - 使用可能な値: total (デフォルト), active, anon, buffers, cached, exec, file, free, inactive, pinned, shared, slab, wired, used, pused (パーセン テージの時), available, pavailable (パーセ ンテージの時利用可能) このパラメーターについては プラットフォ ーム固有 サポートと 追加の詳細 も参照して ください。	この項目は 3 つのカテゴリのパラメータを 受け入れます: 1) total - メモリ総量 2) プラットフォーム固有のメモリタイプ: active, anon, buffers, cached, exec, file, free, inactive, pinned, shared, slab, wired 3) メモリの使用量と使用可能量に関するユ ーザーレベルの見積もり: used, pused, available, pavailable.

Web モニタリングデータ

アイテムキー			
説明	戻り値	パラメータ	コメント
web.page.get [host,<path>,<port>]			
Web ページコ ンテンツ取得	テキストとし ての Web ペ ージソース (ヘ ッダーを含む)	host - ホスト名または URL (scheme://host:port/path として、 host のみが必須) 許可されている URL スキーム: http、 https ⁴ 欠落しているスキームは http として 扱われます。URL が指定されている場合、 path と port は空でなければなりません。 認証が必要なサーバーに接続する際のユ ーザー名/パスワードの指定、たとえ ば:http://user:password@www.example.com は cURL サポートでのみ可能です ⁴ ホスト名で Punycode がサポートされてい ます。 path - HTML ドキュメントへのパス (デフ ォルトは /) port - ポート番号 (HTTP のデフォルトは 80)	host で指定されたりソースが存在しない か利用できない場合、このアイテムはサポ ートされなくなります。 host は、ホスト名、ドメイン名、IPv4 ま たは IPv6 アドレスにすることができます。 ただし、IPv6 アドレスの場合、Zabbix エ ージェントは IPv6 サポートを有効にしてコン パイルする必要があります。 例: => web.page.get[www.example.com,index.php,80] => web.page.get[https://www.example.com] => web.page.get[https://blog.example.com/?s=zabbix]
=> web.page.get[localhost:80] => web.page.get["::1]/server-status"]
web.page.perf [host,<path>,<port>]			
Web ページ全 体の読み込み 時間 (秒単位)	Float	host - ホスト名または URL (scheme://host:port/path として、 host のみが必須) 許可されている URL スキーム: http、 https ⁴ 欠落しているスキームは http として 扱われます。URL が指定されている場合、 path と port は空でなければなりません。 認証が必要なサーバーに接続する際のユ ーザー名/パスワードの指定、たとえ ば:http://user:password@www.example.com は cURL サポートでのみ可能です ⁴ ホスト名で Punycode がサポートされてい ます。 path - HTML ドキュメントへのパス (デフ ォルトは /) port - ポート番号 (HTTP のデフォルトは 80)	host で指定されたりソースが存在しない か利用できない場合、このアイテムはサポ ートされなくなります。 host は、ホスト名、ドメイン名、IPv4 ま たは IPv6 アドレスにすることができます。 ただし、IPv6 アドレスの場合、Zabbix エ ージェントは IPv6 サポートを有効にしてコン パイルする必要があります。 例: => web.page.perf[www.example.com,index.php,80] => web.page.perf[https://www.example.com]
web.page.regexp [host,<path>,<port>,<regexp>,<length>,<output>]			

アイテムキー

Web ページで文字列を検索	一致した文字列、またはオプションの output パラメータで指定された文字列	<p>host - ホスト名または URL (scheme://host:port/path として、host のみが必須)</p> <p>許可されている URL スキーム: http、https⁴ 欠落しているスキームは http として扱われます。URL が指定されている場合、path と port は空でなければなりません。認証が必要なサーバーに接続する際のユーザー名/パスワードの指定、たとえば: http://user:password@www.example.com は cURL サポートでのみ可能です⁴</p> <p>ホスト名で Punycode がサポートされています。</p> <p>path - HTML ドキュメントへのパス (デフォルトは /)</p> <p>port - ポート番号 (HTTP のデフォルトは 80)</p> <p>regexp - 必要なパターンを説明する正規式</p> <p>length - 返される最大文字数</p> <p>output - オプションの出力フォーマットテンプレート。 \0 エスケープシーケンスは、テキストの一致した部分 (一致が始まる最初の文字から一致が終了する文字まで) に置き換えられますが、 \N (N=1... 9) エスケープシーケンスは、N 番目に一致したグループ (または N がキャプチャされたグループの数を超える場合は空の文字列) に置き換えられます。</p>	<p>host で指定されたリソースが存在しないか利用できない場合、このアイテムはサポートされなくなります。</p> <p>host は、ホスト名、ドメイン名、IPv4 または IPv6 アドレスにすることができます。ただし、IPv6 アドレスの場合、Zabbix エージェントは IPv6 サポートを有効にしてコンパイルする必要があります。</p> <p>output パラメータを使用したコンテンツ抽出はエージェントで行われます。</p> <p>output パラメータはバージョン 2.2 からサポートされています。</p> <p>例: => web.page.regexp[www.example.com,index.php,80,OK,2] => web.page.regexp[https://www.example.com,,,OK,2]</p>
----------------	---	---	--

agent 2 固有の item キー

Zabbix agent 2 は、Zabbix agent でサポートされている全ての item キーを Unix(/manual/config/items/itemtypes/zabbix_agent) と Windows(manual/config/items/itemtypes/zabbix_agent/win_keys) でサポートしています。このページでは、Zabbix agent 2 でのみ使用できる追加 item キーの詳細を、所属するプラグインごとにまとめています。

こちらをご参照ください。 [Plugins supplied out-of-the-box](#)

Note:

角括弧のないパラメータは必須です。角括弧 < > でマークされたパラメータはオプションです。

Ceph

キー	戻り値	パラメータ	コメント
ceph.df.details [connString, <user>, <apikey>]			
クラスタのデータ使用量とプール間の分散	JSON object	connString - URI またはセッション名 user,password - Ceph ログイン資格情報	
ceph.osd.stats [connString, <user>, <apikey>]			

キー		
集計および OSD ごとの統 計 ceph.osd.discovery [connString, <user>, <apikey>]	JSON object	connString - URI またはセッション名 user, password - Ceph ログイン資格情報
検出された OSD のリス ト。ローレ ベルディスカバ リに使用 ceph.osd.dump [connString, <user>, <apikey>]	JSON object	connString - URI またはセッション名 user, password - Ceph ログイン資格情報
OSD の使用し き値とステ ータス ceph.ping [connString, <user>, <apikey>]	JSON object	connString - URI またはセッション名 user, password - Ceph ログイン資格情報
Ceph への接 続を確立でき るかテスト	0 - 接続断 (AUTH や設定 の問題を含む エラーが表示 されている) 1 - 接続成功	connString - URI またはセッション名 user, password - Ceph ログイン資格情報
ceph.pool.discovery [connString, <user>, <apikey>]		
検出されたプ ールのリスト。 ローレベルデ ィスカバリで 使用 ceph.status [connString, <user>, <apikey>]	JSON object	connString - URI またはセッション名 user, password - Ceph ログイン資格情報
クラスタ全体 のステータス	JSON object	connString - URI またはセッション名 user, password - Ceph ログイン資格情報

Docker

キー			
説明	戻り値	パラメータ	コメント
docker.container_info [<ID>] コンテナに関 するローレベ ルインフォメ ーション	JSON としてシ リアル化され た Container- Inspect API 呼 び出しの出力	ID - コンテナの ID または名前	Agent2 ユーザー ('zabbix') は、十分な権限 を得るために 'docker' グループ に追加する 必要があります。そうしない場合、チェッ クは失敗します。
docker.container_stats [<ID>]			

キー

コンテナリソースの使用統計	JSONとしてシリアル化された ContainerStats API 呼び出しの出力と CPU 使用率	ID - コンテナの ID または名前	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.containers			
コンテナリスト	JSON としてシリアル化された ContainerList API 呼び出しの出力	-	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.containers.discovery[<options>]			
コンテナリスト。ローレベルディスクバリエーションに使用	JSON オブジェクト	options - 実行中のコンテナをすべて検出するか、実行中のコンテナのみを検出するかを指定します。サポートされている値: true - すべてのコンテナを返します。 false - 実行中のコンテナのみを返します。 (default)	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.data_usage			
現在のデータ使用量に関する情報	JSON としてシリアル化された SystemDataUsage API 呼び出しの出力	-	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.images			
イメージリスト	JSON としてシリアル化された ImageList API 呼び出しの出力	-	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.images.discovery			
画像一覧。ローレベルディスクバリエーションに使用	JSON オブジェクト	-	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.info			
システム情報	JSON としてシリアル化された SystemInfo API 呼び出しの出力	-	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。
docker.ping			
Docker デモーンが動作しているかどうかをテストします	1 - 動作しています 0 - 切断されています	-	Agent2 ユーザー ('zabbix') は、十分な権限を得るために 'docker' グループ に追加する必要があります。そうしない場合、チェックは失敗します。

Ember+

Key

Description	Return value	Parameters	Comments
ember.get[<uri>,<path>]			

Key		
Returns the result of the required device.	JSON object	uri - Ember+ device URI. Default: 127.0.0.1:9998 path - OID path to device. Empty by default, returns root collection data.

Memcached

キー			
説明	戻り値	パラメータ	コメント
memcached.ping	[connString,<user>,<password>]		
接続が有効かどうかをテストします	1 - 有効です 0 - 切断されています (AUTH や設定の問題を含むエラーが表示された場合)	connString - URI またはセッション名	
memcached.stats	[connString,<user>,<password>,<type>]		
STATS コマンドの出力を取得します	JSON - 出力は JSON としてシリアル化されます	connString - URI またはセッション名 user, password - Memcached ログイン資格情報 type - 返される統計タイプ: items,sizes,slabs または settings (デフォルトでは空で、一般統計を返します)	

MongoDB

キー			
説明	戻り値	パラメータ	コメント
mongodb.collection.stats	[connString,<user>,<password>,<database>,collection]		
特定のコレクションのさまざまなストレージ統計を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報 database - データベース名 (デフォルト: admin) collection - コレクション名	
mongodb.collections.discovery	[connString,<user>,<password>]		
発見されたコレクションのリストを返します。ローレベルディスクバリエーションに使用 (/manual/discovery/low_level_discovery/examples/systemd).	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報	
mongodb.collections.usage	[connString,<user>,<password>]		
コレクションの使用統計を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報	
mongodb.connpool.stats	[connString,<user>,<password>]		

キー

現在のデータベースインスタンスからシャードクラスターまたはレプリカセットの他のメンバーへの開いている発信接続に関する情報を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.db.stats[connString,<user>,<password>,<database>]</code>		
特定のデータベースシステムの状態を反映する統計を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報 database - データベース名 (デフォルト: admin)
<code>mongodb.db.discovery[connString,<user>,<password>]</code>		
検出されたデータベースのリストを返します。ローレベルディスクバリエーションに使用 (/manual/discovery/low_level_discovery/examples/systemd).	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.jumbo_chunks.count[connString,<user>,<password>]</code>		
ジャンボチャンクの数を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.oplog.stats[connString,<user>,<password>]</code>		
oplog からポーリングされたデータを使用して、レプリカセットのステータスを返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.ping[connString,<user>,<password>]</code>		
接続が有効かどうかをテストします	1 - 接続有効 0 - 接続無効 (AUTH や構成の問題を含むエラーが表示されている場合)	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.rs.config[connString,<user>,<password>]</code>		
レプリカセットの現在の設定を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.rs.status[connString,<user>,<password>]</code>		
メソッドが実行されるメンバーの観点からレプリカセットのステータスを返します。	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
<code>mongodb.server.status[connString,<user>,<password>]</code>		

キー		
データベースの状態を返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報
mongodb.sh.discovery[connString,<user>,<password>]		
クラスター内に存在する検出されたシャードのリストを返します	JSON object	connString - URI またはセッション名 user, password - MongoDB 認証情報

MQTT

キー			
説明	戻り値	パラメータ	コメント
mqtt.get[<broker_url>,topic,<username>,<password>]			
指定されたブローカーの特定のトピック (ワイルドカードを使用) をサブスクライブし、パブリケーションを待ちます。	トピックの内容によって異なります。 ワイルドカードを使用すると、トピックの内容がJSONとして返されます。	broker_url - MQTT ブローカー URL (空の場合、ポート 1883 の localhost が使用されます) topic - MQTT トピック (必須)。ワイルドカード (+, #) がサポートされています。 username,password - 認証資格情報 (必要な場合)	アイテムは アクティブチェック ('Zabbix エージェント (アクティブ)' アイテムタイプ) として構成する必要があります。 TLS 暗号化証明書は、デフォルトの場所 (Ubuntu の場合は /etc/ssl/certs/ ディレクトリなど) に保存することで使用できます。TLS の場合は、tls:// スキームを使用します。

MSSQL

Key			
Description	Return value	Parameters	Comments
mssql.availability.group.get[URI,<user>,<password>] Returns availability groups.	JSON object	URI - MSSQL server URI (the only supported schema is sqlserver://). Embedded credentials will be ignored; user, password - username, password to send to protected MSSQL server.	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
mssql.custom.query[URI,<user>,<password>,queryName,<args...>] Returns the result of a custom query.	JSON object	URI - MSSQL server URI (the only supported schema is sqlserver://). Embedded credentials will be ignored; user, password - username, password to send to protected MSSQL server; queryName - name of a custom query configured in Plugins.MSSQL.CustomQueriesDir without the .sql extension; args - one or several comma-separated arguments to pass to a query.	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
mssql.db.get Returns all available MSSQL databases.	JSON object		Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
mssql.job.status.get Returns the status of jobs.	JSON object		Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
mssql.last.backup.get			

Key

<code>mssql.local.db.get</code> Returns the last backup time for all databases.	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
<code>mssql.mirroring.get</code> Returns mirroring info.	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
<code>mssql.nonlocal.db.get</code> Returns databases that are participating in an Always On availability group and replica (primary or secondary) located on other servers (the database is not local to the SQL Server instance that the connection was established to).	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
<code>mssql.perfcounter.get</code> Returns the performance counters.	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
<code>mssql.ping</code> Ping the database. Test if connection is correctly configured.	1 - alive, 0 - not alive	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
<code>mssql.quorum.get</code>		

Key		
Returns the quorum info. mssql.quorum.members.get	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
Returns the quorum members. mssql.replica.get	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
Returns the replicas. mssql.version	JSON object	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.
Returns the MSSQL version.	String	Supported since Zabbix 6.0.27. For more information see the MSSQL plugin readme.

MySQL

キー			
説明	戻り値	パラメータ	コメント
mysql.db.discovery[connString, <username>, <password>] MySQL データベースのリスト。ローレベルディスカバリーに使用 (/manual/discovery/low_level_discovery/examples/systemd).	LLD JSON 形式の "show databases" SQL クエリ結果	connString - URI またはセッション名 username, password - MySQL 認証情報	
mysql.db.size[connString, <username>, <password>, dbName] バイト単位のデータベースサイズ	特定のデータベースに対する "select coalesce(sum(data_length + index_length), 0) as size from information_schema.tables where table_schema=?" SQL クエリの結果 (バイト単位)	connString - URI またはセッション名 username, password - MySQL 認証情報 dbName - データベース名	
mysql.get_status_variables[connString, <username>, <password>] グローバル変数の値	JSON 形式の "show global status" SQL クエリの結果	connString - URI またはセッション名 username, password - MySQL 認証情報	
mysql.ping[connString, <username>, <password>]			

キー

接続が有効かどうか	1 - 有効 0 - 無効 (AUTH や構成の問題など、何らかのエラーが表示される場合)	connString - URI またはセッション名 username, password - MySQL 認証情報
mysql.replication.discovery[connString, <username>, <password>]		
MySQL レプリケーションのリスト。ローレベルディスクカバリに使用 (/manual/discovery/low_level_discovery/examples/systemd).	LLD JSON 形式の "show slave status" SQL クエリの結果	connString - URI またはセッション名 username, password - MySQL 認証情報
mysql.replication.get_slave_status[connString, <username>, <password>, <master-Host>]		
レプリケーションステータス	JSON 形式の "show slave status" SQL クエリの結果	connString - URI またはセッション名 username, password - MySQL 認証情報 masterHost - レプリケーションマスターのホスト名。
mysql.version[connString, <username>, <password>]		
MySQL バージョン	MySQL インスタンスのバージョンを含む文字列	connString - URI またはセッション名 username, password - MySQL 認証情報

Oracle

キー

説明	戻り値	パラメーター	コメント
oracle.diskgroups.stats	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
ASM ディスクグループの統計			
oracle.diskgroups.discovery	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
ASM ディスクグループのリスト。ローレベルディスクカバリに使用			
oracle.archive.info	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
アーカイブログの統計			
oracle.cdb.info	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
CDBs 情報			

oracle.custom.query[connString,<user>,<password>,<service>,<args...>]		
カスタムクエリの結果	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名 queryName — name of a custom query (must be equal to a name of an sql file without an extension). args... — one or several comma-separated arguments to pass to a query.
oracle.datafiles.stats[connString,<user>,<password>,<service>]		
データファイルの統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.db.discovery[connString,<user>,<password>,<service>]		
データベースのリスト。ローレベルディスカバリに使用	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.fra.stats[connString,<user>,<password>,<service>]		
FRA 統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.instance.info[connString,<user>,<password>,<service>]		
インスタンス統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.pdb.info[connString,<user>,<password>,<service>]		
PDB 情報	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.pdb.discovery[connString,<user>,<password>,<service>]		
PDB のリスト。ローレベルディスカバリに使用	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.pga.stats[connString,<user>,<password>,<service>]		
PGA 統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.ping[connString,<user>,<password>,<service>]		
Oracle への接続を確立できるかどうかをテストします	0 - 接続失敗 (AUTH や構成の問題を含むエラーが表示された場合) 1 - 接続成功	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名
oracle.proc.stats[connString,<user>,<password>,<service>]		
プロセス統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名

キー

oracle.redolog.info[connString,<user>,<password>,<service>]			
制御ファイルからファイル情報をログに記録します	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
oracle.sga.stats[connString,<user>,<password>,<service>]			
SGA 統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]			
セッション統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名 lockMaxTime - 長時間ロックされたセッションとしてカウントする最大セッションロック期間 (秒単位)。デフォルト: 600 秒	
oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]			
システムメトリック値のセット	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名 duration - システムメトリック値の取得間隔 (秒単位)。使用可能な値: 60 — 長時間 (デフォルト), 15 — 短時間	
oracle.sys.params[connString,<user>,<password>,<service>]			
システムパラメータ値のセット	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
oracle.ts.stats[connString,<user>,<password>,<service>]			
表領域の統計	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
oracle.ts.discovery[connString,<user>,<password>,<service>]			
表領域のリスト。ローレベルディスクバリエーションに使用	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名	
oracle.user.info[connString,<user>,<password>,<service>,<username>]			
表領域のリスト。ローレベルディスクバリエーションに使用	JSON object	connString - URI またはセッション名 user, password - Oracle ログイン資格情報 service - Oracle サービス名 username - 情報が必要なユーザー名。小文字のユーザー名はサポートされていません。デフォルト: 現在のユーザー	

PostgreSQL

キー

説明	戻り値	パラメータ	コメント
pgsql.autovacuum.count[uri,<username>,<password>,<dbName>]			
オートバキュームワーカーの数	Integer	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名	

キー

pgsql.archive[uri,<username>,<password>,<dbName>]

アーカイブされたファイルに関する情報

JSON object

uri - URI またはセッション名
username, password - PostgreSQL 資格情報
dbName - データベース名

返されたデータは依存アイテムによって処理されます:
pgsql.archive.count_archived_files - 正常にアーカイブされた WAL ファイルの数
pgsql.archive.failed_trying_to_archive - WAL ファイルのアーカイブに失敗した回数
pgsql.archive.count_files_to_archive - アーカイブするファイルの数
pgsql.archive.size_files_to_archive - アーカイブするファイルのサイズ

pgsql.bgwriter[uri,<username>,<password>,<dbName>]

データベースクラスタのチェックポイントの合計数。チェックポイントタイプ別に分類されています。

JSON object

uri - URI またはセッション名
username, password - PostgreSQL 資格情報
dbName - データベース名

返されたデータは依存アイテムによって処理されます:
pgsql.bgwriter.buffers_alloc - 割り当てられたバッファの数
pgsql.bgwriter.buffers_backend - バックエンドによって直接書き込まれたバッファの数
pgsql.bgwriter.maxwritten_clean - 書き込みバッファが多すぎたため、バックグラウンドライターがクリーニングスキャンを停止した回数
pgsql.bgwriter.buffers_backend_fsync - バックエンドがバックグラウンドライターの代わりに独自の fsync 呼び出しを実行しなかった回数
pgsql.bgwriter.buffers_clean - バックグラウンドライターによって書き込まれたバッファの数
pgsql.bgwriter.buffers_checkpoint - チェックポイント中に書き込まれたバッファ数
pgsql.bgwriter.checkpoints_timed - 実行されたスケジュール済みチェックポイント数
pgsql.bgwriter.checkpoints_req - 要求されて実行されたチェックポイント数
pgsql.bgwriter.checkpoint_write_time - ファイルがディスクに書き込まれるチェックポイント処理に費やされた合計時間 (ミリ秒単位)
pgsql.bgwriter.sync_time - ファイルがディスクと同期されるチェックポイント処理に費やされた合計時間

pgsql.cache.hit[uri,<username>,<password>,<dbName>]

PostgreSQL バッファキャッシュヒット率

Float

uri - URI またはセッション名
username, password - PostgreSQL 資格情報
dbName - データベース名

pgsql.connections[uri,<username>,<password>,<dbName>]

キー

タイプ別接続.	JSON object	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名	返されたデータは依存アイテムによって処理されます: pgsql.connections.active - バックエンドがクエリを実行中 pgsql.connections.fastpath_function_call - バックエンドが高速パス関数を実行中 pgsql.connections.idle - バックエンドが新しいクライアントコマンドを待っています pgsql.connections.idle_in_transaction - バックエンドはトランザクション内にありますが、現在クエリを実行していません pgsql.connections.prepared - 準備された接続数 pgsql.connections.total - 接続の総数 pgsql.connections.total_pct - PostgreSQL サーバーの 'max\connections' 設定に関する合計接続数のパーセンテージ。 pgsql.connections.waiting - クエリ内の接続数 pgsql.connections.idle_in_transaction_aborted - バックエンドはトランザクション内にありますが、現在クエリを実行しておらず、トランザクション内のステートメントの1つがエラーを引き起こしました
pgsql.custom.query[uri,<username>,<password>,queryName[,args...]]	カスタムクエリの結果を返します	JSON object uri - URI またはセッション名 username, password - PostgreSQL 資格情報 queryName - カスタムクエリの名前。拡張子を除いた SQL ファイル名と一致する必要があります args (オプション) - クエリに渡す引数	
pgsql.dbstat[uri,<username>,<password>,dbName]			

キー

データベースごとに統計を収集します。ローレベルデイスカバリに使用

JSON object

uri - URI またはセッション名
username, password - PostgreSQL 資格情報
dbName - データベース名

返されたデータは依存アイテムによって処理されます:

pgsql.dbstat.numbackends["#{DBNAME}"]

- このデータベースのバックエンドによるデータファイルブロックの読み取り時間 (ミリ秒単位)

pgsql.dbstat.sum.blk_read_time["#{DBNAME}"]

- このデータベースのバックエンドによるデータファイルブロックの読み取り時間 (ミリ秒単位)

pgsql.dbstat.sum.blk_write_time["#{DBNAME}"]

- このデータベースでバックエンドがデータファイルブロックを書き込むのにかった時間 (ミリ秒単位)

pgsql.dbstat.sum.checksum_failures["#{DBNAME}"]

- (共有オブジェクトで) 検出されたデータページチェックサムエラーの数、またはデータチェックサムが有効でない場合は NULL (PostgreSQL バージョン 12 のみ)

pgsql.dbstat.blks_read.rate["#{DBNAME}"]

- このデータベースで読み取られたディスクブロック数

pgsql.dbstat.deadlocks.rate["#{DBNAME}"]

- このデータベースで検出されたデッドロック数

pgsql.dbstat.blks_hit.rate["#{DBNAME}"]

- ディスクブロックがバッファキャッシュ内ですで見つかったため、読み取りが不要になった回数 (これには、オペレーティングシステムのファイルシステムキャッシュではなく、PostgreSQL Pro バッファキャッシュ内のヒットのみが含まれます)

pgsql.dbstat.xact_rollback.rate["#{DBNAME}"]

- このデータベース内のロールバックされたトランザクションの数

pgsql.dbstat.xact_commit.rate["#{DBNAME}"]

- このデータベースでコミットされたトランザクションの数

pgsql.dbstat.tup_updated.rate["#{DBNAME}"]

- このデータベースでクエリによって更新された行数

pgsql.dbstat.tup_returned.rate["#{DBNAME}"]

- このデータベースのクエリによって返される行数

pgsql.dbstat.tup_inserted.rate["#{DBNAME}"]

- このデータベースのクエリによって挿入された行数

pgsql.dbstat.tup_fetched.rate["#{DBNAME}"]

- このデータベースでクエリによってフェッチされた行数

pgsql.dbstat.tup_deleted.rate["#{DBNAME}"]

- このデータベースでクエリによって削除された行数

pgsql.dbstat.conflicts.rate["#{DBNAME}"]

- このデータベースでリカバリとの競合が原因でキャンセルされたクエリの数 (競合はスタンバイサーバーでのみ発生します)

pgsql.dbstat.temp_files.rate["#{DBNAME}"]

- このデータベースでクエリによって作成された一時ファイルの数。log_temp_files の設定や、一時ファイルが作成された理由 (並べ替えやハッシュなど) に関係なく、すべての一時ファイルがカウントされます。

pgsql.dbstat.temp_bytes.rate["#{DBNAME}"]

- このデータベースのクエリによって一時フ

キ-

pgsql.dbstat.sum[uri,<username>,<password>,
<dbName>]

キー

クラスター内のすべてのデータベースの要約データ	JSON object	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名	返されたデータは依存アイテムによって処理されます: pgsql.dbstat.numbackends - このデータベースに現在接続されているバックエンドの数 pgsql.dbstat.sum.blk_read_time - このデータベースのバックエンドによるデータファイルブロックの読み取り時間 (ミリ秒単位) pgsql.dbstat.sum.blk_write_time - このデータベースでバックエンドがデータファイルブロックを書き込む時間 (ミリ秒単位) pgsql.dbstat.sum.checksum_failures - (共有オブジェクトで) 検出されたデータページチェックサムエラーの数、またはデータチェックサムが有効になっていない場合は NULL (PostgreSQL バージョン 12 のみ) pgsql.dbstat.sum.xact_commit - このデータベースでコミットされたトランザクションの数 pgsql.dbstat.sum.conflicts - スタンバイサーバーでのリカバリとの競合によるクエリのキャンセルに関するデータベース統計 pgsql.dbstat.sum.deadlocks - このデータベースで検出されたデッドロックの数 pgsql.dbstat.sum.blks_read - このデータベースで読み取られたディスクブロックの数 pgsql.dbstat.sum.blks_hit - バッファキャッシュ内でディスクブロックが既に見つかったため、読み取りは不要だった回数 (PostgreSQL Pro バッファキャッシュ内のヒットのみが含まれます) pgsql.dbstat.sum.temp_bytes - このデータベースのクエリによって一時ファイルに書き込まれたデータの合計量。log_temp_files の設定や、一時ファイルが作成された理由 (並べ替えやハッシュなど) に関係なく、すべての一時ファイルのデータが含まれます。 pgsql.dbstat.sum.temp_files - このデータベースでクエリによって作成された一時ファイルの数。log_temp_files の設定や、一時ファイルが作成された理由 (並べ替えやハッシュなど) に関係なく、すべての一時ファイルがカウントされます。 pgsql.dbstat.sum.xact_rollback - このデータベースでロールバックされたトランザクションの数 pgsql.dbstat.sum.tup_deleted - このデータベースでクエリによって削除された行数 pgsql.dbstat.sum.tup_fetched - このデータベースでクエリによってフェッチされた行数 pgsql.dbstat.sum.tup_inserted - このデータベースのクエリによって挿入された行数 pgsql.dbstat.sum.tup_returned - このデータベースのクエリによって返される行数 pgsql.dbstat.sum.tup_updated - このデータベースでクエリによって更新された行数
-------------------------	-------------	--	---

キー

pgsql.db.age[uri,<username>,<password>,
<dbName>]

データベース Integer **uri** - URI またはセッション名
の最も古い **username, password** - PostgreSQL 資格
FrozenXID の 情報
経過時間。 **dbName** - データベース名
ローレベルディス
スカバリに使用

pgsql.db.bloating_tables[uri,<username>,<password>,
<dbName>]

データベース Integer **uri** - URI またはセッション名
あたりの肥大 **username, password** - PostgreSQL 資格
化したテーブル 情報
の数。 **dbName** - データベース名
ローレ
レベルディス
スカバリに使用

pgsql.db.discovery[uri,<username>,<password>,
<dbName>]

PostgreSQL デ JSON object **uri** - URI またはセッション名
ータベースの **username, password** - PostgreSQL 資格
リスト。 **dbName** - データベース名
ローレ
レベルディス
スカバリに使用

pgsql.db.size[uri,<username>,<password>,
<dbName>]

データベース Integer **uri** - URI またはセッション名
のサイズ (バ **username, password** - PostgreSQL 資格
イト単位)。 **dbName** - データベース名
ローレ
レベルディス
スカバリに使用

pgsql.locks[uri,<username>,<password>,
<dbName>]

データベース JSON object **uri** - URI またはセッション名
ごとに付与さ **username, password** - PostgreSQL 資格
れたロックに 情報
関する情報。 **dbName** - データベース名
ローレ
レベルディス
スカバリに使用

返されたデータは依存アイテムによって処
理されます:

pgsql.locks.shareupdateexclusive["{#DBNAME}"]
- 共有更新の排他ロックの数
pgsql.locks.accessexclusive["{#DBNAME}"]
- アクセス排他ロックの数
pgsql.locks.accessshare["{#DBNAME}"]
- アクセス共有ロックの数
pgsql.locks.exclusive["{#DBNAME}"]
- 排他ロックの数
pgsql.locks.rowexclusive["{#DBNAME}"]
- 行排他ロックの数
pgsql.locks.rowshare["{#DBNAME}"]
- 行共有ロックの数
pgsql.locks.share["{#DBNAME}"] - 共
有ロックの数
pgsql.locks.sharerowexclusive["{#DBNAME}"]
- 共有行の排他ロックの数

pgsql.oldest.xid[uri,<username>,<password>,
<dbName>]

最古の XID の Integer **uri** - URI またはセッション名
経過時間 **username, password** - PostgreSQL 資格
情報
dbName - データベース名

pgsql.ping[uri,<username>,<password>,
<dbName>]

キー

接続が有効かどうかをテストします	1 - 接続成功 0 - 接続失敗 (AUTH や構成の問題を含むエラーが表示される等)	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名	
pgsql.queries[uri,<username>,<password>,<dbName>,<timePeriod>]			
クエリの実行時間測定	JSON object	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名 timePeriod - スロークエリの実行時間制限 (正の整数である必要があります)	返されたデータは依存アイテムによって処理されます: pgsql.queries.mro.time_max["{#DBNAME}"] - 最大メンテナンスクエリ時間 pgsql.queries.query.time_max["{#DBNAME}"] - 最大クエリ時間 pgsql.queries.tx.time_max["{#DBNAME}"] - 最大トランザクションクエリ時間 pgsql.queries.mro.slow_count["{#DBNAME}"] - スローメンテナンスクエリ数 pgsql.queries.query.slow_count["{#DBNAME}"] - スロークエリ数 pgsql.queries.tx.slow_count["{#DBNAME}"] - スロートランザクションクエリ数 pgsql.queries.mro.time_sum["{#DBNAME}"] - 合計メンテナンスクエリ時間 pgsql.queries.query.time_sum["{#DBNAME}"] - 合計クエリ時間 pgsql.queries.tx.time_sum["{#DBNAME}"] - 合計トランザクションクエリ時間
pgsql.replication.count[uri,<username>,<password>]			
スタンバイサーバー数	Integer	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.replication.process[uri,<username>,<password>]			
各送信プロセスごとのフラッシュラグ、書き込みラグ、リプレイラグ	JSON object	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.replication.process.discovery[uri,<username>,<password>]			
レプリケーションプロセス名検出	JSON object	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.replication.recovery_role[uri,<username>,<password>]			
回復状態	0 - マスターモード 1 - 回復進行中 (スタンバイモード)	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.replication.status[uri,<username>,<password>]			
レプリケーションステータス	0 - ストリーミングダウン 1 - ストリーミングアップ 2 - マスターモード	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.replication_lag.b[uri,<username>,<password>]			

この項目は Zabbix 6.0.3 以降でサポートされています

キー

レプリケーションラグ (バイト単位)	Integer	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.replication_lag.sec[uri,<username>,<password>]			
秒単位のレプリケーションラグ	Integer	uri - URI またはセッション名 username, password - PostgreSQL 資格情報	
pgsql.uptime[uri,<username>,<password>,<dbName>]			
PostgreSQL の稼働時間 (ミリ秒)	Float	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名	
pgsql.wal.stat[uri,<username>,<password>,<dbName>]			
WAL 統計	JSON object	uri - URI またはセッション名 username, password - PostgreSQL 資格情報 dbName - データベース名	返されたデータは依存アイテムによって処理されます: pgsql.wal.count - WAL ファイル数 pgsql.wal.write - 使用された WAL lsn (バイト単位)

Redis

キー

説明	戻り値	パラメータ	コメント
redis.config[connString,<password>,<pattern>]			
パターンに一致する Redis インスタンスの構成パラメーターを取得します	JSON - glob スタイルのパターンが使用された場合 単一値 - パターンにワイルドカード文字が含まれていない場合	connString - URI またはセッション名 password - Redis パスワード pattern - glob スタイルパターン (デフォルトで *)	
redis.info[connString,<password>,<section>]			
INFO コマンドの出力を取得します	JSON - 出力はシリアル化されます	connString - URI またはセッション名 password - Redis パスワード section - セクション の情報 (デフォルトは default)	
redis.ping[connString,<password>]			
接続が有効かどうか	1 - 有効 0 - 無効 (AUTH や構成の問題など、何らかのエラーが表示される場合)	connString - URI またはセッション名 password - Redis パスワード	
redis.slowlog.count[connString,<password>]			
Redis が開始されてからのスローログエントリの数	Integer	connString - URI またはセッション名 password - Redis パスワード	

キー

説明	戻り値	パラメータ	コメント
smart.attribute.discovery S.M.A.R.T. デバイス属性のリストを返します	JSON object		次のマクロとその値が返されます: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH}. HDD、SSD、および NVME ドライブタイプがサポートされています。ドライブは単独でも、RAID で組み合わせてもかまいません。{#NAME} には RAID の場合アドオンがあります。例: {"#NAME": "/dev/sda cciss,2"}
smart.disk.discovery S.M.A.R.T. デバイスのリストを返します	JSON object		次のマクロとその値が返されます: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE}. HDD、SSD、および NVME ドライブタイプがサポートされています。ドライブが RAID に属していない場合、{#RAIDTYPE} は空になります。{#NAME} には RAID の場合のアドオンがあります。例: {"#NAME": "/dev/sda cciss,2"}
smart.disk.get[<path>,<raid_type>] S.M.A.R.T. デバイスの使用可能なすべてのプロパティを返します。	JSON object	path (Zabbix 6.0.4 以降) - ディスクパス。 {#PATH} マクロを値として使用できます raid_type (Zabbix 6.0.4 以降) - RAID タイプ。 {#RAID} マクロを値として使用できます	HDD、SSD、および NVME ドライブタイプがサポートされています。ドライブは単独でも、RAID で組み合わせてもかまいません。 データには、smartctl のバージョンと呼び出しの引数、および追加のフィールドが含まれます: disk_name - RAID 検出に必要なアドオンの名前を保持します。例: {"disk_name": "/dev/sda cciss,2"} disk_type - ディスクタイプ HDD、SSD、または NVME を保持します。例: {"disk_type": "ssd"} パラメータが指定されていない場合、アイテムはすべてのディスクに関する情報を返します。

Systemd

キー

説明	戻り値	パラメータ	コメント
systemd.unit.get[unit name,<interface>] systemd ユニットのすべてのプロパティを返します	JSON object	unit name - ユニット名 (アイテムプロトタイプで {#UNIT.NAME} マクロを使用して名前を見つけることができます) interface - ユニットインターフェイスタイプ。使用可能な値: Unit (デフォルト), Service, Socket, Device, Mount, Automount, Swap, Target, Path	この項目は Linux プラットフォームでのみサポートされています。 Unit インターフェイスの LoadState、ActiveState、および UnitFileState (は、テキストと整数として返されます。 "ActiveState": {"state": 1, "text": "active"}
systemd.unit.info[unit name,<property>,<interface>]			

キー

Systemd ユニ ットインフォ メーション	String	unit name - ユニット名 (アイテムプロト タイプで <code>{#UNIT.NAME}</code> マクロを使用して 名前を見つけることができます) property - unit プロパティ (例: ActiveState (デフォ ルト), LoadState, Description) interface - ユニットインターフェイスタイ プ (例: Unit (デフォルト), Socket, Service)	dbus API で説明されているように、このア イテムを使用すると、特定のタイプのイン ターフェイスから特定のプロパティを取得 できます。 この項目は Linux プラットフォームでのみ サポートされています。 例: => <code>systemd.unit.info["{#UNIT.NAME}"]</code> - - 検出された systemd ユニットのアクティブ 状態 (アクティブ、リロード中、非アクティ ブ、失敗、アクティブ化、非アクティブ化) 情報を収集します => <code>sys- temd.unit.info["{#UNIT.NAME}", LoadState]</code> - 検出された systemd ユニットの負荷状態 情報を収集する => <code>systemd.unit.info[mysql.service, Id]</code> - - サービス技術名の取得 (mysql.service) => <code>sys- temd.unit.info[mysql.service, Description]</code> - サービスの説明を取得する (MySQL Server) => <code>sys- temd.unit.info[mysql.service, ActiveEnterTimestamp]</code> - サービスが最後にアクティブ状態になった 時刻を取得する (1562565036283903) => <code>sys- temd.unit.info[dbus.socket, NConnections, Socket]</code> - このソケットユニットからの接続数を収集 する
systemd.unit.discovery[<type>] systemd ユニ ットとその詳 細のリスト。 ローレベルデ ィスカバリに 使用 (/man- ual/discovery/low_level_discovery/examples/systemd).	JSON object	type - 使用可能な値: all, automount, device, mount, path, service (デフォルト), socket, swap, target	この項目は、Linux プラットフォームでの みサポートされています。

Web 証明書

キー

説明	戻り値	パラメータ	コメント
web.certificate.get[hostname, <port>, <address>]			

キー

証明書を検証し、証明書の詳細を返します	JSON object	<p>hostname - IP または DNS のいずれか URL スキーム (https のみ)、パス (無視されます)、およびポートを含めることができます。最初と 2 番目のパラメーターの両方でポートが指定されている場合、それらの値は一致する必要があります。</p> <p>アドレス (3 番目のパラメーター) が指定されている場合、ホスト名は SNI とホスト名の検証にのみ使用されます。</p> <p>port - ポート番号 (HTTPS のデフォルトは 443)</p> <p>address - IP または DNS のいずれか。指定すると、接続に使用され、ホスト名 (第 1 パラメーター) が SNI とホスト検証に使用されます。</p> <p>1 番目のパラメーターが IP で 3 番目のパラメーターが DNS の場合、1 番目のパラメーターは接続に使用され、3 番目のパラメーターは SNI とホストの検証に使用されます。</p>	<p>このアイテムは、host で指定されたリソースが存在しないか利用できない場合、または TLS ハンドシェイクが無効な証明書以外のエラーで失敗した場合にサポートされなくなります。</p> <p>現在、AIA (Authority Information Access) X.509 拡張機能、CRL および OCSP (OCSP ステータスを含む)、Certificate Transparency、およびカスタム CA トラストストアはサポートされていません。</p>
---------------------	-------------	---	---

Windows 固有の item キー

Item キー

表は、Zabbix Windows agent で使用できる固有の item キーの詳細です。

こちらもご参照ください。 [Minimum permission level for Windows agent items](#)

Key

Description	戻り値	パラメータ	コメント
eventlog[name,<regex>,<severity>,<source>,<eventid>,<maxlines>,<mode>]			

Event log monitoring.	Log	<p>name - name of event log</p> <p>regexp - regular expression describing the required pattern</p> <p>severity - regular expression describing severity</p> <p>This parameter accepts the following values: "Information", "Warning", "Error", "Critical", "Verbose" (since Zabbix 2.2.0 running on Windows Vista or newer)</p> <p>source - regular expression describing source identifier (regular expression is supported since Zabbix 2.2.0)</p> <p>eventid - regular expression describing the event identifier(s)</p> <p>maxlines - maximum number of new lines per second the agent will send to Zabbix server or proxy. This parameter overrides the value of 'MaxLinesPerSecond' in <code>zabbix_agentd.win.conf</code></p> <p>mode - possible values: all (default), skip - skip processing of older data (affects only newly created items).</p>	<p>The item must be configured as an active check.</p> <p>Examples:</p> <pre>=> eventlog[Application] => eventlog[Security,"Failure Audit",,"^(529 680)\$] => event- log[System,"Warning Error"] => eventlog[System,,,,^1\$] => event- log[System,,,,@TWOSHORT] - here a custom regular expression named TWOSHORT is referenced (defined as a Result is TRUE type, the expression itself being ^1\$\ ^70\$).</pre> <p>Note that the agent is unable to send in events from the "Forwarded events" log.</p> <p>The <code>mode</code> parameter is supported since Zabbix 2.0.0. "Windows Eventing 6.0" is supported since Zabbix 2.2.0.</p> <p>Note that selecting a non-Log type of information for this item will lead to the loss of local timestamp, as well as log severity and source information.</p>
net.if.list	Text	<p>Network interface list (includes interface type, status, IPv4 address, description).</p>	<p>Supported since Zabbix agent version 1.8.1. Multi-byte interface names supported since Zabbix agent version 1.8.6. Disabled interfaces are not listed.</p> <p>Note that enabling/disabling some components may change their ordering in the Windows interface name.</p> <p>Some Windows versions (for example, Server 2008) might require the latest updates installed to support non-ASCII characters in interface names.</p>
perf_counter[counter,<interval>]			

Key

Value of any Windows performance counter.	Integer, float, string or text (depending on the request)	counter - path to the counter interval - last N seconds for storing the average value. The <code>interval</code> must be between 1 and 900 seconds (included) and the default value is 1.	Performance Monitor can be used to obtain list of available counters. Until version 1.6 this parameter will return correct value only for counters that require just one sample (like <code>\System\Threads</code>). It will not work as expected for counters that require more than one sample - like CPU utilization. Since 1.6, <code>interval</code> is used, so the check returns an average value for last "interval" seconds every time. See also: Windows performance counters .
perf_counter_en[counter,<interval>] Value of any Windows performance counter in English.	Integer, float, string or text (depending on the request)	counter - path to the counter in English interval - last N seconds for storing the average value. The <code>interval</code> must be between 1 and 900 seconds (included) and the default value is 1.	This item is only supported on Windows Server 2008/Vista and above. You can find the list of English strings by viewing the following registry key: <code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009</code> . Supported since Zabbix agent versions 4.0.13 and 4.2.7.
perf_instance.discovery[object] List of object instances of Windows performance counters. Used for low-level discovery .	JSON object	object - object name (localized)	Supported since Zabbix agent version 5.0.1.
perf_instance_en.discovery[object] List of object instances of Windows performance counters, discovered using object names in English. Used for low-level discovery .	JSON object	object - object name (in English)	Supported since Zabbix agent version 5.0.1.
proc_info[process,<attribute>,<type>]			

Various information about specific process(es).

Float

process - process name
attribute - requested process attribute
type - representation type (meaningful when more than one process with the same name exists)

The following attributes are supported:

- vmsize (default) - size of process virtual memory in Kbytes
- wkset - size of process working set (amount of physical memory used by process) in Kbytes
- pf - number of page faults
- ktime - process kernel time in milliseconds
- utime - process user time in milliseconds
- io_read_b - number of bytes read by process during I/O operations
- io_read_op - number of read operation performed by process
- io_write_b - number of bytes written by process during I/O operations
- io_write_op - number of write operation performed by process
- io_other_b - number of bytes transferred by process during operations other than read and write operations
- io_other_op - number of I/O operations performed by process, other than read and write operations
- gdiobj - number of GDI objects used by process
- userobj - number of USER objects used by process

Valid types are:

- avg (default) - average value for all processes named <process>
- min - minimum value among all processes named <process>
- max - maximum value among all processes named <process>
- sum - sum of values for all processes named <process>

Examples:

```
=>
proc_info[iexplore.exe,wkset,sum]
- to get the amount of physical
memory taken by all Internet
Explorer processes
=>
proc_info[iexplore.exe,pf,avg] -
to get the average number of
page faults for Internet Explorer
processes
```

Note that on a 64-bit system, a 64-bit Zabbix agent is required for this item to work correctly.

Note: io_*, gdiobj and userobj attributes are available only on Windows 2000 and later

Key

service.discovery

List of Windows services. Used
for **low-level discovery**.

JSON
object

Supported since Zabbix agent
version 3.0.

service.info[service,<param>]

Information about a service.

Integer -
with
`param` as
state,
startup

String -
with
`param` as
display-
name,
path, user

Text - with
`param` as
descrip-
tion

Specifically
for state:

0 -
running,
1 -
paused,
2 - start
pending,
3 - pause
pending,
4 -
continue
pending,
5 - stop
pending,
6 -
stopped,
7 -
unknown,
255 - no
such
service

Specifically
for

startup:
0 - auto-
matic,
1 -
automatic
delayed,
2 -
manual,
3 -
disabled,
4 -
unknown,
5 -
automatic
trigger
start,
6 -
automatic
delayed
trigger
start,
7 -

service - a real service name or
its display name as seen in MMC
Services snap-in

param - state (default),
displayname, path, user,
startup or description

Examples:

=> `service.info[SNMPTRAP]` -
state of the SNMPTRAP service
=> `service.info[SNMP Trap]` -
state of the same service, but
with display name specified
=>
`service.info[EventLog,startup]` -
startup type of the EventLog
service

Items `service.info[service,state]`
and `service.info[service]` will
return the same information.

Note that only with `param` as
state this item returns a value
for non-existing services (255).

This item is supported since
Zabbix 3.0.0. It should be used
instead of the deprecated
`service_state[service]` item.

Key

services[<type>,<state>,<exclude>]

Listing of services.

0 - if empty

Text - list of services separated by a newline

type - all (default), automatic, manual or disabled

state - all (default), stopped, started, start_pending, stop_pending, running, continue_pending, pause_pending or paused

exclude - services to exclude from the result. Excluded services should be listed in double quotes, separated by comma, without spaces.

Examples:

=> services[,started] - list of started services
=> services[automatic,stopped] - list of stopped services, that should be run
=> services[automatic,stopped,"service1,service2,service3"] - list of stopped services, that should be run, excluding services with names service1, service2 and service3

The exclude parameter is supported since Zabbix 1.8.1.

wmi.get[<namespace>,<query>]

Execute WMI query and return the first selected object.

Integer, float, string or text (depending on the request)

namespace - WMI namespace
query - WMI query returning a single object

WMI queries are performed with [WQL](#).

Example:

=> wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] - returns the status of the first physical disk

This key is supported since Zabbix 2.2.0.

wmi.getall[<namespace>,<query>]

Execute WMI query and return the whole response.

JSON object

namespace - WMI namespace
query - WMI query

WMI queries are performed with [WQL](#).

Example:

=> wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] - returns status information of physical disks

JSONPath [preprocessing](#) can be used to point to more specific values in the returned JSON.

This key is supported since Zabbix 4.4.0.

vm.vmemory.size[<type>]

Key			
Virtual memory size in bytes or in percentage from total.	Integer - for bytes Float - for percentage	type - possible values: available (available virtual memory), pavailable (available virtual memory, in percent), pused (used virtual memory, in percent), total (total virtual memory, default), used (used virtual memory)	Example: => vm.vmemory.size[pavailable] → available virtual memory, in percentage Monitoring of virtual memory statistics is based on: * Total virtual memory on Windows (total physical + page file size); * The maximum amount of memory Zabbix agent can commit; * The current committed memory limit for the system or Zabbix agent, whichever is smaller. This key is supported since Zabbix 3.0.7 and 3.2.3.

Windows サービスの監視

このチュートリアルでは、Windows サービスの監視を設定するためのステップバイステップの手順を説明します。Zabbix server と agent が設定され、動作していることを前提としています。

Step 1

サービス名を取得します。

この名前は、MMC サービススナップインを開き、サービスのプロパティを表示することで取得できます。
 General タブに 'Service name' というフィールドがあります。
 この後の値は、監視のための項目を設定する際に使用する名前です。

例えば、"workstation" サービスを監視する場合、次のようになります。
 lanmanworkstation

Step 2

サービスを監視するための item の設定 [Configure an item] (/manual/config/items/item) を行います。

service.info[service,<param>] は、特定のサービスの情報を取得する item です。
 必要な情報に応じて、次の値を受け付ける param オプションを指定します：
 displayname、state、path、user、startup または description
 service.info<service> の param が指定されない場合、デフォルト値は state です。

戻り値のタイプは選択された param に依存します:
 state と startup の場合は整数値、displayname、path、user は文字列、description はテキスト。
 説明 (description) には文字列を指定します。

例

- Key: service.info[lanmanworkstation]
- Type of information : Numeric (符号なし)
- Show value: Windows service state 値のマッピングを選択します。

Windows service state と Windows service startup type の 2 つのバリュemapがあり、
 フロントエンドで数値からテキスト表現にマッピングすることができます。

Windows サービスの検出

Low-level discoveryは、コンピュータ上のさまざまなエンティティに対して、
 item、trigger、およびグラフを自動的に作成する方法を提供します。
 Zabbix は自動的にマシンの Windows サービスの監視を開始します。
 サービス名を正確に把握したり、各サービス用の item を手動で作成する必要はありません。

フィルタを使用することで、監視したいサービスのみに対して実際の item、trigger、グラフを生成することができます。

2 SNMP agent

概要

SNMP モニタリングは、プリンター、ネットワークスイッチ、UPS など、通常 SNMP が有効なデバイスで使用することができます。

 OS と Zabbix agent のセットアップを行うことは現実的ではありません。

これらの機器の SNMP エージェントが提供するデータを取得することができます。
 Zabbix server で **initially configured** を行い、SNMP をサポートする必要があります。

SNMP チェックは UDP プロトコルのみで行われます。

Zabbix server と proxy デーモンは、SNMP デバイスに 1 回のリクエストで複数の値を問い合わせます。
 これは全ての種類の SNMP item (通常の SNMP item、動的インデックスを持つ SNMP item、SNMP ローレベルディスクカバリ) に影響し、
 SNMP 処理をより効率的に行えるようになります。
 詳しくは、**bulk processing** セクションを参照してください。
 バルクリクエストは、各インターフェイスの "Use bulk requests" 設定を使用して、
 適切に処理できないデバイスのために無効にすることもできます。

Zabbix server と proxy デーモンが不正な SNMP レスポンスを受信した場合、以下のようなログが記録されます。

ホスト "gateway" からの SNMP レスポンスには、要求されたすべての変数バインディングが含まれていません。

このログは、すべての問題となるケースをカバーしているわけではありませんが、バルクリクエストを無効にすべき
 個々の SNMP デバイスを特定するのに役立ちます。

Zabbix server / proxy は、SNMP ライブラリの再試行メカニズム、または内部の **bulk processing** メカニズムにより、クエリに失敗した後、少なくとも 1 回再試行します。

Warning:

SNMPv3 デバイスを監視する場合、以下のことを確認してください。
 msgAuthoritativeEngineID (snmpEngineID または "Engine ID" と呼ばれる) が 2 つのデバイスで
 共有されていないことを確認してください。RFC2571 (セクション 3.1.1.1) の規定により、
 各デバイスで一意的である必要があります。

Warning:

RFC3414 では、SNMPv3 デバイスが engineBoots を永続化することを要求しています。
 一部のデバイスはこれを行わず、その結果、再起動後にそのデバイスの SNMP メッセージは、再起動後に古いメッセージとして破棄されます。
 このような場合、server / proxy で SNMP キャッシュを手動でクリアするか ([R snmp_cache_reload(/manual/concepts/server#runtime_control)]
 またはサーバー/プロキシを再起動する必要があります。

SNMP モニタリングの設定

SNMP によるデバイスの監視を開始するには、次の手順を実行する必要があります。

Step1

監視する item の SNMP スtring (または OID) を見つけます。

SNMP スtring のリストを取得するには、**snmpwalk** コマンドを使用します
 net-snmp ソフトウェア (Zabbix の一部としてインストールされているはず) を使用します。
 または同等のツールを使用します。

```
shell> snmpwalk -v 2c -c public <ホスト IP>
```

2c は SNMP のバージョンを表し、次のように置き換えることもできます。
 '1' に置き換えて、デバイスの SNMP バージョン 1 を表します。

これにより、SNMP 文字列とその最後の値のリストが表示されるはず。
 もし SNMP の 'community' が標準の 'public' と異なっている可能性があり、その場合、それが何であるかを確認する必要があります。

例えば、ポート 3 のスイッチに入ってくるパケットを監視したい場合、次の行から IF-MIB::ifInOctets.3 という文字列を使用します。この行にある

```
IF-MIB::ifInOctets.3 = Counter32: 3409739121
```

これで、**snmpget** コマンドを使用して、「IF-MIB::ifInOctets.3」の数値 OID を確認することができます。
 'IF-MIB::ifInOctets.3' の数値 OID を調べるために、**snmpget** コマンドを使用します。

文字列の最後の数字が、監視対象のポート番号であることに注意してください。
 こちらも参照してください。Dynamic indexes

これにより、以下のようなものが得られるはず。


```
.1.3.6.1.2.1.2.1.10.3 = Counter32: 3472126941
```

繰り返しますが、OID の最後の数字がポート番号です。

Note:

3COM は 3 桁のポート番号を使うようです。
例えばポート 1 = ポート 101、ポート 3 = ポート 103 のように 3 桁のポート番号を使うようですが、
Cisco はポート 3 = 3 のように普通の数字を使います。

Note:

最もよく使用される SNMP OID のいくつかは、Zabbix によって
 translated automatically to a numeric representation で定義されています。

上記の例では、値のタイプは "Counter32" であり、これは内部的には ASN_COUNTER タイプに相当します。
 サポートされるタイプの完全なリストは以下のとおりです。
 ASN_COUNTER, ASN_COUNTER64, ASN_UNSIGNED64,
 ASN_INTEGER, ASN_INTEGER64, ASN_FLOAT, ASN_DOUBLE, ASN_TIMETICKS,
 ASN_GAUGE, ASN_IPADDRESS, ASN_OCTET_STR and ASN_OBJECT_ID (バージョン 2.2.8, 2.4.3 から)
 これらのタイプは、おおよそ **snmpget** のアウトプットの
 "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks",
 "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" に対応しますが、"STRING", "Hex-STRING", "OID" などと
 表示されることもあります。これらは表示ヒントの有無によって異なります。

Step 2

Create a host は、デバイスに対応します。

Host	Templates	IPMI	Tags	Macros	Inventory	Encryption	Value mapping
* Host name	SNMP device host						
Visible name	SNMP device host						
* Groups	Discovered hosts × type here to search						
Interfaces	Type	IP address	DNS name				
	Agent	127.0.0.1					
	SNMP	127.0.0.1					
* SNMP version	SNMPv2 ▼						
* SNMP community	{\${SNMP_COMMUNITY}}						
	<input checked="" type="checkbox"/> Use bulk requests						

ホストの SNMP インタフェースを追加します:

- IP アドレス/DNS 名とポート番号を入力します。

- ドロップダウンから SNMP バージョンを選択します。

- 選択した SNMP のバージョンに応じて、インターフェイスの認証情報を追加します。

 - SNMPv1, v2 community のみが必要です (一般的には 'public')

 - SNMPv3 より多くの情報が必要です (下記をご参照ください)

- SNMP リクエストのバルクリクエストを可能にするため、Use bulk requests のチェックボックスをマークしたままにします。

SNMPv3 パラメータ	説明
Context name	Enter context name to identify item on SNMP subnet. Context name is supported for SNMPv3 items since Zabbix 2.2. User macros are resolved in this field.
Security name	Enter security name. User macros are resolved in this field.
Security level	Select security level: noAuthNoPriv - no authentication nor privacy protocols are used AuthNoPriv - authentication protocol is used, privacy protocol is not AuthPriv - both authentication and privacy protocols are used
Authentication protocol	Select authentication protocol - MD5, SHA1, SHA224, SHA256, SHA384 or SHA512.
Authentication passphrase	Enter authentication passphrase. User macros are resolved in this field.
Privacy protocol	Select privacy protocol - DES, AES128, AES192, AES256, AES192C (Cisco) or AES256C (Cisco).
Privacy passphrase	Enter privacy passphrase. User macros are resolved in this field.

SNMPv3 の認証情報 (security name , authentication protocol/passphrase , privacy protocol) が間違っている場合、
 Zabbix は net-snmp から ERROR を受け取ります。
 ただし、Privacy passphrase が間違っている場合、Zabbix は net-snmp から TIMEOUT エラーを受信します。

Warning:

Authentication protocol、Authentication passphrase、Privacy protocol、Privacy passphrase を変更します。
 server / proxy 上のキャッシュが有効になった後、キャッシュが手動でクリアされた後 (-R snmp_cache_reload)、server/proxy が再起動された場合、または
 Security name が変更された場合、すべてのパラメータが即座に更新されます。

提供されている SNMP テンプレート (Template SNMP Device など) を使用すると、
 一連の項目が自動的に追加されます。ただし、テンプレートはホストと互換性がない場合があります。
 Add をクリックすると、ホストが保存されます。

Step 3

監視のためのアイテムを作成します。

Zabbix に戻り、先ほど作成した SNMP ホストの item をクリックします。
 ホストを作成する際にテンプレートを使用したかどうかによって、ホストに関連する SNMP アイテムのリストが表示されます。
 または空のリストが表示されます。ここでは、snmpwalk と snmpget を使用して収集した情報を使用して、
 自分でアイテムを作成することを前提に作業を進めますので、Create item をクリックします。

新しいアイテムのフォームで:

- item 名を入力します。

- 'Type' フィールドを 'SNMP agent' に変更します。

- 'Key' に何か意味のあるもの、例えば SNMP-InOctets-Bps を入力します。

- 'Host interface' フィールドにお使いのスイッチ/ルーターがあることを確認します。

- 'SNMP OID' フィールドに、先ほど取得したテキストまたは数値の OID を入力します (例 : .1.3.6.1.2.1.2.1.10.3)

- 'Type of information' を Numeric (float) に設定します。

- '更新間隔' と '履歴の保存期間' をデフォルトと異なる値にしたい場合は、入力します。

- Preprocessing* タブで、Change per second のステップを追加します。
 (重要。さもないと、SNMP デバイスから最新の変更ではなく、累積値を取得することになります)
 必要な場合は、カスタム乗数を選択します。

Item	Tags	Preprocessing
* Name	Interface wlp3s0: Bits received	
Type	SNMP agent	
* Key	net.if.in[ifHCInOctets.3]	
Type of information	Numeric (unsigned)	
* Host interface	127.0.0.1:161	
* SNMP OID	1.3.6.1.2.1.31.1.1.1.6.3	
Units	bps	
* Update interval	3m	

すべての必須入力フィールドには、赤いアスタリスクが表示されます。

アイテムを保存して、Monitoring → Latest data で SNMP のデータを確認します。

例 1

一般的な例:

パラメータ	説明
OID	1.2.3.45.6.7.8.0 (or .1.2.3.45.6.7.8.0)
Key	<Unique string to be used as reference to triggers> For example, "my_param".

OID は、数値でも文字列でも指定できることに注意。ただし、文字列の OID を数値表現に変換する必要がある場合がある。
このような場合には、snmpget ユーティリティを使用することができる。


```
shell> snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Zabbix のソース設定時に --with-net-snmp フラグが指定されている場合、SNMP パラメータの監視が可能です。

例 2

稼働率のモニタリング

パラメータ	説明
OID	MIB::sysUpTime.0
Key	router.uptime
Value type	Float
Units	uptime
Multiplier	0.01

バルクリクエストの内部動作

Zabbix 2.2.3 以降 Zabbix server と proxy は、SNMP 機器に対して 1 回のリクエストで複数の値を問い合わせることができます。
これはいくつかのタイプの SNMP item に影響します。

- 通常の SNMP items
- SNMP items with dynamic indexes
- SNMP low-level discovery rules

同一のパラメータを持つ単一インターフェース上のすべての SNMP アイテムが同時にクエリされるようにスケジュールされます。最初の 2 種類のアイテムは、ポーラーによって最大 128 アイテムのバッチで取得されますが、ローレベルディスクバリ発見ルールは、従来通り個別に処理されます。

下位レベルでは、以下の 2 種類のオペレーションが実行されます。複数の指定されたオブジェクトを取得することと、OID ツリーを walk することです。

"get" については、最大 128 個の変数バインディングを持つ GetRequest-PDU が使用されます。walking では、SNMPv1 では GetNextRequest-PDU を、SNMPv2 および SNMPv3 では、最大 128 の "max-repetitions" フィールドを持つ GetBulkRequest を使用します。

各 SNMP item タイプにおけるバルクリクエストのメリットを示します。

- 通常の SNMP アイテムの場合、"get" の改善によるメリットがあります。
- 動的インデックスを持つ SNMP アイテムの場合、"get" と "walk" の両方の改善効果があります。"get" はインデックスの検証に、"walk" はキャッシュの構築に使用されます。
- SNMP のローレベルディスクバリ発見ルールは、"walk" の改善により恩恵を受けます。

ただし、すべてのデバイスがリクエストごとに 128 の値を返せるわけではないという技術的な問題があります。あるものは常に適切なレスポンスを返します。しかし、他のデバイスは、"tooBig(1)" エラーで応答するか、または、潜在的な応答が "tooBig" を超えると、まったく応答しないというエラーになったり、ある限界値を超えると全く反応しなくなったりします。

特定のデバイスに対して問合せを実行する最適なオブジェクト数を見つけるために、Zabbix は次のような戦略をとっています。慎重に開始し 1 つのリクエストで 1 つの値を問合せます。それが成功した場合、リクエストで 2 つの値を問合せます。再び成功した場合、リクエストで 3 つの値を問合せます。というように、問い合わせるオブジェクトの数を 1.5 倍にしていきます。その結果、リクエストの大きさは次のようになります。1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128

しかし、デバイスが適切なレスポンスを拒否した場合 (例えば、42 変数の場合)、Zabbix は 2 つのことを実行します。

まず、現在の item バッチでは、1 回のリクエストでオブジェクトの数を半分にし、21 個の変数を問合せます。デバイスが生きている場合、28 変数が動作することが知られており、21 はそれよりもかなり少ないからです。しかし、それでも失敗した場合、Zabbix は値を 1 つずつクエリすることに戻ります。この時点でまだ失敗する場合、デバイスは確実に応答しておらず、リクエストサイズは問題ではありません。

Zabbix が 2 つ目に行くことは、後続のアイテムバッチに対して、最後に成功した変数のサイズ (この例では 28) から開始し限界に達するまで、リクエストサイズを 1 ずつ増やし続けます。例えば、最大のレスポンスサイズが 32 変数であると仮定すると、後続のリクエストはサイズ 29, 29, 29 となります。それ以降のリクエストは 29, 30, 31, 32, 33 のサイズになります。最後のリクエストは失敗し、Zabbix はサイズ 33 のリクエストを発行することはありません。この時点から、Zabbix はこのデバイスに対して最大 32 個の変数を問い合わせることになります。

この変数のサイズで大きな問合せが失敗する場合、次の 2 つの意味があります。デバイスが応答サイズを制限するために使用する正確な基準を知ることはできませんが、私たちは変数のサイズを使用してそれを近似しようとします。つまり、第 1 の可能性は、この変数のサイズが一般的な場合、デバイスの実際のレスポンスサイズ制限の範囲ギリギリであること。第 2 つ目の可能性は、どちらかの方向の UDP パケットが単に紛失した可能性です。これらの理由から、Zabbix は失敗したクエリを取得した場合、デバイスの快適な範囲に深く入ろうとするため、変数の最大数を減らします。ただし、最大 2 回までです。(Zabbix 2.2.8 から)

上記の例では、32 個の変数を持つクエリがたまたま失敗した場合、Zabbix はカウントを 31 に減らします。そのクエリも失敗した場合、Zabbix はカウントを 30 に減らします。しかし、Zabbix はカウントを 30 以下に減らすことはありません。なぜなら、それ以上の障害は UDP パケットの損失によるものと判断されるからです。

しかし、デバイスが他の理由でバルクリクエストを適切に処理できない場合、Zabbix 2.4 以降では、各インターフェースに "Use bulk requests" 設定があり、そのインターフェースでそのデバイスのバルクリクエストを無効化することができます。

1 ダイナミックインデックス

概要

必要なインデックス番号 (例えば、ネットワーク・インターフェース) を SNMP OID の中から見つけることができるかもしれませんが、インデックス番号が常に同じであるとは限りません。

インデックス番号は動的であり、時間の経過とともに変化し、その結果、アイテムが動作しなくなる可能性があります。

このようなシナリオを回避するために、インデックス番号が変化する可能性を考慮した OID を定義することが可能です。

たとえば、次のようなインデックス値を取得する必要があるとします。Cisco のデバイスの **GigabitEthernet0/1** インターフェースに対応する **ifInOctets** に対応するインデックス値を取得する必要がある場合、次の OID を使用します。

```
ifInOctets["index", "ifDescr", "GigabitEthernet0/1"]
```

文法

OID の特別な構文が使用されます

<OID of data>["index", "<base OID of index>", "<string to search for>"]

パラメータ	説明
OID of data index	item のデータ検索に使用するメイン OID 処理方法。現在、1つの方式に対応しています: index - インデックスを検索し、データ OID に追記する。
base OID of index string to search for	この OID を検索して、文字列に対応するインデックス値を取得する。 検索時に値と完全一致するために使用する文字列。大文字と小文字を区別する。

例

apache プロセスのメモリ使用量を取得します。

この OID 構文を使用する場合、以下のようになります。

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index", "HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

インデックス番号はここで検索されます。

```
...  
HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"  
HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"  
HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"  
HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"  
...
```

これで、5388 というインデックスができました。
 このインデックスは、データ OID に追加され、目的の値を受け取ります。


```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

index lookup のキャッシュ

動的インデックスアイテムが要求された場合、Zabbix はインデックスのベース OID の下にある
 SNMP テーブル全体を取得し、キャッシュします。
 これは、別のアイテムが同じベース OID を後で参照する場合に備えて行われます。
 Zabbix は監視ホストに再度問い合わせる代わりに、キャッシュからインデックスを検索します。
 各ポーラープロセスは別々のキャッシュを使用していることに注意してください。

その後のすべての値取得操作では、見つかったインデックスのみが検証されます。
 インデックスが変更されていない場合、値が要求されます。
 変更されたインデックスに遭遇した各ポーラは、インデックス SNMP テーブルを再度検索し、キャッシュを再構築します。

2 特殊な OID

最もよく使われる SNMP OID のいくつかは、Zabbix によって自動的に数値表現に変換されます。
 例えば、**ifIndex** は **1.3.6.1.2.1.2.2.1.1** に、**ifIndex.0** は **1.3.6.1.2.1.2.2.1.1.0** に変換されます。

この表は、特殊な OID のリストをです。

特殊 OID	識別子	説明
ifIndex	1.3.6.1.2.1.2.2.1.1	A unique value for each interface.
ifDescr	1.3.6.1.2.1.2.2.1.2	A textual string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the hardware interface.
ifType	1.3.6.1.2.1.2.2.1.3	The type of interface, distinguished according to the physical/link protocol(s) immediately 'below' the network layer in the protocol stack.

特殊 OID	識別子	説明
ifMtu	1.3.6.1.2.1.2.2.1.4	The size of the largest datagram which can be sent / received on the interface, specified in octets.
ifSpeed	1.3.6.1.2.1.2.2.1.5	An estimate of the interface's current bandwidth in bits per second.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	The interface's address at the protocol layer immediately 'below' the network layer in the protocol stack.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	The current administrative state of the interface.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	The current operational state of the interface.
ifInOctets	1.3.6.1.2.1.2.2.1.10	The total number of octets received on the interface, including framing characters.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	The number of subnetwork-unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	The number of non-unicast (i.e., subnetwork- broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
ifInErrors	1.3.6.1.2.1.2.2.1.14	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	The total number of octets transmitted out of the interface, including framing characters.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	The number of outbound packets that could not be transmitted because of errors.

特殊 OID	識別子	説明
ifOutQLen	1.3.6.1.2.1.2.2.1.21	The length of the output packet queue (in packets).

3 MIB ファイル

はじめに

MIB とは、Management Information Base (管理情報ベース) の略です。
 MIB ファイルにより、OID (オブジェクト識別子) をテキストで表現したものです。

例えば

```
ifHCOutOctets
```

が OID のテキスト表現である場合

```
1.3.6.1.2.1.31.1.1.1.10
```

Zabbix で SNMP デバイスを監視する場合、どちらを使っても構いません。テキスト表現の方が使いやすいと感じる場合は、MIB ファイルをインストールする必要があります。

MIB ファイルのインストール

Debian ベースのシステムの場合:

```
# apt install snmp-mibs-downloader
# download-mibs
```

RedHat ベースのシステムの場合:

```
# yum install net-snmp-libs
```

MIB ファイルの有効化

RedHat ベースのシステムでは、mib ファイルはデフォルトで有効になっているはずですが、
 Debian ベースのシステムでは、ファイル /etc/snmp/snmp.conf を編集し、mibs :snmp.conf という行と
mibs:' という行をコメントアウトしてください。


```
# snmp のパッケージにはライセンスの関係で MIB ファイルが含まれていないため、
# MIB の読み込みはデフォルトで無効になっています。
# MIB を追加した場合は、以下の行をコメントアウトすることで、MIB の読み込みを再度有効にすることができます。
#mibs :
```

MIB ファイルのテスト

snmp MIB のテストは、snmpwalk コーティリティを使用して行うことができます。
 もし、インストールされていない場合は、以下の手順で行ってください。

Debian ベースのシステムの場合:

```
# apt install snmp
```

RedHat ベースのシステムの場合:

```
# yum install net-snmp-utils
```

その後、以下のコマンドでネットワークデバイスに問い合わせたときに、エラーが出ないようにします:

```
$ snmpwalk -v 2c -c public <NETWORK DEVICE IP> ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 176137634
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 240375057
IF-MIB::ifInOctets.4 = Counter32: 220893420
[...]
```

Zabbix での MIB の利用について

最も重要なことは、Zabbix のプロセスには MIB ファイルの変更が通知されないということです。
 そのため、変更するたびに Zabbix サーバまたはプロキシを再起動する必要があります。

```
# service zabbix-server restart
```

その後、MIB ファイルに加えられた変更が有効になります。

カスタム MIB ファイルを使用する

すべての GNU/Linux ディストリビューションには、標準的な MIB ファイルが付属しています。
しかし、いくつかのデバイスベンダーは、独自のものを提供しています。

例えば、**CISCO-SMI** MIB ファイルを使いたいとします。以下の手順でダウンロードとインストールを行います。

```
# wget ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my -P /tmp
# mkdir -p /usr/local/share/snmp/mibs
# grep -q '^mibdirs +/usr/local/share/snmp/mibs' /etc/snmp/snmp.conf 2>/dev/null || echo "mibdirs +/usr/local/share/snmp/mibs" >> /etc/snmp/snmp.conf
# cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

これで使えるようになったはずですが、
オブジェクトの名前 `ciscoProducts` を OID に変換してみてください。


```
# snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

OID の代わりにエラーが表示される場合は、前のすべてのコマンドでエラーが返されていないことを確認します。

オブジェクト名の変換がうまくいけば、カスタム MIB ファイルを使用する準備が整います。
クエリで使用される MIB 名のプレフィックス (`CISCO-SMI::`) に注意してください。
これは Zabbix だけでなく、コマンドラインツールを使用する際にも必要です。

この MIB ファイルを Zabbix で使用する前に、Zabbix server / proxy を再起動することを忘れないでください。

Attention:

MIB ファイルには依存関係があることに留意してください。
つまり、ある MIB が別の MIB を必要とする場合があります。
これらの依存関係を満たすために
影響を受けるすべての MIB ファイルをインストールする必要があります。

3 SNMP trap

概要

SNMP トラップの受信は、SNMP 対応デバイスへのクエリとは逆です。

この場合、情報は SNMP 対応デバイスから送信され、Zabbix によって収集または“トラップ”されます。

通常、何らかの条件が変化するとトラップが送信され、エージェントはポート 162 でサーバーに接続します (クエリに使用されるエージェント側のポート 161 とは対照的です)。トラップを使用すると、クエリ間隔中に発生し、クエリデータによって見逃される可能性のあるいくつかの短い障害を検出できます。

Zabbix での SNMP トラップの受信は、**snmptrapd** と、トラップを Zabbix に渡すためのメカニズムの 1 つ (Bash または Perl スクリプトまたは SNMPTT) で動作するように設計されています。

Note:

Zabbix の構成後にトラップ監視をセットアップする最も簡単な方法は、Bash スクリプトソリューションを使用することです。これは、最近のディストリビューションでは Perl と SNMPTT が欠落していることが多く、より複雑な構成が必要になるためです。ただし、このソリューションでは、`traphandle` として構成されたスクリプトが使用されます。本番システムでのパフォーマンスを向上させるには、組み込みの Perl ソリューション (`do perl` オプション付きのスクリプトまたは SNMPTT) を使用してください。

トラップを受信するワークフロー:

1. `snmptrapd` がトラップを受信
2. `snmptrapd` はトラップをレシーバースクリプト (Bash,Perl) または SNMPTT に渡します。
3. 受信者はトラップを解析、フォーマットし、ファイルに書き込みます
4. Zabbix SNMP トラッパーは、トラップファイルを読み取り、解析します。
5. トラップごとに、Zabbix は、受信したトラップアドレスと一致するホストインターフェイスを持つすべての“SNMP トラッパー”アイテムを検索します。マッチングでは、ホストインターフェイスで選択した“IP” または“DNS”のみが使用されることに注意してください。
6. 見つかった項目ごとに、トラップは `snmptrap[regex]` の正規表現と比較されます。トラップは、一致したすべてのアイテムの値として設定されます。一致するアイテムが見つからず、`snmptrap.fallback` アイテムが存在する場合、トラップはその値として設定されます。
7. トラップがアイテムの値として設定されていない場合、Zabbix はデフォルトで、一致しないトラップをログに記録します。(これは“管理” → “一般設定” → “その他”の“マッチしない SNMP トラップをログに記録”で設定されます。)

SNMP トラップの設定

フロントエンドで以下のフィールドを設定することは、このアイテムタイプ特有のものです。

- ホストには SNMP インターフェースが必要です

設定 → ホストの インターフェースフィールドで、正しい IP アドレスまたは DNS を使用して SNMP インターフェースを設定します。受信した各トラップからのアドレスは、対応するホストを見つけるために、すべての SNMP インターフェースの IP および DNS アドレスと比較されます。

- アイテムの設定

キーフィールドでは、SNMP トラップキーのいずれかを使用します。

キー	説明	戻り値	コメント
<code>snmptrap[regex]</code>	<code>regex</code> で指定された正規表現に一致するすべての SNMP トラップをキャッチします。 <code>regex</code> が指定されていない場合、すべてのトラップをキャッチします。	SNMP トラップ	この項目は、SNMP インターフェースに対してのみ設定できます。この項目キーのパラメーターでは、ユーザーマクロとグローバル正規表現がサポートされています。
<code>snmptrap.fallback</code>	そのインターフェースの <code>snmptrap[]</code> アイテムのいずれによってもキャッチされなかったすべての SNMP トラップをキャッチします。	SNMP トラップ	このアイテムは、SNMP インターフェースに対してのみ設定できます。

Note:

現時点では、複数行の正規表現一致はサポートされていません。

タイムスタンプを解析するには、データ型を 'ログ' に設定します。'数値 (整数)' など、他の形式も使用できますが、カスタムトラップハンドラが必要になる場合があることに注意してください。

SNMP トラップ監視を機能させるには、まず正しく設定する必要があります (以下を参照)。

SNMP Trap 監視の設定

Zabbix サーバー/プロキシの設定

トラップを読み取るには、Zabbix サーバーまたはプロキシが、SNMP トラップパープロセスを開始し、SNMPTT または Bash/Perl トラップレシーバーによって書き込まれているトラップファイルを指すように構成されている必要があります。これを行うには設定ファイル (`zabbix_server.conf` または `zabbix_proxy.conf`) を編集します。

```
StartSNMPTrapper=1
```

```
SNMPTrapperFile=[トラップファイル]
```

Warning:

systemd パラメーター `PrivateTmp` が使用されている場合、このファイルは `/tmp` では機能しません。

Bash トラップレシーバーの設定

要件: `snmptrapd` のみ。

Bash トラップレシーバー [スクリプト](#) を使用して、`snmptrapd` から直接 Zabbix サーバーにトラップを渡すことができます。設定するには、`snmptrapd` 設定ファイル (`snmptrapd.conf`) に `traphandle` オプションを追加します。例 ([https://raw.githubusercontent.com/zabbix/zabbix-docker/6.2/Dockerfiles/snmptraps/alpine /conf/etc/snmp/snmptrapd.conf](https://raw.githubusercontent.com/zabbix/zabbix-docker/6.2/Dockerfiles/snmptraps/alpine/conf/etc/snmp/snmptrapd.conf))。

SNMPTT の設定

最初に SNMPTT を使用するよう、`snmptrapd` を設定する必要があります。

Note:

最高のパフォーマンスを得るためには **snmpthandler-embedded** を使用してトラップを渡すデーモンとして SNMPTT を構成する必要があります。SNMPTT の設定手順を参照してください。

SNMPTT がトラップを受信するように構成されている場合は 'snmptt.ini' を設定します。

1. NET-SNMP パッケージの Perl モジュールの使用を有効にします。

```
net_snmp_perl_enable = 1
```

2. Zabbix によって読み取られるトラップファイルにトラップを記録します。

```
log_enable = 1 log_file = [トラップファイル]
```

3. 日付フォーマットを設定します。

```
date_time_format = %H:%M:%S %Y/%m/%d = [日付フォーマット]
```

Warning:

"net-snmp-perl" パッケージは RHEL 8.0-8.2 で削除されましたが、RHEL 8.3 で再度追加されました。詳細については既知の問題を参照してください。

Zabbix が認識できるようにトラップをフォーマットします (snmptt.conf を編集します)。

1. 各 FORMAT ステートメントは "ZBXTRAP [address]" で開始する必要があります。[address] は、Zabbix の SNMP インターフェースの IP および DNS アドレスと比較されます。例:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. 以下の SNMP トラップ形式の詳細を参照してください。

Attention:

未知のトラップは使用しないでください。Zabbix はそれらを認識できません。snmptt.conf: EVENT general .* "General event" Normal で一般イベントを定義することにより、不明なトラップを処理できます。

Perl トラップレシーバーの設定

要件: Perl および --enable-embedded-perl でコンパイルされた Net-SNMP (Net-SNMP 5.4 以降はデフォルトで実行)

Perl トラップレシーバー (misc/snmptrap/zabbix_trap_receiver.pl を参照) を使用して、snmptrapd から直接 Zabbix サーバーにトラップを渡すことができます。

設定するには: - Perl スクリプトを snmptrapd 設定ファイル (snmptrapd.conf) に追加します。例:

```
perl do "[PERL RECEIVER SCRIPT へのフルパス]";
```

- レシーバーを設定します。例:

```
$SNMPTrapperFile = '[トラップファイル]'; $DateTimeFormat = '[日時フォーマット]';
```

Note:

スクリプト名が引用符で囲まれていない場合、snmptrapd は次のようなメッセージで起動を拒否します。

```
Regexp modifiers "/1" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
```

```
Regexp modifier "/1" may not appear twice at (eval 2) line 1, at end of line
```

SNMP トラップのフォーマット

カスタマイズされたすべての Perl トラップレシーバーと SNMPTT トラップ構成は、次の方法でトラップをフォーマットする必要があります。

```
[timestamp] [the trap, part1] ZBXTRAP [address] [the trap, part 2]
```

内訳

- [timestamp] - ログ項目に使用されるタイムスタンプ
- ZBXTRAP - 新しいトラップがこの行で開始することを示すヘッダー
- [address] - このトラップのホストを見つけるために使用される IP アドレス

"ZBXTRAP" と "[address]" は、処理中にメッセージから切り取られることに注意してください。トラップが別の方法でフォーマットされている場合、Zabbix は予期せずトラップを解析する可能性があります。

トラップの例:

```
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - ZBXTRAP 192.168.1.1 Link down on interface 2. Admin state:
1. Operational state: 2
```

これにより、IP=192.168.1.1のSNMPインターフェースに対して次のトラップが発生します。

```
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

システム要件

ラージファイルサポート

Zabbixは、SNMPトラッパーのラージファイルをサポートしています。Zabbixが読み取れる最大ファイルサイズは 2^{63} (8 EiB)です。ファイルシステムによってはファイルサイズに下限があることに注意してください。

ログローテーション

Zabbixはログローテーションシステムを提供していません。ログローテーションはユーザが処理する必要があります。
 ログのローテーションは、トラップが失われることがないようにまず古いファイルの名前を変更し、その後で削除する必要があります。

1. Zabbixは最後に確認された場所にあるTrapファイルを開き、ステップ3へ進みます。
2. Zabbixは現在開いているファイルがローテーションされているかどうか、inode番号と定義されたTrapファイルのinode番号を比較することで確認します。開いているファイルがない場合、Zabbixは最後の場所をリセットし、ステップ1へ進みます。
3. Zabbixは現在開いているファイルからデータを読み込み、新しい場所を設定します。
4. 新しいデータがパースされます。ローテートされたファイルの場合、ファイルは閉じられ、ステップ2に戻ります。
5. 新しいデータがない場合、Zabbixは1秒間スリープし、ステップ2に戻ります。

ファイルシステム

Trapファイルの実装のため、Zabbixはファイルを区別するためにinodeをサポートする
 ファイルシステムを必要とします (この情報はstat()呼出で取得します)。

異なるSNMPプロトコルバージョンを使用したセットアップ例

この例では、snmptrapdとBashレシーバースクリプトを使用して、トラップをZabbixサーバーに渡します。

設定:

1. SNMPトラッパーを起動し、トラップファイルを設定するようにZabbixを構成します。zabbix_server.confに以下を追加:

```
StartSNMPTrapper=1 SNMPTrapperFile=/tmp/my_zabbix_traps.tmp
```

2. Bashスクリプトを/usr/sbin/zabbix_trap_handler.shとしてダウンロード

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/6.2/Dockerfiles/snmptraps/alpine/conf
```

必要に応じて、スクリプトのZABBIX_TRAPS_FILE変数を調整します。デフォルト値を使用するには、最初に親ディレクトリを作成します。

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. 以下をsnmptrapd.confに追加します (参照動作例)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

4. SNMPアイテムTEST:を作成します。

```
ホスト SNMP インターフェース IP: 127.0.0.1 キー: snmptrap["linkup"] ログタイムフォーマット: yyyyMMdd.hhmmss
```

5. 選択したSNMPプロトコルバージョンに対してsnmptrapdを構成し、snmptrapユーティリティを使用してテストトラップを送信します。

SNMPv1, SNMPv2

SNMPv1およびSNMPv2プロトコルは"コミュニティストリング"認証に依存しています。以下の例では、コミュニティストリングとして"secret"を使用します。SNMPトラップ送信側と同じ値に設定する必要があります。

SNMPv2は実稼働環境でまだ広く使用されていますが、暗号化と実際の送信者認証を提供していないことに注意してください。データはプレーンテキストとして送信されるため、これらのプロトコルバージョンは、プライベートネットワークなどの安全な環境でのみ使用し、公共またはサードパーティのネットワークでは決して使用しないでください。

SNMPバージョン1は、64ビットカウンターをサポートしておらず、レガシープロトコルと見なされているため、最近では実際には使用されていません。

SNMPv1 または SNMPv2 トラップの受け入れを有効にするには、次の行を `snmptrapd.conf` に追加する必要があります。“secret” を、SNMP トラップ送信者で構成された SNMP コミュニティ文字列に置き換えます。

```
authCommunity log,execute,net secret
```

次に `snmptrap` を使用してテストトラップを送信します。この例では、共通の“linkUp” OID を使用します。

```
snmptrap -v 2c -c secret localhost 0 linkUp.0
```

SNMPv3

SNMPv3 は、SNMPv1/v2 のセキュリティ問題に対処し、認証と暗号化を提供します。認証方式として SHA または MD5 を使用し、暗号として AES または DES を使用できます。

SNMPv3 の受け入れを有効にするには、次の行を `snmptrapd.conf` に追加します。

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

Attention:

このユーザーセキュリティモデルのスクリプトを実行できるようにする“execute” キーワードに注意してください。

```
# snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x8000000001020304
```

Warning:

AES192 や AES256 などの強力な暗号化方式を使用する場合は、バージョン 5.8 以降の `net-snmp` を使用してください。`configure` オプションで再コンパイルする必要があるかもしれません：`--enable-blumenthal-aes` 古いバージョンの `net-snmp` は AES192/AES256 をサポートしていません。参照：http://www.net-snmp.org/wiki/index.php/Strong_Authentication_or_Encryption

検証

どちらの例でも、`/var/lib/zabbix/snmptraps/snmptraps.log` に同様の行が表示されます：

```
20220805.102235 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:35736->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 0:0:00:00.00
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Zabbix のアイテム値は次のようになります。

```
2022-08-05 10:54:432022-08-05 10:54:41
```

```
20220805.105441 UDP: [127.0.0.1]:44262->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 0:0:00:00.00
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

参照

- [Zabbix blog article on SNMP traps](#)
- [Configuring snmptrapd \(net-snmp の公式ドキュメント\)](#)
- [Configuring snmptrapd to receive SNMPv3 notifications \(net-snmp の公式ドキュメント\)](#)

4 IPMI チェック

概要

Zabbix で IPMI(Intelligent Platform Management Interface) デバイスの健全性と可用性を監視することができます。
 PMI チェックを行うには Zabbix server の初期 `configured` で IPMI をサポートしている必要があります。

IPMI はリモートで“lights-out”(電源断) または“out-of-band” 管理を行うための標準化されたインタフェースです。
 いわゆる“out-of-band” 管理カードから直接ハードウェアの状態を監視することができます。
 OS やマシンの電源の有無に関係なく、いわゆる“out-of-band” 管理カードから直接ハードウェアの状態を監視することができます。

Zabbix IPMI 監視は、IPMI をサポートするデバイス (HP iLO、DELL DRAC、IBM RSA、Sun SSP など) にのみ対応しています。

Zabbix 3.4 以降、IPMI ポーラーによる IPMI チェックをスケジュールするための新しい IPMI マネージャプロセスが追加されました。
 これにより、ホストは常に一度に 1 つの IPMI ポーラーによってポーリングされ、BMC コントローラへのオープン接続数を減らすことができます。
 IPMI ポーラーの数を増やしても、BMC コントローラへの過負荷を心配する必要はありません。
 IPMI マネージャプロセスは、少なくとも 1 つの IPMI ポーラーが起動すると自動的に開始されます。

IPMI チェックについては、[known issues](#) も参照してください。

設定

ホストの設定

IPMI チェックを処理するために、ホストを設定する必要があります。
 IPMI インターフェイスを追加し、それぞれの IP アドレスおよびポート番号を指定し、IPMI 認証パラメーターを定義する必要があります。

詳細は、[configuration of hosts](#) を参照してください。

Server の設定

デフォルトでは、Zabbix サーバは IPMI ポーラーを起動するように設定されていないため、IPMI アイテムを追加しても動作しません。
 これを変更するには、Zabbix サーバ設定ファイル ([zabbix_server.conf](#)) を root ユーザで開き、
 次の行を探します。

```
# StartIPMIPollers=0
```

この行をアンコメントし、ポーラー数を 3 に設定すると、次のようになります。

```
StartIPMIPollers=3
```

ファイルを保存し、その後 `zabbix_server` を再起動します。

item の設定

ホストレベルで[configuring an item](#)を行う場合:

- Type として 'IPMI エージェント' を選択します。
- ホスト内で一意である item **key** (例えば、`ipmi.fan.rpm`) を入力します。
- Host interface には、関連する IPMI インターフェイス (IP アドレスおよびポート番号) を選択します。IPMI インターフェイスはホスト上に存在する必要があることに注意してください。
- IPMI センサー * (Dell Poweredge の 'FAN MOD 1A RPM' など) からメトリックを取得するように指定します。デフォルトでは、センサー ID を指定する必要があります。接頭辞を使用することも可能です。
 - `id:` - センサーの ID を指定します。
 - `name:` - センサーのフルネームを指定します。これは、フルネームを指定することによってのみセンサを区別できる場合に役立ちます。
- それぞれの情報の種類を選択します (この場合は、'Numeric (float)' ディスクリットセンサーの場合は 'Numeric (unsigned)'), 単位 (多くの場合 'rpm')、およびその他の必要な項目属性を選択します。

対応チェック

以下の表は、IPMI エージェントのチェックでサポートされている内蔵 item です。

Item key	説明	戻り値	コメント
▲ ipmi.get	IPMI- sensor related in- formation.	JSON object	This item can be used for the discovery of IPMI sensors. Supported since Zabbix 5.0.0.

タイムアウトとセッションの終了

PMI メッセージのタイムアウトとリトライ回数は、OpenIPMI ライブラリで定義されています。
 現在の OpenIPMI の設計上、Zabbix でこれらの値をインターフェイスや item レベルで設定することはできません。

LAN の IPMI セッションの非アクティブ時のタイムアウトは 60±3 秒です。
 現在のところ、OpenIPMI で定期的な Activate Session コマンドの定期的な送信を実装することはできません。
 BMC で設定されたセッションタイムアウト時間以上、Zabbix から特定の BMC への IPMI item のチェックがない場合
 タイムアウト後の次の IPMI チェックは、個々のメッセージのタイムアウト、再試行、または受信エラーによりタイムアウトします。
 その後、新しいセッションが開かれ、BMC の完全な再スキャンが開始されます。
 もし、BMC の不要な再スキャンを回避するには、IPMI item ポーリング間隔を BMC で設定された
 IPMI セッション非活動時限よりも短く設定することをお勧めします。

IPMI ディスクリートセンサに関する注意事項

ホスト上のセンサーを見つけるには、**DebugLevel=4** を有効にして Zabbix サーバを起動します。数分後、server ログファイルでセンサーの発見記録を確認します。

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

IPMI センサーの種類と状態を解釈するには、IPMI 2.0 の仕様書入手してください。
 <http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-specifications.html>
 (執筆時点では最新のドキュメントは
 <http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/second-gen-interface-spec-v2.pdf>)

最初に始めるパラメータは"reading_type"です。
 仕様書の"表 42-1,Event/Reading Type Code Ranges"を使用し、"reading_type"コードをデコードします。
 この例では、ほとんどのセンサーが"reading_type:0x1"で、"threshold"センサーを意味します。
 "表 42-3, Sensor Type Codes"によると、"type:0x1"は温度センサー、"type:0x2"は電圧センサー、"type:0x3"は温度センサー、
 "type:0x4"はファンなどです。閾値センサーは、温度、電圧、回転数などの連続的なパラメータを測定するため、
 "アナログ"センサーと呼ばれることもあります。

別の例として、"reading_type:0x3"のセンサー
 "表 42-1,Event/Reading Type Code Ranges"によると、読み取りタイプコード 02h-0Ch は、
 "Generic Discrete"センサーを意味します。ディスクリート・センサは、最大 15 個の状態
 (言い換えれば、最大 15 個の意味のあるビット)があります。例えばセンサー"CATERR"の場合、
 "type:0x7"は"表 42-3, Sensor Type Codes"に示されています。このタイプは"プロセッサ"を意味し、
 ビットの意味は次のとおりです。00h (最下位ビット) - IERR、01h - サーマルトリップなど。

この例では、"reading_type:0x6f"を持つセンサーはほとんどありません。
 これらのセンサーの場合、"表 42-1,Event/Reading Type Code Ranges"は、
 ビットの意味を解釈するために、"表 42-3, センサータイプコード"を使用するように助言しています。
 例えば、センサー"Power Unit Stat"のタイプは"type:0x9"であり、"Power Unit"を意味します。
 オフセット 00h は"PowerOff/Power Down"を意味します。つまり、最下位ビットが1であれば
 サーバの電源がオフになっていることを意味します。このビットをテストするには
 マスクが'1'の **bitand** 関数を使用することができます。
 トリガ式は次のようになります。


```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

これはサーバの電源オフを警告します。

OpenIPMI-2.0.16, 2.0.17, 2.0.18, 2.0.19 のディスクリートセンサーの名前に関する注意事項

OpenIPMI-2.0.16, 2.0.17, 2.0.18 のディスクリートセンサーの名前には"0" (または他の数字や文字) を
 末尾に付加していることがよくあります。
 例えば、ipmitool と OpenIPMI-2.0.19 では、センサーの名前は"PhysicalSecurity" や"CATERR" のように表示されます。
 OpenIPMI-2.0.16, 2.0.17, 2.0.18 では"PhysicalSecurity0" または"CATERR0" と表示されます。

OpenIPMI-2.0.16, 2.0.17, 2.0.18 を使用して Zabbix サーバで IPMI item を設定する場合、
 IPMI エージェントアイテムの IPMI sensor フィールドにこれらの名前を 0 として使用してください。
 Zabbix サーバを OpenIPMI-2.0.19 (またはそれ以降) を使用する新しい Linux ディストリビューションにアップグレードした場合、
 これらの IPMI センサーのアイテムは「NOT SUPPORTED」になります。そのため
 IPMI センサー名を変更し (末尾の'0'を削除)、再び"Enabled"になる前に、しばらく待つ必要があります。

閾値とディスクリートセンサーの同時使用に関する注意事項

いくつかの IPMI エージェントは、閾値センサーとディスクリートセンサーを同じ名前で提供しているものがあります。
 2.2.8 および 2.4.3 より前のバージョンの Zabbix では、最初に提供されたセンサーが選択されます。
 バージョン 2.2.8 と 2.4.3 以降

では、常に閾値センサーが優先されます。

接続終了時の注意事項

IPMI チェックが行われなかった場合 (何らかの理由: ホストの IPMI 項目が全て無効/サポートされていない、
 ホストが無効/削除された、ホストがメンテナンス中であるなど)
 Zabbix server / proxy を起動した時間に応じて 3~4 時間後に Zabbix server / proxy から IPMI 接続が切断されます。

5 シンプルチェック

概要

シンプルチェックは、通常、リモートエージェントを使用しないサービスをチェックするために使用します。

なお、シンプルチェックでは Zabbix agent は必要ありません。
 Zabbix server / proxy はシンプルチェックの処理 (外部接続の確立など) を担当します。

シンプルチェックの利用例

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:

シンプルチェック項目の User name および Password フィールドは VMware の監視項目に使用され、それ以外は無視されま

サポートされているシンプルチェック

サポートするシンプルチェックの一覧です:

こちらも参照のこと:

- [VMware monitoring item keys](#)

Key	説明	戻り値	パラメータ	コメント
icmping[<target>,<packets>,<interval>,<size>,<timeout>]	ICMP ping によるホストアクセス	0 - ICMP ping 失敗 1 - ICMP ping 成功	target - ホスト IP または DNS 名 packets - パケット数 interval - 連続するパケット間の時間 (ミリ秒単位) size - パケットサイズ (バイト単位) timeout - タイムアウト (ミリ秒単位)	例: => icmping[4] → 4 つのパケットのうち少なくとも 1 つのパケットが返された場合、アイテムは 1 を返します。 参照: デフォルト値の表
icmpingloss[<target>,<packets>,<interval>,<size>,<timeout>]	パケットロス割合	浮動小数点	target - ホスト IP または DNS 名 packets - パケット数 interval - 連続するパケット間の時間 (ミリ秒単位) size - パケットサイズ (バイト単位) timeout - タイムアウト (ミリ秒単位)	参照: デフォルト値の表
icmpingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>]				

ICMP ping 応答時間 (秒単位)	浮動小数点	target - ホスト IP または DNS 名 packets - パケット数 interval - 連続するパケット間の時間 (ミリ秒単位) size - パケットサイズ (バイト単位) timeout - タイムアウト (ミリ秒単位) mode - 使用可能な値: min, max, avg (デフォルト)	パケットロスやタイムアウトパケットは計算に使用されません。 ホストが使用できない (タイムアウトに達した) 場合、アイテムは 0 を返します。 戻り値が 0.0001 秒未満の場合、値は 0.0001 秒に設定されます。 参照: デフォルト値 の表
net.tcp.service[service,<ip>,<port>] サービスが実行中であり、TCP 接続を受け入れているかどうかを確認します。	0 - サービス停止 1 - サービス実行中	service - 使用可能な値: ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet (参照: 詳細) ip - IP アドレスまたは DNS 名 (デフォルトではホスト IP/DNS が使用されます) port - ポート番号 (デフォルトでは、標準のサービスポート番号が使用されます)	例: => net.tcp.service[ftp,,45] → TCP ポート 45 で FTP サーバーの可用性をテストします ポートを示す tcp サービスは必須であることに注意してください。 これらのチェックにより、システムデーモンのログファイルに追加のメッセージが記録される場合があります (SMTP および SSH セッションは通常ログに記録されます)。 暗号化されたプロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには net.tcp.service[tcp,<ip>,<port>] を使用してください。 https および telnet サービスは、Zabbix 2.0 以降でサポートされています。
net.tcp.service.perf[service,<ip>,<port>] TCP サービスのパフォーマンスをチェックします	浮動小数点 0.000000 - サービス停止 秒 - サービスへの接続にかかった秒数	service - 使用可能な値: ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet (参照: 詳細) ip - IP アドレスまたは DNS 名 (デフォルトではホスト IP/DNS が使用されます) port - ポート番号 (デフォルトでは、標準のサービスポート番号が使用されます)	例: => net.tcp.service.perf[ssh] → SSH サーバーの初期応答速度をテストします。 ポートを示す tcp サービスが必須であることに注意してください。 暗号化されたプロトコルのチェック (ポート 993 の IMAP またはポートの POP など) 995) は現在サポートされていません。回避策として、これらのチェックには net.tcp.service.perf[tcp,<ip>,<port>] を使用してください。 https および telnet サービスは、Zabbix 2.0 以降でサポートされています。 Zabbix 2.0 より前は tcp_perf と呼ばれていました。
net.udp.service[service,<ip>,<port>]			

Key

サービスが実行中であり、UDP 要求に応答しているかどうかを確認します。	0 - サービス停止 1 - サービス実行中	service - 使用可能な値: ntp (参照:詳細) ip - IP アドレスまたは DNS 名 (デフォルトではホスト IP/DNS が使用されます) port - ポート番号 (デフォルトでは、標準のサービスポート番号が使用されます)	例: => net.udp.service[ntp,,45] → UDP ポート 45 で NTP サービスの可用性をテストします。 このアイテムは Zabbix 3.0 以降でサポートされていますが、ntp サービスは以前のバージョンでは net.tcp.service[] アイテムで利用可能でした。
net.udp.service.perf[service,<ip>,<port>] UDP サービスのパフォーマンスをチェックします。	浮動小数点 0.000000 - サービス停止 秒 - サービスからの応答の待機にかかった秒数	service - 使用可能な値: ntp (参照:詳細) ip - IP アドレスまたは DNS 名 (デフォルトではホスト IP/DNS が使用されます) port - ポート番号 (デフォルトでは、標準のサービスポート番号が使用されます)	例: => net.udp.service.perf[ntp] → NTP サービスからの応答時間をテストします。 このアイテムは Zabbix 3.0 以降でサポートされていますが、ntp サービスは以前のバージョンでは net.tcp.service[] アイテムで利用可能でした。

Attention:

LDAP シンプルチェック (例: net.tcp.service[ldap]) で SourceIP をサポートするには、OpenLDAP バージョン 2.6.1
 またはそれ以上が必要です。SourceIP は Zabbix 6.0.1 以降、LDAP シンプルチェックでサポートされています。

タイムアウト処理

Zabbix は、Zabbix server / proxy 設定ファイルに定義されたタイムアウト秒数以上のシンプルチェックは処理しません。

ICMP ping

Zabbix は ICMP ping の処理に外部ユーティリティ **fping** を使用しています。

このユーティリティは Zabbix の配布物には含まれていないため、追加でインストールする必要があります。
 ユーティリティがない場合、パーミッションが正しくない場合、またはユーティリティの場所が
 Zabbix server / proxy 設定ファイル ('FpingLocation' パラメータ) で設定した場所と一致しない場合、ICMP Ping(**icmpping**, **icmppingloss**, **icmppingsec**) は処理されません。

known issues もあわせて参照してください。

fping は、Zabbix daemon が実行されるユーザで実行可能である必要があります。setuid root で実行する必要があります。
 正しいパーミッションを設定するために、ユーザ **root** として以下のコマンドを実行してください。


```
shell> chown root:zabbix /usr/sbin/fping
shell> chmod 4710 /usr/sbin/fping
```

上記 2 つのコマンドを実行した後、**fping** 実行ファイルの所有権を確認します。場合によっては、chmod コマンドを実行することで、所有権をリセットできる場合があります。

また、ユーザ zabbix がグループ zabbix に属しているかどうかを確認してください。

```
shell> groups zabbix
```

zabbix グループに追加されていない場合は、以下を実行してください。

```
shell> usermod -a -G zabbix zabbix
```

ICMP チェックパラメータのデフォルト値、制限値の説明:

パラメータ	単位	説明	Fping の flag	デフォルト値	制限値 by Zabbix	
packets	number	number of request packets to a target	-C	fping	Zabbix 3 1	max 10000
interval	milliseconds	time to wait between successive packets	-p	1000	20	unlimited
size	bytes	packet size in bytes 56 bytes on x86, 68 bytes on x86_64	-b	56 or 68	24	65507
timeout	milliseconds	fping v3.x - timeout to wait after last packet sent, affected by -C flag fping v4.x - individual timeout for each packet	-t	fping v3.x - 500 fping v4.x - inherited from -p flag, but not more than 2000	50	unlimited

さらに、Zabbix は fping オプションとして、`-i interval ms` (上表の item パラメータ `interval` と `
` 混同しないように注意してください) と `-S source IP address` (旧 fping バージョンでは `-l`) を使用します。 `
` これらのオプションは、異なるオプションの組み合わせでチェックを実行することで、自動的に検出されます。 `
` Zabbix は、fping が `-i` で使用できる最小値をミリ秒単位で検出するため、0、1、10 の 3 つの値を試します。 `
` 最初に成功した値がその後の ICMP チェックに使われます。この処理は、各 **ICMP pinger** プロセスが個別に行います。

自動検出された fping オプションは、1 時間ごとに無効化され、次の ICMP 検査時に再度検出されます。 `
` このプロセスの詳細を表示するには、`DebugLevel>=4` に設定します。 `
` server または proxy のログファイルにこのプロセスの詳細が表示されます。 `
`

Warning:

警告: fping のデフォルトは、プラットフォームとバージョンによって異なる場合があります。 `
` fping のドキュメントを確認してください。

Zabbix は、3 つの **icmping** キーでチェックする IP アドレスを一時ファイルに書き込み、 `
` このファイルを **fping** に渡します。 item キーパラメータが異なる項目は、キーパラメータが同一のもの `
` のみ 1 つのファイルに書き込まれます。 `
` 1 つのファイルに書き込まれた全ての IP アドレスを並列に fping でチェックするため、Zabbix の icmp ピンガープロセスは `
` ファイル内の IP アドレスの数に関係なく、一定の時間を費やします。 `
`

VMware 監視アイテムキー

アイテムキー

この表は **VMware 環境** を監視するためのシンプルチェックの詳細を示します。

Key

説明	戻り値	パラメータ	コメント
vmware.cl.perfcounter[<code><url></code> , <code><id></code> , <code><path></code> , <code><instance></code>]			

Key				
VMware cluster performance counter metrics.	Integer	url - VMware service URL id - VMware cluster ID path - performance counter path instance - performance counter instance	id can be received from vmware.cluster.discovery[] as {#CLUSTER.ID}	
vmware.cluster.discovery[<url>] Discovery of VMware clusters.	JSON object	url - VMware service URL		
vmware.cluster.status[<url>,<name>] VMware cluster status.	Integer: 0 - gray; 1 - green; 2 - yellow; 3 - red	url - VMware service URL name - VMware cluster name		
vmware.datastore.discovery[<url>] Discovery of VMware datastores.	JSON object	url - VMware service URL		
vmware.datastore.hv.list[<url>,<datastore>] List of datastore hypervisors.	JSON object	url - VMware service URL datastore - datastore name		
vmware.datastore.read[<url>,<datastore>,<mode>] Amount of time for a read operation from the datastore (milliseconds).	Integer ²	url - VMware service URL datastore - datastore name mode - latency (average value, default), maxlatency (maximum value)		
vmware.datastore.size[<url>,<datastore>,<mode>] VMware datastore space in bytes or in percentage from total.	Integer - for bytes Float - for percentage	url - VMware service URL datastore - datastore name mode - possible values: total (default), free, pfree (free, percentage), uncommitted		
vmware.datastore.write[<url>,<datastore>,<mode>] Amount of time for a write operation to the datastore (milliseconds).	Integer ²	url - VMware service URL datastore - datastore name mode - latency (average value, default), maxlatency (maximum value)		
vmware.dc.discovery[<url>] Discovery of VMware datacenters.	JSON object	url - VMware service URL		
vmware.eventlog[<url>,<mode>] VMware event log.	Log	url - VMware service URL mode - all (default), skip - skip processing of older data	There must be only one vmware.eventlog[] item key per URL.	See also: example of filtering VMware event log records.
vmware.fullname[<url>] VMware service full name.	String	url - VMware service URL		
vmware.hv.cluster.name[<url>,<uuid>] VMware hypervisor cluster name.	String	url - VMware service URL uuid - VMware hypervisor host name		
vmware.hv.cpu.usage[<url>,<uuid>] VMware hypervisor processor usage (Hz).	Integer	url - VMware service URL uuid - VMware hypervisor host name		
vmware.hv.cpu.usage.perf[<url>,<uuid>]				

Key

VMware hypervisor processor usage as a percentage during the interval.	Float	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.cpu.utilization[<url>,<uuid>]		
VMware hypervisor processor usage as a percentage during the interval, depends on power management or HT.	Float	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.datacenter.name[<url>,<uuid>]		
VMware hypervisor datacenter name.	String	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.datastore.discovery[<url>,<uuid>]		
Discovery of VMware hypervisor datastores.	JSON object	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.datastore.list[<url>,<uuid>]		
List of VMware hypervisor datastores.	JSON object	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.datastore.multipath[<url>,<uuid>,<datastore>,<partitionid>]		
Number of available datastore paths.	Integer	url - VMware service URL uuid - VMware hypervisor host name datastore - datastore name partitionid - internal ID of physical device from vmware.hv.datastore.discovery
vmware.hv.datastore.read[<url>,<uuid>,<datastore>,<mode>]		
Average amount of time for a read operation from the datastore (milliseconds).	Integer ²	url - VMware service URL uuid - VMware hypervisor host name datastore - datastore name mode - latency (default)
vmware.hv.datastore.size[<url>,<uuid>,<datastore>,<mode>]		
VMware datastore space in bytes or in percentage from total.	Integer - for bytes Float - for percent- age	url - VMware service URL uuid - VMware hypervisor host name datastore - datastore name mode - possible values: total (default), free, pfree (free, percentage), uncommitted
vmware.hv.datastore.write[<url>,<uuid>,<datastore>,<mode>]		
Average amount of time for a write operation to the datastore (milliseconds).	Integer ²	url - VMware service URL uuid - VMware hypervisor host name datastore - datastore name mode - latency (default)
vmware.hv.discovery[<url>]		
Discovery of VMware hypervisors.	JSON object	url - VMware service URL
vmware.hv.fullname[<url>,<uuid>]		
VMware hypervisor name.	String	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.hw.cpu.freq[<url>,<uuid>]		
VMware hypervisor processor frequency (Hz).	Integer	url - VMware service URL uuid - VMware hypervisor host name
vmware.hv.hw.cpu.model[<url>,<uuid>]		

Key

VMware hypervisor processor model.	String	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.hw.cpu.num[<url>,<uuid>] Number of processor cores on VMware hypervisor.	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.hw.cpu.threads[<url>,<uuid>] Number of processor threads on VMware hypervisor.	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.hw.memory[<url>,<uuid>] VMware hypervisor total memory size (bytes).	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.hw.model[<url>,<uuid>] VMware hypervisor model.	String	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.hw.uuid[<url>,<uuid>] VMware hypervisor BIOS UUID.	String	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.hw.vendor[<url>,<uuid>] VMware hypervisor vendor name.	String	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.maintenance[<url>,<uuid>] VMware hypervisor maintenance status.	Integer	url - VMware service URL uuid - VMware hypervisor host name	Returns '0' - not in maintenance or '1' - in maintenance
vmware.hv.memory.size.ballooned[<url>,<uuid>] VMware hypervisor ballooned memory size (bytes).	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.memory.used[<url>,<uuid>] VMware hypervisor used memory size (bytes).	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.network.in[<url>,<uuid>,<mode>] VMware hypervisor network input statistics (bytes per second).	Integer ²	url - VMware service URL uuid - VMware hypervisor host name mode - bps (default)	
vmware.hv.network.out[<url>,<uuid>,<mode>] VMware hypervisor network output statistics (bytes per second).	Integer ²	url - VMware service URL uuid - VMware hypervisor host name mode - bps (default)	
vmware.hv.perfcounter[<url>,<uuid>,<path>,<instance>] VMware hypervisor performance counter value.	Integer ²	url - VMware service URL uuid - VMware hypervisor host name path - performance counter path ¹ instance - performance counter instance. Use empty instance for aggregate values (default)	
vmware.hv.power[<url>,<uuid>,<max>]			

Key

VMware hypervisor power usage (W).	Integer	url - VMware service URL uuid - VMware hypervisor host name max - maximum allowed power usage	
vmware.hv.sensor.health.state[<url>,<uuid>] VMware hypervisor health state rollup sensor.	Integer: 0 - gray; 1 - green; 2 - yellow; 3 - red	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.sensors.get[<url>,<uuid>] VMware hypervisor HW vendor state sensors.	JSON	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.status[<url>,<uuid>] VMware hypervisor status.	Integer: 0 - gray; 1 - green; 2 - yellow; 3 - red	url - VMware service URL uuid - VMware hypervisor host name	Uses the host system overall status property.
vmware.hv.uptime[<url>,<uuid>] VMware hypervisor uptime (seconds).	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.version[<url>,<uuid>] VMware hypervisor version.	String	url - VMware service URL uuid - VMware hypervisor host name	
vmware.hv.vm.num[<url>,<uuid>] Number of virtual machines on VMware hypervisor.	Integer	url - VMware service URL uuid - VMware hypervisor host name	
vmware.version[<url>] VMware service version.	String	url - VMware service URL	
vmware.vm.cluster.name[<url>,<uuid>] VMware virtual machine name.	String	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.cpu.latency[<url>,<uuid>] Percentage of time the virtual machine is unable to run because it is contending for access to the physical CPU(s).	Float	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.cpu.num[<url>,<uuid>] Number of processors on VMware virtual machine.	Integer	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.cpu.readiness[<url>,<uuid>,<instance>] Percentage of time that the virtual machine was ready, but could not get scheduled to run on the physical CPU.	Float	url - VMware service URL uuid - VMware virtual machine host name instance - CPU instance	
vmware.vm.cpu.ready[<url>,<uuid>] Time (in milliseconds) that the virtual machine was ready, but could not get scheduled to run on the physical CPU. CPU ready time is dependent on the number of virtual machines on the host and their CPU loads (%).	Integer ²	url - VMware service URL uuid - VMware virtual machine host name	

Key

vmware.vm.cpu.swapwait[<url>,<uuid>,<instance>]	Percentage of CPU time spent waiting for swap-in.	Float	url - VMware service URL uuid - VMware virtual machine host name instance - CPU instance
vmware.vm.cpu.usage[<url>,<uuid>]	VMware virtual machine processor usage (Hz).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.cpu.usage.perf[<url>,<uuid>]	VMware virtual machine processor usage as a percentage during the interval.	Float	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.datacenter.name[<url>,<uuid>]	VMware virtual machine datacenter name.	String	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.discovery[<url>]	Discovery of VMware virtual machines.	JSON object	url - VMware service URL
vmware.vm.guest.memory.size.swapped[<url>,<uuid>]	Amount of guest physical memory that is swapped out to the swap space (KB).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.guest.osuptime[<url>,<uuid>]	Total time elapsed since the last operating system boot-up (in seconds).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.hv.name[<url>,<uuid>]	VMware virtual machine hypervisor name.	String	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size[<url>,<uuid>]	VMware virtual machine total memory size (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.ballooned[<url>,<uuid>]	VMware virtual machine ballooned memory size (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.compressed[<url>,<uuid>]	VMware virtual machine compressed memory size (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.consumed[<url>,<uuid>]	Amount of host physical memory consumed for backing up guest physical memory pages (KB).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.private[<url>,<uuid>]	VMware virtual machine private memory size (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.shared[<url>,<uuid>]	VMware virtual machine shared memory size (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.swapped[<url>,<uuid>]	VMware virtual machine swapped memory size (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name

Key

vmware.vm.memory.size.usage.guest[<url>,<uuid>] VMware virtual machine guest memory usage (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.size.usage.host[<url>,<uuid>] VMware virtual machine host memory usage (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.memory.usage[<url>,<uuid>] Percentage of host physical memory that has been consumed.	Float	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.net.if.discovery[<url>,<uuid>] Discovery of VMware virtual machine network interfaces.	JSON object	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.net.if.in[<url>,<uuid>,<instance>,<mode>] VMware virtual machine network interface input statistics (bytes/packets per second).	Integer ²	url - VMware service URL uuid - VMware virtual machine host name instance - network interface instance mode - bps (default)/pps - bytes/packets per second
vmware.vm.net.if.out[<url>,<uuid>,<instance>,<mode>] VMware virtual machine network interface output statistics (bytes/packets per second).	Integer ²	url - VMware service URL uuid - VMware virtual machine host name instance - network interface instance mode - bps (default)/pps - bytes/packets per second
vmware.vm.net.if.usage[<url>,<uuid>,<instance>] VMware virtual machine network utilization (combined transmit-rates and receive-rates) during the interval (KBps).	Integer	url - VMware service URL uuid - VMware virtual machine host name instance - network interface instance
vmware.vm.perfcounter[<url>,<uuid>,<path>,<instance>] VMware virtual machine performance counter value.	Integer ²	url - VMware service URL uuid - VMware virtual machine host name path - performance counter path ¹ instance - performance counter instance. Use empty instance for aggregate values (default)
vmware.vm.powerstate[<url>,<uuid>] VMware virtual machine power state.	Integer: 0 - poweredOff; 1 - poweredOn; 2 - sus- pended	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.storage.committed[<url>,<uuid>] VMware virtual machine committed storage space (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name
vmware.vm.storage.readoio[<url>,<uuid>,<instance>]		

Key

Average number of outstanding read requests to the virtual disk during the collection interval.	Integer	url - VMware service URL uuid - VMware virtual machine host name instance - disk device instance (mandatory)	
vmware.vm.storage.totalreadlatency[<url>,<uuid>,<instance>] The average time a read from the virtual disk takes (milliseconds).	Integer	url - VMware service URL uuid - VMware virtual machine host name instance - disk device instance (mandatory)	
vmware.vm.storage.totalwritelatency[<url>,<uuid>,<instance>] The average time a write to the virtual disk takes (milliseconds).	Integer	url - VMware service URL uuid - VMware virtual machine host name instance - disk device instance (mandatory)	
vmware.vm.storage.uncommitted[<url>,<uuid>] VMware virtual machine uncommitted storage space (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.storage.unshared[<url>,<uuid>] VMware virtual machine unshared storage space (bytes).	Integer	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.storage.writeio[<url>,<uuid>,<instance>] Average number of outstanding write requests to the virtual disk during the collection interval.	Integer	url - VMware service URL uuid - VMware virtual machine host name instance - disk device instance (mandatory)	
vmware.vm.uptime[<url>,<uuid>] VMware virtual machine uptime (seconds).	Integer	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.vfs.dev.discovery[<url>,<uuid>] Discovery of VMware virtual machine disk devices.	JSON object	url - VMware service URL uuid - VMware virtual machine host name	
vmware.vm.vfs.dev.read[<url>,<uuid>,<instance>,<mode>] VMware virtual machine disk device read statistics (bytes/operations per second).	Integer ²	url - VMware service URL uuid - VMware virtual machine host name instance - disk device instance mode - bps (default)/ops - bytes/operations per second	
vmware.vm.vfs.dev.write[<url>,<uuid>,<instance>,<mode>] VMware virtual machine disk device write statistics (bytes/operations per second).	Integer ²	url - VMware service URL uuid - VMware virtual machine host name instance - disk device instance mode - bps (default)/ops - bytes/operations per second	
vmware.vm.vfs.fs.discovery[<url>,<uuid>] Discovery of VMware virtual machine file systems.	JSON object	url - VMware service URL uuid - VMware virtual machine host name	VMware Tools must be installed on the guest virtual machine.
vmware.vm.vfs.fs.size[<url>,<uuid>,<fsname>,<mode>]			

Key				
VMware virtual machine file system statistics (bytes/percentages).	Integer	url - VMware service URL uuid - VMware virtual machine host name fsname - file system name mode - total/free/used/pfree/pused		VMware Tools must be installed on the guest virtual machine.

脚注

¹ VMware パフォーマンスカウンターのパスは `group/counter[rollup]` 形式です:

- `group` - `cpu` などのパフォーマンスカウンターのグループ
- `counter` - `usagemhz` などのパフォーマンスカウンターの名前
- `rollup` - `average` などのパフォーマンスカウンターの丸め方の種類

したがって、上記の例では、次のカウンターパスが得られます: `cpu/usagemhz[average]`

パフォーマンスカウンターのグループ、カウンター名、および丸め方の種類についての説明は、[VMware のドキュメント](#) を参照してください。

関連項目: [VMware のカスタムパフォーマンスカウンター名の作成](#).

² これらの項目の値は VMware パフォーマンスカウンターから取得され、`VMwarePerfFrequency` パラメータを使用して、Zabbix VMware キャッシュ内のデータが更新されます。

- `vmware.hv.datastore.read`
- `vmware.hv.datastore.write`
- `vmware.hv.network.in`
- `vmware.hv.network.out`
- `vmware.hv.perfcounter`
- `vmware.vm.cpu.ready`
- `vmware.vm.net.if.in`
- `vmware.vm.net.if.out`
- `vmware.vm.perfcounter`
- `vmware.vm.vfs.dev.read`
- `vmware.vm.vfs.dev.write`

詳細

Zabbix で VMware 環境を監視するための詳細な設定方法は、[仮想マシン監視](#) を参照してください。

6 ログファイル監視

概要

Zabbix は、ログローテーションサポートの有無にかかわらず、ログファイルの集中監視と分析に利用することができます。

ログファイルに特定の文字列や文字列パターンが含まれる場合、通知機能を使ってユーザーに警告することができます。

ログファイルを監視するには、以下の環境が必要です。

- ホスト上で Zabbix agent が動作していること
- ログ監視 item の設定

Attention:

監視するログファイルのサイズ制限は、`large file support` に依存します。

設定

agent パラメーターの確認

`agent configuration file` で以下を確認します。

- 'Hostname' パラメータがフロントエンドのホスト名と一致していること。
- 'ServerActive' パラメータで、アクティブチェックの処理に使用する `server` が指定されていること。

アイテムの設定

ログ監視のアイテム設定

Item	Tags	Preprocessing								
		<p>* Name <input type="text" value="Log item"/></p> <p>Type <input type="text" value="Zabbix agent (active)"/></p> <p>* Key <input type="text" value="log[/var/log/syslog,error]"/></p> <p>Type of information <input type="text" value="Log"/></p> <p>* Update interval <input type="text" value="30s"/></p> <table border="1"><thead><tr><th>Custom intervals</th><th>Type</th><th>Interval</th><th>Period</th></tr></thead><tbody><tr><td></td><td>Add</td><td></td><td></td></tr></tbody></table> <p>* History storage period <input type="text" value="Do not keep history"/> <input checked="" type="radio"/> Storage period <input type="text" value="3600"/></p> <p>Log time format <input type="text" value="ppppddphh:mm:ss"/></p>	Custom intervals	Type	Interval	Period		Add		
Custom intervals	Type	Interval	Period							
	Add									

すべての必須入力フィールドには赤いアスタリスクが付いています。

具体的には、ログ監視アイテムに対して次のように入力します。

タイプ キー	<p>Zabbix エージェント (アクティブ) を選択 次の項目キーのいずれかを使用します log[] または logrt[]:</p> <p>これら 2 つのアイテムキーを使用すると、ログを監視し、存在する場合はコンテンツの正規表現によってログエントリをフィルター処理できます。 例: <code>log[/var/log/syslog,error]</code> ファイルに 'zabbix' ユーザーの読み取り権限があることを確認してください。そうでない場合、アイテムのステータスは 'サポートされていません' に設定されます。</p> <p>log.count[] または logrt.count[]:</p> <p>これら 2 つの項目キーは一致する行の数のみを返すことができます。 これらのアイテムキーとそのパラメータの使用の詳細については、サポートされている Zabbix エージェントアイテム キーセクションを参照してください。</p>
データ型	<p>自動事前入力: log[] または logrt[] アイテムの場合 - ログ log.count[] または logrt.count[] アイテムの場合 - 数値 (整数)。 オプションで output パラメータを使用する場合は、Log 以外の適切なタイプの情報を手動で選択できます。</p>
監視間隔 (秒)	<p>非ログタイプの情報を選択すると、ローカルタイムスタンプが失われることに注意してください。 このパラメータは、Zabbix エージェントがログファイルの変更をチェックする頻度を定義します。 1 秒に設定するとできるだけ早く新しいレコードを取得できます。</p>

ログの時間の形式	<p>このフィールドでは、オプションで、ログ行のタイムスタンプを解析するためのパターンを指定できます。</p> <p>空白のままにすると、タイムスタンプは解析されません。</p> <p>サポートされているプレースホルダー:</p> <ul style="list-style-type: none"> * y: 年 (0001 -9999) * M: 月 (01-12) * d: 日 (01-31) * h: 時 (00-23) * m: 分 (00-59) * s: 秒 (00-59) <p>たとえば、Zabbix エージェントログファイルの次の行について考えてみましょう:</p> <pre>" 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)."</pre> <p>PID の 6 文字で始まり、その後の日付、時刻、残りの行が続きます。</p> <p>この行のログ時刻の形式は"pppppp:yyyyMMdd:hhmmss" になります。</p> <p>"p" と ":" の文字は単なるプレースホルダーであり、"yMdhms" 以外であれば何でもかまいません。</p>
----------	---

重要な注意事項

- server と agent は、監視対象ログのサイズと最終更新時間 (logrt の場合) のトレースを 2 つのカウンターに保持します。
 - また、agent は内部的に inode 番号 (UNIX/GNU/Linux)、ファイルインデックス (Microsoft Windows 上)、MD5 ログファイルの最初の 512 バイトの合計を使用しています。ログファイルが切り捨てられたり、ローテートした場合の判断を改善するためです。
 - UNIX/GNU/Linux システムでは、ログファイルが保存されているファイルシステムで、ファイルの追跡のために inode 番号を報告していると仮定します。
 - Microsoft Windows の場合、Zabbix agent はログファイルが存在するファイルシステムの種類を判断して使用します。
 - * NTFS ファイルシステムの場合、64 ビットファイルインデックスを使用します。
 - * ReFS ファイルシステムの場合 (Microsoft Windows Server 2012 から)128 ビットのファイル ID。
 - * ファイルインデックスが変更されるファイルシステム (例: FAT32、exFAT) では、ログファイルのローテートにより同じ最終更新時刻を持つ複数のログファイルが発生する場合、不確実な条件下で賢明なアプローチを取るためにフォールバックアルゴリズムが使用されます。
 - inode 番号、ファイルインデックス、MD5 サムは Zabbix agent が内部で収集します。これらは Zabbix server に転送されず、Zabbix agent を停止すると失われます。
 - ログファイルの最終更新時刻を 'touch' で変更したり、ログファイルをコピーして元の名前に戻したりしないでください (ファイルの inode 番号が変更されます)。どちらの場合もファイルが異なるとみなされ、最初から分析されるため、重複して警告が出されることがあります。
 - logrt [] item に一致するログファイルが複数あり、Zabbix agent がそのうちの最新のログファイルを追いかけていて、その最新のログファイルが削除された場合、警告メッセージ "there are no files matching "<regex mask>" in "<directory>" がログに記録されます。Zabbix agent は、チェック対象の logrt [] item の agent が見た最新の更新時刻より短い更新時刻のログファイルを無視します。
- agent は、前回停止した時点からログファイルの読み込みを開始します。
- 解析済みのバイト数 (サイズカウンタ) と最終更新時刻 (タイムカウンタ) は Zabbix データベースに保存され、agent が起動したばかりの時や、以前は無効またはサポートされていなかった item を受け取った時に、agent がこの時点からログファイルを読み始めるようにするために agent に送信されます。ただし、agent が server から 0 以外のサイズカウンターを受信していても、logrt[] または logrt.count[] item が見つからず、一致するファイルがない場合、サイズカウンターを 0 にリセットし、後にファイルが現れた場合に最初から解析するようにします。
- ログファイルが agent によって知られているログサイズカウンターより小さくなるたびに、カウンターはゼロにリセットされ、agent は時間カウンターを考慮して最初からログファイルの読み込みを開始します。
- ディレクトリ内に同じ最終修正時刻を持つ複数のマッチングファイルがある場合、agent は同じ修正時刻を持つすべてのログファイルを正しく分析し、データをスキップしたり同じデータを二度分析することを避けようとはしますが、すべての状況で保証されるわけではありません。agent は、特定のログファイルローテーションスキームを仮定せず、決定もしません。同じ最終修正時刻を持つ複数のログファイルが提示された場合、agent は辞書的に降順でそれらを処理します。したがって、いくつかのローテーションスキームでは、ログファイルは元の順序で分析され報告されます。他のローテーション方式では、元のログファイルの順番は守られず、変更された順番で一致したログファイルのレコードを報告することになります (ログファイルの最終修正時刻が異なる場合はこの問題は起こりません)。
- Zabbix agent は、ログファイルの新しいレコードを Update interval(更新間隔) 秒に 1 回処理します。
- Zabbix agent は 1 秒間に **maxlines** 行を超えるログファイルを送信しません。この制限はネットワークと CPU リソースの過負荷を防ぎ、agent configuration file の **MaxLinesPerSecond** パラメータのデフォルト値より優先されます。
- 必要な文字列を見つけるために、Zabbix は MaxLinesPerSecond で設定された 10 倍以上の新しい行を処理します。したがって、例えば、log [] または logrt [] item の Update interval(更新間隔) が 1 秒の場合、デフォルトでは agent は 200 以上のログファイルレコードを分析せず、1 回のチェックで 20 以上のマッチングレコードを Zabbix サーバに送信しません。エージェント設定ファイルの **MaxLinesPerSecond** を増やすか、アイテムキーの **maxlines** パラメータを設定することで、1 回のチェックでログファイルの解析レコードを最大 10000 件、Zabbix サーバに送信するマッチングレコードを最大 1000 件まで

増やすことが可能です。Update interval(更新間隔) を 2 秒に設定した場合、1 回のチェックの上限は更新間隔 *1 秒の場合の 2 倍となります。

- さらに、ログと log.count の値は、ログ以外の値がない場合でも、常に agent 送信バッファサイズの 50% に制限されます。したがって、1 つの接続で **maxlines** 値を送信するには (複数の接続ではなく)、agent の **BufferSize** パラメータを maxlines x 2 以上にする必要があります。
- ログ item がいない場合、すべての agent バッファサイズは非ログ値に使用されます。ログ値が来ると、指定された 50% を上限として、必要に応じて古い非ログ値を置き換えます。
- 256kB 以上のログファイルレコードの場合、最初の 256kB のみ正規表現と照合され、残りのレコードは無視されます。ただし、長いレコードを処理中に Zabbix agent が停止した場合、agent 内部の状態は失われ、agent の再起動後に長いレコードを再度解析し、異なる結果を得る可能性があります。
- "/" パスセパレータに関する特別な注意事項: file_format が "file//.log" の場合、"file" ディレクトリは存在してはならない。なぜなら、"." がエスケープされているか、ファイル名の最初のシンボルであるかを明確に定義することができないからである。
- logrt の正規表現はファイル名のみサポートされ、ディレクトリの正規表現マッチングはサポートされません。
- UNIX プラットフォームでは、ログファイルが見つかったと期待されるディレクトリが存在しない場合、logrt [] item は NOTSUPPORTED になります。
- Microsoft Windows では、ディレクトリが存在しない場合、item は NOTSUPPORTED になりません。(例えば、item キーのディレクトリのスペルが間違っている場合など)
- logrt [] item のログファイルがない場合でも、NOTSUPPORTED になるわけではありません。logrt [] item のログファイル読み込みのエラーは、Zabbix agent ログファイルに警告として記録されますが、そのアイテムが NOTSUPPORTED になることはありません。
- Zabbix agent ログファイルは、log [] または logrt [] アイテムが NOTSUPPORTED になった理由を見つけるために役立ちます。Zabbix は DebugLevel=4 以外の場合、agent ログファイルを監視することができます。

正規表現のマッチング部分を抽出する

正規表現にマッチしたときに、行全体を返すのではなく、ターゲットファイルから目的の値だけを取り出したい場合があります。

Zabbix 2.2.0 以降、ログアイテムはマッチした行から目的の値を抽出することができます。
これは、log と logrtitem に追加された **output** パラメータによって実現されています。

'output' パラメータを使用すると、マッチした行のうち、興味のある "capturing group" を指定することができます。

そのため、例えば

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1] で
```

以下のログ内容から、エントリ数 (Entries: の数値) を返すことができます。

```
Fr Feb 07 2014 11:07:36.6690 */ Thread Id 1400 (GLEWF) large result
buffer allocation - /Length: 437136/Entries: 5948/Client Ver: >=10/RPC
ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement
```

\1 は最初のキャプチャグループのため、番号のみが返されます。(**[0-9]+**)

また、数値を抽出して返す機能により、その値を使ってトリガーを定義することも可能です。

maxdelay パラメータの使用

ログ項目の 'maxdelay' パラメータを使用すると、ログファイルから古い行を無視して、'maxdelay' 秒以内に分析された最新の行を取得できます。

Warning:

maxdelay' > 0 を指定すると、重要なログファイルの記録が無視されたり、警告が見逃されたりすることがあります。必要なときだけ、自己責任で慎重に使用してください。

デフォルトでは、ログ監視のための項目は、ログファイルに現れるすべての新しい行に従います。しかし、ある状況下でログファイルに膨大な数のメッセージを書き始めるアプリケーションがあります。例えば、データベースや DNS サーバーが利用できない場合、そのようなアプリケーションは、正常な動作が回復するまで、ほぼ同じエラーメッセージを何千もログファイルに書き出します。デフォルトでは、これらのメッセージはすべて忠実に解析され、log と logrt の項目で設定されたとおりにマッチする行がサーバに送信されます。

過負荷に対する組み込みの保護は、構成可能な 'maxlines' パラメータ (一致するログ行が多すぎる受信からサーバーを保護する) と 10* 'maxlines' 制限 (1 回のチェックでエージェントによる過負荷からホスト CPU と I/O を保護する) で構成されます。それでも、組み込みの保護機能には 2 つの問題があります。まず、あまり有益ではない可能性のある多数のメッセージがサーバーに報告され、データベース内のスペースが消費されます。第 2 に、1 秒あたりに分析される行数が限られているため、エージェントは最新のログレコードから数時間遅れる可能性があります。多くの場合、古いレコードを何時間もクローलするよりも、ログファイルの現在の状況についてすぐに知りたいと思うかもしれません。

両方の問題の解決策は、'maxdelay' パラメーターを使用することです。'maxdelay' > 0 が指定されている場合、各チェック中に、処理されたバイト数、残りのバイト数、および処理時間が測定されます。これらの数値から、エージェントは推定遅延 (ログファイル内の残りのすべてのレコードを分析するのにかかる秒数) を計算します。

遅延が'maxdelay' を超えない場合、エージェントは通常どおりログファイルの分析を続行します。

遅延が'maxdelay' よりも大きい場合、エージェントは'maxdelay' 秒以内に残りの行を分析できるように、ログファイルのチャンクを無視して新しい推定位置に"ジャンプ" します。

エージェントは無視された行をバッファに読み込むことさえせず、ファイル内のジャンプ先のおおよその位置を計算することに注意してください。

ログファイルの行をスキップしたという事実は、次のようにエージェントログファイルに記録されます。

```
14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,120.0]" logfile:"/home/
```

"ジャンプ" の後、エージェントはファイル内の位置をログ行の先頭に調整するため、"バイトまで" の数値は概算です。

成長速度とログファイルの分析速度を比較する方法によっては、すべてのチェックで"ジャンプ" が見られない場合、ほとんどまたは頻繁に"ジャンプ" が見られる場合、大きなまたは小さな"ジャンプ" が見られる場合、または小さな"ジャンプ" が見られる場合さえあります。システム負荷とネットワーク遅延の変動も遅延の計算に影響を与えるため、"maxdelay" パラメータに追いつくために先に"ジャンプ" します。

'maxdelay' < 'update interval' を設定することはお勧めしません (小さな"ジャンプ" が頻繁に発生する可能性があります)

'copytruncate' ログファイルローテーションの取り扱いに関する注意点

logrt の copytruncate オプションは、異なるログファイルには異なる記録がある (少なくともタイムスタンプは異なる) と
 仮定しています。したがって、初期ブロック (最初の 512 バイトまで) の MD5 チェックサムは異なることになります。
 初期ブロックの MD5 チェックサムが同じである 2 つのファイルは、一方がオリジナルで、もう一方がコピーであることを意味します。

logrt の copytruncate オプションは、ログファイルのコピーを重複して報告することなく、正しく処理するように努力します。
 しかし、同じタイムスタンプのログファイルを複数作成したり、ログファイルのローテーションが logrt[] item の update interval より
 頻繁になったり、agent を頻繁に再起動することは推奨されません。agent は、これらの状況を合理的に処理しようとはしますが、
 すべての状況でよい結果を保証することはできません。

log*[] item の persistent files に関する注意事項

I/O の負荷

Item の永続化ファイルは、item のデータを含むデータを server に送信するたびに更新されます。例えば、デフォルトの'BufferSize' は 100 です。
 ログアイテムが 70 件のマッチングレコードを発見した場合、最初の 50 件のレコードを一括して送信し、永続化ファイルを更新した後、
 残りの 20 レコードが 2 回目のバッチで送信され、永続化ファイルが再び更新されます。

agent と server 間の通信に失敗した場合の対処法

log[] と logrt[]item からマッチする各行と、各 log.count[] と logrt.count[]' item チェックの結果は、
 agent 送信バッファ内の指定された 50% エリアに空きスロットを必要とします。バッファの要素は定期的に server (または proxy) に送信され、
 バッファのスロットは再び空きます。

agent 送信バッファの指定ログ領域に空き枠があり、agent と server (または proxy) 間の通信に失敗した場合、
 ログ監視結果が送信バッファに蓄積されます。これにより、短時間の通信障害を緩和することができます。

長時間の通信障害時には、すべてのログスロットが占有され、次のような動作が行われます。

- log[] と logrt[]' の item チェックは停止しています。通信が復旧し、バッファの空きスロットができると、チェックは前の位置から再開されます。マッチする行は失われず、後で報告されるだけです。
- maxdelay = 0 (デフォルト) の場合、log.count[] と logrt.count[] のチェックが停止します。動作は上記の log[] と logrt[] の項目と同様です。これは log.count[] と logrt.count[]' の結果に影響を与えることに注意してください。例えば、あるチェックでログファイルの 100 行にマッチしましたが、バッファに空きスロットがないため、チェックは停止されました。通信が回復したとき、agent は同じ 100 のマッチする行と、70 の新しいマッチする行を数えます。agent は、あたかも一度のチェックで見つかったかのように、count = 170 を送信します。
- maxdelay > 0 の log.count[] と logrt.count[] のチェック: チェック中に"jump" がなければ、上記の動作と同じになります。もし、ログファイルの行を"jump" した場合は、"jump" 後の位置を保持し、カウントした結果は破棄されます。したがって、agent は、通信障害が発生しても、ログファイルの増加についていこうとします。

Handling of regular expression compilation and runtime errors

If a regular expression used in log[], logrt[], log.count[] or logrt.count[] item cannot be compiled by PCRE or PCRE2 library then the item goes into NOTSUPPORTED state with an error message. To continue monitoring the log item, the regular expression should be fixed.

If the regular expression compiles successfully, but fails at runtime (on some or on all log records), then the log item remains supported and monitoring continues. The runtime error is logged in the Zabbix agent log file (without the log file record).

Note that the logging of regular expression runtime errors is supported since Zabbix 6.0.21.

The logging rate is limited to one runtime error per check to allow Zabbix agent to monitor its own log file. For example, if 10 records are analyzed and 3 records fail with a regexp runtime error, one record is produced in the agent log.

Exception: if MaxLinesPerSecond=1 and update interval=1 (only 1 record is allowed to analyze per check) then regexp runtime errors are not logged.

zabbix_agentd logs the item key in case of a runtime error, zabbix_agent2 logs the item ID to help identify which log item has runtime errors. It is recommended to redesign the regular expression in case of runtime errors.

永続化ファイルの目的

Zabbix agent の起動時に、Zabbix server または proxy からアクティブチェックのリストを受け取ります。
 log*[] メトリクスでは、ログファイルの監視を開始する場所を見つけるために、処理されたログのサイズと変更時間を受け取ります。
 ファイルシステムから報告された実際のログファイルサイズと変更時間に応じて、agent は処理されたログサイズから
 ログファイル監視を継続するか、ログファイルを最初から再分析するかを決定します。

実行中の agent は、チェックの間に監視されたすべてのログファイルを追跡するために、
 より大きな属性のセットを維持します。このインメモリの状態は、エージェントが停止されると失われます。

新しいオプションパラメータ **persistent_dir** は、この log[], log.count[], logrt[] または logrt.count[] item の状態を
 ファイルに格納するディレクトリを指定するものです。Zabbix agent を再起動すると、ログアイテムの状態は永続化ファイルから復元されます。

主な用途は、ミラーリングされたファイルシステム上にあるログファイルの監視です。ある時点まで、ログファイルは
 両方のミラーに書き込まれます。その後、ミラーは分割されます。アクティブなコピーでは、ログファイルは新しいレコードを
 取得し、まだ成長しています。Zabbix agent はこれを解析し、処理されたログのサイズと更新時刻を server に送信します。
 パッシブコピーでは、ログファイルは同じまま、アクティブコピーよりかなり遅れています。その後、OS と Zabbix エージェントが
 パッシブコピーからレポートされます。Zabbix agent が server から受け取った処理済みのログサイズと更新時刻は、
 パッシブコピー上の状況に対して有効でない場合があります。ファイルシステムのミラースプリット時に agent が停止した場所から
 ログファイルの監視を継続するために、agent は永続ファイルから状態をリストアします。

永続ファイルを使用した agent 操作

起動時、Zabbix agent は永続化ファイルについて何も知りません。Zabbix サーバからアクティブチェックのリストを受け取った後、
 agent はいくつかのログアイテムが指定されたディレクトリの永続ファイルでバックアップされることを認識します。

agent 操作の間、永続ファイルは書き込みのために開かれ (fopen(filename, "w"))、最新のデータで上書きされます。
 上書きとファイルシステムのミラーリングが同時に起こった場合、永続ファイルのデータが失われる可能性は非常に低いので、
 特別な処理は行いません。永続ファイルへの書き込みの後にメディアへの強制同期を行わない (fsync() is not called) は呼び出されないため、
 永続ファイルへの書き込みの後にメディアへの強制同期を行うことはできません。

最新データの上書きは、一致するログファイルのレコードまたはメタデータ (処理されたログのサイズと更新時間) が Zabbix server に正常に報告された後に行われます。ログファイルが変化し続ける場合、item チェックのたびに実行される可能性があります。

agent シャットダウン中の特別な動作はありません。

アクティブなチェックのリストを受け取った後、agent は廃止された永続ファイルを削除するようマークします。
 以下の場合、永続ファイルは廃止されます。
 1) 対応するログ項目が監視されなくなった、
 2) ログ項目が以前と異なる **persistent_dir** 位置で再設定された。

NOTSUPPORTED 状態のログファイルはアクティブチェックのリストに含まれませんが、後に SUPPORTED になる可能性があり、
 その永続ファイルが有用であるため、24 時間遅れて削除されます。

24 時間経過前に agent を停止した場合、Zabbix agent は Zabbix server からファイルの所在情報を取得しなくなるため、
 古くなったファイルは削除されません。

Warning:

agent が停止している間に、ユーザが古い永続ファイルを削除せずに、ログアイテムの **persistent_dir** を古い **persistent_dir** の場所に再構成すると、古い永続ファイルからエージェント状態をリストアすることになり、メッセージの見逃しや誤った警告が発生します。

永続ファイルの命名と場所

Zabbix agent は、アクティブなチェックをキーで区別します。例えば、logrt[/home/zabbix/test.log] と
 logrt[/home/zabbix/test.log,] は異なるアイテムです。フロントエンドの item logrt[/home/zabbix/test.log,,,10] を
 logrt[/home/zabbix/test.log,,,20] に変更すると、item logrt[/home/zabbix/test.log,,,10] が削除されます。
 logrt[/home/zabbix/test.log,,,10] を agent のアクティブチェックリストから削除し、logrt[/home/zabbix/test.log,,,20] item を作成します。
 (いくつかの属性は、agent ではなくフロントエンド/サーバで変更されます)

ファイル名は、item キーの MD5 チェックサムと、衝突の可能性を低減するために item キーの長さを付加したものです。
 例えば、logrt[/home/zabbix50/test.log,,,,,/home/zabbix50/agent_private] item の状態は、
 永続ファイル c963ade4008054813bbc0a650bb8e09266 に保存されます。

persistent_dir は、複数のログ item で同じ値を使用できます。

persistent_dir は、特定のファイルシステムのレイアウト、マウントポイント、マウントオプション、ストレージのミラーリング設定を
 考慮して指定されます。

persistent_dir ディレクトリが作成できないか存在しないか、Zabbix agent のアクセス権でファイルの作成/書き込み/読み取り/削除が
 できない場合、ログ item は NOTSUPPORTED になります。

agent 操作中に永続的ストレージファイルへのアクセス権が削除されたり、他のエラー (例: ディスクフル) が発生した場合、エラーは agent ログファイルに記録されますが、ログ item が NOTSUPPORTED になることはありません。

7 計算 item

概要

計算項目を使用すると、他の項目の値を基に計算を行うことができます。

計算では、次の両方を使用できます。

- 個々の項目の単一の値
- 集計のために複数の項目を選択するための複雑なフィルター (**aggregate calculations**を参照してください)。

このように、計算 item は仮想データソースを作成するための手段です。すべての計算は Zabbix server のみによって行われます。
 値は使用された算術式に基づき定期的に計算されます。

計算結果のデータは他のアイテムと同様に Zabbix データベースに保存され、ヒストリー値とトレンド値の両方が保存され、
 グラフを作成することも可能です。

Note:

計算結果が float 値の場合、計算された item の情報タイプが数値 (符号なし) であれば、整数に切り詰められます。

計算 item は、トリガー式 **expressions** と構文を共有しています。
 計算 item は、文字列との比較も可能です。計算項目は、他の項目型と同様に、マクロや他のエンティティから参照することができます。

計算 item を使用するには、item タイプ **Calculated** を選択します。

設定可能なフィールド

キーは、(ホストごとに) 一意なアイテム識別子です。サポートされているシンボルを使用して、任意のキー名を作成することができます。

計算の定義は、**Formula** の欄に入力してください。数式とキーの間には事実上何の関係もありません。
 キーのパラメータは、数式では一切使用されません。

簡単な式の構文は以下の通りです。

```
function(/host/key,<parameter1>,<parameter2>,...)
```

ここで

function	One of the supported functions : last, min, max, avg, count, etc
host	Host of the item that is used for calculation. The current host can be omitted (i.e. as in function(/key,parameter,...)).
key	Key of the item that is used for calculation.
parameter(s)	Parameters of the function, if required.

Attention:

関数パラメータ、item フィルタパラメータ、定数を参照するために使用される場合、数式内の
 User macros は展開されます。関数、ホスト名、item キー、item キーパラメータ、
 演算子を参照する場合、ユーザーマクロは展開されません。

より複雑な式では、関数、演算子、および括弧を組み合わせて使用することができます。トリガー式でサポートされている
 すべての関数と演算子 **operators** を使用することができます。
 ロジックと演算子の優先順位は全く同じです。

Zabbix はトリガー式とは異なり、新しい値を受け取った時ではなく、item の更新間隔に従って計算された item を処理します。

計算 item 式の中で履歴関数によって参照される item は、すべて存在し、データを収集している必要があります。
 また、参照される item の item キーを変更した場合、そのキーを使用しているすべての数式を手動で更新する必要があります。

計算された item は、いくつかのケースでサポートされなくなることがあります:

- referenced item(s)
 - is not found
 - is disabled
 - belongs to a disabled host
 - is not supported (except with nodata() function and operators with unknown values)
- no data to calculate a function
- division by zero
- incorrect syntax used

使用例

例 1

'/' のディスク空き容量のパーセンテージを計算します。

Use of function **last**:

```
100*last(/vfs.fs.size[/,free])/last(/vfs.fs.size[/,total])
```

Zabbix はフリーディスクスペースとトータルディスクスペースの最新値を取得し、指定された計算式に従ってパーセンテージを計算します。

例 2

Zabbix で処理された値の 10 分間の平均値を計算します。

Use of function **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

長い時間の計算 item を多用すると、Zabbix server のパフォーマンスに影響を与える可能性があることに注意してください。

例 3

eth0 の総帯域幅を計算する。

2 つの関数の合計:

```
last(/net.if.in[eth0,bytes])+last(/net.if.out[eth0,bytes])
```

例 4

受信トラフィックのパーセンテージを計算する。

より複雑な表現:

```
100*last(/net.if.in[eth0,bytes])/(last(/net.if.in[eth0,bytes])+last(/net.if.out[eth0,bytes]))
```

Examples of aggregate calculationsを参照してください。

集計計算

概要

集計計算は、Zabbix サーバーによって複数の項目から情報を収集し、使用する集計関数に応じて集計を計算できる**計算項目**タイプです。

集計計算項目では、符号なしの整数値と浮動小数点値 (情報の種類) のみがサポートされています。

集計計算では、監視対象のホストでエージェントを実行する必要はありません。

文法

集計は、どちらかの作業で取得することができます。:

- history of items:

```
aggregate_function(function(/host/item,parameter),function(/host2/item2,parameter),...)
```

- a **foreach function** を唯一のパラメータとして:

```
aggregate_function(foreach_function(/*/key?[group="host group"],timeperiod))
```

where:

- aggregate_function' は、サポートされている aggregate functions: avg, max, min, sum, etc のいずれかである。
- foreach_function' は、サポートされている foreach 関数 (avg_foreach, count_foreach など) のうちの 1 つです。

Foreach 関数は、複数の項目の履歴を扱うために item filter を使用し、各項目につき 1 つの値を配列で返します。

Note:

集計結果が float 値の場合、集計された情報の item タイプが Numeric(unsigned) であれば、整数に切り捨てられます。

以下の場合、集計計算がサポートされなくなる可能性があります:

- 参照した項目が見つからない (item キーが正しくない、item が存在しない、含まれるグループがすべて正しくない場合に発生する可能性があります)
- 関数を計算するためのデータがない

使用例

集計のためのキーの例です。

例 1

ホストグループ 'MySQL Servers' の総ディスク容量

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

例 2

そのホストで net.if.in[*] に一致するすべての item の最新値の合計

```
sum(last_foreach(/host/net.if.in[*]))
```

例 3

ホストグループ 'MySQL Servers' の平均プロセッサ負荷

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

例 4

ホストグループ 'MySQL Servers' の 1 秒あたりのクエリ数の 5 分平均値

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

例 5

特定のタグを持つ複数のホストグループに属する全ホストの CPU 負荷の平均値

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (tag=tag1)]))
```

例 6

ホストグループ全体の最新の item 値合計で計算します

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[group="video"]))
```

例 7

The total number of unsupported items in host group 'Zabbix servers'. ホストグループ 'Zabbix servers' のサポート対象外 item の総数です。

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

正しい/間違った構文の例

式 (関数呼び出しを含む) は、ヒストリー、トレンド、または foreach 関数 パラメーターとして使用できません。ただし、これらの関数自体は、他の (非ヒストリ) 関数パラメーターで使用できます。

式

例

有効

```
avg(last(/host/key1),last(/host/key2)*10,last(/host/key1)*100)
max(avg(avg_foreach(/*/system.cpu.load?[group="Servers A"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers B"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers C"],5m)))
```

式	例
! 無効	<pre>sum(/host/key,10+2) sum(/host/key, avg(10,2)) sum(/host/key,last(/host/key2))</pre>

次のような表現に注意してください。

```
sum(sum_foreach(/resptime[*],5m))/sum(count_foreach(/resptime[*],5m))
```

式の両方の部分が常に同じ値のセットになるとは限りません。式の一部が評価されている間に、要求された期間の新しい値が到着し、式の他の部分が異なる値のセットを持つことがあります。

8 内部チェック

概要

内部チェックでは、Zabbix の内部処理を監視することができます。つまり、Zabbix サーバーや Zabbix プロキシで何が行われているのかを監視することができます。

内部チェックを算出:

- Zabbix サーバーの場合 - ホストがサーバーによって監視されている場合
- Zabbix プロキシの場合 - ホストがプロキシによって監視されている場合

内部チェックは、ホストのメンテナンス状態に関係なく、サーバーまたはプロキシで処理されます。

このアイテムを使用するには、アイテムの **Zabbix** インターナルタイプを選択してください。

Note:

内部チェックは Zabbix ポーラープロセスによって処理されます

パフォーマンス

いくつかの内部チェックアイテムを使用すると、パフォーマンスに悪影響を及ぼす可能性があります。そのアイテムは以下の通り:

- zabbix[host,,items]
- zabbix[host,,items_unsupported]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[queue]
- zabbix[required_performance]
- zabbix[stats,,queue]
- zabbix[triggers]

The **システム情報** と **キュー** の Web インターフェースも影響を受けます。

サポートするチェック

- 角括弧のないパラメータは定数です。例えば、zabbix[host,<type>,available] の 'host' と 'available' これらはアイテムキーでそのまま使用します。
- プロキシでサポートされていない ("not supported onproxy") アイテムおよびアイテムの引数の値は、ホストがサーバーによって監視されている場合にのみ収集できます。逆に、サーバーでサポートされていない ("not supported on server") 値は、ホストがプロキシ監視されている場合にのみ収集することができます。

Key

▲ 説明	戻り値	コメント
zabbix[boottime] Startup time of Zabbix server or Zabbix proxy process in seconds.	Integer.	
zabbix[cluster,discovery,nodes]		

Key

Discover **high availability**
cluster nodes.

JSON.

This item can be used in low-level discovery.

zabbix[history]

Number of values stored in the HISTORY table.

Integer.

This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! (not supported on proxy)

zabbix[history_log]

Key

	Number of values stored in the HISTORY_LOG table.	Integer.	This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! (not supported on proxy)
zabbix[history_str]	Number of values stored in the HISTORY_STR table.	Integer.	This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! (not supported on proxy)
zabbix[history_text]			

Key

Number of values stored in the HISTORY_TEXT table.

Integer.

This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! (not supported on proxy)

zabbix[history_uint]

Key

Number of values stored in the HISTORY_UINT table.	Integer.	This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! This item is supported since Zabbix 1.8.3. (not supported on proxy)
zabbix[host,,items] Number of enabled items (supported and not supported) on the host.	Integer.	This item is supported since Zabbix 3.0.0.
zabbix[host,,items_unsupported] Number of enabled unsupported items on the host.	Integer.	This item is supported since Zabbix 3.0.0.*
zabbix[host,,maintenance]		

Key

Current maintenance status
of a host.

0 - host in normal state,
1 - host in maintenance with
data collection,
2 - host in maintenance
without data collection.

This item is always processed by Zabbix server regardless of host location (on server or proxy). The proxy will not receive this item with configuration data. The second parameter must be empty and is reserved for future use.

zabbix[host,discovery,interfaces]

Key

Details of all configured interfaces of the host in Zabbix frontend.

JSON object.

This item can be used in **low-level discovery**. This item is supported since Zabbix 3.4.0. (not supported on proxy)

zabbix[host,<type>,available]

Key

Availability of the main interface of a particular type of checks on the host.

0 - not available, 1 - available, 2 - unknown.

Valid **types** are: agent, snmp, ipmi, jmx

The item value is calculated according to configuration parameters regarding host **unreachability/unavailability**.

This item is supported since Zabbix 2.0.0.

zabbix[hosts]

Number of monitored hosts.

Integer.

zabbix[items]

Number of enabled items (supported and not supported).

Integer.

zabbix[items_unsupported]

Number of not supported items.

Integer.

zabbix[java,,<param>]

Key

Information about Zabbix
Java gateway.

If <param> is ping, "1" is
returned. Can be used to
check Java gateway
availability using nodata()
trigger function.

If <param> is version,
version of Java gateway is
returned. Example: "2.0.0".

Valid
val-
ues
for
param
are:
ping,
ver-
sion

Second
pa-
ram-
eter
must
be
empty
and
is re-
served
for
fu-
ture
use.

zabbix[lld_queue]

Count of values enqueued in
the low-level discovery
processing queue.

Integer.

This
item
can
be
used
to
mon-
itor
the
low-
level
dis-
cov-
ery
pro-
cess-
ing
queue
length.

This
item
is
sup-
ported
since
Zab-
bix
4.2.0.

zabbix[preprocessing_queue]

Key

Count of values enqueued in the preprocessing queue.

Integer.

This item can be used to monitor the preprocessing queue length.

This item is supported since Zabbix 3.4.0.

zabbix[process,<type>,<mode>,<state>]

Time a particular Zabbix process or a group of processes (identified by <type> and <mode>) spent in <state> in percentage. It is calculated for the last minute only.

If <mode> is Zabbix process number that is not running (for example, with 5 pollers running <mode> is specified to be 6), such an item will turn into unsupported state. Minimum and maximum refers to the usage percentage for a single process. So if in a group of 3 pollers usage percentages per process were 2, 18 and 66, min would return 2 and max would return 66.

Processes report what they are doing in shared memory and the self-monitoring process summarizes that data each second. State changes (busy/idle) are registered upon change - thus a process that becomes busy registers as such and doesn't change or update the state until it becomes idle. This ensures that even fully hung processes will be correctly registered as 100% busy. Currently, "busy" means "not sleeping", but in the future additional states might be introduced - waiting for locks, performing database queries, etc.

On Linux and most other systems, resolution is 1/100 of a second.

Percentage of time.
Float.

Supported **types** of **server processes**: alert manager, alert syncer, alerter, availability manager, configuration syncer, discoverer, escalator, history poller, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, odbc poller, poller, pre-processing manager, pre-processing

Key

zabbix[proxy,<name>,<param>]

Information about Zabbix proxy.

Integer.

name:
proxy name

Valid values for **param** are:
lastaccess - timestamp of last heart beat message received from proxy delay - how long collected values are un-sent, calculated as "proxy delay" (difference between the current proxy time and the timestamp of the oldest un-sent value on proxy)
+

Key

zabbix[proxy_history]	Number of values in the proxy history table waiting to be sent to the server.	Integer.	(not supported on server)
zabbix[queue,<from>,<to>]	Number of monitored items in the queue which are delayed at least by <from> seconds but less than by <to> seconds.	Integer.	from - default: 6 seconds to - default: infinity Time-unit symbols (s,m,h,d,w) are supported for these parameters.
zabbix[rcache,<cache>,<mode>]			

Key

Availability statistics of
Zabbix configuration cache.

Integer (for size); float (for
percentage).

cache:
buffer

Valid
modes

are:
total

-
total
size
of
buffer
free

- size
of
free

buffer
pfree

- per-
cent-
age
of

free
buffer

used
- size

of
used

buffer
pused

- per-
cent-
age
of

used
buffer

pused
mode

is
sup-
ported

since

Zab-
bix

4.0.0.

zabbix[requiredperformance]

Key

Required performance of Zabbix server or Zabbix proxy, in new values per second expected.

Float.

Approximately correlates with "Required server performance, new values per second" in Reports → [System information](#).

zabbix[stats,<ip>,<port>]

Remote Zabbix server or proxy internal metrics.

JSON object.

ip - IP/DNS/network mask list of servers/proxies to be re-motely queried (default is 127.0.0.1)
port - port of server/proxy to be re-motely queried (default is 10051)

Note that the stats request will only be accepted from the addresses listed in the 'Stat-sAllowedIP' *server/proxy* parameter on the target instance.

A selected set of internal

Key

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

Remote Zabbix server or proxy internal queue metrics (see `zabbix[queue,<from>,<to>]`).

JSON object.

ip - IP/DNS/network mask list of servers/proxies to be remotely queried (default is 127.0.0.1)
port - port of server/proxy to be remotely queried (default is 10051)
from - delayed by at least (default is 6 seconds)
to - delayed by at most (default is infinity)
Note that the stats request will only be accepted from the addresses listed in the

Key

zabbix[tcache,cache,<parameter>]

Key

Effectiveness statistics of the Zabbix trend function cache.

Integer (for size); float (for percentage).

Valid parameters are:
all - total cache requests (default) hits - cache hits phits - percentage of cache hits misses - cache misses pmisses - percentage of cache misses items - the number of cached items requests - the number of cached requests pitems - percentage of cached items from cached items + requests. Low percentage

Key

zabbix[trends]

Number of values stored in the TRENDS table.

Integer.

This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! (not supported on proxy)

zabbix[trends_uint]

Key

Number of values stored in the TRENDS_UINT table.	Integer.	This item is deprecated since Zabbix 6.0. Do not use if MySQL InnoDB, Oracle or PostgreSQL is used! This item is supported since Zabbix 1.8.3. (not supported on proxy)
<code>zabbix[triggers]</code> Number of enabled triggers in Zabbix database, with all items enabled on enabled hosts.	Integer.	(not supported on proxy)
<code>zabbix[uptime]</code> Uptime of Zabbix server or Zabbix proxy process in seconds.	Integer.	
<code>zabbix[vcache,buffer,<mode>]</code>		

Key

Availability statistics of
Zabbix value cache.

Integer (for size); float (for
percentage).

Valid
modes
are:
total
-
total
size
of
buffer
free
- size
of
free
buffer
pfree
- per-
cent-
age
of
free
buffer
used
- size
of
used
buffer
pused
- per-
cent-
age
of
used
buffer

(not
sup-
ported
on
proxy)

zabbix[vcache,cache,<parameter>]

Key

Effectiveness statistics of
Zabbix value cache.

Integer.

With the mode parameter:
0 - normal mode,
1 - low memory mode

Valid **pa-ram-eter** values are: requests - total number of requests hits - number of cache hits (history values taken from the cache) misses - number of cache misses (history values taken from the database) mode - value cache operating mode

This item is supported since Zabbix 2.2.0 and the mode parameter

Key

zabbix[version]	Version of Zabbix server or proxy.	String.	This item is supported since Zabbix 5.0.0.
zabbix[vmware,buffer,<mode>]	Availability statistics of Zabbix vmware cache.	Integer (for size); float (for percentage).	Example of return value: 5.0.0beta1 Valid modes are: total - total size of buffer free - size of free buffer pfree - percentage of free buffer used - size of used buffer pused - percentage of used buffer
zabbix[wcache,<cache>,<mode>]	Statistics and availability of Zabbix write cache.		Specifying <cache> is mandatory.
Cache	Mode		

Key

values	all (default)	Total number of values processed by Zabbix server or Zabbix proxy, except unsupported items.	Integer.	Counter. You may use this key with the Change per second pre-processing step in order to get values per second statistics.
	float	Number of processed float values.	Integer.	Counter.
	uint	Number of processed unsigned integer values.	Integer.	Counter.
	str	Number of processed character/string values.	Integer.	Counter.
	log	Number of processed log values.	Integer.	Counter.
	text	Number of processed text values.	Integer.	Counter.
	not supported	Number of times item processing resulted in item becoming unsupported or keeping that state.	Integer.	Counter.

Key

history	pfree (default)	Percentage of free history buffer.	Float.	History cache is used to store item values. A low number indicates performance problems on the database side.
	free	Size of free history buffer.	Integer.	
	total	Total size of history buffer.	Integer.	
	used	Size of used history buffer.	Integer.	
	pusd	Percentage of used history buffer.	Float.	pusd mode is supported since Zabbix 4.0.0.
index	pfree (default)	Percentage of free history index buffer.	Float.	History index cache is used to index values stored in history cache. Index cache is supported since Zabbix 3.0.0.
	free	Size of free history index history buffer.	Integer.	

Key

	Key	Description	Type	Notes
trend	total	Total size of history index history buffer.	Integer.	
	used	Size of used history index history buffer.	Integer.	
	pusd	Percentage of used history index buffer.	Float.	pusd mode is supported since Zabbix 4.0.0.
	pfree (default)	Percentage of free trend cache.	Float.	Trend cache stores aggregate for the current hour for all items that receive data. (not supported on proxy)
	free	Size of free trend buffer.	Integer.	(not supported on proxy)
	total	Total size of trend buffer.	Integer.	(not supported on proxy)
	used	Size of used trend buffer.	Integer.	(not supported on proxy)

Key	used	Percentage of used trend buffer.	Float.	(not supported on proxy)
				used mode is supported since Zabbix 4.0.0.

9 SSH チェック

概要

SSH チェックはエージェントレスで実行できます。SSH チェックを行うためには Zabbix エージェントは必要ありません。

SSH チェックを実行するには、Zabbix サーバーを構築時の最初に **configure** する際に SSH2(libssh2 または libssh で) をサポートするようにする必要があります。以下、参照：[必要な要件](#)。

Attention:

RHEL 8 以降では、libssh だけがサポートされています。

設定

パスワード認証

SSH チェックには、ユーザーパスワードペアとキーファイルベースの 2 つの認証方法があります。

鍵を使用しない場合、ソースからビルドする場合は、libssh2/libssh を Zabbix にリンクする以外に、追加の設定は必要ありません。

キーファイル認証

SSH アイテムに鍵ベースの認証を使用するためには、サーバーに特定の設定を行うことが必要です。

Zabbix サーバーの設定ファイル (`zabbix_server.conf`) を root 権限で開き、以下の行を確認します。

```
# SSHKeyLocation=
```

コメントを外して、公開鍵と秘密鍵が格納されるフォルダのフルパスを設定します。

```
SSHKeyLocation=/home/zabbix/.ssh
```

ファイルを保存し、その後 `zabbix_server` を再起動します。

`/home/zabbix` ここに `zabbix` ユーザーアカウントのホームディレクトリがあります。`*/.ssh` は、ホームディレクトリ内で `ssh-keygen` コマンドを使用して公開鍵と秘密鍵を生成するディレクトリです。

通常、様々な OS ディストリビューションからの `zabbix-server` インストールパッケージでは、あまり知られていない場所 (システムアカウントのように) にホームディレクトリを持つ `zabbix` ユーザーアカウントを作成します。例えば `/var/lib/zabbix`。

鍵の生成を開始する前に、ホームディレクトリをより良く知られた場所 (直感的に予想できる場所) に再配置するアプローチを検討することができます。これは、前述の `SSHKeyLocation` という Zabbix server の設定パラメータに対応します。

ソースコードからのインストールに従って手動で `zabbix` アカウントを追加した場合、ホームディレクトリが既に `/home/zabbix` にある可能性が高いため、これらの手順は省略することができます。

`zabbix` ユーザーアカウントの設定を変更するには、そのアカウントを使用しているすべての作業プロセスを停止する必要があります。

```
# service zabbix-agent stop
```

```
# service zabbix-server stop
```

ホームディレクトリの場所を変更するには (存在する場合)、コマンドを実行する必要があります。


```
# usermod -m -d /home/zabbix zabbix
```

古い場所にホームディレクトリが存在しない可能性が十分にあるので、新しい場所に作成する必要があります。そのためには以下のように実行します：

```
# test -d /home/zabbix || mkdir /home/zabbix
```

すべてが安全であることを確認するために、追加のコマンドを実行して、ホームディレクトリのパーミッションを設定することができます。

```
# chown zabbix:zabbix /home/zabbix
# chmod 700 /home/zabbix
```

停止していたプロセスを再び起動できるようになります：

```
# service zabbix-agent start
# service zabbix-server start
```

これで、公開鍵と秘密鍵を生成する手順を、コマンドで実行できるようになりました：

```
# sudo -u zabbix ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
Created directory '/home/zabbix/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zabbix/.ssh/id_rsa.
Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
The key fingerprint is:
90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
The key's randomart image is:
+---[ RSA 2048]-----+
|          |
|          |
|   o      |
| .  o     |
|+   . S   |
|. +  o =   |
|E .  * =   |
|=o . . .* .|
|... oo.o+  |
+-----+

```

注：公開鍵と秘密鍵（それぞれ `id_rsa.pub` と `id_rsa`）は、デフォルトで Zabbix サーバーの `SSHKeyLocation` 設定パラメータに対応する `/home/zabbix/.ssh` ディレクトリに生成されています。

Attention:

"rsa" 以外の鍵の種類は、ssh-keygen ツールや SSH サーバーではサポートされていますが、Zabbix で使用されている libssh2 ではサポートされていない場合があります。

シェル設定フォーム

この手順は、SSH チェックで監視する各ホストに対して 1 回だけ実行する必要があります。

次のコマンドを使用して、**public** 鍵ファイルをリモートホスト 10.10.10.10 にインストールし、
 root アカウントで SSH チェックを実行できるようにすることができます。

```
# sudo -u zabbix ssh-copy-id root@10.10.10.10
The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
root@10.10.10.10's password:
Now try logging into the machine, with "ssh 'root@10.10.10.10'", and check in:
 .ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
```

これで、zabbix ユーザーアカウントのデフォルトの秘密鍵 (`/home/zabbix/.ssh/id_rsa`) を使用した SSH ログインを
 確認できるようになります：

```
# sudo -u zabbix ssh root@10.10.10.10
```

ログインに成功した場合は、シェルの設定が終了し、リモート SSH セッションを閉じることができます。

アイテム設定

実行する実際のコマンドは、アイテム設定の 実行するスクリプトフィールドに記載する必要があります。

複数のコマンドを実行するには、複数行にコマンドを記載することで実行できます。この場合、返される値も複数行としてフォーマットされます。

The screenshot shows the configuration form for an SSH agent item in Zabbix. The fields are as follows:

- Name:** SSH test check (without passphrase)
- Type:** SSH agent
- Key:** ssh.run[clear]
- Type of information:** Text
- Host interface:** 10.10.10.10:10050
- Authentication method:** Public key
- User name:** root
- Public key file:** id_rsa.pub
- Private key file:** id_rsa
- Key passphrase:** (empty)
- Executed script:** service mysql-server status
- Update interval:** 1m

すべての必須入力フィールドには赤いアスタリスクが付いています。

SSH アイテムの特定の情報を必要とするフィールドは次のとおりです。

パラメータ	説明	コメント
タイプ キー	SSH エージェントを選択 (ホストごとに) 一意なアイテムキーフォーマット ssh.run[<unique short description>,<ip>,<port>,<encoding>]	<unique short description> は必須であり、ホストごとにすべての SSH アイテムに対して一意である必要があります デフォルトポートは 22 であり、このアイテムが割り当てられているインターフェイスで指定されたポートではありません
認証方法 ユーザー名	"パスワード" または "公開鍵" のいずれか リモートホストで認証するユーザー名 必須	
公開鍵ファイル	認証方式が "公開鍵" の場合の公開鍵ファイル名。 必須	例: id_rsa.pub - ssh-keygen コマンドによって生成されるデフォルトの公開鍵ファイル名
秘密鍵ファイル	認証方式が "公開鍵" の場合の秘密鍵ファイル名。 必須	例: id_rsa - デフォルトの秘密鍵ファイル名

パラメータ	説明	コメント
パスワード	認証するためのパスワードまたはパスフレーズ 秘密鍵に使用された場合	パスフレーズが使用されていない場合は、パスワードフィールドを空にします パスフレーズの使用については 既知の問題 も参照してください
実行するスクリプト	SSH リモートセッションを使用してシェルコマンドを実行します	例: date +%s service mysql-server status ps auxww grep httpd wc -l

Attention:

libssh2 ライブラリは、実行可能スクリプトを ~32kB に切り詰める場合があります。

10 Telnet チェック

概要

Telnet チェックはエージェントレス監視として実行されます。Zabbix agent は必要ありません。

設定可能なフィールド

実行する実際のコマンドは、アイテム設定の 実行するスクリプトフィールドに記載する必要があります。複数のコマンドを実行するには、複数行にコマンドを記載することで実行できます。この場合、返される値も複数行としてフォーマットされます。

シェルプロンプトの最後に使用できるサポートされている文字:

- \$
- #
- >
- %

Note:

これらの文字のいずれかで終わる telnet プロンプト行は、戻り値から削除されますが、コマンドリストの最初のコマンドに対してのみ、つまり telnet セッションの開始時にのみ削除されます。

キー	説明
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	telnet 接続を使用してリモートデバイスでコマンドを実行する

Attention:

telnet チェックが非 ASCII 文字と非 UTF8 エンコーディングの値を返す場合、キーの <encoding> パラメータを適切に指定する必要があります。詳細については、**戻り値のエンコード** ページを参照してください。

11 外部からのチェック

概要

外部チェックとは、Zabbix サーバがシェルの実行 **running a shell script** や **
** バイナリを実行することで行うチェックのことです。ただし、Zabbix proxy でホストを監視している場合、**
** 外部チェックは proxy で実行されます。

外部チェックでは、監視対象のホスト上で agent を実行する必要はありません。

アイテムキーの構文は以下の通りです。

script [<parameter1>,<parameter2>,...]

引数	定義
script	シェルスクリプトまたはバイナリの名前

引数	定義
parameter(s)	オプションのコマンドラインパラメータ

もし、スクリプトにパラメータを渡したくない場合は、以下のようにします。

```
script[] or
script
```

Zabbix サーバは外部スクリプトの場所として定義されたディレクトリ (**Zabbix server configuration file**のパラメータ'ExternalScripts')を検索し、コマンドを実行します。
 コマンドは Zabbix サーバの実行ユーザとして実行されるため、アクセス権限や環境変数は必要に応じてラッパースクリプトで処理し、
 コマンドのパーミッションはそのユーザが実行できるようにする必要があります。指定したディレクトリにあるコマンドのみ実行可能です。

Warning:

外部チェックを使いすぎないようにしましょう！
 各スクリプトは Zabbix server からフォークプロセスを起動する必要があるため、多くのスクリプトを実行すると
 Zabbix のパフォーマンスが大幅に低下する可能性があります。

使用例

最初のパラメーターに'-h'を指定してスクリプト **check_oracle.sh** を実行します。
 第 2 パラメータは、ホストプロパティでの選択に応じて、IP アドレスまたは DNS 名に置き換えられます。


```
check_oracle.sh["-h","{HOST.CONN}"].
```

ホストが IP アドレスを使用するように設定されていると仮定して、Zabbix が実行します

```
check_oracle.sh '-h' '192.168.1.4'
```

外部チェック結果

チェックの戻り値は、標準出力と標準エラーです (Zabbix 2.0 以降は末尾の空白を切り詰めた完全な出力が返されます)。

Attention:

テキスト (文字、ログ、テキスト型情報) item は、標準エラー出力の場合、非対応になることはありません。

要求されたスクリプトが見つからない場合、または Zabbix サーバに実行権限がない場合、
 アイテムは未サポートとなり、対応するエラーメッセージが設定されます。タイムアウトが発生した場合、
 アイテムはサポート対象外となり、対応するエラーメッセージが表示され、スクリプト用にフォークされたプロセスは強制終了されます。

12 Trapper item

概要

Trapper item は、問合せの代わりに受信データを受け取ります。

Zabbix に”プッシュ”したいデータに対して有効です。

Trapper item を使用するには、以下が必要です。

- Zabbix に Trapper item が設定されていること
- Zabbix にデータを送信すること

設定

アイテム設定

トラッパーアイテムを設定するには:

- 設定 → ホストにアクセスします。
- ホストの行の アイテムをクリックします。
- アイテムの作成をクリックします。
- item のパラメータをフォームに入力します。

Item	Tags	Preprocessing
* Name	Trapper item	
Type	Zabbix trapper	
* Key	trap	
Type of information	Text	
* History storage period	Do not keep history	Storage period 3600

すべての必須入力項目には赤いアスタリスクが付けられています。

トラッパーアイテムで特定の情報を必要とするフィールドは、次のとおりです。

タイプ	Zabbix トラッパーを選択します
キー	データ送信時にアイテムを認識するためのキーを入力します
データ型	送信されるデータの形式に対応する情報の種類を選択します
許可されたホスト	コンマ区切りの IP アドレスのリスト。オプションで CIDR 表記またはホスト名を使用します。 指定すると、ここにリストされているホストからの着信接続のみが受け入れられます。 IPv6 サポートが有効になっている場合は、'127.0.0.1', ':::127.0.0.1', '::ffff:127.0.0.1' は同等に扱われ、 ':::/0' は任意の IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' を使用して任意の IPv4 アドレスを許可できます。 "IPv4 互換の IPv6 アドレス"(0000::/96 プレフィックス) はサポートされていますが、 RFC4291 によって非推奨になっていることに注意してください。 例: Server=127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, zabbix.domain Zabbix 2.2.0 以降、このフィールドではスペースと ユーザーマクロ を使用できます。 ホストマクロ: {HOST.HOST}、{HOST.NAME}、{HOST.IP}、{HOST.DNS}、{HOST.CONN} は、Zabbix 4.0.2 以降、このフィールドで許可されています。

Note:

アイテムを保存した後、値を送信する前にサーバーがコンフィギュレーションキャッシュの更新から変更を拾うまで、最大 60 秒待つ必要がある場合があります。

データの送信

最も単純なケースでは、`zabbix_sender`ユーティリティを使用して `
 'テスト値'` を送信することができます。

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

値を送信するには、これらのキーを使用します。

- z - Zabbix サーバの IP アドレスを指定します
- p - Zabbix サーバのポート番号を指定します (デフォルトは 10051)
- s - ホストを指定します (ここでは 'visible' な名前ではなく、必ず 'technical' **host name** を使ってください)
- k - 定義した item のキーを指定します
- o - 実際に送信する値を指定します

Attention:

Zabbix の trapper item で、ターゲットホストの対応する item キーの存在を確認する際、`
` item キーで使用されているマクロが展開されないことがあります。

表示

Monitoring → Latest data での結果です。

☰ Latest data

<input type="checkbox"/>	Host	Name ▲	Last check	Last value	Change
<input type="checkbox"/>	New host	Trapper item	2m 27s	test value	

なお、単一の数値が送られた場合、データグラフにはその数値の時点の左と右に横線が表示されます。

13 JMX 監視

概要

JMX 監視は、Java アプリケーションの JMX カウンターを監視するために使用できます。

JMX 監視は Zabbix 2.0 から導入された "Zabbix Java gateway" と呼ばれる Zabbix daemon によって
 Zabbix でネイティブにサポートされています。

ホスト上の特定の JMX カウンタの値を取得するために、Zabbix server は Zabbix **Java gateway** に問合せをし、
 Zabbix は **JMX management API** を使用して
 対象のアプリケーションにリモートで問い合わせをします。

詳細と設定方法は **Zabbix Java gateway** の項を参照してください。

Warning:

Java gateway と監視対象の JMX アプリケーション間の通信をファイアウォールで遮断してはいけません。

Java アプリケーションの JMX リモート監視の有効化

Java アプリケーションは、追加のソフトウェアをインストールする必要はありませんが、リモート JMX 監視をサポートするためには、
 以下に指定するコマンドラインオプションを使用して起動することが必要です。

最低限、ローカルホスト上の単純な Java アプリケーションをセキュリティなしで監視することから始めたい場合は、
 以下のオプションで起動してください。

```
java \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=12345 \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-Dcom.sun.management.jmxremote.ssl=false \  
-Dcom.sun.management.jmxremote.registry.ssl=false \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

これは、Java がローカルホストからの 12345 番ポート上の着信 JMX 接続をリッスンするようにし、
 認証や SSL を要求しないように指示します。

別のインターフェースでの接続を許可する場合は、-Djava.rmi.server.hostname パラメータを
 そのインターフェースの IP に設定します。

もし、セキュリティにもっと厳しくしたいのであれば、他にも多くの Java のオプションがあります。
 例えば、次の例では、より多用途なオプションのセットでアプリケーションを開始し、ローカルホストだけでなく、
 より広いネットワークに開放しています。

```
java \  
-Djava.rmi.server.hostname=192.168.3.14 \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=12345 \  
-Dcom.sun.management.jmxremote.authenticate=true \  
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \  
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \  
-Dcom.sun.management.jmxremote.ssl=true \  
-
```

```
-Dcom.sun.management.jmxremote.registry.ssl=true \  
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \  
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \  
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \  
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \  
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

これらの設定のほとんど（すべてではないにしても）は、`/etc/java-6-openjdk/management/management.properties`（または別の場所にあるそのファイル）に指定することができます。

SSLを使用したい場合は、`startup.sh` スクリプトを変更して、Java gateway に `-Djavax.net.ssl.*` オプションを追加し、
 鍵とトラストストアの場所を認識させる必要があることに注意してください。

[Monitoring and Management Using JMX](#)を参照してください。

Zabbix フロントエンドでの JMX インターフェースと item の設定

Java gateway が動作し、`server` が Java gateway の場所を認識し、リモート JMX 監視をサポートする Java アプリケーションが
 起動したら、Zabbix GUI でインターフェースと item を設定する必要があります。

JMX インターフェースの設定

The screenshot shows the Zabbix GUI configuration page for a host named 'JMX host'. The 'Host' tab is selected. The 'Host name' field is 'JMX host'. The 'Visible name' field is also 'JMX host'. The 'Groups' field shows 'Java (new)' with a close button. Below this is a search box with the text 'type here to search'. The 'Interfaces' table has two rows: 'Agent' with IP address '127.0.0.1' and 'JMX' with IP address '127.0.0.1'. There is an 'Add' link below the table.

Interfaces	Type	IP address	DNS name
Agent		127.0.0.1	
JMX		127.0.0.1	

まず、対象のホストで JMX タイプのインターフェースを作成します。

必須入力項目にはすべて赤いアスタリスクが付けられています。

JMX エージェント項目の追加

関心のある JMX カウンターごとに、そのインターフェースに接続された **JMX agent** item を追加します。

下のスクリーンショットのキーには、次のように書かれています。`jmx["java.lang:type=Memory", "HeapMemoryUsage.used"]`

Item	Tags	Preprocessing
* Name	Used heap memory	
Type	JMX agent	
* Key	jmx["java.lang:type=Memory","HeapMemoryUsage.used"]	
Type of information	Numeric (unsigned)	
* Host interface	127.0.0.1:12345	
* JMX endpoint	service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi	
User name	{JMX_USERNAME}	
Password	{JMX_PASSWORD}	
Units		

必須入力項目には、赤いアスタリスクが表示されています。

JMX item で特定の情報を必要とするフィールドは、以下の通りです。

型	JMX agent をここにセット
Key	The <code>jmx[]</code> item key contains three parameters: object name - the object name of an MBean attribute name - an MBean attribute name with optional composite data field names separated by dots unique short description - a unique description that allows multiple JMX items with the same object name and attribute name on the host (optional) See below for more detail on JMX item keys. Since Zabbix 3.4, you may discover MBeans and MBean attributes using a <code>jmx.discovery[]</code> low-level discovery item.
JMX endpoint	You may specify a custom JMX endpoint. Make sure that JMX endpoint connection parameters match the JMX interface. This can be achieved by using <code>{HOST.*}</code> macros as done in the default JMX endpoint. This field is supported since 3.4.0. <code>{HOST.*}</code> macro and user macros are supported.
User name	Specify the user name, if you have configured authentication on your Java application. User macros are supported.
Password	Specify the password, if you have configured authentication on your Java application. User macros are supported.

"true" または "false" のブール値カウンターを監視したい場合は、情報のタイプを "Numeric (unsigned)" に指定し、
 プリプロセッシングタブで "Boolean to decimal" プリプロセスステップを選択します。
 サーバーはブール値をそれぞれ 1 または 0 として格納します。

JMX item キーの詳細

単純な属性

MBean オブジェクト名は Java アプリケーションで定義する文字列にすぎません。一方、属性名はもっと複雑なことができます。
 属性がプリミティブなデータ型 (整数、文字列など) を返す場合、何も心配する必要はなく、キーは次のようになります。


```
jmx[com.example:type=Hello,weight]
```


この例では、オブジェクト名は"com.example.Type=Hello"で、属性名は"weight"、戻り値の型はおそらく
 "Numeric (float)" であるべきです。

複合データを返す属性

例えば、属性名が"apple"で、"weight"、"color"などのパラメータを表すハッシュを返す場合、属性はより複雑になります。キーは次のようになります。

```
jmx[com.example.Type=Hello,apple.weight]
```

このように、属性名とハッシュ・キーは、ドット記号で区切られます。同じように、属性がネストした複合データを返す場合は、ドットで区切ります。

```
jmx[com.example.Type=Hello,fruits.apple.weight]
```

表形式データを返す属性

表形式データ属性は、1つまたは複数の複合属性から構成されます。属性名パラメータにそのような
 属性が指定された場合、この item 値はその属性の完全な構造を JSON 形式で返します。
 表形式データ属性の中の個々の要素値は、プリプロセスによって取得することができます。

表形式データ属性の例:

Item 値:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

ドットの問題

ここまではいい。しかし、属性名やハッシュキーにドット記号が含まれている場合はどうでしょうか。以下はその例です。

```
jmx[com.example.Type=Hello,all.fruits.apple.weight]
```

これは問題です。属性名が"all"ではなく"all.fruits"であることを Zabbix に伝えるにはどうすればよいですか？名前の一部であるドットと属性名やハッシュキーの区切りであるドットを区別する方法は？

2.0.4 以前の Zabbix Java gateway はこのような状況に対応することができず、ユーザは UNSUPPORTED item が表示されたままになっていました。2.0.4 以降では、名前の一部であるドットをバックスラッシュでエスケープすることで、この問題を解決することができます。

```
jmx[com.example.Type=Hello,all\.fruits.apple.weight]
```

同じように、ハッシュキーにドットが含まれている場合は、それをエスケープします。

```
jmx[com.example.Type=Hello,all\.fruits.apple.total\.weight]
```

その他の問題

属性名中のバックスラッシュ文字はエスケープする必要があります:

```
JMX[com.example.type=Hello,c:\documents]
```

JMX item キーのその他の特殊文字の扱いについては、アイテムキー形式 [section] (/manual/config/items/item/key##parameter_-_quoted_string) を参照してください。

実はこれだけなんです。それでは、JMX の監視をお楽しみください。

非プリミティブなデータ型

Zabbix 4.0.0 以降、**toString()** メソッドをオーバーライドした非プリミティブなデータ型を返すカスタム MBeans を使用することができます。

JBoss EAP 6.4 でのカスタムエンドポイントの使用

カスタムエンドポイントを使用すると、デフォルトの RMI 以外の異なるトランスポートプロトコルで作業することができます。

この可能性を説明するために、例として JBoss EAP 6.4 モニタリングを構成してみましょう。まず、いくつかの前提条件を設定します。

- Zabbix の Java gateway はすでにインストールされています。インストールされていない場合、[documentation](#)に従ってインストールすることができます。
- Zabbix server と Java gateway は /usr/local/ というプレフィックスでインストールされます。
- JBoss は /opt/jboss-eap-6.4/ にインストール済みで、スタンドアロンモードで動作しています。
- これらのコンポーネントはすべて同一ホスト上で動作していると仮定します。
- ファイアウォールと SELinux は無効になっています (または適宜設定されています)

zabbix_server.conf で簡単な設定をしてみましょう。

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

そして、zabbix_java/settings.sh 設定ファイル (または zabbix_java_gateway.conf) に追加します。

```
START_POLLERS=5
```

JBoss がその標準的なマネジメントポートをリッスンしていることを確認します。

```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

では、Zabbix で JMX インターフェース 127.0.0.1:9999 を持つホストを作成してみましょう。

Host	Templates	IPMI	Tags	Macros	Inventory	Encryption	Value mapping
* Host name <input type="text" value="jboss"/>							
Visible name <input type="text" value="jboss"/>							
* Groups Java (new) X <input type="button" value="Select"/>							
<input type="text" value="type here to search"/>							
Interfaces	Type	IP address	DNS name	Connect to	Port		
Agent		<input type="text" value="127.0.0.1"/>	<input type="text"/>	<input type="button" value="IP"/> <input type="button" value="DNS"/>	<input type="text" value="10050"/>		
JMX		<input type="text" value="127.0.0.1"/>	<input type="text"/>	<input type="button" value="IP"/> <input type="button" value="DNS"/>	<input type="text" value="9999"/>		

このバージョンの JBoss は、RMI の代わりに JBoss Remoting プロトコルを使用することがわかっているので、
 それに応じて JMX テンプレートの item の JMX エンドポイントパラメータを大量に更新することができます。

```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```

Mass update

Item Tags Preprocessing

Type Original

JMX endpoint

設定キャッシュを更新してみましょう。

```
$ /usr/local/sbin/zabbix_server -R config_cache_reload
```

なお、最初にエラーが発生する場合があります。

```
3. mc [root@centos7-dev]/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
    at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
    at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
    at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
    ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
    at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
    at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
    at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
    ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885)
as stopped
2017-11-07 13:52:26.167 [main] INFO com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has
tarted
```

“Unsupported protocol: remoting-jmx” は、Java gateway が指定されたプロトコルをどのように動作させるか知らないことを意味します。そのため、以下のような内容で、“need/need_modules.txt” を作成することで、解決することができます。

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio</pre>
```

そして、以下のコマンドを実行します：

```
$ for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname ${i}*.jar -exec cp {} /usr/local
```

このように、Java gateway には jmx-remoting を使用するために必要なモジュールがすべて含まれています。あとは、Java gateway を再起動し、少し待つと、JMX 監視データが Zabbix に届き始めます。Latest data も参照ください。

14 ODBC 監視

概要

ODBC 監視は、Zabbix フロントエンドの Database monitor item タイプに相当します。

ODBC はデータベース管理システム (DBMS) にアクセスするための C 言語のミドルウェア API です。ODBC のコンセプトは Microsoft によって開発され、その後他のプラットフォームにも移植されました。

Zabbix は ODBC でサポートされている全てのデータベースに問い合わせをすることができます。そのため、Zabbix はデータベースに直接接続せず、ODBC インターフェースと ODBC で設定されたドライバを利用します。この機能により、例えば、特定のデータベースキューや利用統計の確認など、複数の目的で異なるデータベースをより効率的に監視することが可能です。Zabbix は最も一般的に利用されているオープンソースの ODBC API 実装の 1 つである unixODBC をサポートしています。

Attention:

ODBC チェックについては、known issues も参照してください。

unixODBC のインストール

unixODBC をインストールする推奨される方法は、Linux オペレーティングシステムのデフォルトパッケージリポジトリを
 使用することです。最も人気のある Linux ディストリビューションでは、unixODBC はデフォルトで
 パッケージリポジトリに含まれています。
 もし利用できない場合は、unixODBC のホームページで入手することができます。
 <http://www.unixodbc.org/download.html>.

RedHat/Fedora ベースのシステムで yum パッケージマネージャを使用して unixODBC をインストールします。

```
shell> yum -y install unixODBC unixODBC-devel
```

SUSE ベースのシステムで zypper パッケージマネージャを使用して unixODBC をインストールします。

```
# zypper in unixODBC-devel
```

Note:
unixODBC-devel パッケージは、unixODBC をサポートする Zabbix をコンパイルするために必要です。

unixODBC ドライバのインストール

監視するデータベースには、unixODBC データベースドライバをインストールする必要があります。
 unixODBC には、サポートされているデータベースとドライバのリストがあります。
 <http://www.unixodbc.org/drivers.html>
 Linux ディストリビューションによっては、データベースドライバがパッケージリポジトリに含まれている場合があります。

RedHat/Fedora ベースのシステムで、yum パッケージマネージャを使用して MySQL データベースドライバをインストールする場合:

```
shell> yum install mysql-connector-odbc
```

SUSE ベースのシステムで、zypper パッケージマネージャを使用して MySQL データベースドライバをインストールする場合:

```
zypper で MyODBC-unixODBC
```

unixODBC の設定

ODBC の設定は、**odbcinst.ini** と **odbc.ini** ファイルを編集することによって行われます。
 設定ファイルの場所を確認するには、次のように入力します。

```
shell> odbcinst -j
```

odbcinst.ini は、インストールされている ODBC データベースドライバーの一覧を表示するために使用されます。

```
[mysql]
Description = ODBC for MySQL
Driver       = /usr/lib/libmyodbc5.so
```

パラメータの詳細:

属性	説明
mysql	データベースドライバ名
Description	データベースドライバーの説明
Driver	データベースドライバーライブラリの場所

odbc.ini は、データソースの定義に使用されます:

```
[test]
Description = MySQL test database
Driver       = mysql
Server      = 127.0.0.1
User        = root
Password    =
Port        = 3306
Database    = zabbix
```

パラメータの詳細:

属性	説明
test	データソース名 (DSN)
Description	データソースの説明。
Driver	データベースドライバー名 - odbcinst.ini で指定されているとおり
Server	データベースサーバーの IP/DNS
User	接続用のデータベースユーザー

属性	説明
Password	データベースユーザーパスワード
Port	データベース接続ポート
Database	データベース名

ODBC 接続が正常に動作しているかどうかを確認するために、データベースへの接続をテストする必要があります。
これは **isql** ユーティリティ (unixODBC パッケージに含まれています) で行うことができます。

```
shell> isql test
+-----+
| Connected! |
| |
| sql-statement |
| help [tablename] |
| quit |
| |
+-----+
SQL>
```

ODBC サポート付き Zabbix のコンパイル

ODBC サポートを有効にするには、以下のフラグを付けて Zabbix をコンパイルする必要があります。

```
--with-unixodbc[=ARG] use odbc driver against unixODBC package
```

Note:

Zabbix のインストールについて詳しくは、[source code](#)を参照してください。

Zabbix フロントエンドでの item 設定

データベース監視itemを設定します。

The screenshot shows the Zabbix web interface for configuring a new item. The 'Item' tab is selected. The configuration fields are as follows:

- Name:** MySQL host count
- Type:** Database monitor (dropdown menu)
- Key:** db.odbc.select[mysql-simple-check,test]
- Type of information:** Numeric (unsigned) (dropdown menu)
- User name:** zabbix
- Password:** (empty field)
- SQL query:** select count(*) from hosts

必須入力項目には、赤いアスタリスクが付けられています。

特にデータベース監視の項目について、入力する必要があります:

タイプ

ここでデータベースモニタを選択

キー

サポートされている 2 つのアイテムキーのいずれかを入力します:

db.odbc.select[< 一意の短い説明 >,<dsn>,< 接続文字列 >] - このアイテムは SQL クエリ結果の最初の行の最初の列、1 つの値を返すように設計されています。クエリが複数の列を返す場合、最初の列のみが読み取られます。クエリが複数の行を返す場合、最初の行のみが読み取られます。

db.odbc.get[< 一意の短い説明 >,<dsn>,< 接続文字列 >] - この項目は JSON 形式で複数の行/列を返すことができます。したがって、1 回のシステムコールですべてのデータを収集するマスターアイテムとして使用できますが、JSONPath 前処理を依存アイテムで使用して個々の値を抽出できます。詳細については、ローレベルディスカバリーで使用され、返される形式の例を参照してください。この項目は Zabbix 4.4 以降でサポートされています。

一意の説明は、トリガーなどで項目を識別するのに役立ちます。

dsn と connection string はオプションのパラメータですが、少なくとも 1 つが存在する必要があります。データソース名 (DSN) と接続文字列の両方が定義されている場合、DSN は無視されます。

データソース名を使用する場合は、odbc.ini で指定されているとおりに設定する必要があります。

接続文字列には、ドライバー固有の引数が含まれる場合があります。

例 (MySQL ODBC driver 5 の接続):

```
=> db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1"]
```

データベースのユーザー名を入力してください

odbc.ini でユーザーが指定されている場合、このパラメータはオプションです。

接続文字列が使用され、ユーザー名フィールドが空でない場合、UID=<user> として接続文字列に追加されます

データベースユーザーのパスワードを入力してください

パスワードが odbc.ini で指定されている場合、このパラメータはオプションです。

接続文字列が使用され、Password フィールドが空でない場合、接続文字列に PWD=<password> として追加されます。

SQL クエリを入力します。

db.odbc.select [] アイテムを使用すると、クエリは 1 つの値のみを返す必要があることに注意してください。

ここで正しく選択できるよう、クエリによって返される情報の種類を把握しておく必要があります。データ型が間違っていると、アイテムはサポート対象外になります。

ユーザー名

パスワード

SQL クエリ

データ型

重要な注意事項

- server または proxy 設定で odbc poller プロセスが起動されていない場合、データベース監視項目はサポートされなくなります。ODBC ポラーを起動するには、Zabbix server 設定ファイルで StartODBCPollers パラメータを設定するか、proxy で実行するチェックの場合は Zabbix proxy (/manual/appendix/config/zabbix_proxy) 設定ファイルで StartODBCPollers パラメータを設定します。
- Zabbix はクエリの実行時間を制限していません。妥当な時間で実行できるクエリを選択することはユーザにゆだねられています。
- Zabbix server の Timeout パラメータの値は ODBC ログインタイムアウトとして使用されます (ODBC ドライバによっては、ログインタイムアウト設定が無視される場合があることに注意してください)。
- SQL コマンドは、select ... を使用した他のクエリと同様に結果セットを返す必要があります。問い合わせの構文は、それを処理する RDBMS に依存します。ストレージプロシージャへのリクエストの構文は call キーワードで開始する必要があります。

エラーメッセージ

ODBC のエラーメッセージは、詳細な情報を提供するためにフィールドに分かれて構成されています。
たとえば、以下のよう

なものです。

Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executing

	Native error code	Native error message
	SQLState	
Zabbix message	ODBC return code	

エラーメッセージの長さは 2048 バイトに制限されているため、メッセージは切り捨てられる可能性があることに注意してください。

 複数の ODBC 診断レコードがある場合、Zabbix はそれらを (| で区切って) 制限された長さになるように連結しようと試みます。

1 MySQL の推奨 UnixODBC 設定

インストール

*** Red Hat Enterprise Linux**:

```
# yum install mysql-connector-odbc
```

***Debian/Ubuntu**:

対応するプラットフォームのデータベースドライバをダウンロードするには、[MySQL documentation](#)を参照し、
 対応するプラットフォーム用の必要なデータベースドライバをダウンロードしてください。

いくつかの追加情報について、[unixODBC のインストール](#)を参照してください。

設定

ODBC の設定は、**odbcinst.ini** と **odbc.ini** ファイルを編集することによって行われます。
 これらの設定ファイルは、/etc フォルダにあります。**odbcinst.ini** ファイルが存在しない場合があります、
 この場合は手動で作成する必要があります。

odbcinst.ini

```
[mysql]
Description = General ODBC for MySQL
Driver       = /usr/lib64/libmyodbc5.so
Setup       = /usr/lib64/libodbcmyS.so
FileUsage   = 1
```

以下の **odbc.ini** の設定パラメータの例をご検討ください。

- IP 接続の例:

```
[TEST_MYSQL]
Description = MySQL database 1
Driver      = mysql
Port        = 3306
Server      = 127.0.0.1
```

- IP 接続とクレデンシャルを使用した場合の例です。デフォルトで Zabbix データベースを使用します。

```
[TEST_MYSQL_FILLED_CRED]
Description = MySQL database 2
Driver      = mysql
User        = root
Port        = 3306
Password    = zabbix
Database    = zabbix
Server      = 127.0.0.1
```

ソケットを使用した接続とクレデンシャルを使用した例です。デフォルトでは、Zabbix データベースを使用します。

```
[TEST_MYSQL_FILLED_CRED_SOCKET]
Description = MySQL database 3
Driver      = mysql
User        = root
Password    = zabbix
Socket      = /var/run/mysqld/mysqld.sock
Database    = zabbix
```

その他の設定パラメータオプションは、[MySQL official documentation](#)のウェブページを参照してください。

2 PostgreSQL の推奨 UnixODBC 設定

インストール

- **Red Hat Enterprise Linux:**

```
# yum install postgresql-odbc
```

- **Debian/Ubuntu:**

対応するプラットフォームのデータベースドライバは、[PostgreSQL documentation](#)を参照し、
 対応するプラットフォーム用のデータベースドライバをダウンロードしてください。

追加情報については、[unixODBC のインストール](#)を参照してください。

設定

ODBC の設定は、**odbcinst.ini** と **odbc.ini** ファイルを編集することによって行われます。
 これらの構成ファイルは、/etc フォルダにあります。**odbcinst.ini** ファイルがない場合があります、
 この場合は手動で作成する必要があります。

以下の例を検討してください。
 odbcinst.ini

```
[postgresql]
Description = General ODBC for PostgreSQL
Driver       = /usr/lib64/libodbcpsql.so
Setup        = /usr/lib64/libodbcpsqlS.so
FileUsage    = 1
# Since 1.6 if the driver manager was built with thread support you may add another entry to each driver e
# This entry alters the default thread serialization level.
Threading    = 2
```

odbc.ini

```
[TEST_PSQL]
Description = PostgreSQL database 1
Driver      = postgresql
#CommLog    = /tmp/sql.log
Username    = zbx_test
Password    = zabbix
# Name of Server. IP or DNS
Servername  = 127.0.0.1
# Database name
Database    = zabbix
# Postmaster listening port
Port        = 5432
# Database is read only
# Whether the datasource will allow updates.
ReadOnly    = No
# PostgreSQL backend protocol
# Note that when using SSL connections this setting is ignored.
# 7.4+: Use the 7.4(V3) protocol. This is only compatible with 7.4 and higher backends.
Protocol    = 7.4+
# Includes the OID in SQLColumns
ShowOidColumn = No
# Fakes a unique index on OID
FakeOidIndex = No
# Row Versioning
# Allows applications to detect whether data has been modified by other users
# while you are attempting to update a row.
# It also speeds the update process since every single column does not need to be specified in the where c
RowVersioning = No
# Show SystemTables
# The driver will treat system tables as regular tables in SQLTables. This is good for Access so you can s
ShowSystemTables = No
# If true, the driver automatically uses declare cursor/fetch to handle SELECT statements and keeps 100 rc
```



```
Fetch = Yes
# Booleans as Char
# Booleans are mapped to SQL_CHAR, otherwise to SQL_BIT.
BooleansAsChar = Yes
# SSL mode
SSLmode = Yes
# Send to backend on connection
ConnSettings =
```

3 Oracle の推奨 UnixODBC 設定

インストール方法

インストール方法については、[Oracle documentation](#) を参照してください。

追加情報については、以下を参照してください。[Installing unixODBC](#) を参照してください。

4 MSSQL の推奨 UnixODBC 設定

インストール方法

- **** Red Hat Enterprise Linux****:

```
# yum -y install freetds unixODBC
```

- **Debian/Ubuntu**:

[FreeTDS user guide](#) を参照して、対応するプラットフォームに必要なデータベースドライバーをダウンロードしてください。

その他の情報については、[unixODBC のインストール](#) を参照してください。

Configuration

ODBC の設定は、**odbcinst.ini** と **odbc.ini** ファイルを編集することによって行われます。
これらの構成ファイルは、/etc フォルダにあります。**odbcinst.ini** ファイルがない場合があり、
この場合は手動で作成する必要があります。

以下の例を検討してください。

odbcinst.ini

```
$ vi /etc/odbcinst.ini
[FreeTDS]
Driver = /usr/lib64/libtdsodbc.so.0
```

odbc.ini

```
$ vi /etc/odbc.ini
[sqll1]
Driver = FreeTDS
Server = <SQL server 1 IP>
PORT = 1433
TDS_Version = 8.0
```

15 依存アイテム

概要

1つのアイテムで複数のメトリクスを同時に収集する場合や、関連するメトリクスを同時に収集する方が理にかなっている場合などがある。例えば：

- 各コアの CPU 使用率
- ネットワークトラフィックの受信/送信/総量

メトリクスの一括収集と複数の関連項目での同時利用を可能にするため、Zabbix は依存アイテムをサポートしています。依存アイテムは親アイテムに依存し、1つのクエリで同時にデータを収集します。マスターアイテムの新しい値は自動的に依存するアイテムの値に反映されます。依存アイテムは、マスターアイテムと異なる更新間隔を設定することはできません。

Zabbix の保存前処理オプションを使用すると、マスターアイテムから依存アイテムに必要な部分を抽出することができます。

保存前処理は、Zabbix3.4 で追加された preprocessing manager プロセスと、保存前処理のステップを実行するワーカーによって管理されます。異なるデータ収集装置からのすべての値（保存前処理のあり/なし）は、ヒストリーキャッシュに追加される前に preprocessing manager を通過します。データ収集処理（ポーラ、トラッパーなど）と保存前処理の間にはソケットベースの IPC 通信が使用されます。

Zabbix サーバーまたは Zabbix プロキシ（プロキシ経由でホストを監視している場合）は、保存前処理の手順を実行し、依存アイテムを処理しています。

依存アイテムであっても、あらゆるタイプのアイテムをマスターアイテムとして設定することができます。
 依存アイテムのレベルを追加することで、既存の依存アイテムの値からより小さな部分を抽出することができます。

制限事項

- 同一ホスト（テンプレート）への依存関係のみが許可される。
- アイテムのプロトタイプは、同じホストの別のアイテムのプロトタイプまたは通常のアイテムに依存することができます。
- 1つのマスターアイテムに対する依存アイテムの最大数は 29999 に制限されます。（依存レベルの数には関係ありません）
- 依存レベルは最大 3 つまで。
- テンプレートにあるマスターアイテムを持つホストの依存アイテムは、XML にエクスポートされません。

アイテムの設定

依存アイテムは、データのためにマスターアイテムに依存します。そのため、マスターアイテムを最初に設定する（存在させる）必要があります。

- 設定 → ホストにアクセスします。
- ホストの行にある、アイテムをクリックします。
- アイテムの作成をクリックします。
- アイテムのパラメータをフォームに入力します。

Item	Tags	Preprocessing
* Name	Apache server status	
Type	Zabbix agent	
* Key	web.page.get[127.0.0.1/server-status]	
Type of information	Text	
* Host interface	127.0.0.1:1050	
* Update interval	30s	

必須入力項目には、赤いアスタリスクが表示されています。

追加をクリックして、マスターアイテムを保存します。

次に、依存アイテムを設定することができます。

Item	Tags	Preprocessing
* Name	Apache server uptime	
Type	Dependent item	
* Key	apache.server.uptime	
Type of information	Text	
* Master item	Apache: Apache server status	

必須入力項目には、赤いアスタリスクが表示されています。

依存アイテムに対して特定の情報を必要とするフィールドは、以下のとおりです。

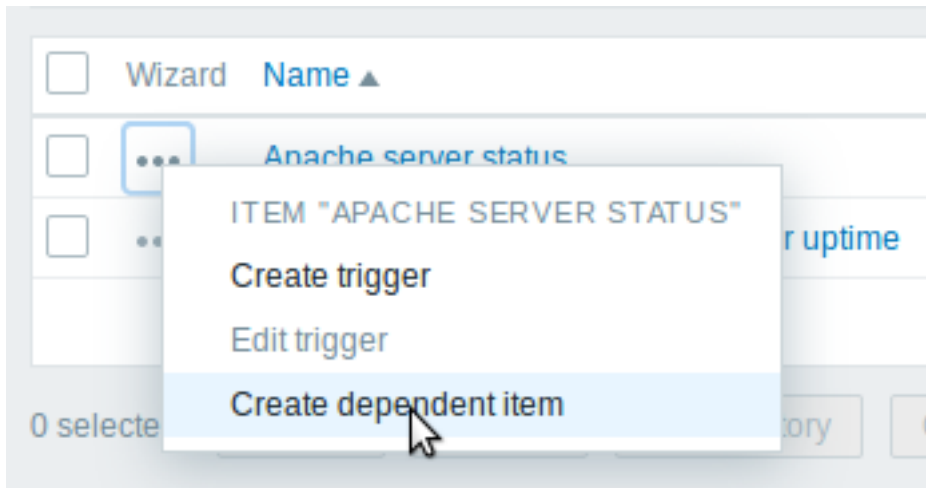
タイプ	ここでは依存アイテムを選択します。
キー	アイテムを認識するために使用するキーを入力します。
マスターアイテム	マスターアイテムを選択します。マスターアイテムの値は、依存アイテムの値を設定するために使用されます。
データ型	保存するデータの型に合わせてデータ型を選択します。

アイテムの値の保存前処理の機能を使用してマスターアイテムの値の必要な部分のみ抽出できます。

保存前処理を行わない場合、依存アイテムの値はマスターアイテムの値とまったく同じになります。

追加をクリックして、依存アイテムを保存します。

依存アイテムをすばやく作成するには、アイテムリストでウィザードを使用するのが近道です：



表示

アイテムリストでは、依存アイテムはマスターアイテム名をプレフィックスとして表示されます。

<input type="checkbox"/>	Name ▲	Triggers	Key
<input type="checkbox"/>	... Apache server status		web.page.get[127.0.0.1/server-status]
<input type="checkbox"/>	... Apache server status: Apache server uptime		apache.server.uptime

マスターアイテムが削除された場合、そのアイテムに依存する依存アイテムもすべて削除されます。

16 HTTP agent

概要

この item タイプでは、HTTP/HTTPS プロトコルを使用したデータポーリングが可能です。Zabbix sender または Zabbix sender プロトコルを利用したトラッピングも可能です。

HTTP item のチェックは Zabbix server で実行されます。ただし、Zabbix proxy でホストを監視している場合、HTTP item チェックは proxy で実行されます。

HTTP item チェックは、監視対象のホスト上で agent を実行する必要はありません。

HTTP agent は HTTP と HTTPS の両方をサポートしています。Zabbix はオプションでリダイレクトを追跡します (下記の Follow redirects オプションを参照)。
 リダイレクトの最大数は 10 にハードコードされています (cURL オプション CURLOPT_MAXREDIRS を使用)。

HTTPS プロトコルを使用する場合については、[known issues](#) も参照してください。

Attention:

Zabbix server / proxy は、cURL(libcurl) をサポートした状態で初期設定されている必要があります。

設定

HTTP item を設定するには:

- Configuration → Hosts にアクセスします。
- ホストの行にある Items をクリックします。
- Create item をクリックします。
- item のパラメータをフォームに入力します。

The screenshot shows the 'Create item' form in Zabbix. The 'Name' field is 'HTTP agent item'. The 'Type' is 'HTTP agent'. The 'Key' is 'http_value_search'. The 'Type of information' is 'Numeric (unsigned)'. The 'URL' is 'http://localhost:8200/api/values_search'. The 'Request type' is 'POST'. The 'Request body type' is 'JSON data'. The 'Request body' is a JSON object:

```
{  "query": {    "host": "  "trend": {    "items": [      "trend"  ]  }  }
```

. The 'Required status codes' is '200'. The 'Follow redirects' checkbox is checked. The 'Retrieve mode' is 'Body'. The 'Convert to JSON' checkbox is unchecked. The 'HTTP proxy' is 'zabbix@[user:password@[proxy.example.com:port]]'. The 'HTTP authentication' is 'None'. The 'Host interface' is '127.0.0.1:1050'. The 'Update interval' is '1m'. The 'Custom intervals' section has 'Type' set to 'Flexible' and 'Interval' set to '5m'. The 'History storage period' is '90d'. The 'Trend storage period' is '365d'. The 'Value mapping' field is empty. The 'Populate host inventory field' is 'None'. The 'Enabled' checkbox is checked. At the bottom, there are 'Add', 'Test', and 'Cancel' buttons.

必須入力項目には、赤いアスタリスクが表示されています。

HTTP item で特定の情報を必要とする項目は、以下の通りです。

Type

Key

Select **HTTP agent** here.

Enter a unique item key.

URL	<p>URL to connect to and retrieve data. For example: https://www.example.com http://www.example.com/download</p> <p>Domain names can be specified in Unicode characters. They are automatically punycode-converted to ASCII when executing the HTTP check.</p> <p>The Parse button can be used to separate optional query fields (like ?name=Admin&password=mypassword) from the URL, moving the attributes and values into Query fields for automatic URL-encoding.</p> <p>Limited to 2048 characters.</p> <p>Supported macros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, user macros, low-level discovery macros.</p> <p>This sets the CURLOPT_URL cURL option.</p>
Query fields	<p>Variables for the URL (see above).</p> <p>Specified as attribute and value pairs.</p> <p>Values are URL-encoded automatically. Values from macros are resolved and then URL-encoded automatically.</p> <p>Supported macros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, user macros, low-level discovery macros.</p> <p>This sets the CURLOPT_URL cURL option.</p>
Request type Timeout	<p>Select request method type: GET, POST, PUT or HEAD</p> <p>Zabbix will not spend more than the set amount of time on processing the URL (1-60 seconds). Actually this parameter defines the maximum time for making a connection to the URL and maximum time for performing an HTTP request. Therefore, Zabbix will not spend more than 2 x Timeout seconds on one check.</p> <p>Time suffixes are supported, e.g. 30s, 1m.</p> <p>Supported macros: user macros, low-level discovery macros.</p> <p>This sets the CURLOPT_TIMEOUT cURL option.</p>
Request body type	<p>Select the request body type:</p> <p>Raw data - custom HTTP request body, macros are substituted but no encoding is performed</p> <p>JSON data - HTTP request body in JSON format. Macros can be used as string, number, true and false; macros used as strings must be enclosed in double quotes. Values from macros are resolved and then escaped automatically. If "Content-Type" is not specified in headers then it will default to "Content-Type: application/json"</p> <p>XML data - HTTP request body in XML format. Macros can be used as a text node, attribute or CDATA section. Values from macros are resolved and then escaped automatically in a text node and attribute. If "Content-Type" is not specified in headers then it will default to "Content-Type: application/xml"</p>
Request body	<p>Note that selecting XML data requires libxml2.</p> <p>Enter the request body.</p> <p>Supported macros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, user macros, low-level discovery macros.</p>
Headers	<p>Custom HTTP headers that will be sent when performing a request.</p> <p>Specified as attribute and value pairs.</p> <p>Supported macros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, user macros, low-level discovery macros.</p> <p>This sets the CURLOPT_HTTPHEADER cURL option.</p>

Required status codes	<p>List of expected HTTP status codes. If Zabbix gets a code which is not in the list, the item will become unsupported. If empty, no check is performed.</p> <p>For example: 200,201,210-299</p> <p>Supported macros in the list: user macros, low-level discovery macros.</p>
Follow redirects	<p>This uses the CURINFO_RESPONSE_CODE cURL option.</p> <p>Mark the checkbox to follow HTTP redirects.</p> <p>This sets the CURLOPT_FOLLOWLOCATION cURL option.</p>
Retrieve mode	<p>Select the part of response that must be retrieved:</p> <p>Body - body only</p> <p>Headers - headers only</p>
Convert to JSON	<p>Body and headers - body and headers</p> <p>Headers are saved as attribute and value pairs under the "header" key.</p> <p>If 'Content-Type: application/json' is encountered then body is saved as an object, otherwise it is stored as string, for example:</p>
HTTP proxy	<pre data-bbox="810 714 1393 1025"> { "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> } </pre> <p>You can specify an HTTP proxy to use, using the format <code>[protocol://][username[:password]@]proxy.example.com[:port]</code>. The optional <code>protocol://</code> prefix may be used to specify alternative proxy protocols (e.g. <code>https</code>, <code>socks4</code>, <code>socks5</code>; see documentation; the protocol prefix support was added in cURL 7.21.7). With no protocol specified, the proxy will be treated as an HTTP proxy. If you specify the wrong protocol, the connection will fail and the item will become unsupported.</p> <p>By default, 1080 port will be used.</p> <p>If specified, the proxy will overwrite proxy related environment variables like <code>http_proxy</code>, <code>HTTPS_PROXY</code>. If not specified, the proxy will not overwrite proxy-related environment variables. The entered value is passed on "as is", no sanity checking takes place.</p> <p>Note that only simple authentication is supported with HTTP proxy.</p> <p>Supported macros: <code>{HOST.IP}</code>, <code>{HOST.CONN}</code>, <code>{HOST.DNS}</code>, <code>{HOST.HOST}</code>, <code>{HOST.NAME}</code>, <code>{ITEM.ID}</code>, <code>{ITEM.KEY}</code>, <code>{ITEM.KEY.ORIG}</code>, user macros, low-level discovery macros.</p> <p>This sets the CURLOPT_PROXY cURL option.</p> <p>Authentication type:</p>
HTTP authentication	<p>None - no authentication used.</p> <p>Basic - basic authentication is used.</p> <p>NTLM - NTLM (Windows NT LAN Manager) authentication is used.</p> <p>Kerberos - Kerberos authentication is used. See also: Configuring Kerberos with Zabbix.</p> <p>Digest - Digest authentication is used.</p> <p>Selecting an authentication method will provide two additional fields for entering a user name and password, where user macros and low-level discovery macros are supported.</p> <p>This sets the CURLOPT_HTTPAUTH cURL option.</p>

SSL verify peer	<p>Mark the checkbox to verify the SSL certificate of the web server. The server certificate will be automatically taken from system-wide certificate authority (CA) location. You can override the location of CA files using Zabbix server or proxy configuration parameter <code>SSLCALocation</code>.</p> <p>This sets the <code>CURLOPT_SSL_VERIFYPEER</code> cURL option.</p>
SSL verify host	<p>Mark the checkbox to verify that the Common Name field or the Subject Alternate Name field of the web server certificate matches.</p> <p>This sets the <code>CURLOPT_SSL_VERIFYHOST</code> cURL option.</p>
SSL certificate file	<p>Name of the SSL certificate file used for client authentication. The certificate file must be in PEM¹ format. If the certificate file contains also the private key, leave the SSL key file field empty. If the key is encrypted, specify the password in SSL key password field. The directory containing this file is specified by Zabbix server or proxy configuration parameter <code>SSLCertLocation</code>.</p> <p>Supported macros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, user macros, low-level discovery macros.</p> <p>This sets the <code>CURLOPT_SSLCERT</code> cURL option.</p>
SSL key file	<p>Name of the SSL private key file used for client authentication. The private key file must be in PEM¹ format. The directory containing this file is specified by Zabbix server or proxy configuration parameter <code>SSLKeyLocation</code>.</p> <p>Supported macros: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, user macros, low-level discovery macros.</p> <p>This sets the <code>CURLOPT_SSLKEY</code> cURL option.</p>
SSL key password	<p>SSL private key file password.</p> <p>Supported macros: user macros, low-level discovery macros.</p> <p>This sets the <code>CURLOPT_KEYPASSWD</code> cURL option.</p>
Enable trapping	<p>With this checkbox marked, the item will also function as trapper item and will accept data sent to this item by Zabbix sender or using Zabbix sender protocol.</p>
Allowed hosts	<p>Visible only if Enable trapping checkbox is marked.</p> <p>List of comma delimited IP addresses, optionally in CIDR notation, or hostnames.</p> <p>If specified, incoming connections will be accepted only from the hosts listed here.</p> <p>If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally and ':::0' will allow any IPv4 or IPv6 address.</p> <p>'0.0.0.0/0' can be used to allow any IPv4 address.</p> <p>Note, that "IPv4-compatible IPv6 addresses" (0000::/96 prefix) are supported but deprecated by RFC4291.</p> <p>Example: Server=127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, zabbix.domain</p> <p>Spaces and user macros are allowed in this field.</p> <p>Host macros: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} are allowed in this field.</p>

Note:

*HTTP proxy フィールドを空にした場合、proxy 関連の環境変数を設定することで HTTP proxy を使用する別の方法があります。

HTTP の場合 - Zabbix server ユーザの環境変数 `http_proxy` を設定します。例: `http_proxy=http://proxy_ip:proxy_port`。

HTTPS の場合 - 環境変数 `HTTPS_PROXY` を設定します。例: `HTTPS_PROXY=http://proxy_ip:proxy_port` 詳細は、シェルコマンド `# man curl` を実行することで確認できます。

Attention:

[1] Zabbix は証明書と秘密鍵のファイルを PEM フォーマットでのみサポートしています。
 証明書と秘密鍵のデータが PKCS#12 形式のファイル (通常、拡張子は " *.p12" または "/*.pfx") の場合、以下のコマンドで PEM ファイルを
 生成することが可能です。

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes -out ssl-cert.key
```

例

例 1

Elasticsearch などのサービスからデータを取得するために、シンプルな GET リクエストを送信します。

- URL を指定して GET item を作成します。localhost:9200/?pretty で GET item を作成します。
- レスポンスに注目してください。

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "kH4CYqh5QfgeTsjh2F9zg",
  "version" : {
    "number" : "6.1.3",
    "build_hash" : "af51318",
    "build_date" : "2018-01-26T18:22:55.523Z",
    "build_snapshot" : false,
    "lucene_version" : "7.1.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You know, for search"
}
```

- ここで、JSONPath のプリプロセスを利用してバージョン番号を抽出します。\$.version.number

例 2

Elasticsearch などのサービスからデータを取得するために、シンプルな POST リクエストを送信します。

- URL を指定して POST アイテムを作成する。http://localhost:9200/_search?scroll=10s
- プロセッサの負荷 (各コア 1 分平均) を取得するために、以下の POST ボディを設定します。

```
{
  "query": {
    "bool": {
      "must": [{
        "match": {
          "itemid": 28275
        }
      }],
      "filter": [{
        "range": {
          "clock": {
            "gt": 1517565836,
            "lte": 1517566137
          }
        }
      }
    ]
  }
}
```

- Received:

```
{
  "_scroll_id": "DnF1ZXJ5VGh1bkZldGN0BQAAAAAAAAAaF1lRM1ZBWS1UU1pxTmdEeGVwQjRBTfEAAAAAAAAAJRZZUTJWQVktVFN",
  "took": 18,
  "timed_out": false,
  "_shards": {
```



```

    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1.0,
    "hits": [{
      "_index": "dbl",
      "_type": "values",
      "_id": "dqX9VWEBV6sEKSMYk6sw",
      "_score": 1.0,
      "_source": {
        "itemid": 28275,
        "value": "0.138750",
        "clock": 1517566136,
        "ns": 25388713,
        "ttl": 604800
      }
    }
  ]
}

```

- ここで、JSONPath のプリプロセスを利用して item 値を抽出します。`$.hits.hits[0]._source.value`

例 3

`apiinfo.version` を使用して、Zabbix API が活着しているかどうかをチェックします。

- Item 設定

Item Tags Preprocessing

* Name

Type

* Key

Type of information

* URL

Query fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Add](#)

Request type

* Timeout

Request body type

Request body

```
{
  "jsonrpc": "2.0",
  "method": "apiinfo.version",
  "params": [],
  "id": 1
}
```

Headers

Name	Value
<input type="text" value="Content-Type"/>	<input type="text" value="application/json-rpc"/>

[Add](#)

Required status codes

Follow redirects

Retrieve mode

JSON データで POST メソッドを使用し、リクエストヘッダを設定し、ヘッダのみを返すように要求することに注意してください。

- 項目値を正規表現で前処理し、HTTP コードを取得します:

Item Tags Preprocessing 1

Preprocessing steps

Name	Parameters
1: <input type="text" value="Regular expression"/>	<input type="text" value="HTTPV1.1 ([0-9]+)"/>

[Add](#)

- Latest data の結果をチェックします:

Filter

Host groups

Hosts
type here to search

Application

Name

Show items without data

Show details

<input type="checkbox"/> Host	Name ▲	Last check	Last value	Change
▼ <u>Zabbix server</u> - other - (1 Item)				
<input type="checkbox"/>	Check Zabbix API version	2018-05-16 23:50:34	OK (200)	Graph

例 4

Openweathermap の公開サービスに接続して天気情報を取得する。

- 1 つの JSON でデータを一括収集するためのマスター item を設定する。

Item

Tags

Preprocessing

* Name

Type

* Key

Type of information

* URL

Query fields

Name	⇒	Value
<input type="text" value="units"/>	⇒	<input type="text" value="metric"/>
<input type="text" value="lat"/>	⇒	<input type="text" value="{ \$LAT }"/>
<input type="text" value="lon"/>	⇒	<input type="text" value="{ \$LON }"/>
<input type="text" value="APPID"/>	⇒	<input type="text" value="{ \$WEATHER_APIKEY }"/>
<input type="text" value="lang"/>	⇒	<input type="text" value="{ \$WEATHER_LANG }"/>

[Add](#)

Request type

* Timeout

Request body type

Request body

クエリフィールドのマクロの使い方に注意してください。
 記入方法は、[Openweathermap API](#)を参照してください。

HTTP agent のレスポンスで返される JSON のサンプル:

```
{
  "body": {
    "coord": {
      "lon": 40.01,
```

```

    "lat": 56.11
  },
  "weather": [{
    "id": 801,
    "main": "Clouds",
    "description": "few clouds",
    "icon": "02n"
  }],
  "base": "stations",
  "main": {
    "temp": 15.14,
    "pressure": 1012.6,
    "humidity": 66,
    "temp_min": 15.14,
    "temp_max": 15.14,
    "sea_level": 1030.91,
    "grnd_level": 1012.6
  },
  "wind": {
    "speed": 1.86,
    "deg": 246.001
  },
  "clouds": {
    "all": 20
  },
  "dt": 1526509427,
  "sys": {
    "message": 0.0035,
    "country": "RU",
    "sunrise": 1526432608,
    "sunset": 1526491828
  },
  "id": 487837,
  "name": "Stavrovo",
  "cod": 200
}
}

```

次のタスクは、JSON からデータを抽出する従属 item を設定することです。

- 湿度に関するサンプルの従属 item を設定します。

Item	Tags	Preprocessing
		<p>* Name <input type="text" value="Humidity"/></p> <p>Type <input type="text" value="Dependent item"/></p> <p>* Key <input type="text" value="humidity"/></p> <p>Type of information <input type="text" value="Numeric (float)"/></p> <p>* Master item <input type="text" value="Apache: Get weather"/></p> <p>Units <input type="text"/></p>

その他、「気温」などの気象指標も同様に追加されます。

- JSONPath を使った従属 item 値のプリプロセスの例:

Item Tags **Preprocessing 1**

Preprocessing steps

Name	Parameters
1: JSONPath	\$.body.main.humidity

[Add](#)

- Latest data で気象データの結果を確認します:

Host	Name	Inter...	History	Trends	Type	Last check	Last value
weather	Weather (8 Items)						
<input type="checkbox"/>	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{'body': {'coord': {'lon...
<input type="checkbox"/>	Get weather HTTP response code get_weather.http_code	7d	0		Depende...	2018-05-17 01:23:45	OK (200)
<input type="checkbox"/>	Humidity humidity	90d	365d		Depende...	2018-05-17 01:23:45	66 %
<input type="checkbox"/>	Temperature temp	90d	365d		Depende...	2018-05-17 01:23:45	15.14 C
<input type="checkbox"/>	Weather weather	90d			Depende...	2018-05-17 01:23:45	Clouds
<input type="checkbox"/>	Weather condition id weather.condition.id	7d	0		Depende...	2018-05-17 01:23:45	801
<input type="checkbox"/>	Weather description weather.description	90d			Depende...	2018-05-17 01:23:45	few clouds
<input type="checkbox"/>	Wind speed wind.speed	90d	365d		Depende...	2018-05-17 01:23:45	1.86 m/s

例 5

Nginx のステータスページに接続し、そのメトリクスを一括で取得する。

- [official guide](#)に従って Nginx を設定します。
- 一括データ収集のためのマスター item を設定する。

Item Tags **Preprocessing**

* Name

Type

* Key

Type of information

* URL

Query fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Add](#)

Request type

* Timeout

Request body type Raw data JSON data XML data

Nginx stub のステータス出力のサンプル:

```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

次のタスクは、データを抽出する従属 item を設定することです。

- サンプルとして、1 秒あたりのリクエストの従属 item を設定します。

The screenshot shows the configuration for a new item named "Client requests per second". The configuration is as follows:

- Name:** Client requests per second
- Type:** Dependent item
- Key:** nginx_requests_rps
- Type of information:** Numeric (unsigned)
- Master item:** Nginx by HTTP: Nginx: Get stub status page

- 正規表現による従属 item 値のプリプロセッシング例 `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:

The screenshot shows the preprocessing steps configuration for the item. It includes two steps:

Preprocessing steps	Name	Parameters
1:	Regular expression	requests\s+([0-9]+) ([0-9]+) ([0-9]+)
2:	Change per second	

There is also an "Add" button below the steps.

- スタブモジュールの全結果は Latest data でご確認ください。

Host	Name ▲	Last check	Last value
nginx	Nginx (8 Items)		
<input type="checkbox"/>	Accepted client connections	2018-05-18 17:54:53	568
<input type="checkbox"/>	Active connections	2018-05-18 17:54:53	1
<input type="checkbox"/>	Client requests per second	2018-05-18 17:54:53	0 rps
<input checked="" type="checkbox"/>	Get Nginx stub status	2018-05-18 17:54:53	HTTP/1.1 200 OK Se...
<input type="checkbox"/>	Handled connections per second	2018-05-18 17:54:53	0
<input type="checkbox"/>	Reading	2018-05-18 17:54:53	0
<input type="checkbox"/>	Waiting	2018-05-18 17:54:53	0
<input type="checkbox"/>	Writing	2018-05-18 17:54:53	1

17 Prometheus チェック

概要

Zabbix は Prometheus のラインフォーマットで公開されているメトリクスを問合せすることができます。

Prometheus のデータ収集を開始するには、2つのステップが必要です。

- 適切なデータエンドポイントを指す **HTTP master item** (例: `https://<prometheus host>/metrics`)
- Prometheus のプリプロセッシングオプションを使用して、マスター item によって収集されたメトリクスから必要なデータを問合せする従属 item

Prometheus のデータ前処理オプションは 2 つあります。

- Prometheus pattern - Prometheus のデータを問い合わせるために通常の item で使用されます。

- Prometheus to JSON - 通常の item とローレベルディスカバリーに使用されます。この場合、照会された Prometheus データは JSON フォーマットで返されます。

バルクプロセッシング

依存 item はバルクプロセッシングに対応しています。
 キャッシュとインデックスを有効にするには、Prometheus pattern のプリプロセスを最初のプリプロセスステップにする必要があります。
 Prometheus pattern が最初のプリプロセスステップである場合、パースされた Prometheus データは、
 Prometheus pattern プリプロセスステップの最初の <label>==<value> 条件によってキャッシュされ、インデックスが作成されます。
 このキャッシュは、このバッチ内の他の依存 item を処理する際に再利用されます。
 最適なパフォーマンスを得るために、最初のラベルは最も異なる値を持つものにすべきです。

最初のステップの前に他のプリプロセスがある場合は、マスター item か、従属 item のマスター item として使用される
 新しい従属 item のどちらかに移動させる必要があります。

設定

HTTP マスター item が設定されていることを前提に、Prometheus のプリプロセスステップを使用する
 dependent item を作成する必要があります。

- 設定フォームに一般的な従属項目パラメータを入力する
- Preprocessing タブに移動します
- Prometheus の Preprocessing オプション (Prometheus pattern or Prometheus to JSON) を選択します。

以下のパラメータは、Prometheus pattern Preprocessing オプションに固有のもので:

パラメータ	説明	例
Pattern	<p>To define the required data pattern you may use a query language that is similar to Prometheus query language (see comparison table), e.g.:</p> <p><metric name> - select by metric name {__name__="<metric name>} - select by metric name {__name__=~<regex>} - select by metric name matching a regular expression {<label name>="<label value>","...} - select by label name {<label name>=~<regex>","...} - select by label name matching a regular expression {__name__=~".*"}==<value> - select by metric value</p> <p>Or a combination of the above: <metric name> {<label1 name>="<label1 value>",<label2 name>=~<regex>","...}==<value></p> <p>Label value can be any sequence of UTF-8 characters, but the backslash, double-quote and line feed characters have to be escaped as \\, \" and \n respectively; other characters shall not be escaped.</p>	<pre>wmi_os_physical_memory_free_bytes cpu_usage_system{cpu="cpu-total"} cpu_usage_system{cpu=~".*"} cpu_usage_system{cpu="cpu-total",host=~".*"} wmi_service_state{name="dhcp"}==1 wmi_os_timezone{timezone=~".*"}==1</pre>

パラメータ	説明	例
Result processing	Specify whether to return the value, the label or apply the appropriate function (if the pattern matches several lines and the result needs to be aggregated): value - return metric value (error if multiple lines matched) label - return value of the label specified in the Label field (error if multiple metrics are matched) sum - return the sum of values min - return the minimum value max - return the maximum value avg - return the average value count - return the count of values This field is only available for the Prometheus pattern option.	See also examples of using parameters below.
Output	Define label name (optional). In this case the value corresponding to the label name is returned. This field is only available for the Prometheus pattern option, if 'Label' is selected in the Result processing field.	

パラメータ使用の例

1. 最も一般的な使用例は、値を返すことです。/var/db からの値を返すには、次のようにします。

```
node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096
```

以下のパラメータを使用します:

- Pattern - node_disk_usage_bytes{path="/var/db"}
 - Result processing - select 'value'
2. また、すべての node_disk_usage_bytes パラメータの 平均値に興味があるかもしれません:
 - Pattern - node_disk_usage_bytes
 - Result processing - select 'avg'
 3. Prometheus は数値データのみをサポートしていますが、関連するテキストの説明も返すことができるワークアラウンドを
 使用することが一般的です。これは、フィルタとラベルを指定することで実現できます。つまり、'color' ラベルの値を返すには、
 以下のようにします。
 > elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1
 > elasticsearch_cluster_health_status{cluster="elasticsearch",color="red"} 0
 > elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0

以下のパラメータを使用します:

- Pattern - elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1
- Result processing - select 'label'
- Label - specify 'color'

フィルター (数値'1'に基づく) は適切な行にマッチし、ラベルはヘルステータスの説明を返す (現在は'緑'だが、'赤'や'黄'の可能性もある)。

Prometheus から JSON へ

Prometheus からのデータローレベルディスクカバリーに使用することができます。この場合、JSON 形式のデータが必要で、
 Prometheus to JSON プリプロセッシングオプションはまさにそのデータを返します。

詳しくは、[Discovery using Prometheus data](#)を参照してください。

クエリ言語の比較

以下の表は、PromQL と Zabbix Prometheus プリプロセッシングクエリ言語の相違点と類似点をリストアップしたものです。

PromQL instant vector selector	Zabbix Prometheus preprocessing
<p>差 異</p> <p>Query Prometheus server target</p> <p>Return instant vector</p> <p>Label <code>#, !=, =~, !~</code></p> <p>matching operators</p> <p>Regular expression used in label or metric name matching</p> <p>Comparison operators</p> <p>Similarities</p> <p>Selecting <code><metric name></code> or <code>{_name_="<metric name>"}</code> by metric name that equals string</p> <p>Selecting <code>{_name_ =~ "<regex>"}</code> by metric name that matches regular expression</p>	<p>Plain text in Prometheus exposition format</p> <p>Metric or label value (Prometheus pattern)</p> <p>Array of metrics for single value in JSON (Prometheus to JSON)</p> <p><code>=, !=, =~, !~</code></p> <p>PCRE</p> <p>Only <code>==</code> (equal) is supported for value filtering</p> <p><code><metric name></code> or <code>{_name_="<metric name>"}</code></p> <p><code>{_name_ =~ "<regex>"}</code></p>

PromQL instant vector selector	Zabbix Prometheus preprocessing
Select by label name=>"<label value>","...}	{<label name>=>"<label value>","...}
Select by label name=~"<regex>","...}	{<label name>=~"<regex>","...}
Select by time__=~".*"} == <value>	{__name__=~".*"} == <value>

18 スクリプト item

概要

スクリプト item は、ユーザー定義の JavaScript コードを実行してデータを収集するために使用されます。
 HTTP/HTTPS でデータを取得する機能を持つ、ユーザー定義の JavaScript コードを実行することによって、
 データを収集することができます。スクリプトの他に、オプションでパラメータ（名前と値のペア）のリストとタイムアウトを
 指定することができます。

この item タイプは、複数のステップや複雑なロジックを必要とするデータ収集シナリオに有用であると思われます。
 例えば、スクリプト item は、HTTP コールを行い、最初のステップで受信したデータを何らかの方法で処理し、
 変換された値を 2 番目の HTTP コールに渡すように設定することが可能です。

スクリプトアイテムは Zabbix server または proxy ポーラによって処理されます。

設定

item configuration form の Type フィールドで Script を選択し、必要事項を入力します。

Item	Tags	Preprocessing												
* Name	<input type="text" value="Data collector script"/>													
Type	<input type="text" value="Script"/>													
* Key	<input type="text" value="script.data.collector"/>													
Type of information	<input type="text" value="Text"/>													
Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td><input type="text" value="host"/></td> <td><input type="text" value="{HOST.CONN}"/></td> <td>Remove</td> </tr> <tr> <td><input type="text" value="endpoint"/></td> <td><input type="text" value="{SENDPOINT}"/></td> <td>Remove</td> </tr> <tr> <td colspan="3">Add</td> </tr> </tbody> </table>		Name	Value	Action	<input type="text" value="host"/>	<input type="text" value="{HOST.CONN}"/>	Remove	<input type="text" value="endpoint"/>	<input type="text" value="{SENDPOINT}"/>	Remove	Add		
Name	Value	Action												
<input type="text" value="host"/>	<input type="text" value="{HOST.CONN}"/>	Remove												
<input type="text" value="endpoint"/>	<input type="text" value="{SENDPOINT}"/>	Remove												
Add														
* Script	<input type="text" value="var request = new HttpRequest();..."/>													
* Timeout	<input type="text" value="3s"/>													

すべての必須入力項目には、赤いアスタリスクが付けられています。

スクリプトアイテムで特定の情報を必要とするフィールドは次のとおりです。

フィールド	説明
Key	Enter a unique key that will be used to identify the item.
Parameters	Specify the variables to be passed to the script as the attribute and value pairs. Built-in macros {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} and user macros are supported.
Script	Enter JavaScript code in the block that appears when clicking in the parameter field (or on the view/edit button next to it). This code must provide the logic for returning the metric value. The code has access to all parameters, it may perform HTTP GET, POST, PUT and DELETE requests and has control over HTTP headers and request body. See also: Additional JavaScript objects , JavaScript Guide .
Timeout	JavaScript execution timeout (1-60s, default 3s); exceeding it will return error. Time suffixes are supported, e.g. 30s, 1m. Depending on the script it might take longer for the timeout to trigger.

例

シンプルなデータ収集

https://www.example.com/release_notes の内容を収集します。

- タイプ "Script" の item を作成します。
- Script 欄に、次のコードを入力します。

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

パラメータによるデータ収集

パラメータ値として {HOST.CONN} マクロを使用し、マクロを展開してレスポンスを取得します。

- タイプ "Script" の item を作成します。
- 以下のパラメータを作成します: Name: host
Value: {HOST.CONN}

- script の欄に、以下のコードを入力します:

```
var request = new HttpRequest();
return request.post("https://postman-echo.com/post", JSON.parse(value));
```

複数の HTTP リクエスト

https://www.example.com と https://www.example.com/release_notes の両方の内容を収集します。

- タイプ "Script" の item を作成します。
- Script の欄に、以下のコードを入力します:

```
var request = new HttpRequest();  
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

ロギング

Zabbix server ログに "Log test" エントリを追加し、アイテム値 "1" を返信します。

- タイプ "Script" の item を作成します。
- Script フィールドに、以下のコードを入力します。

```
Zabbix.log(3, 'Log test');  
return 1;
```

4 ヒストリとトレンド

概要

Zabbix では、収集したデータを保存する方法として、ヒストリとトレンドの 2 つがあります。

ヒストリは収集した各値を保持するのに対し、トレンドは 1 時間単位で平均化された情報を保持するため、リソースをあまり消費しません。

ヒストリの保持

ヒストリを何日間残すかを設定することができます。

- アイテムの [設定画面](#) で設定できます。
- アイテムの一括更新時
- データの保存期間の [設定](#) を行う時

古いデータは、housekeeper プロセスによって削除されます。

一般的なアドバイスとして、ヒストリはできるだけ少ない日数で保存し、多くのヒストリの値でデータベースに負荷をかけないようにすることが重要です。

例えば、ヒストリを 14 日間、トレンドを 5 年間保存することができます。

ヒストリとトレンドのデータでどれくらいの容量が必要かは、[データベースのサイジング](#)を参照してください。

グラフは古いデータを表示するためにトレンドの値を使用するので、短いヒストリの保存期間であっても、グラフで古いデータを確認することができます。

Attention:

ヒストリの保存期間を '0' に設定すると、アイテムが更新されるのは依存アイテムとインベントリのみを更新します。
 トリガー条件式の評価はヒストリの値にのみ基づいて行われるため、トリガー関数は評価されません。

Note:

ヒストリを保存する別の方法として、ローダブルモジュールの [ヒストリエクスポート](#) 機能を使うことを検討してください。

トレンドの保持

トレンドは、数値データタイプの 1 時間あたりの最小値、最大値、平均値、および合計値を保存するために組み込まれたヒストリデータ削減メカニズムです。

トレンドは何日間保存されるかを設定することができます。

- アイテムの [設定画面](#) で設定できます。
- アイテムの一括更新時
- データの保存期間を設定するとき

トレンドは通常、ヒストリよりもはるかに長い期間保持することができます。古いデータは housekeeper プロセスによって削除されます。

Zabbix サーバーは、データが流入するにつれて、実行時にトレンドデータをトレンドキャッシュに蓄積します。サーバーは、以下のような場合、すべてのアイテムの前の **1** 時間のトレンドをデータベース (Web インターフェースが検索可能な場所) にフラッシュします:

- サーバーがそのアイテムの最初の現在時刻の値を受信したとき
- 現在の時間が残り 5 分以下で、まだそのアイテムの現在の時間の値がないとき
- サーバーが停止するとき

グラフでトレンドを見るには、少なくとも次の時間の始まりまで (アイテムが頻繁に更新される場合)、最大でも次の時間の終わりまで (アイテムがほとんど更新されない場合)、つまり最大 2 時間待つ必要があります。

サーバーがトレンドキャッシュをフラッシュし、データベースにその時間のトレンドが既にある場合
 (例えば、サーバーが時間の途中で再起動した場合)、サーバーは単純な挿入の代わりに更新ステートメントを使用する必要があります。
 したがって、大規模なインストールで再起動が必要な場合は、トレンドデータの重複を避けるために、ある時間の終わりにサーバーを停止し、次の時間の初めに開始することが望ましいです。

履歴テーブルはトレンドの生成には一切関与しません。

Attention:

トレンドの保存期間が '0' に設定されている場合、Zabbix サーバーはトレンドの計算や保存を全く行いません。

Note:

トレンドは元の値と同じデータ型で計算され、保存されます。その結果、符号なしデータ型の値の平均値の計算は丸められ、値の間隔が狭いほど結果の精度は低くなります。例えば、項目が 0 と 1 の値を持つ場合、平均値は 0.5 ではなく、0 になります。

また、サーバーを再起動すると、符号なしデータ型の平均値の計算の精度が落ちることがあります。

5 ユーザパラメータ

概要

Zabbix にあらかじめ定義されていないエージェントチェックを実行したい場合があります。
 このような場合、ユーザパラメータが役に立ちます。

必要なデータを取得するコマンドを作成し、そのデータをユーザパラメータとして **agent configuration file** ('UserParameter') という設定パラメータを作成します。

ユーザパラメータは次のような構文になります。

```
UserParameter=<key>,<command>
```

このように、ユーザパラメータにはキーも含まれます。キーは、item を設定する際に必要となります。
 参照しやすい任意のキーを入力します (ホスト内で一意でなければなりません)。

agent を再起動するか、agent **runtime control** オプションを使用して、新しいパラメータを取得します。

```
zabbix_agentd -R userparameter_reload
```

次に、**configuring an item**の際に、実行したいユーザパラメータからコマンドを参照するためのキーを入力します。

ユーザパラメータとは、Zabbix agent が実行するコマンドのことです。item のプリプロセスステップの前に
 最大 512KB のデータを返すことができます。ただし、最終的にデータベースに保存できるテキスト値は、
 MySQL では 64KB に制限されていることに注意してください。
 (他のデータベースについては、**table**を参照してください)。

UNIX オペレーティングシステムでは、コマンドラインインタプリタとして **/bin/sh** が使用されます。
 ユーザーパラメータはエージェントチェックのタイムアウトに従います。タイムアウトに達すると
 フォークされたユーザーパラメータ・プロセスが終了します。

こちらをご覧ください。

- [ユーザーパラメータを利用するためのStep-by-step tutorial](#)
- [Command execution](#)

単純なユーザーパラメータの例

単純なコマンド:

```
UserParameter=ping,echo 1
```

agent は、'ping' キーを持つ item に対して常に'1'を返します。

より複雑な例:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

agent は、MySQL サーバが生きている場合は'1'を、そうでない場合は'0'を返します。

フレキシブルユーザーパラメータ

フレキシブルユーザーパラメータは、キーでパラメータを受け取ります。
このように、フレキシブルユーザーパラメータは、複数の item を作成するためのベースとなることができます。

フレキシブルユーザーパラメータの構文は以下の通りです。

```
UserParameter=key[*],command
```

パラメータ	説明
Key	Unique item key. The [*] defines that this key accepts parameters within the brackets.
Command	Parameters are given when configuring the item. Command to be executed to evaluate value of the key. For flexible user parameters only: You may use positional references \$1...\$9 in the command to refer to the respective parameter in the item key. Zabbix parses the parameters enclosed in [] of the item key and substitutes \$1,...,\$9 in the command accordingly. \$0 will be substituted by the original command (prior to expansion of \$0,...,\$9) to be run. Positional references are interpreted regardless of whether they are enclosed between double (") or single (') quotes. To use positional references unaltered, specify a double dollar sign - for example, awk '{print \$\$2}'. In this case \$\$2 will actually turn into \$2 when executing the command.

Attention:

Zabbix agent では、柔軟なユーザパラメータの場合のみ、\$ 記号を含む位置参照が検索され、置換されます。
シンプルなユーザパラメータでは、このような参照処理はスキップされるため、\$ 記号の引用は必要ありません。

Attention:

デフォルトでは、ユーザパラメータに特定のシンボルを使用することはできません。全リストは [UnsafeUserParameters](#) ドキュメントを参照してください。

例 1

とてもシンプルなもの:

```
UserParameter=ping[*],echo $1
```

ping[something] というフォーマットを持つ監視項目は無制限に定義することができる。

- ping[0] - 常に'0'を返す
- ping[aaa] - 常に'aaa'を返す

例 2

もっとセンスを磨こう!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

このパラメータは、MySQL データベースの可用性を監視するために使用することができます。
ユーザー名とパスワードを渡すことができます:

```
mysql.ping[zabbix,our_password]
```

例 3

正規表現にマッチする行はファイル中に何行あるか?

```
UserParameter=wc[*],grep -c "$2" $1
```

このパラメータは、ファイル内の行数を計算するために使用することができます。

```
wc [/etc/passwd,root]
wc [/etc/services,zabbix]
```

コマンドの結果

コマンドの戻り値は、標準出力と標準エラー出力です。

Attention:

標準エラー出力の場合、テキスト（文字、ログ、テキスト型情報）item は未サポートになりません。

テキスト（文字、ログ、テキスト型情報）を返すユーザーパラメータは、空白を返すことができます。
無効な結果を返した場合、その item はサポートされなくなります。

Zabbix agent の機能拡張

このチュートリアルでは、**user parameter**を使用して Zabbix agent の機能を拡張する方法をステップバイステップで説明します。

ステップ 1

必要なパラメータを取得するためのスクリプトまたはコマンドラインを記述します。

例えば、MySQL サーバで実行されたクエリの総数を取得するために、以下のようなコマンドを記述することができます。

```
mysqladmin -uroot status | cut -f4 -d":". | cut -f1 -d "S"
```

このコマンドを実行すると、SQL クエリの総数が返されます。

ステップ 2

zabbix_agentd.conf にコマンドを追加します。

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":". | cut -f1 -d "S"
```

mysql.questions は一意な識別子です。それは任意の有効なキーであることができます。例えば、queries などです。

このパラメータをテストするには、Zabbix agent を「-t」フラグで使用します (root 権限で実行する場合)。ただし、デーモンとして起動した場合、agent のパーミッションが異なる可能性があります。

```
zabbix_agentd -t mysql.questions
```

ステップ 3

以下のコマンドを実行して、設定ファイルからユーザパラメータを再ロードします。

```
zabbix_agentd -R userparameter_reload
```

実行時制御コマンドの代わりに、agent を再起動することもできます。

zabbix_get ユーティリティを使用して、パラメータをテストします。

ステップ 4

監視するホストに Key=mysql.questions の新しい item を追加します。
item のタイプは、Zabbix agent または Zabbix agent (active) のいずれかにする必要があります。

Zabbix server で返される値のタイプが正しく設定されている必要があることに注意してください。
そうでない場合、Zabbix は値を受け付けません。

6 ローダブルモジュール

概要

ローダブルモジュールは、Zabbix の機能を拡張するためのパフォーマンス重視のオプションです。

Zabbix の機能を拡張する方法として、すでに以下のようなものがあります。

- **user parameters** (agent メトリクス)
- **external checks**(agent レス監視)
- **system.run [] Zabbix agent item.**

これらは非常にうまく動作しますが、1つ大きな欠点があります。それは `fork()` です。
 Zabbix はユーザメトリクスを処理するたびに新しいプロセスを `fork` する必要があり、これはパフォーマンスにとって良いことではありません。通常は大きな問題ではありませんが、組み込みシステムの監視、多数の監視パラメータ
 複雑なロジックや長い起動時間を持つ重いスクリプトを持つ場合、深刻な問題となる可能性があります。

ロードダブルモジュールのサポートは、パフォーマンスを犠牲にすることなく、Zabbix エージェント、server、proxy を
 拡張する方法を提供します。

ロードダブルモジュールとは、基本的に Zabbix デーモンが使用する共有ライブラリで、起動時にロードされます。
 ライブラリには特定の関数が含まれている必要があり、Zabbix プロセスはファイルが本当にロードして動作するモジュールで
 あることを検出することができます。

ロード可能なモジュールには多くの利点があります。優れたパフォーマンスや任意のロジックを実装できることも非常に重要ですが、
 おそらく最も重要な利点は、Zabbix モジュールを開発、利用、共有できることです。これはトラブルのないメンテナンスに貢献し、
 Zabbix のコアコードベースから独立して、より簡単に新しい機能を提供するのに役立ちます。

モジュールのライセンスとバイナリ形式での配布は GPL ライセンスに準拠します (モジュールはランタイムに Zabbix とリンクし、
 Zabbix ヘッダーを使用しています。) バイナリ互換性は Zabbix によって保証されていません。

モジュール API の安定性は Zabbix LTS(Long Term Support) [release](#) の 1 サイクルにおいて保証されます。
 Zabbix API の安定性は保証されていません (技術的にはモジュールから Zabbix 内部関数を呼び出すことは可能ですが、そのようなモジュールが
 動作する保証はありません)。

モジュール API

共有ライブラリを Zabbix モジュールとして扱うには、いくつかの関数を実装し、エクスポートする必要があります。
 現在、Zabbix モジュール API には 6 つの関数があり、そのうち 1 つだけが必須で、他の 5 つはオプションです。

必須インターフェース

必須の関数は `zbx_module_api_version()` だけです。

```
int zbx_module_api_version(void);
```

この関数は、このモジュールが実装している API バージョンを返し、モジュールをロードするためには、このバージョンが
 Zabbix がサポートするモジュール API バージョンと一致する必要があります。Zabbix がサポートするモジュール API のバージョンは、
 `ZBX_MODULE_API_VERSION` です。したがって、この関数はこの定数を返す必要があります。この目的で使用されていた古い定数
 `ZBX_MODULE_API_VERSION_ONE` は、ソースの互換性を保つために `ZBX_MODULE_API_VERSION` と同じに定義されていますが、
 この使用は推奨されません。

1 オプションのインターフェース

オプションのインターフェースは、以下の通りです。`zbx_module_init()`、`zbx_module_item_list()`、`zbx_module_item_timeout()`、`zbx_module_history_write_cbs()`、`zbx_module_uninit()`

```
int zbx_module_init(void);
```

この関数は、モジュールに必要な初期化を行います (もしあれば)。成功した場合は、`ZBX_MODULE_OK` を返します。
 失敗した場合、`ZBX_MODULE_FAIL` を返します。後者の場合、Zabbix は起動しません。


```
ZBX_METRIC *zbx_module_item_list(void);
```

この関数は、モジュールがサポートする item のリストを返す必要があります。各項目は `ZBX_METRIC` 構造体で定義されます。リストの最後には、“key” フィールドが `NULL` の `ZBX_METRIC` 構造体を返します。

```
void zbx_module_item_timeout(int timeout);
```

モジュールが `zbx_module_item_list()` をエクスポートしている場合、この関数は Zabbix によって使用されます。
 モジュールが実装する item チェックが従うべきタイムアウト設定を Zabbix 設定ファイルで指定するために使用されます。
 ここで、“timeout” パラメータは秒単位です。


```
ZBX_HISTORY_WRITE_CBS zbx_module_history_write_cbs(void);
```

この関数は、Zabbix サーバが異なるデータ型の history をエクスポートする際に使用するコールバック関数を返します。
 コールバック関数は `ZBX_HISTORY_WRITE_CBS` 構造体のフィールドとして提供され、モジュールが特定のタイプの history に
 興味がない場合、フィールドは `NULL` にすることが可能です。


```
int zbx_module_uninit(void);
```

この関数は、割り当てられたリソースの解放、ファイルディスクリプタのクローズなど、必要な初期化を行います。

すべての関数は、Zabbix 起動時にモジュールがロードされると一度だけ呼び出されます。`zbx_module_uninit()` は Zabbix のシャットダウン時にモジュールがアンロードされると一度だけ呼び出されます。

item の定義

各 item は、ZBX_METRIC 構造体で定義されます。

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int         (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

ここで、**key** は item のキー (例えば "dummy.random") で、**flags** は CF_HAVEPARAMS または 0 (item がパラメータを受け入れるかどうか依存)、**function** は項目を実装する C 関数です。item (例: zbx_module_dummy_random), **test_param** は、Zabbix agent を「-p」フラグで起動したときに使用されるパラメータリストです (例: 「1,1000」、NULL も可能)

定義例は以下のとおりです。このようになります。

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

項目を実装する各関数は、2 つのポインタパラメータを受け取る必要があります。パラメータを受け取る必要があります。1 つ目は AGENT_REQUEST 型、2 つ目はタイプ AGENT_RESULT です。

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

これらの関数は項目の値が正常に取得された場合は SYSINFO_RET_OK を返します。それ以外の場合は SYSINFO_RET_FAIL を返します。AGENT_REQUEST から情報を取得する方法と、AGENT_RESULT に情報を設定する方法の詳細については、以下のサンプル "dummy" モジュールを参照してください。

2 history のエクスポートコールバックを提供する

Attention:

モジュール経由の履歴エクスポートは Zabbix 4.0.0 以降、Zabbix プロキシではサポートされなくなりました。

モジュールは history データをエクスポートする関数をタイプ別に指定することができます。数値 (float)、Numeric (unsigned)、Character、Text、Log

```
typedef struct
{
    void (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CB;
```

それぞれ、"history_num" 要素の "history" 配列を引数として受け取る必要があります。エクスポートされる history のデータ型に応じて、"history" は以下の構造体の配列となります。

```
typedef struct
{
    zbx_uint64_t    itemid;
    int             clock;
    int             ns;
    double          value;
```

```

}
ZBX_HISTORY_FLOAT;

typedef struct
{
    zbx_uint64_t    itemid;
    int            clock;
    int            ns;
    zbx_uint64_t    value;
}
ZBX_HISTORY_INTEGER;

typedef struct
{
    zbx_uint64_t    itemid;
    int            clock;
    int            ns;
    const char     *value;
}
ZBX_HISTORY_STRING;

typedef struct
{
    zbx_uint64_t    itemid;
    int            clock;
    int            ns;
    const char     *value;
}
ZBX_HISTORY_TEXT;

typedef struct
{
    zbx_uint64_t    itemid;
    int            clock;
    int            ns;
    const char     *value;
    const char     *source;
    int            timestamp;
    int            logeventid;
    int            severity;
}
ZBX_HISTORY_LOG;

```

コールバックは Zabbix server の history 同期プロセスで使用されます。
 Zabbix データベースにデータが書き込まれ、値キャッシュに保存された後、history 同期処理の終了時に使用されます。

Attention:

history エクスポートモジュールに内部エラーが発生した場合、回復するまで監視全体をブロックするのではなく、
代わりにデータを破棄して Zabbix サーバの実行を継続させるようにモジュールを記述することを推奨します。

モジュールのビルド

現在、モジュールは Zabbix のソースツリー内でビルドされることになっています。
なぜなら、モジュール API は Zabbix ヘッダで定義されているいくつかのデータ構造に依存しているからです。

ロード可能なモジュールにとって最も重要なヘッダーは、これらのデータ構造を定義している **include/module.h** です。
include/module.h が正しく動作するために必要な他のシステムヘッダは、**stdlib.h** と **stdint.h** です。

この情報を頭に入れば、モジュールをビルドするための準備はすべて整ったことになります。
モジュールは、**stdlib.h**、**stdint.h**、**module.h** が含まれている必要があり、ビルドスクリプトは、
これらのファイルがインクルードパスにあることを確認する必要があります。
詳細については、以下の”ダミー”モジュールの例を参照してください。

このヘッダには、ロギングとデバッグに使用できる **zabbix_log()** 関数が定義されています

設定パラメータ

Zabbix agent、server、proxy は、モジュールを処理するために 2 つの **parameters** をサポートしています。

- LoadModulePath - ロード可能なモジュールの場所へのフルパスです。

- LoadModule - 起動時にロードするモジュール。モジュールは、LoadModulePath で指定されたディレクトリに配置されているか、
 パスがモジュール名より前にある必要があります。先行するパスが絶対パス ('/' で始まる) の場合、LoadModulePath は無視されます。
 複数の LoadModule パラメータを含めることができます。

例えば、Zabbix エージェントを拡張する場合、以下のように追加することができます。

```
LoadModulePath=/usr/local/lib/zabbix/agent/  
LoadModule=mariadb.so  
LoadModule=apache.so  
LoadModule=kernel.so  
LoadModule=/usr/local/lib/zabbix/dummy.so
```

エージェントの起動時に、/usr/local/lib/zabbix/agent ディレクトリから mariadb.so, apache.so, kernel.so モジュールをロードし、
 /usr/local/lib/zabbix から dummy.so をロードします。モジュールが見つからない場合、不正なパーミッションの場合、
 共有ライブラリが Zabbix モジュールでない場合、失敗します。

フロントエンドの設定

ロード可能なモジュールは Zabbix agent、server、proxy でサポートされているため、Zabbix フロントエンドの item タイプはモジュールがロードされる場所に依存します。モジュールが agent にロードされている場合、item のタイプは "Zabbix agent" または "Zabbix agent(active)" になります。モジュールが server または proxy にロードされている場合、item のタイプは "シンプルチェック" になります。

Zabbix モジュールによる history のエクスポートには、フロントエンドの設定は必要ありません。モジュールが server に正常にロードされ、少なくとも 1 つの非 NULL コールバック関数を返す **zbx_module_history_take_cbs()** 関数を提供する場合、history のエクスポートが自動的に有効化されます。

ダミーモジュール

Zabbix には C 言語で書かれたサンプルモジュールが含まれています。このモジュールは src/modules/dummy にあります：

```
alex@alex:~trunk/src/modules/dummy$ ls -l  
-rw-rw-r-- 1 alex alex 9019 Apr 24 17:54 dummy.c  
-rw-rw-r-- 1 alex alex 67 Apr 24 17:54 Makefile  
-rw-rw-r-- 1 alex alex 245 Apr 24 17:54 README
```

このモジュールはドキュメントが充実しており、独自のモジュールのテンプレートとして使用することができます。

上記のように Zabbix ソースツリーのルートで ./configure を実行した後、 **make** を実行し、 **dummy.so** をビルドしてください。

```
/*  
** Zabbix  
** Copyright (C) 2001-2020 Zabbix SIA  
**  
** This program is free software; you can redistribute it and/or modify  
** it under the terms of the GNU General Public License as published by  
** the Free Software Foundation; either version 2 of the License, or  
** (at your option) any later version.  
**  
** This program is distributed in the hope that it will be useful,  
** but WITHOUT ANY WARRANTY; without even the implied warranty of  
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
** GNU General Public License for more details.  
**  
** You should have received a copy of the GNU General Public License  
** along with this program; if not, write to the Free Software  
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.  
**/  
  
####include <stdlib.h>  
####include <string.h>  
####include <time.h>  
####include <stdint.h>  
  
####include "module.h"
```

```

/* the variable keeps timeout setting for item processing */
static int item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/* KEY          FLAG          FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,          dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS, dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS, dummy_random, "1,1000"},
    {NULL}
};

/*****
*
* Function: zbx_module_api_version
*
* Purpose: returns version number of the module interface
*
* Return value: ZBX_MODULE_API_VERSION - version of module.h module is
*             compiled with, in order to load module successfully Zabbix
*             MUST be compiled with the same version of this header file
*
*****/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/*****
*
* Function: zbx_module_item_timeout
*
* Purpose: set timeout value for processing of items
*
* Parameters: timeout - timeout in seconds, 0 - no timeout set
*
*****/
void zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/*****
*
* Function: zbx_module_item_list
*
* Purpose: returns list of item keys supported by the module
*
* Return value: list of item keys
*
*****/
ZBX_METRIC *zbx_module_item_list(void)
{
    return keys;
}

```

```

static int dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char *param;

    if (1 != request+nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters.));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/*****
 *
 * Function: dummy_random
 *
 * Purpose: a main entry point for processing of an item
 *
 * Parameters: request - structure that contains item key and parameters
 *             request+key - item key without parameters
 *             request+nparam - number of parameters
 *             request+params[N-1] - pointers to item key parameters
 *             request+types[N-1] - item key parameters types:
 *                 REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty)
 *                 REQUEST_PARAMETER_TYPE_ARRAY (array)
 *                 REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string)
 *
 *             result - structure that will contain result
 *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked
 *              as not supported by zabbix
 *              SYSINFO_RET_OK - success
 *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth
 *          parameter starting from 0 (first parameter). Make sure it exists
 *          by checking value of request+nparam.
 *          In the same manner get_rparam_type(request, N-1) can be used to
 *          get a parameter type.
 *
 *****/
static int dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char *param1, *param2;
    int from, to;

    if (2 != request+nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters.));
    }
}

```

```

    return SYSINFO_RET_FAIL;
}

param1 = get_rparam(request, 0);
param2 = get_rparam(request, 1);

/* there is no strict validation of parameters and types for simplicity sake */
from = atoi(param1);
to = atoi(param2);

if (from > to)
{
    SET_MSG_RESULT(result, strdup("Invalid range specified.));
    return SYSINFO_RET_FAIL;
}

SET_UI64_RESULT(result, from + rand() % (to - from + 1));

return SYSINFO_RET_OK;
}

/*****
 *
 * Function: zbx_module_init
 *
 * Purpose: the function is called on agent startup
 *          It should be used to call any initialization routines
 *
 * Return value: ZBX_MODULE_OK - success
 *              ZBX_MODULE_FAIL - module initialization failed
 *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL
 *
 *****/
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/*****
 *
 * Function: zbx_module_uninit
 *
 * Purpose: the function is called on agent shutdown
 *          It should be used to cleanup used resources if there are any
 *
 * Return value: ZBX_MODULE_OK - success
 *              ZBX_MODULE_FAIL - function failed
 *
 *****/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/*****
 *
 * Functions: dummy_history_float_cb
 *            dummy_history_integer_cb
 *
 *****/

```

```

*      dummy_history_string_cb      *
*      dummy_history_text_cb       *
*      dummy_history_log_cb        *
*
* Purpose: callback functions for storing historical data of types float, *
*         integer, string, text and log respectively in external storage *
*
* Parameters: history      - array of historical data                    *
*             history_num - number of elements in history array         *
*
*****/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/*****

```

```

*
* Function: zbx_module_history_write_cbs
*
* Purpose: returns a set of module functions Zabbix will call to export
* different types of historical data
*
* Return value: structure with callback function pointers (can be NULL if
* module is not interested in data of certain types)
*
*****/
ZBX_HISTORY_WRITE_CBS zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}

```

このモジュールでは、新たに次の3つの項目をエクスポートします。

- `dummy.ping` - 常に'1'を返します。
- `dummy.echo[param1]` - 1番目のパラメータをそのまま返します。例えば `dummy.echo[ABC]` は ABC を返します。
- `dummy.random[param1, param2]` - 1番目の引数と2番目の引数の間の任意の整数を返します。例えば `dummy.random[1, 1000000]`

制限事項

ローダブルモジュールのサポートは、Unix プラットフォームにのみ実装されています。つまり、Windows agent では動作しません。

場合によっては、モジュールに関連する設定パラメータを `zabbix_agentd.conf` から読み込む必要があります。現在、これはサポートされていません。モジュールで設定パラメータを使用する必要がある場合は、モジュール固有の設定ファイルのパーズを実装する必要があります。

7 Windows パフォーマンスカウンタ

概要

Windows のパフォーマンスカウンタは、`perf_counter[]` キーを使用して効率的に監視することができます。

例えば:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

あるいは

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

このキーまたは英語版の同等物 `perf_counter_en` の使用に関する詳細については、`[Windows-specific item keys](/manual/config/items/it` を参照してください。

監視可能なパフォーマンスカウンタの全リストを取得するには、次のコマンドを実行します。

```
typeperf -qx
```

また、ローレベルディスクバリエーション機能を使用して、複数の `object instances` を検出し、複数のインスタンスオブジェクトに対する `perf_counter item` の作成を自動化することも可能です。

数値表現

Windows はオブジェクトとパフォーマンスカウンタ名の数値表現 (インデックス) を保持しています。Zabbix はこれらの数値表現を `perf_counter`, `perf_counter_en` item キーのパラメータとして、また `PerfCounter`, `PerfCounterEn` 設定パラメータとしてサポートしています。

しかし、数値インデックスが特定のホストで正しい文字列にマップされることを保証できない限り、これらを使用することは推奨されません。ローカライズされた Windows バージョンの異なるホストで動作するポータブルな item を作成する必要がある場合は、perf_counter_en キーまたは PerfCounterEn 設定パラメータを使用して、システムのロケールにかかわらず英語名を使用することが可能です。

数値に相当するものを見つけるには、regedit を実行し、以下を見つけます。HKEY_LOCAL_MACHINE_SOFTWARE_Microsoft_Windows_NT_CurrentVersion_Perflib009

レジストリエントリには、このような情報が含まれています。

```
1
1847
2
システム
4
メモリ
6
プロセッサ時間
10
ファイル読み込み操作/秒
12
ファイル書き込み動作/秒
14
ファイルコントロール動作/秒
16
ファイル読み込みバイト数/秒
18
ファイル書き込みバイト数/秒
....
```

ここで、'System\% Processor Time' のように、パフォーマンスカウンタの各文字列部分に対応する数値が表示されます。

```
システム -> 2
% プロセッサ時間 -> 6
```

そして、この数字を使って、パスを数字で表すことができます。

```
\2\6
```

パフォーマンスカウンタパラメータ

Windows のパフォーマンスカウンタを監視するために、いくつかの PerfCounter パラメータを導入することができます。

例えば、Zabbix エージェント設定ファイルに以下を追加することができます。

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
または
PerfCounter=UserPerfCounter2,"\4\24",30
```

このようなパラメータを設定すると、それぞれの item を作成するためのキーとして UserPerfCounter1 または UserPerfCounter2 を使用するだけでよくなります。

設定ファイルを変更した後は、Zabbix エージェントを再起動することを忘れないでください。

8 一括アップデート

概要

複数の item の属性を一度に変更したい場合があります。このような場合、個々の item を開いて編集する代わりに、一括アップデート機能を使用することができます。

一括更新の使用

いくつかのアイテムを一括で更新するには、次のようにします。

- リスト内の更新するアイテムのチェックボックスをオンにする
- リストの下にある一括更新をクリックする
- 必要な属性 (アイテム、タグ、保存前処理) のあるタブに移動する
- 更新する属性のチェックボックスにマークを付ける
- 属性の新しい値を入力する

Mass update

Item Tags Preprocessing

Private key file Original

Password Original

Update interval Original

History storage period Do not keep history Storage period

Trend storage period Original

Status Original

Log time format Original

Value mapping Original

Enable trapping Original

Mass update

Item Tags Preprocessing

Tags Add Replace Remove

Name

Value

tag

value

[Add](#)

タグオプションを使用すると、以下のことが可能です。

- 追加 - 指定したタグをアイテムに追加します（同じ名前でも異なる値のタグは「重複」とみなされず、同じホストに追加できます）。
- 置換 - 指定されたタグを削除し、新しい値のタグを追加します。
- 削除 - 指定されたタグを項目から削除します。

以下のユーザーマクロはタグで使用できます。{INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}, {HOST.ID}

Mass update

Item Tags **Preprocessing**

Preprocessing steps

Name

Parameters



1:

JavaScript



script



2:

JSONPath



\$.path.to.node

[Add](#)

完了したら、更新をクリックします。

9 値のマッピング

概要

受信した値をより“人間的”に表現するために、数値/文字列値と文字列表現のマッピングを含む値マップを使用することができます。

バリューマップは Zabbix フロントエンドとメディアタイプから送信される通知の両方で使用することができます。
メディアタイプから送信される通知の両方で使用することができます。例えば、値が'0' または'1' である item は、
値マップを使って人間が読みやすい形で値を表現することができます:

- '0' => 'Not Available(利用不可)'
- '1' => 'Available(利用可)'

あるいは、バックアップに関連するバリューマップは次のようになります:

- 'F' => '全部'
- 'D' => 'デフォレンシャル (差分)'
- 'I' => 'インクリメンタル (増分)'

別の例では、電圧の値域をマッピングすることができます:

- '<=209' => 'Low'
- '210-230' => 'OK'
- '>=231' => 'High'

値のマッピングは、テンプレートまたはホストのレベルで定義されます。このような場合、“Constraints” の項を参照してください。バリューマップの継承はありません。ホスト上のテンプレート item は、テンプレートからのバリューマップをそのまま使用します。

configuring itemsでは、バリューマップを使用して、アイテムの値の表示方法を“人間工学的に”変更することができます。そのためには、バリューマッピングのフィールドで、以前に定義されたバリューマップの名前を参照します。

Note:

数値 (符号なし)、数値 (浮動)、文字 * の情報を持つ項目で使用可能です。

値のマッピングは、それぞれのテンプレートまたはホストでエクスポート/インポートすることができます。

値のマッピングは一括更新が可能です。**host** および**template**一括更新フォームには、
バリューマップを一括更新するためのバリューマッピングタブがあります。

設定

バリューマップを定義するには:

- ホストまたはテンプレート設定フォームを開く
- Value mapping タブに移動します。
- 新しいマップを追加するために Add をクリックします。
- 既存のマップの名前をクリックして、編集します。

Value mapping

* Name VMware status

* Mappings

Type	Value	Mapped to
⋮ equals ▾	0	⇒ gray
⋮ equals ▾	1	⇒ green
⋮ equals ▾	2	⇒ yellow
⋮ equals ▾	3	⇒ red

[Add](#)

Update

バリュemapのパラメータ:

パラメータ	説明
Name	Unique name of a set of value mappings.
Mappings	Individual mapping rules for mapping numeric/string values to string representations. Mapping is applied according to the order of mapping rules. It is possible to reorder mappings by dragging. Only numeric value types are supported for mapping ranges (is greater than or equals, is less than or equals, in range mapping types).
Type	Mapping type: equals - equal values will be mapped is greater than or equals - equal or greater values will be mapped is less than or equals - equal or smaller values will be mapped in range - values in range will be mapped; the range is expressed as <number1>-<number2>, or <number>. Multiple ranges are supported (e.g. 1-10,101-110,201) regexp - values corresponding to the regular expression will be mapped (global regular expressions are not supported) default - all outstanding values will be mapped, other than those with specific mappings
Value	Incoming value. Depending on the mapping type, may also contain a range or regular expression.
MappedString representation for the incoming value. to	

すべての必須入力フィールドには、赤いアスタリスクが表示されます。

バリュemapをリストに表示すると、その最初の 3 つのマッピングのみが表示され、3 つのドットはさらに多くのマッピングが

 存在することを示します。

Name	Value
VMware status	=0 ⇒ gray =1 ⇒ green =2 ⇒ yellow ...

[Add](#)

仕組み

例えば、定義済みエージェントアイテムの1つ'Zabbix agent ping'は、'Zabbix agent ping status'というテンプレートレベルの
値マップを使用して値を表示します。

Value mapping

* Name

* Mappings

Type	Value	Mapped to
equals	1	⇒ Up

item configuration form では、Value mapping の欄に、このバリューマップへの参照を見ることができます。:

Value mapping

そこで、Monitoring → Latest data でマッピングを活用し、'Up' (括弧内は生値) を表示します。

Host	Name	Last check	Last value
Zabbix server	Monitoring agent (1 Item)		
	Zabbix agent ping ?	02/23/2021 04:27:07 PM	Up (1)

Latest data では、表示されている値が 20 シンボルに短縮されます。値のマッピングが使用されている場合、
この短縮はマッピングされた値には適用されず、生の値のみに個別に適用されます (括弧内に表示)。

Note:

また、値が読みやすい形で表示されることで、通知を受ける際にも分かりやすくなります。

あらかじめ定義されたバリューマップがなければ、このようにしかならないでしょう:

Host	Name	Last check	Last value
Zabbix server	Monitoring agent (1 Item)		
	Zabbix agent ping ?	02/23/2021 06:00:07 PM	1

この場合、'1' が何を表しているかを推測するか、ドキュメントを検索して調べる必要があるわけです。

10 キュー

概要

キューには、更新待ちの item が表示されます。キューはデータの論理的表現にすぎません。
 Zabbix には IPC キューやその他のキューの仕組みはありません。

プロキシで監視されている item もキューに含まれ、proxy のヒストリーデータ更新期間中はキューにカウントされます。

キューに表示されるのは、更新時刻が設定されている item のみです。つまり、以下の item タイプはキューから除外されます。

- ログ、logrt、イベントログのアクティブな Zabbix agent item
- SNMP trap item
- Trapper item
- Web モニタリングアイテム
- 依存 item

キューが示す統計情報は、Zabbix server のパフォーマンスを示す良い指標となります。

キューは JSON プロトコルを使用して Zabbix server から直接取得されます。この情報は Zabbix サーバが動作している場合のみ
 利用可能です。

キューを読む

キューを読むには、Administration → Queue に進みます。

☰ Queue overview ▾

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

この画像は概ね"OK" なので、server は正常に動作していると考えてよいでしょう。

キューには、最大で 30 秒待っている item があります。これらの item が何であるかを知ることは素晴らしいことです。

そのためには、タイトルのドロップダウンから Queue details を選択してください。これで、これらの遅延 item の
 リストが表示されます。

☰ Queue details ▾

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

これらの情報があれば、なぜこれらの item が遅れているのかが分かるかもしれません。

1つ、2つの item が遅れていても、おそらく心配することはありません。すぐに更新されるかもしれません。
 しかし、あまりにも長い間、多くの item が遅延している場合、より深刻な問題がある可能性があります。

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

キュー item

特別な内部 item `zabbix[queue,<from>,<to>]` を使用すると、Zabbix のキューの状態を監視することができます。設定された時間だけ遅延したアイテムの数が返されます。詳細については、[Internal items](#) を参照してください。

11 値のキャッシュ

概要

トリガー式、計算 item、一部のマクロの計算を高速化するために、Zabbix server では値キャッシュオプションがサポートされています。

このインメモリキャッシュは、データベースに直接 SQL を呼び出す代わりに、過去のデータへのアクセスに使用することができます。過去の値がキャッシュに存在しない場合、不足分の値はデータベースから要求され、それに応じてキャッシュが更新されます。

値キャッシュ機能を有効にするために、オプションの `ValueCacheSize` パラメータが Zabbix server の `configuration` ファイルでサポートされています。

値キャッシュを監視するために、2 つの内部 item がサポートされています。 `zabbix[vcache,buffer,<mode>]` と `zabbix[vcache,cache,<parameter>]` です。詳細は、[internal items](#) を参照してください。

12 即時実行

概要

Zabbix の新しい item 値のチェックは、設定された更新間隔に基づき、周期的に行われます。多くの item は更新間隔が短いですが、ローレベルディスクバリエーションなど更新間隔が長いものもあり、実際の状況下では、発見可能なリソースの変更を把握するために、より早く新しい値をチェックする必要がある場合があります。このような必要性に対応するために、`passive` チェックを再スケジュールし、新しい値をすぐに取得することが可能です。

この機能は、`passive` チェックにのみサポートされています。以下の item タイプがサポートされています。

- Zabbix agent (passive)
- SNMPv1/v2/v3 agent
- IPMI agent
- Simple check
- Zabbix internal
- External check
- Database monitor
- JMX agent
- SSH agent
- Telnet
- Calculated
- HTTP agent

Attention:

詳細は、[CacheUpdateFrequency](#)を参照してください。チェックを実行する前に、
 設定キャッシュは更新されないため、item / ディスカバリールール設定の最新の変更は検出されません。したがって、作成中または
 作成直後の item / ルールの新しい値をチェックすることもできません。このような場合は、item 設定時に Test オプションを
 使用してください。

設定

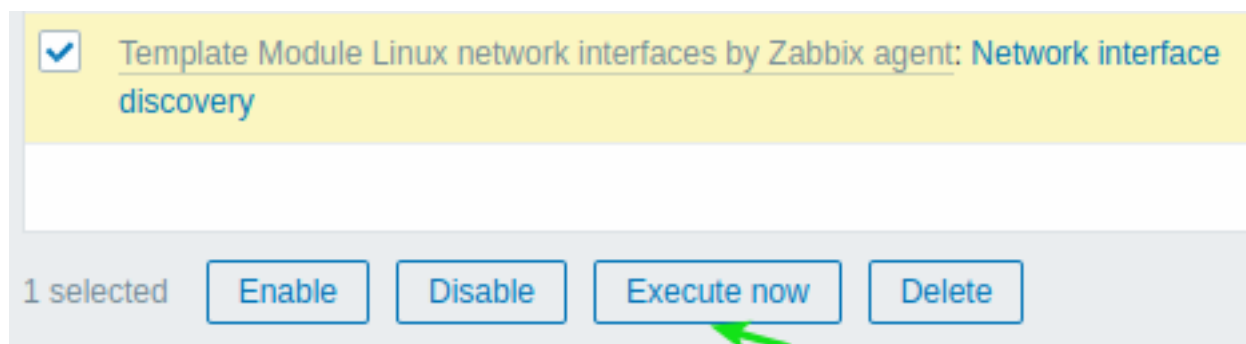
passive チェックをすぐに実行するには:

- 既存の item (またはディスカバリールール) 設定フォームで、Execute now をクリックします。

Enabled



- item / ディスカバリールールのリストで選択された item / ルールに対して、Execute now をクリックします。:



後者の場合、複数の item / ルールを選択し、一度に Execute now することができます。

13 agent チェックの制限

概要

item のブラックリスト、ホワイトリスト、またはホワイトリストとブラックリストの組み合わせを作成することで、agent 側で
 チェックを制限することができます。

そのためには、2 つの agent **configuration** パラメータを組み合わせで使用します。

- AllowKey=<pattern> - どの検査を許可するか、<pattern> はワイルドカード (*) 式を使用して指定します。
- DenyKey=<pattern> - どの検査を拒否するか、<pattern> はワイルドカード (*) 式を使用して指定します。

なお:

- deny キーが指定されていない場合でも、すべての system.run[*] アイテム (リモートコマンド、スクリプト) はデフォルトで無効化されます。
- Zabbix 5.0.2 以降、EnableRemoteCommands agent パラメータは:
 - * Zabbix agent では非推奨
 - * Zabbix agent 2ではサポートされていません

そのため、リモートコマンドを許可するには、許可するコマンドごとに AllowKey=system.run[<command>,*] を指定します
 (* は wait および nowait モードを表します)。AllowKey=system.run[*] パラメータを指定すると、wait および nowait
 モードを持つすべてのコマンドを許可することも可能です。特定のリモートコマンドを許可しないようにするには、
 AllowKey=system.run[*] パラメータの前に system.run[] コマンドを含む DenyKey パラメータを追加してください。

重要ルール

- deny ルールのないホワイトリストは、system.run[*] アイテムに対してのみ許可されます。その他のアイテムでは、DenyKey パラメータがない場合、AllowKey パラメータを使用することはできません。
- 順番は重要です。指定されたパラメータは、設定ファイル内の表示順に従って 1 つずつチェックされます:
 - item キーが allow / deny ルールに一致すると、item は許可または拒否され、ルールチェックは停止されます。したがって、ある item が allow ルールと deny ルールの両方にマッチした場合、どちらのルールが先に来るかによって結果が異なります。

- この順序は、EnableRemoteCommands パラメータにも影響します。(使用する場合)
- AllowKey / DenyKey パラメータは無制限に使用できます。
- AllowKey、DenyKey ルールは、HostnameItem、HostMetadataItem、HostInterfaceItem 構成パラメータには影響しません。
- キーパターンは、ワイルドカード (*) 文字が特定の位置にある任意の数の文字と一致するワイルドカード式です。キー名とパラメータの両方で使用することができます。
- 特定の item キーが agent 構成で許可されていない場合、その item はサポートされていないと報告されます。(理由のヒントは与えられません)。
- Zabbix agent に --print (-p) コマンドラインオプションを指定すると、設定により許可されていないキーは表示されません。
- Zabbix agent の --test(-t) コマンドラインオプションを使用すると、設定で許可されていないキーに対して "Unsupported item key." というステータスが返されます。
- 拒否されたリモートコマンドは agent ログに記録されません。(LogRemoteCommands=1 の場合)

使用例

特定のチェックを拒否する

- DenyKey パラメータで特定のチェックをブラックリスト化します。一致するキーは許可されません。一致しないキーはすべて許可されます。ただし、system.run[] 項目を除きます。

例えば:

```
# セキュアデータアクセスを拒否する
DenyKey=vfs.file.contents[/etc/passwd,*]
```

Attention:

Zabbix の新しいバージョンでは、既存の設定で明示的に制限されていない新しいキーが使用される可能性があるため、
ブラックリストは良い選択とは言えないかもしれません。これはセキュリティ上の欠陥を引き起こす可能性があります。

特定のコマンドを拒否し、他のコマンドを許可する

- 特定のコマンドをブラックリスト化するには、DenyKey パラメータを使用します。AllowKey パラメータを使用して、他のすべてのコマンドをホワイトリストに登録します。

```
# 特定のコマンドを許可しない
DenyKey=system.run[ls -l /]
```

```
# 他のスクリプトを許可する
AllowKey=system.run[*]
```

特定のチェックを許可し、他のチェックを拒否する

- AllowKey パラメータで特定のチェックをホワイトリスト化し、DenyKey=* で他のチェックを拒否する。

例:

```
# Allow reading logs:
AllowKey=vfs.file.*[/var/log/*]
```

```
# Allow localtime checks
AllowKey=system.localtime[*]
```

```
# Deny all other keys
DenyKey=*
```

パターンの例

パターン	説明	一致	非一致
*	Matches all possible keys with or without parameters.	Any	None
vfs.file.contents	Matches vfs.file.contents without parameters.	vfs.file.contents	vfs.file.contents[/etc/passwd]
vfs.file.contents[]	Matches vfs.file.contents with empty parameters.	vfs.file.contents[]	vfs.file.contents
vfs.file.contents[*]	Matches vfs.file.contents with any parameters; will not match vfs.file.contents without square brackets.	vfs.file.contents[] vfs.file.contents[/path/to/file]	vfs.file.contents

パターン	説明	一致	非一致
<code>vfs.file.contents[/etc/passwd]</code>	Matches <code>vfs.file.contents</code> with first parameters matching <code>/etc/passwd</code> and all other parameters having any value (also empty).	<code>vfs.file.contents[/etc/passwd]</code> <code>vfs.file.contents[/etc/passwd,utf8]</code>	<code>vfs.file.contents[/etc/passwd]</code> <code>vfs.file.contents[/var/log/zabbix_server.log,utf8]</code> <code>vfs.file.contents[]</code>
<code>vfs.file.contents[*passwd*]</code>	Matches <code>vfs.file.contents</code> with first parameter matching <code>*passwd*</code> and no other parameters.	<code>vfs.file.contents[/etc/passwd]</code>	<code>vfs.file.contents[/etc/passwd,]</code> <code>vfs.file.contents[/etc/passwd,utf8]</code>
<code>vfs.file.contents[*passwd*]*</code>	Matches <code>vfs.file.contents</code> with only first parameter matching <code>*passwd*</code> and all following parameters having any value (also empty).	<code>vfs.file.contents[/etc/passwd]</code> <code>vfs.file.contents[/etc/passwd,utf8]</code>	<code>vfs.file.contents[/etc/passwd]</code> <code>vfs.file.contents[/tmp/test]</code> <code>utf8]</code>
<code>vfs.file.contents[/var/log/zabbix_server.log,abc]</code>	Matches <code>vfs.file.contents</code> with first parameter matching <code>/var/log/zabbix_server.log</code> , third parameter matching <code>'abc'</code> and any (also empty) second parameter.	<code>vfs.file.contents[/var/log/zabbix_server.log,abc]</code> <code>vfs.file.contents[/var/log/zabbix_server.log,utf8,abc]</code>	<code>vfs.file.contents[/var/log/zabbix_server.log,abc]</code>
<code>vfs.file.contents[/etc/passwd,utf8]</code>	Matches <code>vfs.file.contents</code> with first parameter matching <code>/etc/passwd</code> , second parameter matching <code>'utf8'</code> and no other arguments.	<code>vfs.file.contents[/etc/passwd,utf8]</code>	<code>vfs.file.contents[/etc/passwd,]</code> <code>vfs.file.contents[/etc/passwd,utf16]</code>
<code>vfs.file.*</code>	Matches any keys starting with <code>vfs.file.</code> without any parameters.	<code>vfs.file.contents</code> <code>vfs.file.size</code>	<code>vfs.file.contents[]</code> <code>vfs.file.size[/var/log/zabbix_server.log]</code>
<code>vfs.file.*[*]</code>	Matches any keys starting with <code>vfs.file.</code> with any parameters.	<code>vfs.file.size.bytes[]</code> <code>vfs.file.size[/var/log/zabbix_server.log,utf8]</code>	<code>vfs.file.size.bytes</code>
<code>vfs.*.contents</code>	Matches any key starting with <code>vfs.</code> and ending with <code>.contents</code> without any parameters.	<code>vfs.mount.point.file.contents</code> <code>vfs..contents</code>	<code>vfs.contents</code>

system.run と AllowKey

`myscript.sh` のような仮想スクリプトは、Zabbix agent を介してホスト上でいくつかの方法で実行することができます:

1. passive チェックや active チェックの item キーとして、例えば:

- `system.run[myscript.sh]`
- `system.run[myscript.sh,wait]`
- `system.run[myscript.sh.nowait]`

ここで、“wait”、“nowait” を追加するか、第 2 引数を省略し、`system.run[]` のデフォルト値を使用することができる。

2. グローバルスクリプトとして (フロントエンドまたは API でユーザーによって開始される)

ユーザはこのスクリプトを Administration → Scripts で “Commands” 入力フィールドに “myscript.sh” を入力し、
 “Execute on: Zabbix agent” を設定します。フロントエンドまたは API から呼び出されると、Zabbix サーバは agent に以下を
 送信します。:

- `system.run[myscript.sh,wait]` - Zabbix 5.0.4 以前
- `system.run[myscript.sh]` - 5.0.5 から

ここでは、ユーザーは “wait”/“nowait” パラメータを制御しません。

3. アクションからのリモートコマンドとして Zabbix server は agent に送信します:

- `system.run[myscript.sh.nowait]`

ここでも、ユーザーは “wait”/“nowait” パラメータを制御しません。

どうということかという、AllowKey を次のように設定した場合です。:

```
AllowKey=system.run[myscript.sh]
```

では

- `system.run[myscript.sh]` - 許可されます。
- `system.run[myscript.sh,wait]`, `system.run[myscript.sh.nowait]` 許可されません - アクションのステップとして起動された場合、スクリプトは実行されません。

上記のすべてのバリエーションを許可するには:

```
AllowKey=system.run[myscript.sh,*]
DenyKey=system.run[*]
```

を agent / agent 2 パラメータに設定します。

14 プラグイン

概要

プラグインは、Zabbix の監視機能を拡張するオプションを提供します。プラグインは Go プログラミング言語で記述されており、Zabbix エージェント 2 のみでサポートされています。

Plugins provide an alternative to **loadable modules** (written in C), and other methods for extending Zabbix functionality, such as **user parameters** (agent metrics), **external checks** (agent-less monitoring), and **system.run[] Zabbix agent item**. プラグインは**ロード可能モジュール** (C 言語で記述) に代わるもので、Zabbix の機能を拡張するその他の方法として、**ユーザーパラメータ** (agent メトリクス)、**外部チェック** (エージェントレスモニタリング) や **system.run[] Zabbix エージェントアイテム** があります。

以下の機能は、Zabbix agent 2 とそのプラグインに固有のものです:

- パッシブチェックとアクティブチェックの両方について柔軟なスケジュール間隔のサポート
- スケジュールとタスクの同時実行に関するタスクキュー管理
- プラグインレベルのタイムアウト
- 起動時の Zabbix エージェント 2 とそのプラグインの互換性チェック。

Zabbix 6.0.0 以降、プラグインをエージェント 2 に直接統合する必要はなく、ロード可能なプラグインとして追加できるため、新しい監視メトリックを収集するための追加プラグインの作成プロセスが簡単になります。

このページでは、Zabbix のネイティブでロード可能なプラグインを一覧表示し、ユーザーの観点からプラグイン構成の原則について説明します。独自のプラグインを作成する手順については、[プラグイン開発ガイドライン](#)を参照してください。

プラグインの設定

このセクションでは、一般的なプラグイン設定の原則とベストプラクティスについて説明します。

すべてのプラグインは `Plugins.*` パラメータを使用して設定します。このパラメータは、Zabbix agent 2 の [configuration file] (/manual/appendix/config/zabbix_agent2) の一部、
 またはプラグイン自身の [configuration file] (/manual/appendix/config/zabbix_agent2_plugins) のいずれかに設定することが
 できます。プラグインが別の設定ファイルを使用する場合、Zabbix agent 2 設定ファイルの `Include` パラメータにこのファイルへの
 パスを指定する必要があります。

各プラグインのパラメータは、以下のような構造になっている必要があります:

```
Plugins.<PluginName>.<Parameter>=<Value>
```

パラメータ名は、以下の要件を満たす必要があります:

- プラグイン名は大文字にすることを推奨します。
- パラメータは大文字にする必要があります。
- 特殊文字は使用できません。
- ネストは最大レベルによって制限されません。
- パラメータの数は制限されません。

名前付きセッション

名前付きセッションは、プラグインパラメータの追加レベルを表し、監視対象の各インスタンスに対して個別の認証パラメータ
 セットを定義するために使用することができます。各名前付きセッションのパラメータは、以下のような構造になっている
 必要があります:

```
Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>
```

URI、ユーザー名、パスワードを別々に指定する代わりに、セッション名を `connString` の item キーパラメータとして
 使用することができます。item キーでは、最初のパラメータは `connString` または `Uri` のいずれかにすることができます。
 最初のキーパラメータが設定ファイルで指定されたセッション名と一致する場合、名前付きセッションパラメータを使用して
 チェックが実行されます。最初のキーパラメータがどのセッション名にも一致しない場合、それは `Uri` として扱われます。

ただし、:

- キーパラメータに `connString` (セッション名) を指定する場合、ユーザー名とパスワードのキーパラメータは空である必要があります。
- 埋め込み URI の認証情報を渡すことはサポートされていないので、代わりに名前付きセッションの使用を検討してください。
- 名前付きセッションで認証パラメータが指定されない場合、ハードコードされたデフォルト値が使用されます。

使用可能な名前付きセッションパラメータのリストはプラグインによって異なります。詳細は
 各プラグインの **configuration files** を参照してください。

例: 2つのインスタンス“MySQL1”と“MySQL2”の監視は、以下のように設定することができます。:

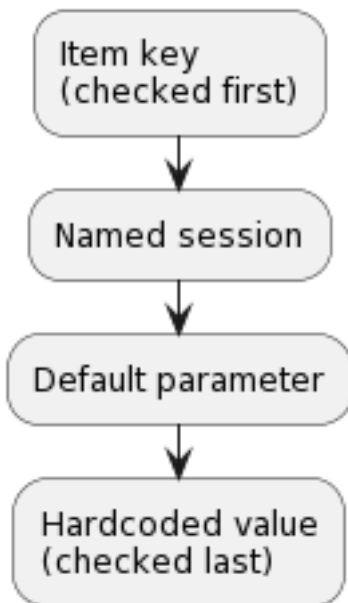
```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=<UsernameForMySQL1>
Plugins.Mysql.Sessions.MySQL1.Password=<PasswordForMySQL1>
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://127.0.0.1:3307
Plugins.Mysql.Sessions.MySQL2.User=<UsernameForMySQL2>
Plugins.Mysql.Sessions.MySQL2.Password=<PasswordForMySQL2>
```

これらの名前は、URIの代わりにキーの connString として使用することができます:

```
mysql.ping[MySQL1]
mysql.ping[MySQL2]
```

Parameter priority

Since version 6.4.3, Zabbix agent 2 plugins search for connection-related parameter values in the following order:



1. The first item key parameter is compared to session names. If no match is found it is treated as an actual value; in this case, step 3 will be skipped. If a match is found, the parameter value (usually, a URI) must be defined in the named session.
2. Other parameters will be taken from the item key if defined.
3. If an item key parameter (for example, password) is empty, plugin will look for the corresponding named session parameter.
4. If the session parameter is also not specified, the value defined in the corresponding **default parameter** will be used.
5. If all else fails, the plugin will use the hardcoded default value.

Example 1

Monitoring of two instances “MySQL1” and “MySQL2”.

Configuration parameters:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

Item keys: mysql.ping[MySQL1], mysql.ping[MySQL2]

Example 2

Providing some of the parameters in the item key (supported since Zabbix 6.0.17).

Configuration parameters:

```
Plugins.Postgres.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.Postgres.Sessions.Session1.User=old_username
Plugins.Postgres.Sessions.Session1.Password=session_password
```

Item key: `pgsql.ping[session1,new_username,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the following parameters:

- URI from session parameter: 192.0.2.234:5432
- Username from the item key: new_username
- Password from session parameter (since it is omitted in the item key): session_password
- Database name from the item key: postgres

Example 3

Collecting a metric using default configuration parameters.

Configuration parameters:

```
Plugins.Postgres.Default.Uri=tcp://192.0.2.234:5432
Plugins.Postgres.Default.User=zabbix
Plugins.Postgres.Default.Password=password
```

Item key: `pgsql.ping[,,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the parameters:

- Default URI: 192.0.2.234:5432
- Default username: zabbix
- Default password: password
- Database name from the item key: postgres

ハードコードされたデフォルト値

認証に必要なパラメータが item キーや名前付きセッションパラメータで提供されない場合、プラグインはハードコードされた
デフォルト値を使用します。

コネクション

プラグインによっては、複数のインスタンスから同時にメトリクスを収集することをサポートしています。
ローカルとリモートの両方のインスタンスを監視することができます。TCP および Unix ソケット接続がサポートされています。

インスタンスへの接続をオープン状態に保つようにプラグインを設定することをお勧めします。その利点は、ネットワークの輻輳、
レイテンシー、および CPU とメモリの使用量を削減できることです。クライアントライブラリは、この処理を行います。

Note:

未使用の接続をオープンにしておく時間は、Plugins.<PluginName>.KeepAlive パラメータで決定されます。例: Plugins.Memcached.KeepAlive

プラグイン

Zabbix agent 2 でサポートされているすべてのメトリックは、プラグインによって収集されます。

Built-in

The following plugins for Zabbix agent 2 are available out-of-the-box. Click on the plugin name to go to the plugin repository with additional information.

Plugin name	Description	Supported item keys	Comments
Agent	Metrics of the Zabbix agent being used.	agent.hostname, agent.ping, agent.version	Supported keys have the same parameters as Zabbix agent keys .
Ceph	Ceph monitoring.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	

Plugin name	Description	Supported item keys	Comments
CPU	System CPU monitoring (number of CPUs/CPU cores, discovered CPUs, utilization percentage).	system.cpu.discovery, system.cpu.num, system.cpu.util	Supported keys have the same parameters as Zabbix agent keys .
Docker	Monitoring of Docker containers.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	See also: Configuration parameters
File	File metrics collection.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Supported keys have the same parameters as Zabbix agent keys .
Kernel	Kernel monitoring.	kernel.maxfiles, kernel.maxproc	Supported keys have the same parameters as Zabbix agent keys .
Log	Log file monitoring.	log, log.count, logrt, logrt.count	Supported keys have the same parameters as Zabbix agent keys . See also: Plugin configuration parameters (Unix/Windows)
Memcached	Memcached server monitoring.	memcached.ping, memcached.stats	
Modbus	Reads Modbus data.	modbus.get	Supported keys have the same parameters as Zabbix agent keys .
MQTT	Receives published values of MQTT topics.	mqtt.get	
MySQL	Monitoring of MySQL and its forks.	mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	To configure encrypted connection to the database, use named sessions and specify the TLS parameters for the named session in the agent configuration file. Currently, TLS parameters cannot be passed as item key parameters.
Netif	Monitoring of network interfaces.	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	Supported keys have the same parameters as Zabbix agent keys .

Plugin name	Description	Supported item keys	Comments
Oracle	Oracle Database monitoring.	oracle.diskgroups.stats, ora-cle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info	Install the Oracle Instant Client before using the plugin.
Proc	Process CPU utilization percentage.	proc.cpu.util	Supported key has the same parameters as Zabbix agent key .
Redis	Redis server monitoring.	redis.config, redis.info, redis.ping, redis.slowlog.count	
Smart	S.M.A.R.T. monitoring.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	Sudo/root access rights to smartctl are required for the user executing Zabbix agent 2. The minimum required smartctl version is 7.1. Supported keys can be used with Zabbix agent 2 only on Linux/Windows, both as a passive and active check. See also: Configuration parameters
Swap	Swap space size in bytes/percentage.	system.swap.size	Supported key has the same parameters as Zabbix agent key .
SystemRun	Runs specified command.	system.run	Supported key has the same parameters as Zabbix agent key . See also: Plugin configuration parameters (Unix/Windows)
Systemd	Monitoring of systemd services.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP connection availability check.	net.tcp.port	Supported key has the same parameters as Zabbix agent key .
UDP	Monitoring of the UDP services availability and performance.	net.udp.service, net.udp.service.perf	Supported keys have the same parameters as Zabbix agent keys .
Uname	Retrieval of information about the system.	system.hostname, system.sw.arch, system.uname	Supported keys have the same parameters as Zabbix agent keys .
Uptime	System uptime metrics collection.	system.uptime	Supported key has the same parameters as Zabbix agent key .

Plugin name	Description	Supported item keys	Comments
VFSDev	VFS metrics collection.	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	Supported keys have the same parameters as Zabbix agent keys .
WebCertificate	Monitoring of TLS/SSL website certificates.	web.certificate.get	
WebPage	Web page monitoring.	web.page.get, web.page.perf, web.page.regex	Supported keys have the same parameters as Zabbix agent keys .
ZabbixAsync	Asynchronous metrics collection.	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	Supported keys have the same parameters as Zabbix agent keys .
ZabbixStats	Zabbix server/proxy internal metrics or number of delayed items in a queue.	zabbix.stats	Supported keys have the same parameters as Zabbix agent keys .
ZabbixSync	Synchronous metrics collection.	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	Supported keys have the same parameters as Zabbix agent keys .

読み込み可

Note:

ロード可能なプラグインを以下のオプションで起動した場合: `--version` - プラグインのバージョンとライセンス情報を出
力します。 `--help` - ヘルプ情報を出力します。

プラグイン名をクリックして、追加情報を含むプラグインリポジトリに移動します。

プラグイン名	説明	対応アイテムキー	コメント
MongoDB	MongoDB サーバーとクラスターの監視 (ドキュメントベース、分散データベース)	mongodb.collection.stats、 mon- godb.collections.discovery、 mon- godb.collections.usage、 mon- godb.connpool.stats、 mongodb.db.stats、 mon- godb.db.discovery、 mon- godb.jumbo_chunks.count、 mongodb.oplog.stats、 mongodb.ping、 mongodb.rs.config、 mongodb.rs.status、 mon- godb.server.status、 mongodb.sh.discovery	
PostgreSQL	PostgreSQL とそのフォークの監視	pgsql.autovacuum.count、 pgsql.archive、 pgsql.bgwriter、 pgsql.cache.hit、 pgsql.connections、 pgsql.custom.query、 pgsql.dbstat、 pgsql.dbstat.sum、 pgsql.db.age、 pgsql.db.bloating_tables、 pgsql.db.discovery、 pgsql.db.size、 pgsql.locks、 pgsql.oldest.xid、 pgsql.ping、 pgsql.queries、 pgsql.replication.count、 pgsql.replication.process、 pgsql.replication.process.discovery、 pgsql.replication.recovery_role、 pgsql.replication.status、 pgsql.replication_lag.b、 pgsql.replication_lag.sec、 pgsql.uptime、 pgsql.wal.stat	このプラグインは Zabbix 6.0.10 以降でロード可能です (以前はビルトインでした)。 データベースへの暗号化された接続を構成するには、 <code>named_sessions</code> を作成し、エージェント構成ファイルで名前付きセッションの TLS パラメーターを指定します。現在、TLS パラメーターをアイテムキーパラメーターとして渡すことはできません。

参照: [ロード可能なプラグインの構築](#).

Footnotes

¹ - Since Zabbix 6.0.13, loadable plugins started using the same versioning system as Zabbix itself. As a result, MongoDB plugin version has changed from 1.2.1 to 6.0.13.

1 ローダブルプラグインのビルド

概要

このページでは、ソースからロード可能なプラグインのバイナリを構築するために必要な手順を説明します。

ソース tarball をダウンロードしていれば、プラグインをオフラインで、つまりインターネットに接続せずに構築できます。

PostgreSQL プラグインが例として使用されます。他のロード可能なプラグインも同様の方法で構築できます。

手順

1. [Zabbix Cloud Images and Appliances](#)からソースをダウンロードする。公式なダウンロードページはすぐに用意します。

2. プラグインをビルドするサーバーにアーカイブを転送する。

3. tar ボールを以下のようにして展開する：

```
tar xvf zabbix-agent2-plugin-postgresql-1.0.0.tar.gz
```

ファイル名"zabbix-agent2-plugin-postgresql-1.0.0.tar.gz" は、ダウンロードしたファイルのファイル名に読み替えてください。

4. 展開したディレクトリ内に移動する：

```
cd <path to directory>
```

5. 実行：

```
make
```

6. プラグインの実行可能ファイルは、Zabbix エージェント 2 が読み込める場所であればどこでも配置できます。プラグインの設定ファイルでプラグインのバイナリへのパスを設定します。例えば、PostgreSQL 用のプラグインでは、postgresql.conf に記述します：

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

7. プラグイン設定ファイルへのパスは、Zabbix エージェント 2 の設定ファイルの Include パラメーターで指定する必要があります：

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Makefile のターゲット

Zabbix が用意しているローダブルモジュールの基本的な Makefile には以下のようなターゲットを用意しています：

ターゲット	説明
make	プラグインをビルドします。
make clean	プラグインのビルド時に通常作成されたすべてのファイルを削除します。
make check	セルフテストを実行します。実際の PostgreSQL データベースが必要です。
make style	'golangci-lint' を用いて Go 言語の文法チェックを行います。
make format	'go fmt' を用いて Go 言語のコードのフォーマットを行います。
make dist	プラグインのソースと、プラグインとそのセルフテストのビルドに必要なすべてのパッケージのソースを含むアーカイブを作成

3 トリガー

概要

トリガーは、アイテムによって収集されたデータを"評価"し、現在のシステム状態を表す論理式です。

アイテムはシステムデータを収集するために使用されますが、これらのデータを常に追跡し、警戒すべき状態や注意が必要な状態を待つのは非常に現実的ではありません。データを「評価」する処理は、トリガーの条件式に任せることができます。

トリガーの条件式は、どのような状態のデータが"許容範囲"であるかの閾値を定義することができます。したがって、取得したデータが許容範囲を超えると、トリガーは"起動"され、ステータスが障害に変更されます。

トリガーは、次のようなステータスを持つことができます：

状態	説明
正常	これは正常なトリガーの状態です。
障害	何かが起こった状態です例えば、プロセッサの負荷が高すぎる。
不明	トリガーの値が計算できない。 不明な状態参照

単純なトリガーでは、CPU 負荷などの一部のデータの 5 分間の平均値にしきい値を設定したいかもしれません。これは、トリガーの条件式を以下のように定義することによって実現できます。

- アイテムキーで受け取った値に'avg'関数を適用する
- 評価期間として 5 分間を使用する
- しきい値を'2'に設定する

```
avg(host/key,5m)>2
```

このトリガーは、5 分間の平均が 2 を超えたら、“発火”します。(障害の状態になる)

より複雑なトリガーでは、条件式は複数の関数と複数の閾値の組み合わせを含むことができます。**トリガー条件式 Trigger expression**を参照してください。

Note:

トリガーを有効にした (構成ステータスを 無効から 有効に変更) 後、トリガー条件式は、その項目が値を受け取るか、時間ベースの関数を処理する時間が来るとすぐに評価されます。

ほとんどのトリガー関数はアイテムの値の**履歴データ**に基づいて評価されますが、一部のトリガー関数は長期的な分析に使用されます。**trendavg()**、**trendcount()**などはトレンドデータを使用します。

計算時間

Zabbix サーバーが式の一部である新しい値を受け取るたびに、トリガーが再計算されます。新しい値を受け取ると、式に含まれる各関数が再計算されます (新しい値を受け取った関数だけではありません)。

さらに、式で時間ベースの関数が使用されている場合は、新しい値が受信されるたび、おおよび 30 秒ごとにトリガーが再計算されます。

時間ベースの関数は、**nodata()**、**date()**、**dayofmonth()**、**dayofweek()**、**time()**、**now()** です。;これらは、Zabbix 履歴同期プロセスによって 30 秒ごとに再計算されます。

トレンド関数のみを参照するトリガーは、式の最小期間ごとに 1 回評価されます。**トレンド関数**も参照してください。

評価期間

評価期間とは、アイテムの履歴を参照する関数で使用されるもので、対象とする期間を指定します。評価期間は、期間 (30 秒、10 分、1 時間)、値域 (#5 - 最新値 5 個) のいずれでも指定可能です。

評価期間は“now ” まで計測され、ここで“now ” は trigger の最新の再計算時間です (上記の**計算時間**を参照)。“now ” はサーバーの“now” 時間ではありません。

評価期間では、どちらかを指定します：

- “now-time period ” と“now ” の間のすべての値 (または、タイムシフトを使用する場合、“now-time shift-time period ” と“now-time_shift ” の間) を考慮する。
- 過去の値のうち、“now” までの num カウントを考慮しない。
 - 指定された期間や数に対して、利用可能な値が 0 である場合、この関数を使用するトリガーや計算アイテムは、サポートされません。

以下に注意：

- トリガーで単一の関数 (履歴データの参照) のみが使用されている場合、“now” は常に最新の受信値である。例えば、1 時間前に最後の値を受信した場合、評価期間は最新の値までとみなされます。
- 新しいトリガーは、最初の値が受信されるとすぐに計算されます (履歴関数)、時間ベースの機能では 30 秒以内に計算されます。したがって、トリガーが作成されてから、設定された評価期間 (たとえば 1 時間) が経過していても、トリガーは計算されます。また、評価範囲が例えば最新の 10 個の値に設定されていたとしても、トリガーは最初の値の後に計算されます。

状態不明

次の場合は、トリガー式に不明なオペランドが現れる可能性があります。

- サポートされていないアイテムが使用されている
- サポートされているアイテムの関数評価がエラーになる

この場合トリガーは一般的に“不明”と評価されます (例外もあります)。詳細については**オペランドが不明な式**を参照してください。

未知のトリガーについて**通知を受ける**ことも可能です。

1 トリガーの設定

概要

trigger を設定するには、次のようにします。

- Configuration → Hosts にアクセスします。
- ホストの行にある Triggers をクリックします。
- 右側の Create trigger をクリックします (または、既存の trigger を編集するために trigger 名をクリックします)
- フォームに trigger のパラメータを入力します。

trigger とその計算時間については、**general information**も参照してください。

設定

Trigger タブには、trigger の基本的な属性がすべて含まれています。

Trigger
Tags
Dependencies 1

*** Name**

Event name

Operational data

Severity Not classified Information Warning Average High Dis

*** Expression**

[Expression constructor](#)

OK event generation Expression Recovery expression None

PROBLEM event generation mode Single Multiple

OK event closes All problems All problems if tag values match

*** Tag for matching**

Allow manual close

URL

Description

Enabled

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
Name	Trigger name. Supported macros are: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.LASTVALUE}, {ITEM.LOG.*} and {MACRO} user macros. \$1, \$2...\$9 macros can be used to refer to the first, second...ninth constant of the expression. Note: \$1-\$9 macros will resolve correctly if referring to constants in relatively simple, straightforward expressions. For example, the name "Processor load above \$1 on {HOST.NAME}" will automatically change to "Processor load above 5 on New host" if the expression is last(/New host/system.cpu.load[percpu,avg1])>5

パラメータ	説明
Event name	<p>If defined, this name will be used to create the problem event name, instead of the trigger name.</p> <p>The event name may be used to build meaningful alerts containing problem data (see example).</p> <p>The same set of macros is supported as in the trigger name, plus {TIME} and {?EXPRESSION} expression macros.</p> <p>Supported since Zabbix 5.2.0.</p>
Operational data	<p>Operational data allow to define arbitrary strings along with macros. The macros will resolve dynamically to real time data in Monitoring → Problems. While macros in the trigger name (see above) will resolve to their values at the moment of a problem happening and will become the basis of a static problem name, the macros in the operational data maintain the ability to display the very latest information dynamically.</p> <p>The same set of macros is supported as in the trigger name.</p>
Severity Expression	<p>Set the required trigger severity by clicking the buttons.</p> <p>Logical expression used to define the conditions of a problem. A problem is created after all the conditions included in the expression are met, i.e. the expression evaluates to TRUE. The problem will be resolved as soon as the expression evaluates to FALSE, unless additional recovery conditions are specified in Recovery expression.</p>
OK event generation	<p>OK event generation options:</p> <p>Expression - OK events are generated based on the same expression as problem events;</p> <p>Recovery expression - OK events are generated if the problem expression evaluates to FALSE and the recovery expression evaluates to TRUE;</p> <p>None - in this case the trigger will never return to an OK state on its own.</p>
Recovery expression	<p>Logical expression (optional) defining additional conditions that have to be met before the problem is resolved, after the original problem expression has already been evaluated as FALSE.</p> <p>Recovery expression is useful for trigger hysteresis. It is not possible to resolve a problem by recovery expression alone if the problem expression is still TRUE.</p> <p>This field is only available if 'Recovery expression' is selected for OK event generation.</p>
PROBLEM event generation mode	<p>Mode for generating problem events:</p> <p>Single - a single event is generated when a trigger goes into the 'Problem' state for the first time;</p> <p>Multiple - an event is generated upon every 'Problem' evaluation of the trigger.</p>
OK event closes	<p>Select if OK event closes:</p> <p>All problems - all problems of this trigger</p> <p>All problems if tag values match - only those trigger problems with matching event tag values</p>
Tag for matching	<p>Enter event tag name to use for event correlation.</p> <p>This field is displayed if 'All problems if tag values match' is selected for the OK event closes property and is mandatory in this case.</p>
Allow manual close	<p>Check to allow manual closing of problem events generated by this trigger. Manual closing is possible when acknowledging problem events.</p>
URL	<p>If not empty, the URL entered here is available as a link in several frontend locations, e.g. when clicking on the problem name in Monitoring → Problems (URL option in the Trigger menu) and Problems dashboard widget.</p> <p>The same set of macros is supported as in the trigger name, plus {EVENT.ID}, {HOST.ID} and {TRIGGER.ID}. Note that user macros with secret values will not be resolved in the URL.</p>

パラメータ	説明
Description	Text field used to provide more information about this trigger. May contain instructions for fixing specific problem, contact detail of responsible staff, etc.
Enabled	The same set of macros is supported as in the trigger name. Unchecking this box will disable the trigger if required. Problems of a disabled trigger are no longer displayed in the frontend, but are not deleted.

Tags タブでは、trigger-level tagsを定義します。この trigger のすべての問題は、ここで入力された値でタグ付けされます。

さらに、Inherited and trigger tags オプションにより、テンプレートレベルで定義されたタグを、
 trigger がそのテンプレートから来る場合に表示することができます。同じタグを持つ複数のテンプレートがある場合、
 これらのタグは一度に表示されません。
 テンプレート名はカンマで区切られます。trigger は、ホストレベルのタグを”継承”して表示することはありません。

パラメータ	説明
Name/Value	<p>Set custom tags to mark trigger events.</p> <p>Tags are a pair of tag name and value. You can use only the name or pair it with a value. A trigger may have several tags with the same name, but different values.</p> <p>User macros, user macro context, low-level discovery macros and macro functions with <code>{ITEM.VALUE}</code>, <code>{ITEM.LASTVALUE}</code> and low-level discovery macros are supported in event tags.</p> <p>Low-level discovery macros can be used inside macro context. <code>{TRIGGER.ID}</code> macro is supported in trigger tag values. It may be useful for identifying triggers created from trigger prototypes and, for example, suppressing problems from these triggers during maintenance.</p> <p>If the total length of expanded value exceeds 255, it will be cut to 255 characters.</p> <p>See all macros supported for event tags.</p> <p>Event tags can be used for event correlation, in action conditions and will also be seen in Monitoring → Problems or the Problems widget.</p>

Dependencies タブには、trigger のすべての **dependencies** が含まれています。

新しい依存関係を追加するには、Add をクリックします。

Note:

また、既存の trigger を開いて Clone ボタンを押し、別の名前でも保存することによっても、trigger を構成することができます。

評価式のテスト

設定した trigger 式が、受け取った値によってどのような式結果になるかをテストすることが可能である。

例として、公式テンプレートにある以下の式を挙げる。

```
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.#{SNMPINDEX}],5m)>{TEMP_WARN}
```

または

```
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.#{SNMPINDEX}])={TEMP_WARN_STATUS}
```

式をテストするには、式フィールドの下にある Expression constructor をクリックします。

Trigger Tags Dependencies

* Name Cisco IOS SNMPv2: Temperature is too high

Event name Cisco IOS SNMPv2: Temperature is too high

Operational data

Severity Not classified Information **Warning** Average High Disaster

* Expression

```
avg(/Cisco IOS
SNMPv2/sensor.temp.value[ciscoEnvMonTemperature
Value.#{SNMPINDEX}],5m)>{TEMP_WARN}
or
last(/Cisco IOS
SNMPv2/sensor.temp.status[ciscoEnvMonTemperatur
eState.#{SNMPINDEX}])={TEMP_WARN_STATUS}
```

Add

[Expression constructor](#)

Expression コンストラクタでは、個々の式がすべてリストアップされます。テスト・ウィンドウを開くには、
 式リストの下にある Test をクリックします。

Target Expression

Or

A avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.#{SNMPINDEX}],5m)>{TEMP_WARN}

B last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.#{SNMPINDEX}])={TEMP_WARN_STATUS}

[Test](#)

テスト画面では、サンプル値（この例では'80'、'70'、'0'、'1'）を入力し、Test ボタンをクリックすると、式の結果を確認することができます。

Test

Expression Variable Elements	Result type	Value
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.#{SNMPINDEX}],5m)	Numeric (float)	80
{TEMP_WARN}	Any	70
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.#{SNMPINDEX}])	Numeric (integer)	0
{TEMP_WARN_STATUS}	Any	1

Expression	Result	Error
Or	TRUE	
A avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.#{SNMPINDEX}],...	TRUE	
B last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.#{SNMPINDEX}],...	FALSE	
A or B	TRUE	

Test Cancel

個々の表現だけでなく、全体の表現の結果も見ることができます。

"TRUE" は、指定された式が正しいことを意味する。このケース A では、"80" は {TEMP_WARN} の指定値 (この例では "70" です) より大きい。
 予想通り、"TRUE" の結果が表示される。

"FALSE" は、指定された式が正しくないことを意味します。この場合この特定のケース B では、
 {TEMP_WARN_STATUS} の "1" が "1" と等しい必要がある。"1" は、指定された値、この例では "0" と等しくなければならない。
 予想通り、"FALSE" の結果が表示される。

式のタイプは "OR" である。指定された条件のうち少なくとも 1 つ (この例では A または B) が
 指定された条件 (この例では A または B) の少なくとも 1 つが "TRUE" であれば、全体の結果も "TRUE" となります。
 つまり、現在値が警告値を超えており、問題が発生していることを意味します。

2 トリガー条件式

Overview

triggers で使用される式は、柔軟性があります。
 これらを使用して、監視する統計に関する複雑な論理テストを作成することができます。

単純な式は、いくつかのパラメータを持つ項目に適用される関数を使用します。この関数は、演算子と定数を使用して、
 閾値と比較される結果を返します。

単純で便利な式のシンタックスは `function(/host/key,parameter)<operator><constant>` です。

例:

```
min(/Zabbix_server/net.if.in[eth0,bytes],5m)>100K
```

は、過去 5 分間の受信バイト数が常に 100 キロバイト以上であった場合に trigger されます。

構文は全く同じですが、機能的な観点からは、2 種類のトリガー式があります:

- problem expression - 問題の条件を定義します。
- recovery expression (オプション) - 問題解決のための追加条件を定義します。

問題式を単独で定義した場合、この式は問題閾値と問題回復閾値の両方として使用されます。
 問題式が "TRUE" と評価されるとすぐに、問題が発生します。問題式が "FALSE" と評価されるとすぐに、問題は解決されます。

問題式と回復式の両方を定義する場合、問題解決はより複雑になります:
 問題式が FALSE であるだけでなく、回復式も TRUE でなければなりません。これは、**hysteresis** を作成して、
 トリガーのバタつきを回避するのに便利です。

関数

収集した値の計算 (平均、最小、最大、合計)、文字列の検索、現在時刻の参照などを行うことができる関数です。

supported functions の完全なリストが利用可能です。

通常、関数は比較のために数値を返します。文字列を返す場合、= および <> 演算子で比較が可能です。
 (example を参照してください)

関数パラメータ

関数パラメータでは、以下の項目を指定できます。

- ホストと item キー (ホストの item history を参照する機能のみ)
- 関数固有のパラメータ
- その他の式 (ホスト item の history を参照する関数では使用できません。例については **other expressions** を参照してください)

ホストと item のキーは /host/key として指定することができます。参照する item はサポートされている状態でなければなりません。
 (ただし、**nodata()** 関数はサポートされていないアイテムについても計算されます)

関数パラメータとしての他の trigger 式は、trigger における非 history 関数に制限されますが、
 calculated items では、この制限は適用されません。

関数別パラメータ

関数別パラメータは、item キーの後に置かれ、item キーとカンマで区切られます。
 これらのパラメータの一覧は、**supported functions** を参照してください。

数値関数のほとんどは、パラメータとして時刻を受け付けます。この場合、秒、または **time suffixes** を使って
 時間を表すことができます。
 ハッシュタグを前に付けると、そのパラメータは別の意味を持ちます。

式	説明
<code>sum(/host/key,10m)</code>	過去 10 分間の値の総和
<code>sum(/host/key,#10)</code>	直近の 10 件の値の合計値

ハッシュタグの付いたパラメータは、関数 **last** では異なる意味を持ちます - N 番目の前の値を示します。

- `last(/host/key,#2)` は '7' を返します。
- `last(/host/key,#5)` は '5' を返します。

タイムシフト

オプションのタイムシフトは、関数パラメータとして時間または値カウンタでサポートされます。このパラメータを使用すると、過去のある期間のデータを参照できます。

タイムシフトは、現在の時間を指定する `now` で始まり、その後 `+N<time unit>` または `-N<time unit>` が続き、N 時間、加算または減算します。

たとえば `avg(/host/key,1h:now-1d)` は 1 日前の 1 時間の平均値を返します。

絶対時間帯による時間シフト

タイムシフトパラメータでは、絶対時間帯がサポートされています。たとえば、1 日の場合は午前 0 時から午前 0 時、1 週間の場合は月曜日から日曜日、1 か月の場合は月の最初の日から最後の日です。

絶対時間帯の時間シフトは `now` で始まります。現在の時間を指定し、その後任意の数の時間操作が続きます: `<time unit> - 時間単位` の開始と終了を定義します。たとえば 1 日の午前 0 時から午前 0 時まで。 `+N<time unit>` または `-N<time unit>` - N 時間単位の加算または減算。

タイムシフトの値は 0 以上で、期間の最小値は 1 であることに注意してください。

パラメータ	説明
<code>1d:now/d</code>	昨日
<code>1d:now/d+1d</code>	当日
<code>2d:now/d+1d</code>	過去 2 日間
<code>1w:now/w</code>	先週
<code>1w:now/w+1w</code>	今週

その他の式

関数のパラメータは、次のように他の式を含むことができます。

`min(min(/host/key,1h),min(/host2/key2,1h)*10)`

関数が `item history` を参照する場合、他の式を使用することはできませんのでご注意ください。
 例えば、次のような構文は使用できません。

`min(/host/key,#5*10)`

演算子

trigger では、以下の演算子がサポートされています (実行優先度の高い順) :

優先度	演算子	定義	【不明な値】の注意事項 (<code>/manual/config/triggers/expression#expressions_with_unsupported_items</code> and <code>unknown_values</code>)	オペランドを強制的に <code>unknown</code> にする
1	-	単項負	-Unknown → Unknown	Yes
2	not	論理否定	not Unknown → Unknown	Yes
3	*	乗算	0 * Unknown → Unknown (yes, Unknown, not 0 - 算術演算では不明) 1.2 * Unknown → Unknown	Yes
	/	分割	Unknown / 0 → error Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Yes
4	+	算術プラス	1.2 + Unknown → Unknown	Yes

優先度	演算子	定義	【不明な値】の注意事項 (/manual/config/triggers/expression#expressions_with_unsupported_items_and_unknown_values)	オペランドを強制的にUnknownにする
	-	算術マイナス	1.2 - Unknown → Unknown	Yes
5	<	未満。演算子は次のように定義されます:	1.2 < Unknown → Unknown	Yes
	<=	以下。演算子は次のように定義されます:	Unknown <= Unknown → Unknown	Yes
	>	超過。演算子は次のように定義されます:		Yes
	>=	以上。演算子は次のように定義されます:		Yes

優先度	演算子	定義	【不明な値】の注意事項 (/manual/config/triggers/expression#expressions_with_unsupported_items_and_unknown_values)	オペランドを強制的に1 and 1 known_values)
6	=	等しい。演算子は次のように定義されます: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		No ¹
	<>	等しくない。演算子は次のように定義されます: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		No ¹
7	and	論理積	0 and Unknown → 0 1 and Unknown → Unknown Unknown and Unknown → Unknown	Yes
8	or	論理和	1 or Unknown → 1 0 or Unknown → Unknown Unknown or Unknown → Unknown	Yes

¹ 文字列オペランドが数値にキャストされるのは、以下の場合です:

- 他のオペランドが数値である場合
- オペランドに = または <> 以外の演算子が使用されている場合。

(キャストに失敗した場合、数値オペランドは文字列オペランドにキャストされ、両方のオペランドは文字列として比較されます)。

not, **and**, **or** 演算子は大文字と小文字を区別する。また、空白または括弧で囲む必要があります。

単項の-と **not** を除くすべての演算子は、左から右への連想性を持っています。
 単項の-と **not** は非連想性である (つまり、-(-1) と **not (not 1)** は、--1 と **not 1** の代わりに使用すべきです。)

評価結果:

- <, <=, >, >=, <> 演算子は、指定された関係が真であれば trigger に '1' を、偽であれば '0' をもたらすものとする。少なくとも一つのオペランドが不明である場合、結果は不明です。
- 未知のオペランドに対して、**and** は、オペランドの両方が '0' と等しくない比較をした場合、'1' を生成し、それ以外の場合は、'0' を生成します。
- 未知のオペランドに対しては、**or** は一方のオペランドが '0' と等しくない場合にのみ '1' を出力し、そうでない場合は '0' を出力します。
- 既知のオペランドに対する論理否定演算子 **not** の結果は、オペランドの値が '0' と等しくない場合は '0' となり、オペランドの値が '0' と等しい場合は '1' となる。未知のオペランドに対しては、**not** は 'Unknown' を生成します。

値のキャッシュ

trigger 評価に必要な値は、Zabbix server によってキャッシュされます。
 このため、server 再起動後のしばらくの間、trigger 評価によるデータベースの負荷が高くなります。
 item history の値を削除しても (手動または housekeeper で) キャッシュはクリアされないため、trigger 関数で定義した期間より古い値になるか、
 server が再起動されるまで、キャッシュされた値が使用されます。

trigger の例

例 1

Zabbix server のプロセッサの負荷が高すぎます。

```
last(/Zabbix server/system.cpu.load[all,avg1])>5
```

関数'last()' を使用することで、最新の値を参照することができます。
 /Zabbix server/system.cpu.load[all,avg1] には監視するパラメータのショートネームが入ります。
 ホストが Zabbix server で、監視するキーが system.cpu.load[all,avg1] であることを指定します。最後に、>5 は、Zabbix server からの
 最新のプロセッサ負荷測定値が 5 より大きい場合、トリガーが PROBLEM 状態になることを意味します。

例 2

www.example.com は、過負荷になっています。

```
last(/www.example.com/system.cpu.load[all,avg1])>5 or min(/www.example.com/system.cpu.load[all,avg1],10m)>2
```

この式は、現在のプロセッサ負荷が 5 以上であるか、過去 10 分間のプロセッサ負荷が 2 以上であった場合に"TRUE" となります。

例 3

/etc/passwd が変更された。

```
(last(/www.example.com/vfs.file.cksum[/etc/passwd],#1)<>last(/www.example.com/vfs.file.cksum[/etc/passwd],#1))
```

この式は、/etc/passwd のチェックサムの前回の値が最新の値と異なる場合に"TRUE" となります。

同様の式は、/etc/passwd、/etc/inetd.conf、/kernel などの重要なファイルの変更を監視するのに便利でしょう。

例 4

誰かがインターネットから大きなファイルをダウンロードしている。

min 関数を使って:

```
min(/www.example.com/net.if.in[eth0,bytes],5m)>100K
```

eth0 の受信バイト数が過去 5 分以内に 100KB 以上であれば、この式は"TRUE" となります。

例 5

クラスタ化された SMTP サーバの両ノードがダウン

1 つの式に 2 つの異なるホストが使用されていることに注意してください。

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

smtp1.example.com と smtp2.example.com の両方で SMTP サーバがダウンしている場合、この式は成立します。

例 6

Zabbix agent のアップグレードが必要

関数 find() を使用して:

```
find(/example.example.com/agent.version,,"like","beta8")=1
```

この式は、Zabbix agent のバージョンが beta8 である場合に"TRUE" となります。

例 7

サーバにアクセスできない

```
count(/example.example.com/icmpping,30m,,"0")>5
```

この式は、ホスト"example.example.com" が過去 30 分間に 5 回以上到達不能になった場合に"TRUE" となる。

例 8

過去 3 分以内に heartbeats がない。

関数 nodata() を使用して:

```
nodata(/example.example.com/tick,3m)=1
```

この trigger を使用するには、Zabbix **trapper** item として 'tick' を定義する必要があります。
 ホストはこの item のデータを zabbix_sender を使用して定期的に送信する必要があります。180 秒以内にデータが受信されないと、
 trigger の値が PROBLEM になります。

注意 'nodata' は任意の item タイプに使用できることに注意してください。

例 9

夜間の CPU アクティビティ

関数 time() を使用し、

```
min(/Zabbix_server/system.cpu.load[all,avg1],5m)>2 and time(>000000 and time(<060000
```

夜間 (00:00 ~ 06:00) にのみトリガーの状態を true に変更できます。

例 10

例外を除く任意の時点での CPU アクティビティ

関数 time() と **not** 演算子を使用:

```
min(/zabbix/system.cpu.load[all,avg1],5m)>2  
and not (dayofweek(=7 and time(>230000)  
and not (dayofweek(=1 and time(<010000)
```

トリガーは、各週の 2 時間 (日曜日の 23:00 から月曜日の 01:00) を除いて、いつでも状態を true に変更できます。

例 11

クライアントのローカルタイムが Zabbix サーバのタイムと同期しているかどうかを確認します。

関数 fuzzytime() を使用して:

```
fuzzytime(/MySQL_DB/system.localtime,10s)=0
```

MySQL_DB と Zabbix server のローカルタイムが 10 秒以上異なる場合、trigger は障害状態に変更されます。'system.localtime' は **パ**
ツ**シ****ブ****チ****ェ****ク** として設定する必要があることに注意してください。

例 12

今日の平均負荷と昨日の同時刻の平均負荷を比較します (タイムシフトを now-1d として使用)。

```
avg(/server/system.cpu.load,1h)/avg(/server/system.cpu.load,1h:now-1d)>2
```

この式は、過去 1 時間の平均負荷が昨日の同じ時間の平均負荷を 2 回以上超えた場合に起動します。

例 13

別のアイテムの値を使用して、トリガーのしきい値を取得します。

```
last(/Template PfSense/hrStorageFree[#{SNMPVALUE}])<last(/Template PfSense/hrStorageSize[#{SNMPVALUE}])*0.
```

空き容量が 10% を下回ると、トリガーが起動します。

例 14

評価結果 を使用して、しきい値を超えるトリガーの数を取得します。

```
(last(/server1/system.cpu.load[all,avg1])>5) + (last(/server2/system.cpu.load[all,avg1])>5) + (last(/serve
```

式内のトリガーのうち少なくとも 2 つが 5 を超えている場合にトリガーが起動します。

例 15

2 つのアイテムの文字列値の比較 - ここでのオペランドは、文字列を返す関数です。

問題: ホストによって Ubuntu のバージョンが異なる場合にアラートを作成する

```
last(/NY Zabbix server/vfs.file.contents[/etc/os-release])<>last(/LA Zabbix server/vfs.file.contents[/etc/
```

例 16

2 つの文字列値の比較 - オペランドは次の通りです:

- 文字列を返す関数
- マクロと文字列の組み合わせ

問題：DNS クエリの変更を検出する

アイテムキーは次の通りです：

```
net.dns.record[8.8.8.8,{$WEBSITE_NAME},{DNS_RESOURCE_RECORD_TYPE},2,1]
```

次のように定義されたマクロを使用します。

```
{WEBSITE_NAME} = example.com  
{DNS_RESOURCE_RECORD_TYPE} = MX
```

通常は以下を返します。

```
example.com          MX          0 mail.example.com
```

したがって、DNS クエリの結果が期待される結果から逸脱しているかどうかを検出するためのトリガー式は次のとおりです：

```
last(/Zabbix server/net.dns.record[8.8.8.8,{$WEBSITE_NAME},{DNS_RESOURCE_RECORD_TYPE},2,1])<>"{WEBSITE_NAME}
```

2 番目のオペランドが引用符で囲まれていることに注意してください。

例 17

2 つの文字列値の比較 - オペランドは次のとおりです：

- 文字列を返す関数
- 特殊文字 \および" を含む文字列定数

問題： /tmp/hello ファイルの内容が以下と等しいかどうかを検出します。

```
\ " //hello ?\"
```

オプション 1) 文字列を直接書き込む

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\" //hello ?\""
```

文字列が直接比較される時に、\ と" の文字がどのようにエスケープされるか注意してください。

オプション 2) マクロを使用する

```
{HELLO_MACRO} = \ " //hello ?\"
```

式：

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])={HELLO_MACRO}
```

例 18

長期間の比較

問題：先月、Exchange server の負荷が 10% 以上増加した。

```
trendavg(/Exchange/system.cpu.load,1M:now/M)>1.1*trendavg(/Exchange/system.cpu.load,1M:now/M-1M)
```

trigger 設定の **Event name** フィールドを使って、例えば次のようなものを受け取るために、意味のある警告メッセージを構築することも可能です。

```
"Load of Exchange server increased by 24% in July (0.69) comparing to June (0.56)"
```

以下のようにイベント名を定義する必要があります：

```
Load of {HOST.HOST} server increased by {?100*trendavg(/system.cpu.load,1M:now/M)/trendavg(/system.cpu.load,1M:now/M-1M)}
```

また、このような問題のために、trigger 設定において手動で閉じることができるようにすることも有効です。

ヒステリシス (Hysteresis)

問題状態と回復状態の間に単純な閾値ではなく、インターバルが必要な場合があります。
例えば、サーバーームの温度が 20°C 以上になったときに問題を報告する trigger を定義し、温度が 15°C 以下に下がるまで
問題状態を維持したい場合、20°C の単純な trigger 閾値では十分ではありません。

その代わりに、最初に問題が発生した問題事象 (温度が 20°C 以上) に対する trigger 式を定義する必要があります。
次に、回復条件 (温度 15°C 以下) を追加で定義する必要があります。
これは、次のように定義することで行われます。Recovery expression パラメータを定義し、**defining trigger**
を定義します。

この場合、問題の回復は 2 つのステップで行われます。

- まず、問題式 (20°C 以上の温度) は、"FALSE" に評価されなければなりません。
- 次に、リカバリー式 (温度 15°C 以下) を "TRUE" に評価します。

回復式は、問題事象が先に解決された場合にのみ評価されます。

Warning:

リカバリー式が"TRUE" であることだけでは、問題式がまだ"TRUE" である場合に問題を解決することができません！

例 1

サーバールームの温度が高すぎる

問題式:

```
last(/server/temp)>20
```

回復式:

```
last(/server/temp)<=15
```

例 2

空きディスク容量が少なすぎる

問題式: 直近 5 分間の空きディスク容量が 10GB 以下

```
max(/server/vfs.fs.size[/,free],5m)<10G
```

回復式: 問題式: 直近 10 分間の空きディスク容量が 40GB 以上

```
min(/server/vfs.fs.size[/,free],10m)>40G
```

未知のオペランドを持つ式

通常、未知のオペランド (サポートされていないアイテムなど) は、トリガー値をすぐに '不明' にレンダリングします。

ただし、未知のオペランド (サポートされていないアイテム、関数エラー等) が式の評価に認められる場合があります。

- `nodata()` 関数は、参照された項目がサポートされているかどうかに関係なく評価されます。
 - OR および AND を使用した論理式は、未知のオペランドに関係なく、次の 2 つのケースで既知の値に評価できます。
 - ケース 1: "1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ..." は既知の結果 ('1' または "障害") に評価できます。
 - ケース 2: "0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ..." は、既知の結果 ('0' または "OK") に評価できます。
- Zabbix は、サポートされていないアイテムを未知のオペランドとして取得することで、論理式を評価しようとしています。上記の 2 つのケースでは、既知の値が生成されます (それぞれ "障害" または "OK") が、その他すべてのケースでは、トリガーは '不明' と評価されます。
- サポートされているアイテムの関数評価でエラーが発生した場合、関数の値は不明になり、以降の式の評価で不明なオペランドとして使用されます。

上記のように、未知のオペランドは論理式でのみ「消える」場合があることに注意してください。算術式では、未知のオペランドの結果は常に不明になります (0 による除算を除く)。

Attention:

不明になる式は、トリガーの状態 ("障害/OK") を変更しません。したがって、状態が "障害" (ケース 1 を参照) であった場合、既知の部分が解決されても ('1' が '0' になる)、トリガー状態を変更せず同じ障害状態のままになります。

サポートされていないアイテムがいくつかあるトリガー式が不明と評価される場合、フロントエンドのエラーメッセージは、最後に評価されたサポートされていないアイテムを参照します。

3 トリガーの依存関係

概要

あるホストの可用性が他のホストに依存することがあります。ルータの背後にあるサーバは、ルータがダウンすると到達不能になります。
 両方のホストに対して trigger を設定した場合、2 つのホストがダウンしたことを通知されるかもしれませんが - ルーターだけが悪者です。

ここで、ホスト間の依存関係が役に立つかもしれません。依存関係を設定することで、依存関係にあるホストの通知を保留し、
 根本的な問題に対する通知のみを送信することができます。

Zabbix はホスト間の依存関係を直接サポートしていませんが、より柔軟な別の方法である trigger 依存関係で定義することができます。
 trigger は 1 つ以上の trigger を持つことができ、その trigger に依存します。

この例では、server trigger の設定フォームを開き、ルータのそれぞれの trigger に依存するように設定します。
 そのような依存関係では、server の trigger は、依存する trigger が'PROBLEM' 状態である限り、状態を変更しません。
 従って、依存するアクションは取られず、通知も送られません。

server とルータの両方がダウンしており、依存関係が存在する場合、Zabbix は依存する trigger のアクションを実行しません。

依存する trigger のアクションは、依存する trigger が以下の状態である場合、実行されません。

- 状態が'PROBLEM' から'UNKNOWN' に変更された場合。
- 手動で、相関関係で、または時間ベースの関数の助けを借りて、閉じられます。
- 依存する trigger に関与していない item の値によって解決される。
- 無効にされた、無効にされた item がある、または無効にされた item があるホスト

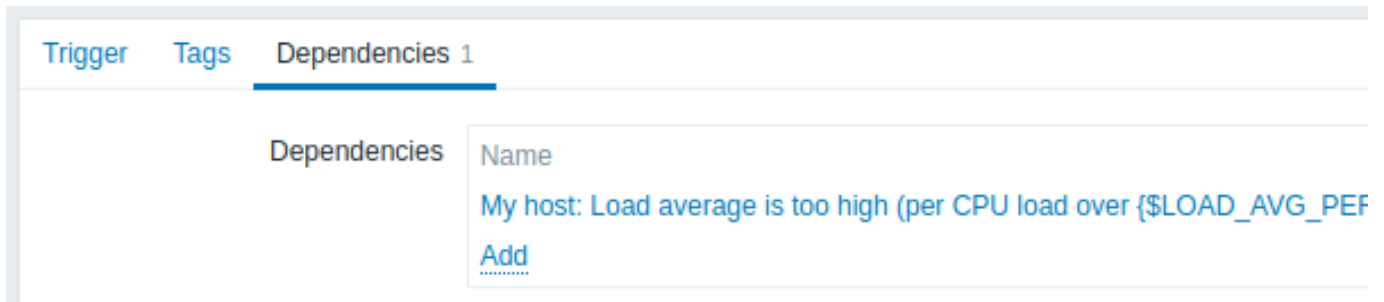
上記のケースの"二次" (従属) トリガは、すぐには更新されないことに注意してください。親 trigger が PROBLEM 状態の間、
 その従属 trigger は、信頼できない値を報告するかもしれません。従って、従属 trigger は、親 trigger が OK 状態にあり、
 信頼できるメトリクスを受け取った後でのみ、再評価され、状態が変更されます。

また

- trigger の依存関係は、循環的な依存関係にならない限り、任意のホスト trigger から他の任意のホスト trigger に追加されるかもしれません。
- trigger の依存関係は、テンプレートからテンプレートに追加されるかもしれません。テンプレート A からの trigger がテンプレート B からの trigger に依存する場合、テンプレート A はテンプレート B とともにホスト (または別のテンプレート) にのみリンクされるかもしれませんが、テンプレート B は単独でホスト (または別のテンプレート) にリンクされるかもしれません。
- テンプレートの trigger からホストの trigger に trigger 依存性を持たせることもできます。この場合、そのようなテンプレートをホストにリンクすると、template trigger が依存していたのと同じ trigger に依存するホスト trigger が作成されます。これは、例えば、いくつかの trigger がルーター (ホスト) trigger に依存するテンプレートを持つことができます。このテンプレートにリンクされたすべてのホストは、その特定のルータに依存します。
- ホスト trigger からテンプレート trigger への trigger 依存は、追加することができません。
- trigger プロトタイプから別の trigger プロトタイプ (同じローレベルディスカバリルール内) または実際の trigger に trigger 依存性を追加することができます。trigger プロトタイプは、別の LLD ルールからの trigger プロトタイプや、trigger プロトタイプから作成された trigger に依存することはできません。ホスト trigger プロトタイプは、テンプレートからの trigger に依存することはできません。

設定

依存関係を定義するには、トリガー **configuration form** で Dependencies タブを開きます。'Dependencies' ブロックの Add をクリックし、trigger が依存する 1 つ以上の trigger を選択します。



Update をクリックします。これで、trigger はリスト内の依存関係を表示するようになりました。

Template Module Linux CPU by Zabbix agent: High CPU utilization (over { \$CPU_UTIL.CRIT }% for 5m)
Depends on:
My host: Load average is too high (per CPU load over { \$LOAD_AVG_PER_CPU.MAX.WARN } for 5m)

複数の依存関係がある場合の例

例えば、あるホストがルータ 2 の背後にあり、ルータ 2 がルータ 1 の背後にある場合です。

Zabbix - ルータ1 - ルータ2 - ホスト

ルータ 1 がダウンした場合、当然ホストとルータ 2 も到達不能になりますが、ホスト、ルータ 1、ルータ 2 がすべてダウンしたという通知を
 3 回受け取りたくはありません。

そこで、この場合、2つの依存関係を定義します。

Host is down' trigger は 'Router2 is down' trigger に依存する。

Router2 is down' trigger は 'Router1 is down' trigger に依存する。

'ホストがダウンしている' trigger のステータスを変更する前に、Zabbix は対応する trigger の依存関係を確認します。もし、見つかった場合、それらの trigger の1つが '問題' 状態であれば、trigger のステータスは変更されず、アクションは実行されず、通知も送信されません。

Zabbix はこのチェックを再帰的に実行します。ルータ 1 またはルータ 2 が到達不能の場合、Host trigger は更新されません。

4 トリガーの深刻度

トリガーの深刻度は、トリガーがどの程度重要であるかを定義します。Zabbix は以下のトリガー重要度をサポートしています:

重要度	定義	色
Not classified	Unknown severity.	Gray
Information	For information purposes.	Light blue
Warning	Be warned.	Yellow
Average	Average problem.	Orange
High	Something important has happened.	Light red
Disaster	Disaster. Financial losses, etc.	Red

深刻度は、以下の用途で使用されます:

- trigger を視覚的に表現するため、重要度によって色を変える。
- グローバルアラームの音声。重要度によって異なる音声
- ユーザーメディア。重要度によって異なるメディア (通知チャンネル) 例えば、SMS - 重大度、電子メール - その他。
- trigger の重大度に対する条件によるアクションの制限

trigger の深刻度の名前と色をカスタマイズすることができます。 [customize trigger severity names and colors](#).

5 トリガー深刻度のカスタマイズ

GUI に関連づけられているトリガーの深刻度の名称および色については、管理 → 一般 → トリガー表示オプションで設定することができます。色はすべての GUI テーマで共有されます。

カスタマイズされた重大度名の翻訳

Attention:

Zabbix フロントエンドの翻訳を使用している場合、カスタマイズした重大度名は、デフォルトで翻訳された名前より優先されます。

デフォルトのトリガー重大度名は、すべてのロケールにおいて翻訳可能です。重大度名を変更した場合、すべてのロケールにおいてカスタム名が使用されるため、手動での追加翻訳が必要です。

カスタム重大度名の翻訳手順

- 必要なカスタム重大度名、例えば 'Important' を設定します。
- `<frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po` を編集します。
- 2行追加:

```
msgid "Important"  
msgstr "<翻訳する文>"
```

上記編集をしてファイルを保存してください。

- `<frontend_dir>/locale/README` で説明されているように .mo ファイルを作成します。

ここで **msgid** は新しいカスタム重大度名と一致し、**msgstr** は特定の言語での翻訳でなければなりません。

この手順は重大度名を変更するたびに実行する必要があります。

6 一括アップデート

概要

一括アップデートでは、一度に多くの trigger の属性を変更することができ、編集のために個々の trigger を開く手間を省くことができます。

一括アップデートの利用

いくつかの trigger を一括でアップデートするには、以下のようにします。

- リスト内の更新したい trigger のチェックボックスにマークを付けます。
- リストの下にある Mass update をクリックします。
- 必要な属性 (Trigger、Tags、Dependencies) のあるタブに移動します。
- 更新する属性のチェックボックスにマークを付けます。

Mass update

Trigger **Tags** Dependencies

Severity Not classified Information Warning Average **High** Disaster

Allow manual close Original

Mass update

Trigger **Tags** Dependencies

Tags Add Replace Remove

Name	Value
<input type="text" value="tag"/>	<input type="text" value="value"/>

[Add](#)

タグ更新の各ボタンを選択すると、以下のオプションが利用できます。

- Add - トリガーに新しいタグを追加することができます。
- Replace - トリガーから既存のタグを削除し、以下で指定するタグに置き換えます。
- Remove - 指定したタグをトリガーから削除します。

同じ名前で異なる値を持つタグは、'重複'とはみなされず、同じトリガーに追加することができることに注意してください。

Mass update

Trigger Tags **Dependencies**

Replace dependencies Name
Zabbix server: Lack of available memory (< 20M of 7.72 GB)

[Add](#)

Replace dependencies - トリガーから既存の依存関係を削除し、指定されたものに置き換えます。

変更を適用するには、Update をクリックしてください。

7 予兆 trigger 関数

概要

時には、問題が発生する予兆があることがあります。これらの兆候を発見することで、事前に対策を講じ、問題の影響を防ぐ、
または少なくとも最小限に抑えることができます。

Zabbix は history データに基づいて監視システムの将来の挙動を予測するツールを備えています。
これらのツールは予兆トリガー機能で実現されます。

関数

問題状態を定義する方法と、対策を講じるのに必要な時間の 2 つを知る必要があります。それから、潜在的な望ましくない状況について
信号を送る trigger をセットアップする方法が 2 つあります。
1 番目に、trigger は、“time to act” の後にシステムが問題状態になると予想されるときに発火しなければなりません。
2 番目は、システムが“time to act” よりも短い時間で問題状態に到達しそうなときに、trigger が発火することです。
対応する trigger 関数は、**forecast** と **timeleft** です。基本的な統計解析は両関数とも基本的に同じであることに注意してください。
好きな方法で trigger をセットアップして、同様の結果を得ることができます。

1 パラメータ

両関数とも、ほぼ同じパラメータを使用します。[supported functions](#)を参考にしてください。

時間間隔

まず最初に、Zabbix が予測値を算出するために分析する期間を指定する必要があります。
これは、**avg**、**count**、**delta**、**max**、**min**、**sum** 関数と同様に、time period パラメータとオプションの
タイムシフトによって行うことが可能です。

予測の水平線

(**forecast** のみ) パラメータ time は、Zabbix が history データから発見した依存関係を、どの程度未来に推定するかを指定します。
time_shift を使用するかどうかに関係なく、time は常に現在の瞬間からカウントされます。

1.1 到達する閾値

(**timeleft** のみ) パラメータ threshold は、分析対象が到達すべき値を指定します。上からでも下からでも違いはありません。
f(t) が決まったら (後述)、方程式 $f(t) = \text{threshold}$ を解いて、今より右側にあるルート、またはそのようなルートがない場合は
999999999.9999 を返さなければなりません。

Note:

item の値がしきい値に近づき、それを越えた場合、**timeleft** はその交点がすでに過去のものであると仮定し、
もしあれば threshold レベルで次の交点に切り替わります。ベストプラクティスは、予測を通常の問題診断の補完として
使うことであり、代替としては使わないことです^a。

^aHttpOnly が 'true' の場合、Cookie は HTTP プロトコルを介してのみアクセス可能になります。これは、JavaScript などのスクリプト言語から Cookie にアクセスできないことを意味します。この設定は、XSS 攻撃による個人情報の盗難を効果的に減らすのに役立ちます (ただし、すべてのブラウザでサポートされているわけではありません)

1.2 fit 関数

デフォルトの fit は linear 関数です。しかし、監視するシステムがより複雑な場合は、より多くのオプションから選択することができます。

fit	$x = f(t)$
linear	$x = a + b*t$
polynomialN ¹	$x = a_0 + a_1*t + a_2*t^2 + \dots + a_n*t^n$
exponential	$x = a*\exp(b*t)$
logarithmic	$x = a + b*\log(t)$
power	$x = a*t^b$

1.3 モード

(**forecast** のみ) trigger 関数は評価されるたびに、指定された history 期間からデータを取得し、そのデータに指定された
関数を当てはめるものです。そのため、データが少し違えば、フィットする関数も少し違ってきます。単純に、もし、将来の
指定された時間にフィットした関数の値を計算するだけなら、分析された item が現在から将来のその時点までの間に
どのように動く

¹Secure は、Cookie がクライアントからの安全な HTTPS 接続を介してのみ送信される必要があることを示します。'true' に設定すると、Cookie は安全な接続が存在する場合にのみ設定されます。

作すると予想されるかについて何も知らないこととなります。
 いくつかの fit オプション (polynomial など) では、未来からの単純な値は誤解を招くかもしれません。

モード	forecast 結果
value	$f(\text{now} + \text{time})$
max	$\max_{\text{now} \leq t \leq \text{now} + \text{time}} f(t)$
min	$\min_{\text{now} \leq t \leq \text{now} + \text{time}} f(t)$
delta	max - min
avg	average of $f(t)$ ($\text{now} \leq t \leq \text{now} + \text{time}$) according to definition

詳細

膨大な数の計算を避けるために、指定された周期の最初の値のタイムスタンプに 1 ns を加えたものを新しいゼロ時刻と見なします。
 (現在のエポックタイムは 10^9 のオーダー、エポック 2 乗は 10^{18} 、倍精度は約 10^{-16})
 1 ns は、 $\log(t)$ の計算を含む logarithmic および power fit のためのすべての正の時間値を提供するために追加されます。
 タイムシフトは、linear, polynomial, exponential* には影響しませんが、logarithmic と power 関数の形状を変更します。
 (計算がより簡単で正確になることは別として)

潜在的なエラー

このような場合、関数は -1 を返します。

- 指定された評価期間内にデータがない。
- 算術演算の結果が定義されていない²
- 数値の複雑さ (残念ながら、入力データによっては倍精度浮動小数点フォーマットの範囲と精度が不十分となる)³があります。

Note:

選択された fit が提供されたデータをうまく説明できない場合や、正確な予測のためにデータが少なすぎる場合、
 警告やエラーは表示されません。

例とエラーの処理

ホストの空きディスク容量がなくなりそうときに警告を表示させるには、次のような trigger 式を使用することができます:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

しかし、エラーコード-1 が出て、trigger が問題状態になることがあります。一般的には、予測が正しく動作しておらず、
 その原因を見つけるためにもっと徹底的に調べるべきだという警告を得ることができるので、これは良いことです。
 しかし、時には、-1 が単に過去 1 時間に取得されたホストの空きディスク容量に関するデータがなかったことを意味することもあり、
 悪いこともあります。もし、あまりにも多くの偽陽性 (false positive) アラートが出るようであれば、より複雑なトリガー式を
 使用することを検討してください。⁴:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)>-1
```

この状況は、**forecast** では少し難しくなります。まず第一に、-1 は、`forecast(/host/item,...)<...` のような式か
 `forecast(/host/item,...)>...` のようにするかによって trigger を問題状態に置かずか置かないか決まります。

さらに、item の値が負であることが普通であれば、-1 は有効な予測であるかもしれません。しかし、現実の状況でこのような状況に
 なる確率はごくわずかです (**how the operator = works** を参照してください)
 そこで、-1 を問題として扱いたい、あるいは扱いたくない場合は、それぞれ `...or forecast(/host/item,...)=-1` または
 `...and forecast(/host/item,...)>-1` を追加してください。

4 イベント

概要

Zabbix で生成されるイベントにはいくつかの種類があります。

²例えば、指数関数や 冪乗関数のフィッティングでは、項目値の $\log()$ を計算します。 $\log()$ は正の値に対してのみ定義されているので、データにゼロや負の数が含まれているとエラーになります。

³線形フィット、指数フィット、対数フィット、パワーフィットの場合、必要な計算をすべて明示的に記述することができます。多項式フィットでは、追加のステップなしに value だけが計算できます。avg の計算には、多項式の反次導関数を (解析的に) 計算する必要があります。max, min, delta の計算には、多項式微分の計算 (解析的) とその根の探索 (数値的) が含まれます。 $f(t) = 0$ を解くには、多項式の根を (数値的に) 求めます。

⁴しかし、この場合、-1 はトリガーが問題状態から回復する原因となることがあります。完全に保護するために使用します。 `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)>-1 or {TRIGGER.VALUE}=1)`

- trigger イベント - trigger のステータスが変化する (OK→PROBLEM→OK) たびに発生するイベントです。
- サービスイベント - サービスのステータスが変更された時 (OK→PROBLEM→OK) に発生するイベントです。
- 発見イベント - ホストやサービスが検出されたときに発生するイベントです。
- 自動登録イベント - アクティブな agent が server によって自動登録されたときに発生するイベントです。
- 内部イベント - item / ローレベルディスクバリエーションがサポートされなくなったとき、または trigger が未知の状態になったときに発生するイベントです。

Note:

内部イベントは Zabbix2.2 バージョンからサポートされています。

イベントはタイムスタンプを持ち、通知メールの送信などのアクションの基礎となります。

フロントエンドでイベントの詳細を表示するには、Monitoring → Problems をご覧ください。この画面でイベントの日時をクリックすると、イベントの詳細が表示されます。

より詳細な情報は、以下の通りです。

- [trigger events](#)
- [other event sources](#)

1 トリガーイベントの生成

概要

trigger の状態変化は、最も頻繁で最も重要なイベントの発生源です。trigger がその状態を変更するたびに、イベントが生成されます。このイベントには、trigger の状態の変化の詳細が含まれます。イベントには、trigger の状態の変化の詳細、つまり、いつ起こったのか、新しい状態が何なのかが含まれます。

trigger によって 2 種類のイベントが生成されます - Problem と OK です。

問題発生イベント

次の場合に、問題イベントが作成されます：

- trigger が OK 状態の場合、トリガ式が TRUE と評価されたとき。
- 複数の問題イベント生成が trigger で有効になっている場合、trigger 式が TRUE に評価されるごとに。

OK イベント

OK イベントは、関連する問題イベントを終了させるもので、3つのコンポーネントで作成されます。

- トリガー - 'OK イベント生成' と 'OK イベントクローズ' の設定に基づきます。
- イベントの相関関係が生まれたとき。
- タスクマネージャー - イベントが **manually closed** されたとき。

トリガー

トリガーには、OK イベントの生成方法を制御する 'OK イベント生成' 設定があります。

- Expression - 問題のある状態のトリガーに対して、その式が FALSE と評価されると OK イベントが生成されます。これは、最も単純な設定で、デフォルトで有効になっています。
- Recovery expression - OK イベントは、その式が FALSE と評価され、回復式が TRUE と評価されたとき、問題状態のトリガーに対して生成されます。これは、トリガーの回復基準が問題基準と異なる場合に使用できます。
- None - OK イベントは、決して生成されません。これは、複数の問題イベント生成と組み合わせて、何かが起こったときに単に通知を送るために使用することができます。

さらに、トリガーには 'OK イベントクローズ' の設定があり、どの問題イベントがクローズされるかを制御します。

- All problems - OK イベントは、トリガーによって作成されたすべての未解決問題を閉じます。
- All problems if tag values match - OK イベントは、トリガーによって作成され、少なくとも 1 つの一致するタグ値を持つ未解決の問題を閉じます。タグは、'Tag for matching' トリガ設定によって定義されます。閉じるべき問題イベントがない場合、OK イベントは生成されません。これは、トリガレベルイベント相関と呼ばれます。

イベント相関

イベント相関 (グローバルイベント相関とも呼ばれる) は、カスタムイベント終了 (結果的に OK イベントを生成する) ルールを設定する方法です。

このルールでは、新しい問題イベントと既存の問題イベントの組み合わせ方法を定義し、対応する OK イベントを生成して、新しいイベントまたは一致したイベントを閉じることができます。

しかし、イベント相関は、イベント処理のパフォーマンスに悪影響を及ぼしたり、設定を誤ると意図したよりも多くのイベントを
 クローズしてしまう（最悪の場合、すべての問題イベントをクローズしてしまう）ため、非常に慎重に設定する必要があります。
 設定のヒントをいくつか紹介します。

1. コントロールイベント（古いイベントと対になるイベント）に一意のタグを設定し、'新しいイベントタグ' 相関条件を使用することで、常に相関範囲を縮小します。
2. 'close old event' オペレーションを使用する際に、古いイベントに基づいた条件を追加することを忘れないでください。
3. 異なる相関設定によって使用される共通のタグ名を使用しないようにします。

タスクマネージャー

トリガーに対して'Allow manual close(手動で閉じることを許可する)' 設定が有効になっている場合、トリガーによって生成された
 問題イベントを手動で閉じることが可能です。これは、フロントエンドで **updating a problem** を行う際に行われます。
 イベントは直接的には閉じられません。代わりに'close event(イベントを閉じる)' タスクが作成され、まもなくタスクマネージャーに
 よって処理されます。タスクマネージャーは対応する OK イベントを生成し、問題イベントは閉じられます。

2 その他のイベントソース

サービスイベント

サービスイベントは、これらのイベントに対するサービスアクションが有効な場合にのみ生成されます。
 この場合、サービスステータスの変更ごとに新しいイベントを生成します。

- 問題イベント - サービスステータスが OK から PROBLEM に変更されたとき
- OK イベント - サービスステータスが PROBLEM から OK に変更されたとき

このイベントには、サービス状態の変化の詳細（いつ起こったか、新しい状態は何か）が含まれています。

ディスカバリイベント

Zabbix は、ネットワークディスカバリルールで定義された IP 範囲を定期的にスキャンします。チェックの頻度は、ルールごとに個別に設定できます。ホストまたはサービスが検出されると、ディスカバリイベント（または複数のイベント）が生成されます。

Zabbix は次のイベントを生成します。

イベント	発生条件
Service Up	Zabbix がアクティブなサービスを検出するたびに
Service Down	Zabbix がサービスを検出できないたびに
Host Up	少なくとも 1 つのサービスが IP に対して稼働している場合
Host Down	すべてのサービスが応答していない場合
Service Discovered	サービスがダウンタイム後に戻った場合、または初めて検出された場合
Service Lost	起動後にサービスが失われた場合
Host Discovered	ホストがダウンタイム後に復帰した場合、または初めて検出された場合
Host Lost	起動後にホストが失われた場合

アクティブ agent 自動検出イベント

アクティブ agent の自動登録は Zabbix にイベントを作成します。

設定されている場合、以前は知られていなかったアクティブ agent がチェックを要求した場合、またはホストのメタデータが
 変更された場合、アクティブ agent の自動登録イベントが作成されます。server は、受信した agent の IP アドレスとポートを
 使用して、新しい自動登録されたホストを追加します。

詳細については、 [active agent autoregistration](#) ページを参照してください。

内部イベント

内部イベントは、次のような場合に発生します。

- アイテムの状態が'normal' から'unsupported' に変わったとき
- 項目が'unsupported' から'normal' に状態を変更したとき
- ローレベルディスカバリルールが'normal' から'unsupported' に状態を変更したとき
- ローレベルディスカバリルールが状態を'unsupported' から'normal' に変更したとき
- トリガーの状態が'normal' から'unsupported' に変更されたとき
- トリガーの状態が'unsupported' から'normal' に変更されたとき

Zabbix 2.2 以降、内部イベントがサポートされています。内部イベントを導入する目的は、アイテムがサポートされなくなり、データの収集が停止するなど、内部イベントが発生したときにユーザーに通知できるようにすることです。

内部イベントは、これらのイベントの内部アクションが有効になっている場合にのみ作成されます。内部イベントの生成を停止するには (アイテムがサポートされなくなるなど)、設定 → アクション → 内部イベントアクションで内部イベントのすべてのアクションを無効にします。

Note:

内部アクションが無効になっている場合、オブジェクトが '未対応' 状態であっても、このオブジェクトの回復イベントは引き続き作成されます。
オブジェクトが '未対応' 状態にある間に内部アクションが有効になっている場合、オブジェクトに対して '障害イベント' が作成されていなくても、このオブジェクトの回復イベントが作成されます。

参考:【非対応品のお知らせを受け取る】 (/manual/config/notifications/unsupported_item)

3 問題の手動クローズ

概要

一般的に問題イベントは、トリガーのステータスが 'Problem' から 'OK' になったときに自動的に解決されますが、
 トリガー式によって問題が解決されたかどうかを判断することが困難な場合もあります。そのような場合、手動で問題を
 解決する必要があります。

例えば、syslog は、いくつかのカーネルパラメータが最適なパフォーマンスのために調整される必要があることを
 報告するかもしれませんが、この場合、問題は Linux 管理者に報告され、管理者は問題を修正した後、問題を手動で解決します。

問題を手動で閉じることができるのは、手動クローズを許可オプションが有効になっているトリガーだけです。

問題が "手動で閉じる" 場合、Zabbix は Zabbix server の新しい内部タスクを生成します。そして、タスクマネージャプロセスが
 このタスクを実行し、OK イベントを生成し、問題イベントを終了します。

手動で問題を閉じても、基礎となるトリガーが二度と '問題' 状態にならないわけではありません。トリガー式は再評価され、
 問題が発生する可能性があります:

- トリガ式に含まれるいずれかの項目について、新しいデータが到着したとき (スロットリングの前処理ステップによって破棄された値は、受信したとはみなされず、トリガ式の再評価の原因にならないことに注意する)
- 時間ベースの関数が式で使用されている場合。タイムベース関数の完全なリストは、**トリガー** で見ることができます。

設定

手動で問題を閉じるには、2つのステップが必要です。

トリガー設定

トリガー設定で、手動クローズを許可オプションを有効にします。

Allow manual close

問題の更新ウィンドウ

手動クローズフラグの付いたトリガーに問題が発生した場合、**問題の認識**ポップアップウィンドウを開いて、
 手動で問題を閉じることができます。

問題を閉じるには、フォームの Close problem (障害のクローズ) オプションをチェックし、Update (更新) をクリックします。

Update problem ✕

Message

History

Scope

Only selected problem

Selected and all other problems of related triggers 1 event

Change severity Not classified Information Warning Average High Disaster

Acknowledge

Close problem

* At least one update operation or message must exist.

すべての必須入力項目には赤いアスタリスクが表示されます。

リクエストは Zabbix サーバによって処理されます。通常、問題を解決するのに数秒かかります。この間、問題のステータスとして
 監視データ → 障害に 解決済が表示されます。

確認

問題が手動で解決されたことを確認することができます。:

- 監視データ → 障害からイベントの詳細で確認できます。
- 通知メッセージの {EVENT.UPDATE.HISTORY} マクロを使用して、この情報を提供します。

5 イベント相関

概要

イベント相関は、問題発生とその解決の相関を、非常に正確かつ柔軟に行うことができる機能です。

イベントの相関関係を以下のように定義することができます。

- [on trigger level]((/manual/config/event_correlation/trigger) - 1 つの trigger を使って、別々の問題を解決に関連付けることができます。
- **globally** - グローバル相関ルールを使用して、異なる trigger / ポーリングメソッドから問題を解決に関連させることができます。

1 トリガーによるイベント相関

概要

トリガーベースのイベント相関では、1 つのトリガーによって報告された別々の問題を相関させることができます。

一般的に OK イベントは、1 つのトリガーによって作成された全ての問題イベントを閉じることができますが、より詳細な
 アプローチが必要な場合もあります。例えば、ログファイルを監視しているときに、ログファイル内の特定の問題を発見し、
 まとめて閉じるのではなく、個別に閉じたい場合があります。

これは、Multiple Problem Event Generation が有効になっているトリガーがそうです。このようなトリガーは通常、
 ログ監視、トラップ処理などに使用されます。

Zabbix では、tagging に基づいて問題事象を関連付けることが可能です。タグは問題事象の値を抽出し、
 識別を行うために使用されます。これを利用して、一致するタグに基づいて問題を個別にクローズすることも可能です。

言い換えれば、同じトリガーが、イベントタグによって識別される別のイベントを作成することができます。したがって、
 問題イベントは、イベントタグによる識別に基づいて、1 つ 1 つ識別され、別々に閉じることができます。

仕組み

ログモニタリングでは、以下のような行が表示されることがあります：

```
Line1: Application 1 stopped
Line2: Application 2 stopped
Line3: Application 1 was restarted
Line4: Application 2 was restarted
```

イベント関連の考え方は、Line1 からの問題イベントと Line3 からの解決、Line2 からの問題イベントと Line4 からの解決を
 一致させ、これらの問題を 1 つずつ解決することができることです：

```
Line1: Application 1 stopped
Line3: Application 1 was restarted #problem from Line 1 closed
```

```
Line2: Application 2 stopped
Line4: Application 2 was restarted #problem from Line 2 closed
```

そのためには、これらの関連するイベントに、例えば"アプリケーション 1" や"アプリケーション 2" のようなタグを付ける
 必要があります。これは、ログラインに正規表現を適用して、タグの値を抽出することで実現できます。その後、イベントが
 作成されると、それぞれ"アプリケーション 1" と"アプリケーション 2" というタグが付けられ、問題を解決するために
 照合することができます。

設定

item

始めに、例えばログファイルを監視するような項目を設定するのがよいでしょう：

```
log[/var/log/syslog]
```

Item	Tags	Preprocessing
* Name		Syslog
Type		Zabbix agent (active) ▾
* Key		log[/var/log/syslog]
Type of information		Text ▾
* Update interval		30s

item が設定された状態で、設定変更が反映されるまで 1 分ほど待ってから、Latest data にアクセスして、
 item がデータ収集を開始したことを確認します。

トリガ

item が動作している状態で、以下の設定を行う必要があります。trigger を設定します。
 ログファイルのどのエントリに注意を払うべきかを定めることが重要です。例えば、次のトリガー式は、潜在的な問題を
 知らせるために、'Stopping' のような文字列を検索します。

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:

"Stopping" という文字列を含む各行が問題とみなされるようにするには、トリガー設定の Problem event generation mode を
 'Multiple' に設定します。

次に、リカバリー式を定義します。次のリカバリー式は、文字列"Starting" を含むログ行が見つかった場合、すべての問題を
 解決します:

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

そうならないためには、すべての問題を解決するのではなく、対応する根本的な問題を解決するようにすることが重要です。
 そこで役立つのがタグ付けです。

トリガー設定でタグを指定することにより、問題と解決策を一致させることができます。以下の設定を行う必要があります:

- Problem event generation mode: Multiple
- OK event closes: タグの値が一致した場合のすべての問題
- イベントマッチングのためのタグの名前を入力します

Trigger	Tags	Dependencies
* Name	Service {{ITEM.VALUE}}.regex("^. * service ([a-zA-Z]*) .*\$", "\1")} stopped	
Event name	Service {{ITEM.VALUE}}.regex("^. * service ([a-zA-Z]*) .*\$", "\1")} stopped	
Operational data		
Severity	<input type="checkbox"/> Not classified <input type="checkbox"/> Information <input type="checkbox"/> Warning <input type="checkbox"/> Average <input checked="" type="checkbox"/> High <input type="checkbox"/> Disas	
* Problem expression	<pre>find(/My host/log[/var /log/syslog],,"regexp","Stopping")=1</pre>	<input type="button" value="Add"/>
	Expression constructor	
OK event generation	<input type="checkbox"/> Expression <input checked="" type="checkbox"/> Recovery expression <input type="checkbox"/> None	
* Recovery expression	<pre>find(/My host/log[/var /log/syslog],,"regexp","Starting")=1</pre>	<input type="button" value="Add"/>
	Expression constructor	
PROBLEM event generation mode	<input type="checkbox"/> Single <input checked="" type="checkbox"/> Multiple	
OK event closes	<input type="checkbox"/> All problems <input checked="" type="checkbox"/> All problems if tag values match	
* Tag for matching	Service	

- tagsを設定し、ログラインからタグの値を抽出する。

Trigger Tags 2 Dependencies

Trigger tags Inherited and trigger tags

Name	Value
Datcenter	value
Service	{{ITEM.VALUE}.regex("^.* service ([a-zA-Z]*) .*\$", "1")}

Add

正常に構成された場合、アプリケーションごとにタグ付けされた問題イベントが、Monitoring → Problems に表示され、
 解決に結びつきます。

Problems Export to CSV Filter

Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
15:28:13	High	15:28:25	RESOLVED	Zabbix server	Service Apache stopped		12s	No		Service: Apache Webservers

Warning:

設定ミスの可能性があるので、関係ない問題で似たようなイベントタグが作成される場合、以下のケースを確認してください。

- 2つのアプリケーションが同じログファイルにエラーとリカバリメッセージを書いている場合、ユーザーは、{ITEM.VALUE} マクロからアプリケーション A とアプリケーション B の名前を抽出するためにタグ値で別々の正規表現を使用することで、同じトリガーで異なるタグ値で2つのアプリケーションタグを使うことを決定することができます。(例:メッセージフォーマットが異なる場合)しかし、正規表現にマッチするものがない場合、予定通りには動作しないことがあります。正規表現にマッチしない場合、空のタグ値が生成され、問題と OK イベントの両方で1つの空のタグ値があれば、それらを関連付けるのに十分です。したがって、アプリケーション A からの回復メッセージは、アプリケーション B からのエラーメッセージを誤って閉じてしまう可能性があります。
- 実際のタグとタグの値は、トリガーが起動したときのみ表示されます。使用された正規表現が無効な場合、それは黙って *UNKNOWN* 文字列に置き換えられます。タグの値を持つ最初の問題イベントを見逃すと、同じタグの値を持つ後続の OK イベントが現れ、本来閉じられるはずのない問題イベントが閉じられることがあります。
- タグの値として、マクロ関数を使用しない {ITEM.VALUE} をタグの値として使用する場合、255 文字の制限が適用されます。ログメッセージが長く、最初の 255 文字が非特異的である場合、これはまた、無関係な問題のための類似のイベントタグになる可能性があります。

2 グローバルイベント 関連

概要

グローバルイベント 関連は、Zabbix が監視するすべてのメトリクスにアクセスし、関連を作成することができます。

全く異なるトリガーで作成されたイベントを関連させ、それら全てに同じ操作を適用することが可能です。
 インテリジェントな関連ルールを作成することで、何度も繰り返される通知から解放され、問題の根本的な原因に焦点を
 当てることができます。何千もの繰り返しの通知から解放され、問題の根本的な原因に集中することができます。

グローバルイベント 関連は、1つのトリガーに基づく問題解決ロジックから自分を解き放つことができる、強力なメカニズムです。
 これまでは、1つの問題事象が1つのトリガーで作成され、問題解決のためにその同じトリガーに依存していました。
 あるトリガーで発生した問題を、別のトリガーで解決することはできませんでした。しかし、イベント・タグ付けに基づく
 イベント 関連を利用すれば、それが可能になります。

例えば、ログトリガーはアプリケーションの問題を報告し、ポーリングトリガーはアプリケーションが稼働していることを
 報告するかもしれません。イベントタグを利用して、ログトリガーを Status: Down とタグ付けすることができます。
 その間、ポーリングトリガーを Status: Up とタグ付けすることができます。次に、グローバル 関連ルールで、これらのトリガーを
 関連付け、古いイベントを閉じるなど、この 関連に適切な操作を割り当てることができます。

別の使い方として、グローバル 関連は類似のトリガーを識別し、同じ操作を適用することができます。
 ネットワーク・ポートの問題ごとに1つの問題レポートしか得られないとしたらどうでしょう。全部を報告する必要はありません。
 グローバルなイベ

ント相関を使えば、それも可能です。

グローバルイベントの相関は、**correlation rules** で設定されます。correlation rules では、新しい問題イベントを既存の
 問題イベントとどのように組み合わせ、一致した場合にどうするかを定義します（新しいイベントを閉じ、対応する OK イベントを
 生成して一致した古いイベントを閉じます）。グローバル相関によって問題がクローズされると、Monitoring → Problems の Info カラムに報告されます。

グローバル相関ルールの設定は、Super Admin レベルのユーザーのみ可能です。

Attention:

イベント相関は非常に慎重に設定する必要があります。イベント処理のパフォーマンスに悪影響を及ぼしたり、設定を誤ると
 意図した以上のイベントがクローズされる可能性があるからです。（最悪の場合、すべての問題イベントがクローズされる
 可能性もあります）

グローバル相関を 安全に設定するために、次の重要なヒントを守ってください:

- 相関範囲を小さくする。古いイベントと対になる新しいイベントには常に一意のタグを設定し、New event tag の相関条件を使用します。
- 古いイベントを閉じる * 操作を使用する場合は、古いイベントに基づく条件を追加します（さもなければ、既存のすべての問題が閉じられる可能性があります）。
- 異なる相関設定によって使用される可能性のある共通のタグ名を使用しないようにします。
- 相関ルールの数は、本当に必要なものに限定してください。

参照: [known issues](#).

設定

イベント相関ルールをグローバルに設定するには

- Configuration → Event correlation に進みます。
- 右側の Create correlation をクリックします（または、既存のルールを編集するために相関名をクリックします）
- 相関ルールのパラメータをフォームに入力します。

* Name	Close old events										
Type of calculation	And ▼ A and (B and C) and D										
* Conditions	<table border="1"><thead><tr><th>Label</th><th>Name</th></tr></thead><tbody><tr><td>A</td><td>Value of old event tag <i>Application</i> equals value of new event tag <i>Application</i></td></tr><tr><td>B</td><td>Value of old event tag <i>Application</i> equals ABC</td></tr><tr><td>C</td><td>Value of old event tag <i>State</i> equals Down</td></tr><tr><td>D</td><td>Value of new event tag <i>State</i> equals Up</td></tr></tbody></table> Add	Label	Name	A	Value of old event tag <i>Application</i> equals value of new event tag <i>Application</i>	B	Value of old event tag <i>Application</i> equals ABC	C	Value of old event tag <i>State</i> equals Down	D	Value of new event tag <i>State</i> equals Up
Label	Name										
A	Value of old event tag <i>Application</i> equals value of new event tag <i>Application</i>										
B	Value of old event tag <i>Application</i> equals ABC										
C	Value of old event tag <i>State</i> equals Down										
D	Value of new event tag <i>State</i> equals Up										
Description	Close old events for application ABC if an event with State=Up happens.										
Operations	<input checked="" type="checkbox"/> Close old events <input type="checkbox"/> Close new event										
	* At least one operation must be selected.										
Enabled	<input checked="" type="checkbox"/>										

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
Name	Unique correlation rule name.
Type of calculation	The following options of calculating conditions are available: And - all conditions must be met Or - enough if one condition is met And/Or - AND with different condition types and OR with the same condition type Custom expression - a user-defined calculation formula for evaluating action conditions. It must include all conditions (represented as uppercase letters A, B, C, ...) and may include spaces, tabs, brackets (), and (case sensitive), or (case sensitive), not (case sensitive).
Conditions	List of conditions. See below for details on configuring a condition.
Description	Correlation rule description.
Operations	Mark the checkbox of the operation to perform when event is correlated. The following operations are available: Close old events - close old events when a new event happens. Always add a condition based on the old event when using the Close old events operation or all existing problems could be closed. Close new event - close the new event when it happens
Enabled	If you mark this checkbox, the correlation rule will be enabled.

新しい条件の詳細を設定するには、条件ブロックの [Add](#) をクリックします。
 ポップアップウィンドウが開き、条件の詳細を編集することができます。

New condition ✕

Type

Tag

Operator equals does not equal contains does not contain

Value

Add
Cancel

パラメータ	説明
New condition	<p>Select a condition for correlating events. Note that if no old event condition is specified, all old events may be matched and closed. Similarly if no new event condition is specified, all new events may be matched and closed. The following conditions are available:</p> <p>Old event tag - specify the old event tag for matching. New event tag - specify the new event tag for matching. New event host group - specify the new event host group for matching. Event tag pair - specify new event tag and old event tag for matching. In this case there will be a match if the values of the tags in both events match. Tag names need not match. This option is useful for matching runtime values, which may not be known at the time of configuration (see also Example 1). Old event tag value - specify the old event tag name and value for matching, using the following operators: equals - has the old event tag value does not equal - does not have the old event tag value contains - has the string in the old event tag value does not contain - does not have the string in the old event tag value New event tag value - specify the new event tag name and value for matching, using the following operators: equals - has the new event tag value does not equal - does not have the new event tag value contains - has the string in the new event tag value does not contain - does not have the string in the new event tag value</p>

Warning:

設定ミスの可能性があるため、関係ない問題で似たようなイベントタグが作成される場合
 以下のケースを確認してください。

- 実際のタグとタグの値は、トリガーが起動したときにのみ表示されます。使用された正規表現が無効な場合、それは黙って *UNKNOWN* 文字列に置き換えられます。タグの値を持つ最初の問題イベントを見逃すと、同じタグの値を持つ後続の OK イベントが現れ、本来閉じられるべきでない問題イベントを閉じることがあります。
- タグの値として、マクロ関数を使用しない {ITEM.VALUE} をタグの値として使用する場合、255 文字の制限が適用されます。ログメッセージが長く、最初の 255 文字が非特異的である場合、これはまた、無関係な問題のための類似のイベントタグになる可能性があります。

例 1

同じネットワークポートから繰り返し発生する問題事象を停止します。

* Name

Type of calculation A and B

* Conditions

Label	Name
A	Value of old event tag <i>Port</i> equals value of new event tag <i>Port</i>
B	Value of old event tag <i>Host</i> equals value of new event tag <i>Host</i>
Add	

Description

Operations Close old events
 Close new event

* At least one operation must be selected.

Enabled

このグローバルな相関ルールは、Host と Port タグ値がトリガに存在し、それらが元のイベントと新しいイベントで
 同じであれば、問題を相関させます。

この操作は、同じネットワークポート上の新しい問題イベントを閉じ、元の問題のみを開いたままにします。

6 タグ付け

概要

Zabbix には様々なエンティティにタグを付けることができるオプションがあります。タグは以下のものに定義することができます:

- テンプレート
- ホスト
- item
- web シナリオ
- トリガー
- サービス
- テンプレート items および triggers
- ホスト, item と trigger プロトタイプ

タグにはいくつかの用途がありますが、特にイベントをマークするのに適しています。
 もしエンティティにタグが付けられたら、対応する新しいイベントもそれに応じてマークされます:

- タグ付けされたテンプレート - このテンプレートの関連エンティティ (item、トリガーなど) によって作成されたすべてのホストの問題がマークされます。
- タグ付きホスト - ホストのあらゆる問題がマークされます。
- タグ付きアイテム、Web シナリオ - この item、Web シナリオのあらゆるデータ/問題がマークされます。

- タグ付きトリガー - このトリガーのすべての問題がマークされます。

問題のあるイベントは、テンプレート、ホスト、item、Web シナリオ、トリガーのチェーン全体からすべてのタグを引き継ぎます。

 完全に同一の tag:value の組み合わせ（マクロを解決した後）は、イベントをマークする際に重複するのではなく、
 1 つにマージされます。

カスタムイベントタグを持つことで、より柔軟な対応が可能になります。重要なのは、イベントタグに基づいてイベントを
 correlated することができることです。他の用途では、タグ付けされたイベントに基づいて
 アクションを定義することができます。item の問題は、タグに基づいてグループ化することができます。
 問題タグは、問題を **services** にマップするために使用することもできます。

タグ付けは、タグ名と値のペアとして実現されます。名前だけを使うことも、値と対にすることもできます。

MySQL, Service:MySQL, Services, Services:Customer, Applications, Application:Java, Priority:High

エンティティ（テンプレート、ホスト、item、Web シナリオ、トリガー、イベント）は、同じ名前異なる値のタグを
 付けることができますが、これらのタグは '重複したもの' とはみなされません。同様に、値のないタグと値のある同じタグを
 同時に使用することができます。

使用例

この機能の使用例としては、以下のようなものがあります：

1. フロントエンドでトリガーイベントをマーク
 - トリガーレベルでタグを定義します。
 - Monitoring → Problems で、すべてのトリガー問題がこれらのタグでどのようにマークされるかを確認します。
2. テンプレートに継承されるすべての問題をマーク
 - テンプレートレベルでタグを定義します。（例：'App=MySQL'）
 - このテンプレートからトリガーによって作成されたホスト問題が、どのように Monitoring → Problems でこれらのタグでマークされるかを確認します。
3. すべてのホストの問題をマーク
 - ホストレベルで、例えば 'Service=JIRA' のようなタグを定義します。
 - Monitoring → Problems で、ホストトリガーのすべての問題がこれらのタグでどのようにマークされるかを確認します。
4. 関連するアイテムのグループ化
 - 例えば 'MySQL' のように、アイテムレベルでタグを定義します。
 - タグフィルターを使用して、Latest data で 'MySQL' としてタグ付けされたすべてのアイテムを確認します。
5. ログファイル内の問題を特定し、個別にクローズ
 - {ITEM.VALUE<N>}.regsub()} マクロによる値抽出を使用してイベントを特定するタグを、ログトリガーに定義します。
 - トリガーの設定で、複数の問題のイベント生成モードを持ちます。
 - トリガー設定で、**even correlation** を使います：OK イベントがマッチするイベントだけを閉じるというオプションを選び、マッチングのためのタグを選択します。
 - タグで作成され、個別に閉じられた問題イベントを参照してください。
6. 通知のフィルタリングに使用
 - トリガーレベルでタグを定義し、異なるタグでイベントをマークします。
 - アクションコンディションでタグフィルタリングを使用し、タグデータに一致するイベントのみ通知を受けることができます。
7. item の値から抽出した情報をタグの値として利用
 - タグ値に {ITEM.VALUE<N>}.regsub()} マクロを使用します。
 - item 値から抽出したデータとして、Monitoring → Problems のタグ値を参照します。
8. 通知で問題をよりよく特定
 - トリガーレベルでタグを定義します。
 - 問題通知で {EVENT.TAGS} マクロを使用します。
 - 通知がどのアプリケーション/サービスに属するかをより簡単に識別できます。
9. テンプレートレベルでタグを使用し、設定作業を簡素化
 - テンプレートトリガーレベルでタグを定義します。
 - テンプレートトリガーから作成されたすべてのトリガーにこれらのタグが表示されます。
10. ローレベルディスカバリー（LLD）からタグを使用してトリガーを作成
 - トリガープロトタイプにタグを定義します。
 - タグの名前または値で LLD マクロを使用します。
 - トリガープロトタイプから作成されたすべてのトリガーで、これらのタグを参照してください。

設定

タグは、トリガー設定などの専用タブで入力することができます：

Trigger tags		Inherited and trigger tags
Name	Value	Action
Cloud	value	Remove
Service	MySQL	Remove
Customers	value	Remove
Host	{{ITEM.VALUE2}.iregsub(pattern, output)}	Remove

[Add](#)

マクロのサポート

トリガータグでは、以下のマクロを使用することができます：

- {ITEM.VALUE}, {ITEM.LASTVALUE}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} および {HOST.ID} マクロは、タグ名またはタグ値を入力するために使用することができます。
- トリガータグで 1 つまたは複数のホストからホスト・インベントリ値を参照するには、{INVENTORY.*}[macros] (/manual/appendix/macros/supported_by_location) を使用することができます。
- **User macros** とタグ名/値に対してユーザーマクロコンテキストがサポートされています。ユーザーマクロコンテキストは、ローレベルディスカバリーマクロを含む場合があります。
- トリガータグプロトタイプタグにローレベルディスカバリーマクロを使用することができます。

トリガータグの通知では、以下のマクロを使用することができます：

- {EVENT.TAGS} と {EVENT.RECOVERY.TAGS} マクロは、イベントタグまたは回復イベントタグのカンマ区切りリストに解決されます。
- {EVENT.TAGSJSON} と {EVENT.RECOVERY.TAGSJSON} マクロは、イベントタグ **object** または回復イベントタグオブジェクトを含む JSON 配列に解決されるでしょう。

テンプレート、ホスト、item、Web シナリオで使用できるマクロは次のとおりです：

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} and {HOST.ID} macros
- {INVENTORY.*} **macros**
- **User macros**
- アイテムプロトタイプタグにローレベルディスカバリーマクロを使用することができます

ホストプロトタイプタグには、以下のマクロを使用することができます：

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} and {HOST.ID} macros
- {INVENTORY.*} **macros**
- **User macros**
- **Low-level discovery macros** は、ディスカバリープロセス中に解決され、発見されたホストに追加されるでしょう。

トリガータグのサブstring抽出

サブstring抽出は、タグ名またはタグ値を入力するために、マクロ **function**
 - {ITEM.VALUE}, {ITEM.LASTVALUE} によって得られた値に正規表現を適用する - によってサポートされています。
 たとえば、以下のようになります：

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}
```

```
{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

タグ名、タグ値はマクロ分解後、255 文字を超えると 255 文字にカットされます。

こちらをご参照ください。イベントタグ付けのための **low-level discovery macros** でマクロ関数を使用すること。

イベントタグの表示

タグが定義されている場合、新しいイベントで確認することができます。

- Monitoring → Problems
- Monitoring → Problems → Event details
- Monitoring → Dashboard → Problems widget (問題名の上にマウスを置くと開くポップアップ・ウィンドウ)

Status	Info	Host	Problem	Duration	Ack	Actions	Tags
PROBLEM	New host	Nodata on 'New host' for two minutes	39s	No	Cloud Customers Host: HP-Pro		
							Cloud Customers Host: HP-Pro Service: MySQL

タグエントリーは、最初の3つだけが表示されます。3つ以上のタグエントリーがある場合は、3つのドットで表示されます。この3つのドットの上にマウスを乗せると、すべてのタグエントリーがポップアップウィンドウに表示されます。

タグの表示順は、タグのフィルタリングと、Monitoring → Problems または Problems ダッシュボードウィジェットのフィルタにおける Tag display priority オプションに影響されることに注意してください。

7 視覚化

1 グラフ

概要

Zabbix にはたくさんのデータが流れてきますが、数字だけでなく、何が起きているのかを視覚的に見ることができれば、ユーザはとても楽になります。

そこで、グラフの出番です。グラフを使うことで、データの流れを一目で把握することができ、問題の関連付けや、いつから何かが始まったのか、いつから問題になりそうなのかのプレゼンテーションが可能になります。

Zabbix はユーザに以下を提供します。

- 組込の 1 item のデータに対する **simple graphs**
- より複雑な **customized graphs** を作成可能
- **ad-hoc graphs** では、複数のアイテムの比較に素早くアクセスすることができます。
- カスタマイズ可能な **vector graphs**

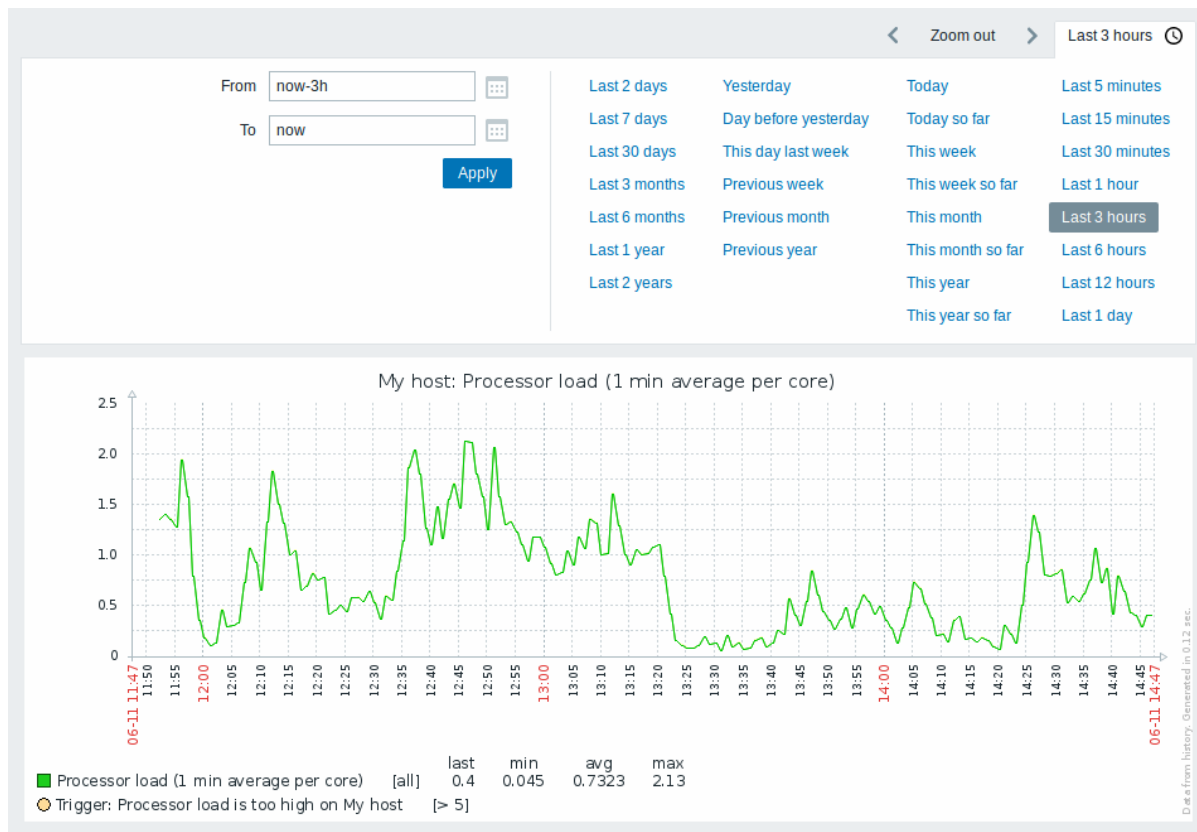
1 シンプルグラフ

概要

item で収集したデータを視覚化するためのシンプルなグラフを提供します。

シンプルグラフを表示するために、ユーザ側で設定する必要はありません。Zabbix はシンプルグラフを自由に利用することができます。

Monitoring → Latest data から各 item のグラフリンクをクリックするだけで、グラフが表示されます。



Note:

数値項目については、シンプルなグラフが用意されています。文字項目については、Monitoring → Latest data に履歴へのリンクが用意されています。

期間セクター

グラフの上にある期間セクターを使えば、マウスを 1 回クリックするだけで必要な期間を選択できます。

今日、今週、今月、今年のようなオプションは、将来の時間/日を含む全期間を表示することに注意してください。一方、今日（現在まで）は、経過時間のみが表示されます。

期間を選択したら、矢印ボタンをクリックすることで、期間を前後に移動できます。ズームアウトボタンを押すと、期間を 2 倍、または各方向に 50% ずつズームアウトすることができます。グラフ内をダブルクリックすることでもズームアウトが可能です。選択した期間の文字列を含むタブラベルをクリックすることで、期間セクター全体を折りたたむことができます。

From/To フィールドには、選択した期間を次のいずれかで表示します。

- 絶対時間構文形式 Y-m-d H:i:s
- 相対時間構文例: now-1d

相対形式の日付は、例えば now-1d や now-1d-2h+5m のように、1 つまたは複数の数学演算 (- や +) を含むことができます。相対時間では、以下の省略形がサポートされています:

- now
- s (seconds)
- m (minutes)
- h (hours)
- d (days)
- w (weeks)
- M (months)
- y (years)

精度は時間フィルターでサポートされています (例:now-1d/M のような式)。精度の詳細:

精度	開始	終了
m	Y-m-d H:m:00	Y-m-d H:m:59
h	Y-m-d H:00:00	Y-m-d H:59:59
d	Y-m-d 00:00:00	Y-m-d 23:59:59
w	Monday of the week 00:00:00	Sunday of the week 23:59:59

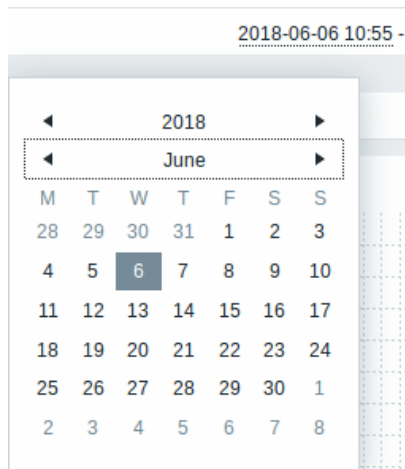
精度	開始	終了
M	First day of the month 00:00:00	Last day of the month 23:59:59
y	1st of January of the year 00:00:00	31st of December of the year 23:59:59

例:

開始	終了	選択される期間
now/d	now/d	当日 00:00 - 当日 23:59
now/d	now/d+1d	当日 00:00 - 翌日 23:59
now/w	now/w	当週月曜 00:00:00 - 当週土曜 23:59:59
now-1y/w	now-1y/w	去年の月曜 00:00:00 - 去年の日曜 23:59:59

日付ピッカー

From/To フィールドの横にあるカレンダーアイコンをクリックして、特定の開始日/終了日を選択することができます。この場合、日付ピッカーのポップアップが開きます。



日付ピッカー内では、Tab および Shift+Tab を使用して、年/月/日のブロック間を移動できます。キーボードの矢印または矢印ボタンを使用して、目的の値を選択できます。Enter を押す (または目的の値をクリックする) と、選択が有効になります。

表示時間を制御するもう 1 つの方法は、グラフ内の領域をマウスの左ボタンで強調表示することです。マウスの左ボタンを放すと、強調表示された領域にグラフがズームインします。

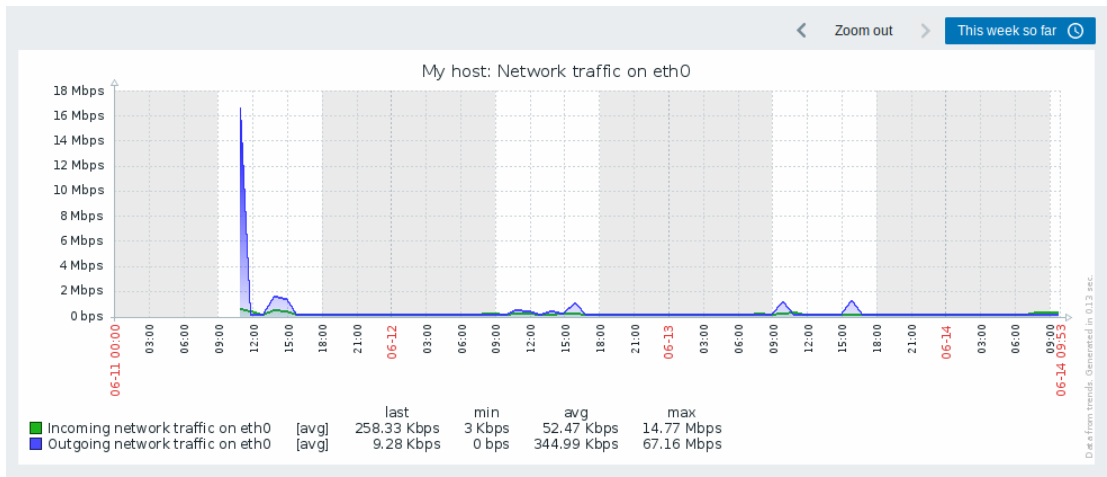
時間値が指定されていない場合、またはフィールドが空白のままの場合、時間値は「00:00:00」に設定されます。これは、今日の日付選択には適用されません。その場合、時間は現在の値に設定されます。

最近のデータと長い期間のデータ

ごく最近のデータでは、受信した各値を結ぶ 1 本の線が描かれます。1 本の線は、1 つの値に対して少なくとも 1 つの
 水平ピクセルが利用可能である限り、描かれます。

濃い緑色の線は平均値を、薄いピンク色と薄い緑色の線はその時点の最大値と最小値を示しています。最大値と最小値の間は、
黄色の背景で埋められます。

稼働時間 (稼働日) はグラフでは白い背景で表示され、非稼働時間はグレーで表示されます。
 (Original blue デフォルトのフロントエンドのテーマで表示されます)



シンプルなグラフでは常に稼働時間が表示されますが、**custom graphs** では、ユーザーの `
` 好みで表示することができます。

3ヶ月以上のグラフの場合、稼働時間は表示されません。

トリガー線

単純なトリガーは、トリガーの重要度の色に黒いダッシュが付いた線で表示されます。グラフ上の青い線と凡例に表示された `
` トリガー情報に注意してください。グラフ上に最大3つのトリガーラインを表示することができます。もし、より多くの `
` トリガーがある場合、重要度の低いトリガーが優先されます。トリガーは常にシンプルなグラフで表示されますが、`
` **custom graphs** で表示することはユーザーの好みによります。 `
`



history / trends からの生成

グラフは、item **history or trends** のいずれかに基づいて描画することができます。

フロントエンドの **debug mode** が有効な場合、右下にグレーの縦長のキャプションが `
` 表示されます。キャプションがグラフの右下に表示され、データの出所を示します。

トレンドの履歴が使用されるかどうかは、いくつかの要因に影響されます。

- item history の寿命。例えば、item history は 14 日間保持することができます。この場合、14 日間より古いデータは trends から来ることになります。
- グラフのデータが輻輳しています。横軸のグラフピクセルに表示する秒数が 3600/16 を超えると、trends データが表示されます (同じ期間の item history が残っている場合もあります)。

- trendsが無効の場合、その期間で item history が利用可能であれば、グラフ作成に使用されます。これは Zabbix2.2.1 からサポートされています (以前は trends が無効の場合、item history が利用可能でもその期間のグラフは空白でした)

データ不在の場合

定期更新の item は、データが収集されないとグラフに何も表示されません。

ただし、Trapper item と定期更新 item (定期更新間隔を 0 に設定) については、最初の収集値までと最後の収集値から
 グラフの終わりまで、それぞれ最初の値/最後の値の高さに直線が引かれます。

生の値への切り替え

右上のドロップダウンで、単純なグラフから Values/500 latest values のリストへ切り替えることができます。これは、グラフを構成する数値を表示するのに便利です。

ここで表示される数値は生のものであり、単位や数値の後処理は行われていません。しかし、値のマッピングは適用されます。

既知の問題

グラフについては、[known issues](#)を参照してください。

2 カスタムグラフ

概要

カスタムグラフは、その名の通り、カスタマイズ機能を提供するものです。

単純なグラフは、1つの項目のデータを見るには適していますが、設定機能はありません。

そのため、グラフのスタイルや線の表示方法を変更したり、複数の項目、例えばトラフィックの入出力を 1つのグラフで
 比較したい場合は、カスタムグラフが必要になります。

カスタムグラフは、手動で設定します。

カスタムグラフは、1つのホスト、複数のホスト、または 1つのテンプレートに対して作成することができます。

カスタムグラフの設定

カスタムグラフを作成するには、以下のようになります。

- Configuration → Hosts (もしくは Templates) に移動します。
- 目的のホストまたはテンプレートの横の行で Graphs をクリックします。
- グラフ画面で Create graph をクリック
- グラフの属性を編集する

Graph
Preview

* Name

* Width

* Height

Graph type Normal

Show legend

Show working time

Show triggers

Percentile line (left)

Percentile line (right)

Y axis MIN value Fixed

Y axis MAX value Calculated

* Items	Name	Function	Draw style	Y axis side	Color	Action
1:	My host: Outgoing network traffic on eth0	avg	Filled region	Left	00C800	Remove
2:	My host: Incoming network traffic on eth0	avg	Bold line	Left	C80000	Remove

[Add](#)

必須入力項目には、赤いアスタリスクが表示されています。

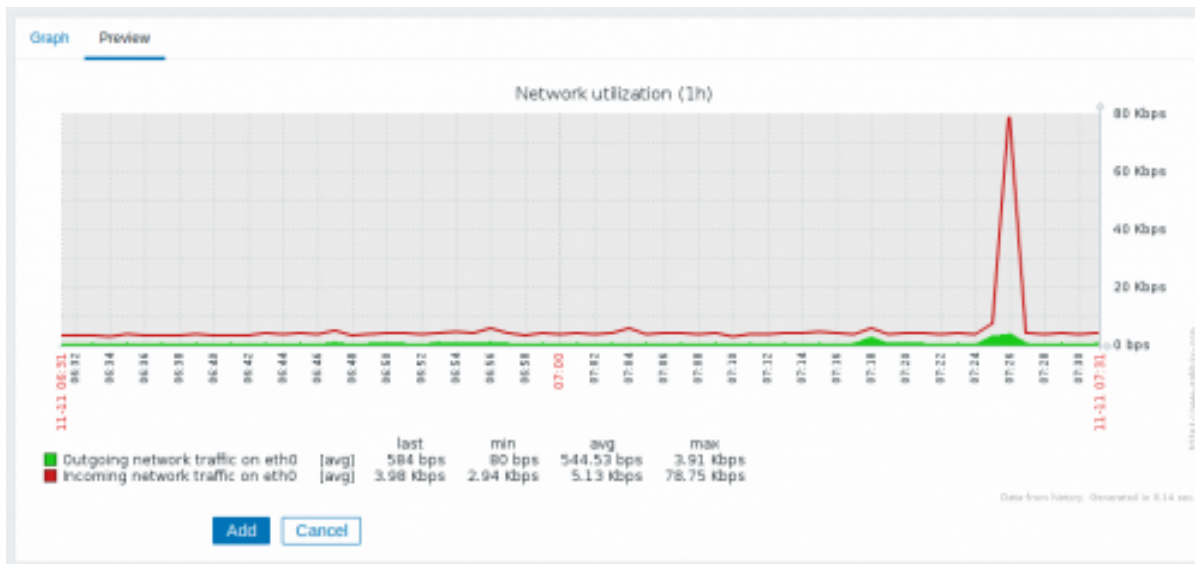
グラフの属性:

パラメータ	説明
Name	<p>Unique graph name.</p> <p>Expression macros are supported in this field, but only with <code>avg</code>, <code>last</code>, <code>min</code> and <code>max</code> functions, with time as parameter (for example, <code>{?avg(/host/key, 1h)}</code>).</p> <p><code>{HOST.HOST<1-9>}</code> macros are supported for the use within this macro, referencing the first, second, third, etc. host in the graph, for example <code>{?avg(/{HOST.HOST2}/key, 1h)}</code>.</p> <p>Note that referencing the first host with this macro is redundant, as the first host can be referenced implicitly, for example <code>{?avg(/key, 1h)}</code>.</p>
Width	Graph width in pixels (for preview and pie/exploded graphs only).
Height	Graph height in pixels.
Graph type	<p>Graph type:</p> <p>Normal - normal graph, values displayed as lines</p> <p>Stacked - stacked graph, filled areas displayed</p> <p>Pie - pie graph</p> <p>Exploded - "exploded" pie graph, portions displayed as "cut out" of the pie</p>
Show legend	Checking this box will set to display the graph legend.
Show working time	If selected, non-working hours will be shown with a gray background. Not available for pie and exploded pie graphs.
Show triggers	If selected, simple triggers will be displayed as lines with black dashes over trigger severity color. Not available for pie and exploded pie graphs.
Percentile line (left)	Display percentile for left Y-axis. If, for example, 95% percentile is set, then the percentile line will be at the level where 95 percent of the values fall under. Displayed as a bright green line. Only available for normal graphs.
Percentile line (right)	Display percentile for right Y-axis. If, for example, 95% percentile is set, then the percentile line will be at the level where 95 percent of the values fall under. Displayed as a bright red line. Only available for normal graphs.
Y axis MIN value	<p>Minimum value of Y-axis:</p> <p>Calculated - Y axis minimum value will be automatically calculated</p> <p>Fixed - fixed minimum value for Y-axis. Not available for pie and exploded pie graphs.</p> <p>Item - last value of the selected item will be the minimum value</p>
Y axis MAX value	<p>Maximum value of Y-axis:</p> <p>Calculated - Y axis maximum value will be automatically calculated</p> <p>Fixed - fixed maximum value for Y-axis. Not available for pie and exploded pie graphs.</p> <p>Item - last value of the selected item will be the maximum value</p>
3D view	Enable 3D style. For pie and exploded pie graphs only.
Items	Items, data of which are to be displayed in this graph. Click on Add to select items. You can also select various displaying options (function, draw style, left/right axis display, color).
Sort order	<p>Draw order. 0 will be processed first. Can be used to draw lines or regions behind (or in front of) another.</p> <p>You can drag and drop items by the arrow at the beginning of (0→100)a line to set the sort order or which item is displayed in front of the other.</p>
Name	Name of the selected item is displayed as a link. Clicking on the link opens the list of other available items.

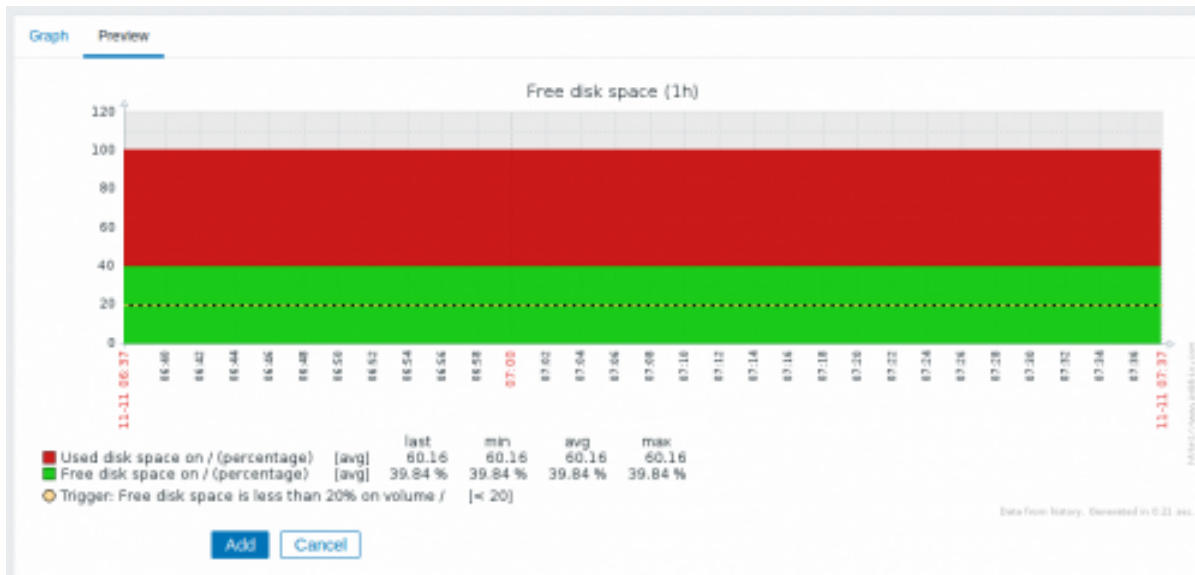
パラメータ	説明
Type	Type (only available for pie and exploded pie graphs): Simple - the value of the item is represented proportionally on the pie Graph sum - the value of the item represents the whole pie Note that coloring of the "graph sum" item will only be visible to the extent that it is not taken up by "proportional" items.
Function	Select what values will be displayed when more than one value exists per vertical graph pixel for an item: all - display all possible values (minimum, maximum, average) in the graph. Note that for shorter periods this setting has no effect; only for longer periods, when data congestion in a vertical graph pixel increases, 'all' starts displaying minimum, maximum, and average values. This function is only available for Normal graph type. See also: Generating graphs from history/trends . avg - display the average values last - display the latest values. This function is only available if either Pie/Exploded pie is selected as graph type. max - display the maximum values min - display the minimum values
Draw style	Select the draw style (only available for normal graphs; for stacked graphs filled region is always used) to apply to the item data - Line, Bold line, Filled region, Dot, Dashed line, Gradient line.
Y axis side	Select the Y axis side to show the item data - Left, Right.
Color	Select the color to apply to the item data.

グラフプレビュー

Preview タブでは、グラフのプレビューが表示され、作成中のグラフをすぐに確認することができます。



なお、プレビューでは、テンプレートアイテムのデータは表示されません。



この例では、トリガレベルを表示する破線の太線と、凡例に表示されるトリガ情報に注目します。

Note:

表示できるトリガーは 3 行までです。それ以上のトリガーがある場合、重要度の低いトリガーが優先的に表示されます。グラフの高さが 120 ピクセル以下に設定されている場合、凡例にはトリガーが表示されません。

3 アドホックグラフ

概要

simple graph は 1 つの item のデータにアクセスするのに適しており、**custom graphs**はカスタマイズオプションを
 提供しますが、少ない労力でメンテナンスなしに複数の item の比較グラフを素早く作成できるものではありません。

この問題を解決するために、Zabbix2.4 から、複数の item のアドホックグラフを非常に迅速に作成することができるようになりました。

設定

アドホックグラフを作成するには、次のようにします:

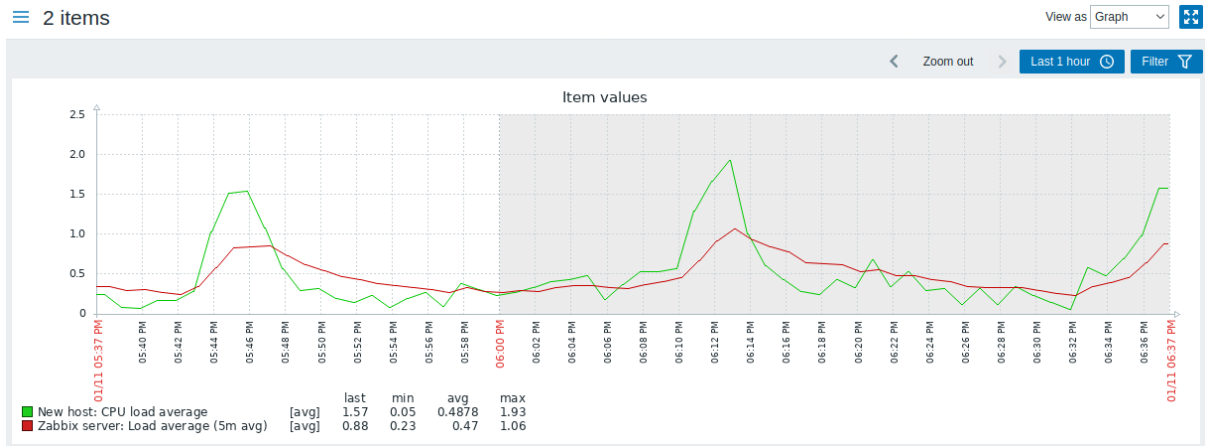
- Monitoring → Latest data に移動します。
- フィルタを使用して、必要な項目を表示します。
- グラフにしたい項目のチェックボックスをマークします。
- Display stacked graph または Display graph ボタンをクリックする

☰ Latest data

<input type="checkbox"/>	Host ▲	Name	Last check	Last val
<input checked="" type="checkbox"/>	New host	CPU load average	05/24/2021 10:46:5...	0.86
<input type="checkbox"/>	Zabbix server	Load average (1m avg)	05/24/2021 10:47:1...	0.73
<input type="checkbox"/>	Zabbix server	Load average (15m avg)	05/24/2021 10:47:1...	0.93
<input checked="" type="checkbox"/>	Zabbix server	Load average (5m avg)	05/24/2021 10:47:1...	0.93

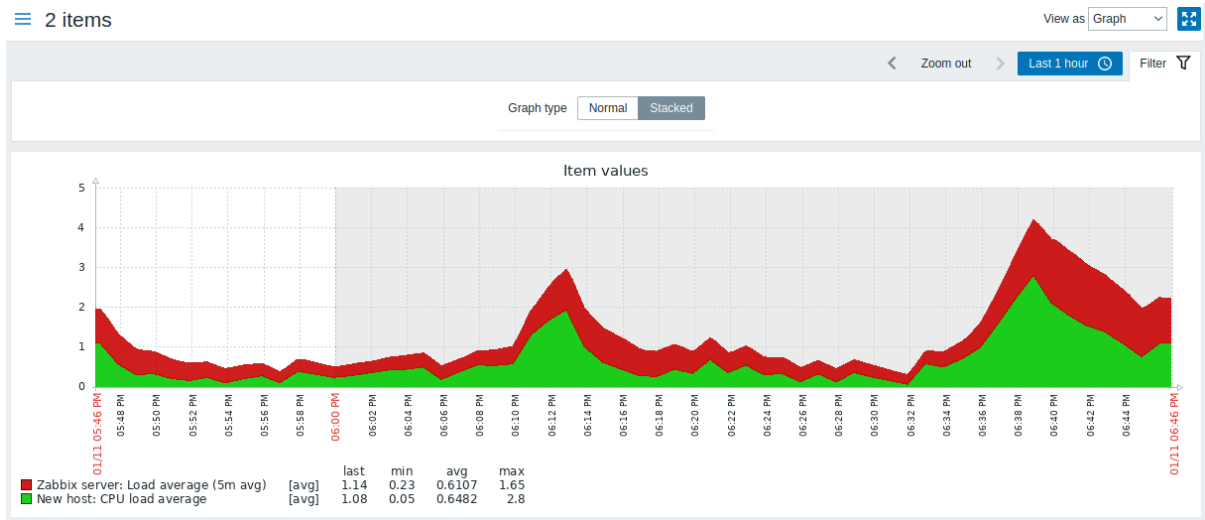
2 selected

グラフは瞬時に作成されます:



なお、グラフの線が多くなりすぎないように、各項目の平均値のみを表示します。(最小値/最大値の線は表示されません)
 trigger と trigger 情報はグラフに表示されません。

作成されたグラフウィンドウでは、time period selector が利用でき、“通常の”折れ線グラフから
 積み上げグラフに切り替えることができます。(戻ることも可能です)



4 グラフの集計

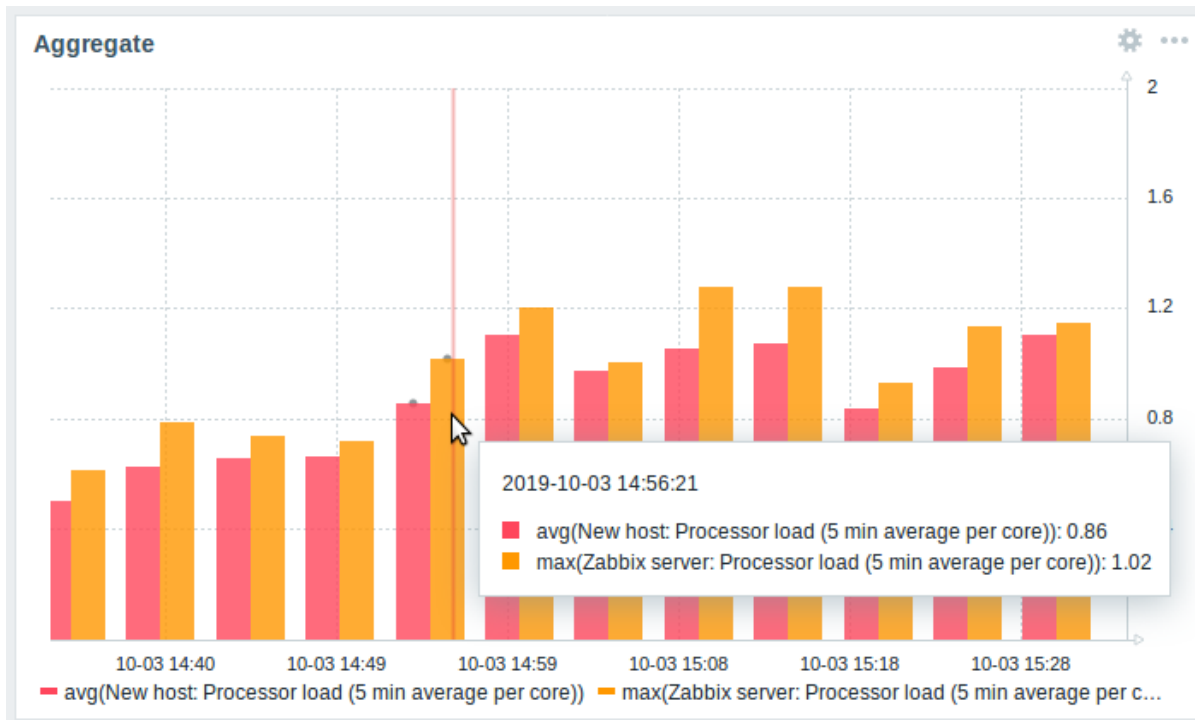
概要

ダッシュボードのグラフウィジェットで利用できる集計機能では、すべての値ではなく、選択した間隔 (5 分、1 時間、1 日) の
 集計値を表示することができます。

集計オプションは以下のとおりです:

- min
- max
- avg
- count
- sum
- first (first value displayed)
- last (last value displayed)

データ集計の最もエキサイティングな使い方は、ある期間のデータを並べてきれいに比較することができることです:



グラフのある時点にカーソルを合わせると、item とその集計値に加え、日付と時刻が表示されます。項目は括弧内に
 表示され、その前に使用した集計関数が表示されます。これはグラフ内のポイントの日時であり、実際の値ではないことに
 注意してください。

設定

graph widgetを構成する際に、
 データセット設定で集計のオプションを利用することが可能です。

Missing data None Connected T

Y-axis Left Right

Time shift

Aggregation function

Aggregation interval

Aggregate Each item Data set

集計関数と時間間隔を選択することができます。データセットが複数の item から構成されている場合、各 item の集計データを
 個別に表示するか、すべてのデータセット item を 1 つの集計値として表示するか、別のオプションを使用することも可能です。

使用例

Nginx サーバーへの平均リクエスト数

Nginx サーバーへの 1 日あたりの平均リクエスト数 (秒) を表示します。

- データセットに秒間リクエスト数の item を追加します。
- 集約関数 avg を選択し、インターバル 1d を指定します。
- 棒グラフが表示され、各棒は一日あたりの平均秒間リクエスト数を表します。

クラスタ間の最小週次ディスク容量

1 週間におけるクラスタ間の最小ディスク容量を表示します。

- データセットに追加します: ホスト cluster*, キー "Free disk space on /data"

- 集約関数 `min` を選択し、間隔 `1w` を指定します。
- 棒グラフが表示され、各棒はクラスタの各 `/data` ボリュームの 1 週間あたりの最小ディスク容量を表します。

2 ネットワークマップ

概要

ネットワークの管理をしている場合、インフラの概要をどこかで把握したいと思うかもしれません。
 するために、Zabbix ではネットワークや好きなもののマップを作成することができます。

すべてのユーザがネットワークマップを作成することができます。
 マップはパブリック (すべてのユーザが利用可能) またはプライベート (選択されたユーザが利用可能) にすることができます。

[configuring a network map](#)に進みます。

1 ネットワークマップの設定

概要

Zabbix でマップを設定するには、まず一般的なパラメータを定義してマップを作成し、その後、実際のマップに要素と
 そのリンクを設定する必要があります。

マップには、ホスト、ホストグループ、トリガー、画像、別のマップの要素を入れることができます。

アイコンは、マップの要素を表現するために使用されます。アイコンと一緒に表示される情報を定義し、最近の問題が
 特別な方法で表示されるように設定できます。アイコンをリンクさせ、リンクに表示される情報を定義することができます。

アイコンをクリックするとアクセスできるカスタム URL を追加することができます。したがって、ホストアイコンをホスト
 プロパティに、マップアイコンを別のマップにリンクさせることができます。

マップは [Monitoring → Maps](#) で管理され、設定、管理、閲覧が可能です。
 モニタリングビューでは、アイコンをクリックして、いくつかのスクリプトや URL へのリンクを利用することができます。

ネットワークマップは Zabbix3.4 からベクターグラフィック (SVG) をベースにしています。

パブリックマップとプライベートマップ

Zabbix の全てのユーザ (管理者以外のユーザも含む) はネットワークマップを作成することができます。マップには所有者
 (作成したユーザ) が存在します。マップはパブリックまたはプライベートに設定することができます。

- Public マップは全てのユーザが見ることができますが、見るためには少なくとも 1 つのマップ要素への読み取り権限が
 必要です。公開マップは、ユーザー/ユーザーグループがこのマップに対する読み取り/書き込み権限と、対応するマップの
 すべての要素 (リンクのトリガーを含む) に対する読み取り権限を持っている場合に、編集することができます。
- Private マップは、その所有者と、所有者がマップを `shared` したユーザー/ユーザーグループにのみ表示されます。一般ユーザー (Super admin 以外) は、自分が所属するグループとユーザーとのみ共有できます。Admin レベルのユーザーは、所有者であるか、共有ユーザーリストに属しているかに関係なく、プライベートマップを表示できます。プライベートマップは、マップの所有者が編集でき、ユーザー/ユーザーグループがこのマップに対する読み取り/書き込み権限と、リンク内のトリガーを含む対応するマップのすべての要素に対する少なくとも読み取り権限を持つ場合、編集できます。

ユーザーが読み取り権限を持たないマップ要素は、グレー表示され、要素上のすべてのテキスト情報は非表示になります。
 ただし、トリガーラベルは、ユーザーがトリガーに対する権限を持っていなくても表示されます。

マップに要素を追加するには、ユーザーはその要素に対して少なくとも読み取り権限を持っていない限りなりません。

マップの作成

マップを作成するには次の手順を実行します:

- 監視データ → マップに移動します
- すべてのマップビューに移動します
- マップの作成をクリックします

既存のマップの設定フォームで複製およびすべて複製ボタンを使用して、新しいマップを作成することもできます。複製をクリックすると、元のマップの一般的なレイアウト属性が保持されますが、要素は保持されません。すべて複製では、一般的なレイアウト属性と元のマップのすべての要素の両方が保持されます。

マップタブには、一般的なマップ属性が含まれています。

Map **Sharing**

* Owner

* Name

* Width

* Height

Background image

Automatic icon mapping [show icon mappings](#)

Icon highlight

Mark elements on trigger status change

Display problems Expand single problem Number of problems Number of p

Advanced labels

Host group label type

Host label type

Trigger label type

Map label type

Image label type

Map element label location

Problem display

Minimum severity Not classified Information Warning Average High

Show suppressed problems

URLs

Name	URL
<input type="text" value="Latest data"/>	<input type="text" value="https://localhost/zabbix/latest.php"/>
Add	

すべての必須入力フィールドには赤いアスタリスクが付いています。

一般的なマップ属性:

パラメータ	説明
所有者 名前	マップ所有者名 ユニークなマップ名
幅	マップの幅 (pixels)
高さ	マップの高さ (pixels)
背景のイメージ	背景画像を使用: イメージは存在しません - 背景画像なし (白い背景) 画像 - 背景画像として使用する選択した画像。スケーリングは実行されません。地理的な地図やその他の画像を使用して、マップを強化することができます。
アイコンの自動マッピング	管理 → 一般設定 → アイコンマッピングで構成された自動アイコンマッピングを使用するように設定できます。アイコンマッピングでは、特定のホストインベントリフィールドに対して特定のアイコンをマッピングできます。
アイコンのハイライト	このボックスをオンにすると、マップ要素が強調表示されます。 アクティブなトリガーを持つ要素は、最も深刻度の高いトリガーと同じ色の丸い背景で表示されます。また、すべての問題が確認されると、円の周りに太い緑色の線が表示されます。 "無効" または "メンテナンス中" ステータスの要素は、それぞれ正方形の背景、グレー、オレンジになります。 参照: マップの表示
トリガーの状態が変化した要素をマーク	トリガーステータスの最近の変更 (最近の障害または解決済) は、ラベルのない要素アイコンの3つの側面にあるマーカー (内側を向いた赤い三角形) で強調表示されます。マーカーは 30 分間表示されます。
障害数の表示	マップ要素で障害を表示する方法を選択します: 障害が一件の時にトリガー名を表示 - 障害が1つしかない場合は、障害の名前が表示されます。それ以外の場合は、障害の総数が表示されます。 障害の数のみ - 障害の総数が表示されます 障害の数と最も深刻度の高いトリガー名を表示 - 最も深刻な障害の名前と障害の総数が表示されます。 '最重要' は、障害の深刻度に基づいて決定され、等しい場合は障害のイベント ID (ID が大きい) か、後に登録された障害が先に表示されます) に基づいて決定されます。トリガーマップ要素の場合、障害の深刻度に基づいており、等しい場合は、トリガースト内のトリガー位置に基づいています。同一トリガーの障害が複数ある場合は、最新のものを表示します。
拡張ラベル アイコンのラベルタイプ	このボックスをチェックすると、個別の要素タイプに対して個別のラベルタイプを定義できます。 マップ要素に使用されるラベルタイプ: ラベル - マップ要素のラベル IP アドレス - IP アドレス アイコン名 - 要素名 (ホスト名など) ステータスのみ - ステータスのみ (OK または障害) なし - ラベルは表示されません
アイコンのラベルの位置	マップ要素に対するラベルの位置: 下 - マップ要素の下 左 - 左 右 - 右 ** 上 ** - マップ要素の上
障害の表示	問題数の表示: すべて - 全ての問題数が表示されます 障害数と未確認数を表示 - 未確認の障害数は合計の障害数と分離して表示されます 未確認のみ - 未確認の障害数のみが表示されます
最小の深刻度	選択した最小重大度レベルを下回る問題は、マップに表示されません。 たとえば、警告を選択すると、情報および未分類レベルトリガーによる変更はマップに反映されません。 このパラメータは、Zabbix 2.2 以降でサポートされています。
メンテナンス中の障害を表示 URL	チェックボックスをオンにすると、ホストのメンテナンス時に抑制される (表示されない) 障害が表示されます。 各要素タイプの URL を定義できます (ラベル付き)。これらは、ユーザーが地図表示モードで要素をクリックすると、リンクとして表示されます。 地図 URL の名前と値でマクロを使用できます。完全なリストについては、 サポートされているマクロ を参照し 'マップ URL の名前と値' を検索してください。

共有

共有タブには、プライベートマップのマップタイプと共有オプション (ユーザーグループ、ユーザー) が含まれています。

Map **Sharing** ●

Type Private Public

List of user group shares

User groups	Permissions
MySQL administrators	<input type="radio"/> Read-only <input checked="" type="radio"/> Read-write
Add	

List of user shares

Users	Permissions
Admin (Zabbix Administrator)	<input type="radio"/> Read-only <input checked="" type="radio"/> Read-write
Add	

パラメータ	説明
タイプ	マップの種類を選択: 非公開 - マップは選択したユーザーグループとユーザーのみに表示されます 公開 - マップはすべてのユーザーに表示されます
共有するユーザーグループ	マップにアクセスできるユーザーグループを選択します。 読み取り専用または読み書きアクセスを許可できます。
共有するユーザー	マップにアクセスできるユーザーを選択します。 読み取り専用または読み書きアクセスを許可できます。

追加をクリックしてこのマップを保存すると、名前、寸法、および特定の設定を含む空のマップが作成されます。ここで、いくつかの要素を追加する必要があります。そのためには、マップリストの **変更** をクリックして、編集可能な領域を開きます。

要素の追加

要素を追加するには、アイコンの横にある **追加** をクリックします。新しい要素がマップの左上隅に表示されます。好きなところにドラッグ&ドロップしてください。

グリッドオプションを“ON”にすると、アイコンは常にグリッドに整列することに注意してください（ドロップダウンからさまざまなグリッドサイズを選択したり、グリッドを非表示/表示したりすることもできます）。グリッドに沿った配置をせずに、要素を好きな位置に配置したい場合は、オプションを“OFF”にします。（ランダムな要素は [グリッドに揃える] ボタンを使用して、後で再びグリッドに整列させることができます。）

いくつかの要素が配置されたので、名前などを付けてそれらを区別したいと思うかもしれません。要素をクリックすると、フォームが表示され、要素のタイプを設定したり、名前を付けたり、別のアイコンを選択したりできます。

Map element: [Add](#) / [Remove](#) Shape: [Add](#) / [Remove](#) Link: [Add](#) / [Remove](#) Expand macros: [Off](#) Grid: [Shown](#) / [On](#) 50x50 [Align map elements](#) [Update](#)

Map element

Type: Host

Label: New element

Label location: Default

* Host: My host [Select](#)

Tags: [And/Or](#) [Or](#)

tag Contains value [Remove](#)

[Add](#)

Automatic icon selection

Icons: Default: Server_(64)

Problem: Default

Maintenance: Default

Disabled: Default

Coordinates X: 224 Y: 91

URLs: Name URL Action [Remove](#)

マップ要素の属性:

パラメータ	説明
タイプ	<p>要素のタイプ:</p> <ul style="list-style-type: none"> ホスト - 選択したホストのすべてのトリガーのステータスを表すアイコン マップ - マップのすべての要素のステータスを表すアイコン トリガー - 1つ以上のトリガーのステータスを表すアイコン ホストグループ - 選択したグループに属するすべてのホストのすべてのトリガーのステータスを表すアイコン イメージ - どのリソースにもリンクされていないアイコン
ラベル	<p>アイコンラベル。任意の文字列</p> <p>マクロと複数行の文字列が使用できます。</p> <p>このフィールドでは式マクロがサポートされていますが、パラメータとして時間を指定した avg、last、min、max 関数のみがサポートされています (例: {?avg(/host/key) ,1h})</p> <p>サポートされているマクロの完全なリストについては、サポートされているマクロ を参照し、'マップエレメントラベル' を検索してください。</p>
ラベルの位置	<p>アイコンに対するラベルの位置:</p> <ul style="list-style-type: none"> デフォルト - マップのデフォルトのラベルの位置 下 - アイコンの下 左 - 左 ** 右 ** - 右 上 - アイコンの上

パラメータ	説明
ホスト	要素タイプが 'ホスト' の場合、ホストを入力します。このフィールドはオートコンプリートなので、ホストの名前を入力し始めると、一致するホスト名がドロップダウンで表示されるのでスクロールして選択します。選択したものを削除するには、[x] をクリックします。
マップ トリガー	要素タイプが 'マップ' の場合、マップを選択します。 要素タイプが 'トリガー' の場合、下の 新規トリガーフィールドで 1 つ以上のトリガーを選択し、追加をクリックします。 選択したトリガーの順序は変更できますが、トリガーの申告度が同じ範囲内でのみ変更できます。複数のトリガーを選択すると、設定モードと表示モードの両方で {HOST.*} マクロの解決にも影響します。 //1 設定モードでは// 最初に表示された {HOST.*} マクロは、リスト内の最初のトリガー (トリガーの深刻度に基づく) に応じて解決されます。 //2 ビューモード// は、一般的なマップ属性の 表示の問題 パラメータに依存します。 * 障害が一件の時トリガー名を表示モードが選択されている場合、最初に表示された {HOST.*} マクロは、最後に検出された障害トリガー (深刻度に関係なく) またはリスト内の最初のトリガー (障害が検出されなかった場合) に応じて解決されます。); * 障害の数と最も深刻度が高いトリガー名を表示モードが選択されている場合、最初に表示された {HOST.*} マクロは、トリガーの深刻度に応じて解決されます。
ホストグループ	要素タイプが 'ホストグループ' の場合、ホストグループを入力します。このフィールドはオートコンプリートなので、グループの名前を入力し始めると、一致するグループ名がドロップダウンで表示されるので、スクロールして選択します。選択したものを削除するには、[x] をクリックします。
タグ	タグを指定して、ウィジェットに表示される障害の数を制限します。特定のタグとタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の照合では、常に大文字と小文字が区別されます。 条件ごとに複数の演算子を使用できます。 存在する - 指定したタグ名を含める 等しい - 指定したタグ名を含め、値 (大文字と小文字を区別) 含む - タグ値に入力された文字列が含まれる指定されたタグ名を含めます (部分文字列の一致、大文字と小文字は区別されません) 存在しない - 指定されたタグ名を除外しますタグ名 等しくない - 指定されたタグ名と値を除外します (大文字と小文字を区別します) 含まない - タグ値に入力された文字列が含まれる指定されたタグ名を除外します (部分文字列の一致、大文字と小文字の区別なし) 条件には次の 2 つの計算タイプがあります。 And/Or - すべての条件が満たされる必要があります。同じタグ名を持つ条件は、Or 条件によってグループ化されます。 Or - 1 つの条件が満たされていれば十分です このフィールドは、ホストおよびホストグループの要素タイプで使用できます。 チェックすると表示するアイコンを決定するためにアイコンマッピングが使用されます。 デフォルト、障害、メンテナンス、無効の場合に表示するアイコンを選択できます 座標 X 地図要素の X 座標 座標 Y 地図要素の Y 座標 URL 要素固有の URL を設定できます。ユーザーがマップ表示モードで要素をクリックすると、リンクとして表示されます。要素に独自の URL があり、定義されたタイプの地図レベルの URL がある場合、同じメニューに結合されます。 地図要素の名前と値にマクロを使用できます。完全なリストについては、 サポートされているマクロ を参照し、'マップ URL の名前と値' を検索してください。

Attention:

追加された要素は自動的に保存されません。ページから移動すると、すべての変更が失われる可能性があります。変更後は右上隅にある 更新ボタンをクリックすることをお勧めします。クリックすると、次のポップアップでの選択に関係なく、変更が保存されます。選択したグリッドオプションも各マップと共に保存されます。

要素の選択

要素を選択するには、1 つを選択してから Ctrl を押しながら他の要素を選択します。

また、編集可能な領域で矩形をドラッグし、その中のすべての要素を選択することで、複数の要素を選択することができます。

複数の要素を選択すると、要素のプロパティフォームが一括更新モードに移行し、選択した要素の属性を一括して変更することが可能です。そのためには、チェックボックスを使って属性をマークし、その属性に新しい値を入力します。
ここでマクロを使用することもできます (例えば、要素のラベルに {HOST.NAME} を使用)。

Map element: [Add / Remove](#) Shape: [Add / Remove](#) Link: [Add / Remove](#) Expand macros: [Off](#) Grid: [Shown / On](#) 50x50 [Align map elements](#) [Update](#)

The screenshot shows a Zabbix map editor interface. At the top, there are navigation links for 'Map element', 'Shape', and 'Link', each with 'Add / Remove' options. There are also settings for 'Expand macros' (set to 'Off'), 'Grid' (set to 'Shown / On'), a '50x50' grid size selector, and buttons for 'Align map elements' and 'Update'. The main area is a grid with X and Y coordinates ranging from 50 to 700. Two server icons are placed on the grid. The first icon is labeled 'New element'. The second icon is labeled with macros: {HOST.NAME} and {HOST.CONN}. A 'Mass update elements' dialog box is open in the foreground, showing a table of selected elements and various configuration options.

Selected elements	Type	Name
	Host	My host
	Host	vcenter.zabbix.lan

Label {HOST.NAME}
 {HOST.CONN}

Label location Top

Automatic icon selection

Icon (default) Cloud_(24)

Icon (problem) Default

Icon (maintenance) Default

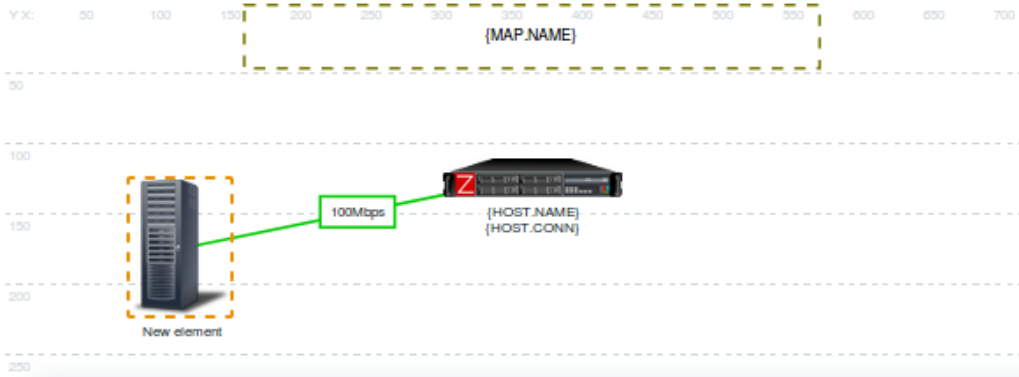
Icon (disabled) Default

[Apply](#) [Remove](#) [Close](#)

要素のリンク

要素をマップに配置したら、それらをリンクすることもできます。2つの要素をリンクするには、最初に2つの要素を選択した状態で、[リンク](#)の横にある[追加]をクリックします。

リンクが作成されると、単一要素フォームに追加のリンクセクションが含まれるようになります。編集をクリックして、リンク属性を編集します。



Map element

Type **Host**

Label

Label location **Default**

* Host [Select](#)

Application [Select](#)

Automatic icon selection

Icons

Default	<input type="text" value="Server_(96)"/>
Problem	<input type="text" value="Default"/>
Maintenance	<input type="text" value="Default"/>
Disabled	<input type="text" value="Default"/>

Coordinates X Y

URLs

Name	URL	Action
<input type="text"/>	<input type="text"/>	Remove

[Add](#)

[Apply](#) [Remove](#) [Close](#)

Links

Element name	Link indicators	Action
vcenter.zabbix.lan		Edit

Label

Connect to

Type (OK) **Bold line**

Color (OK)

Link indicators

Trigger	Type	Color	Action
Add			

[Apply](#) [Remove](#) [Close](#)

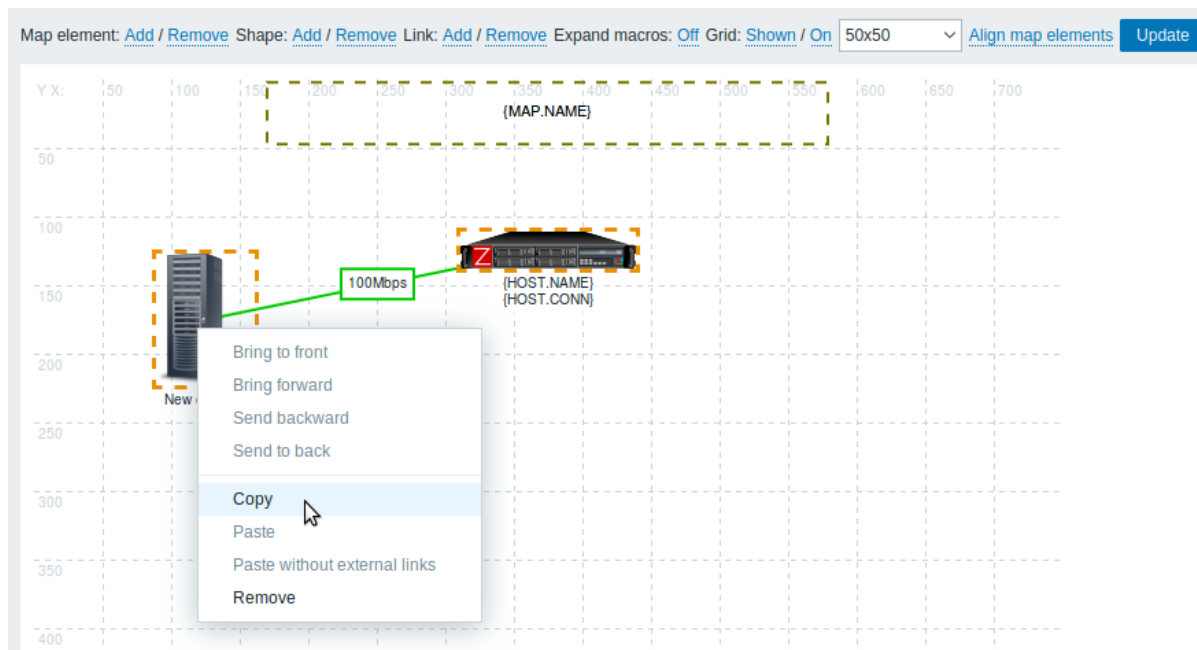
リンク属性:

パラメータ	説明
ラベル	リンクの上に表示されるラベル。 このフィールドでは式マクロがサポートされていますが、パラメータとして時間を指定した avg、last、min、max 関数のみがサポートされています (例: {?avg(/host/key) ,1h})
接続方法 タイプ (正常 OK)	リンクが接続する要素 デフォルトのリンクスタイル: Line - シングルライン Bold line - 太線 Dot - ドット Dashed line - 破線
色 (正常) 障害発生時の条件設定	デフォルトのリンク色 リンクにリンクされたトリガーのリスト。トリガーのステータスが障害の場合、そのスタイルがリンクに適用されます。

要素の移動とコピーペースト

選択した要素の 1 つをクリックし、マウスボタンを押したままカーソルを目的の場所に移動すると、選択した複数の要素を
 マップ内の別の場所に移動することができます。

要素を選択し、マウスの右ボタンで選択した要素をクリックし、メニューから **Copy** を選択することで、1 つまたは
 複数の要素をコピーすることができます。



要素を貼り付けるには、マウスの右ボタンで地図上の領域をクリックし、メニューから Paste を選択します。Paste without external links を選択すると、選択した要素間のリンクのみを残して貼り付けられます。

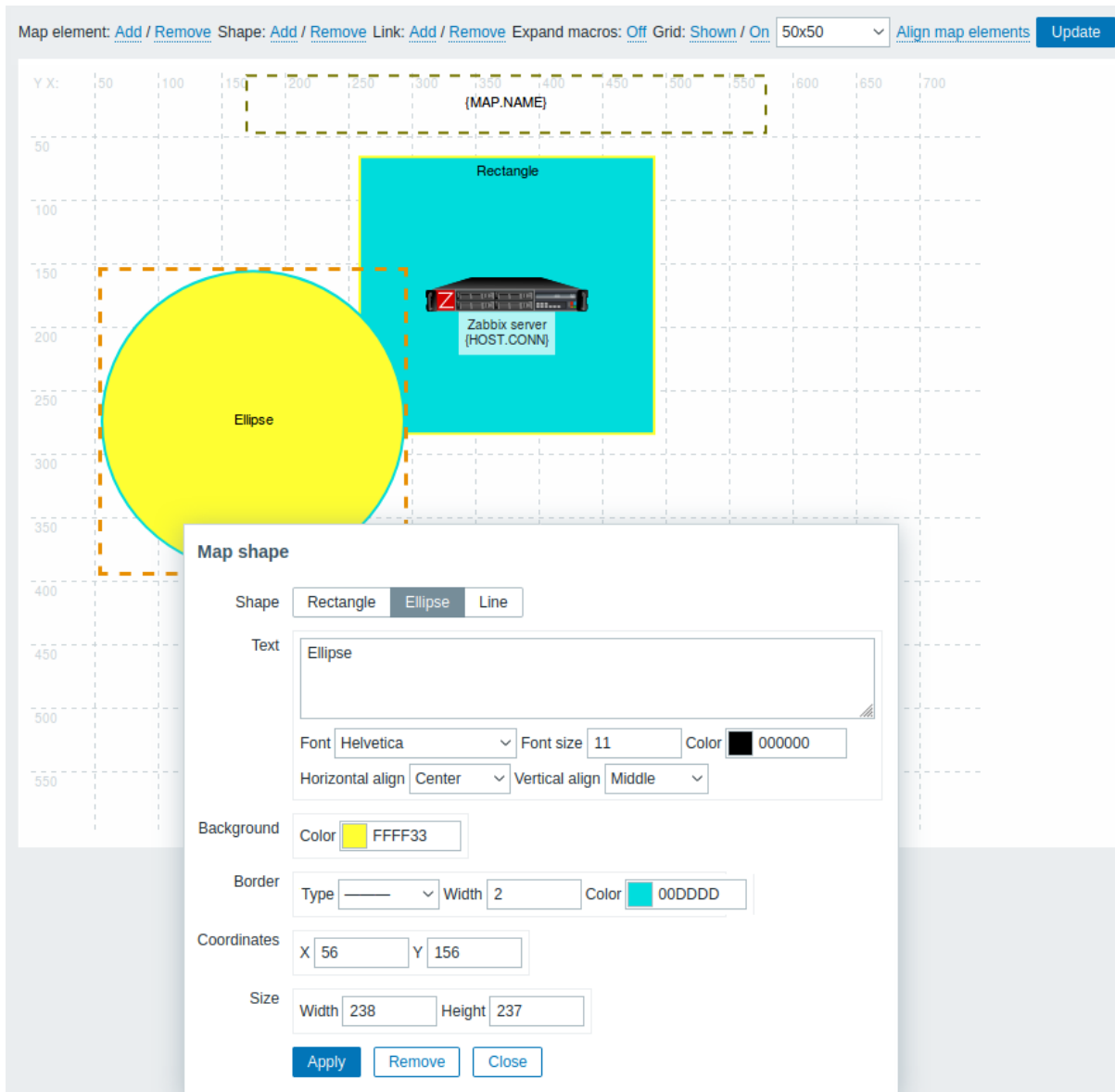
コピー&ペーストは、同じブラウザウィンドウ内で動作します。キーボードショートカットはサポートされていません。

図形の追加

地図要素に加えて、いくつかの図形を追加することができます。
 図形は地図の要素ではなく、視覚的な表現に過ぎません。例えば、長方形の図形は、いくつかのホストをグループ化するための
 背景として使用することができます。矩形と楕円の形状を追加することができます。

図形を追加するには、図形の隣にある Add をクリックします。新しい図形がマップの左上隅に表示されます。
 好きな場所にドラッグ&ドロップしてください。

新しいシェイプは、デフォルトの色で追加されます。シェイプをクリックすると、フォームが表示され、シェイプの見え方を
 カスタマイズしたり、テキストを追加したりすることができます。

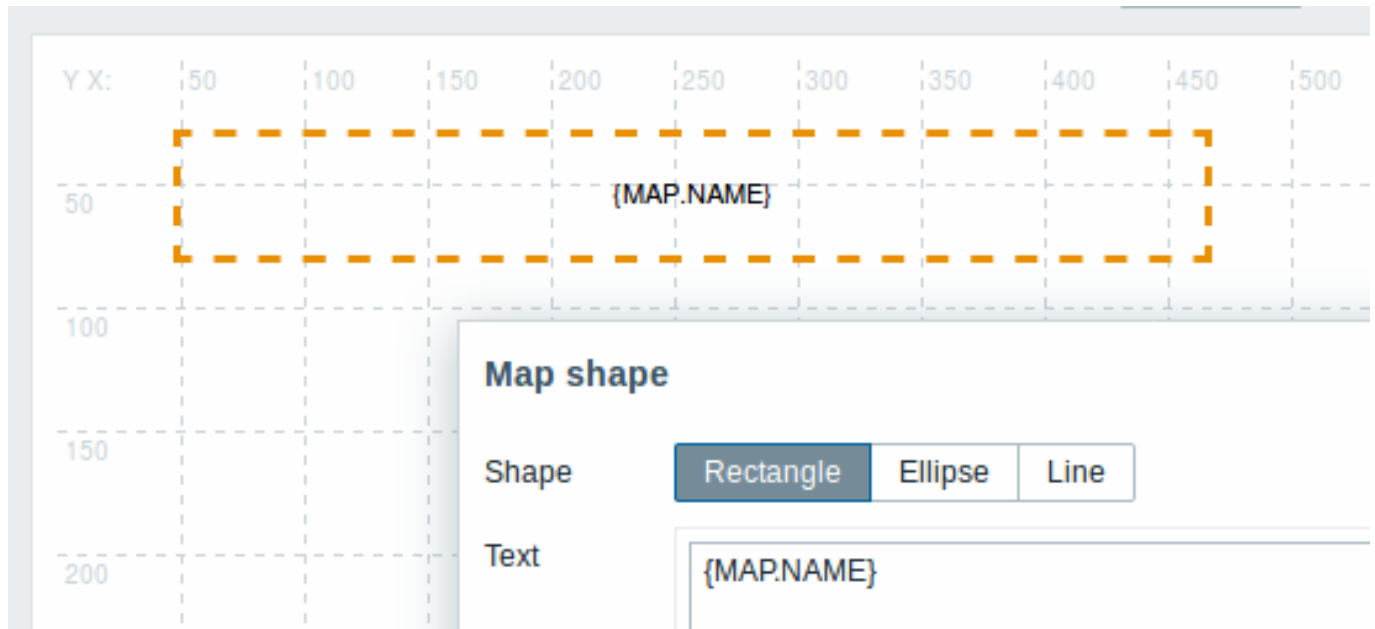


図形を選択するには、1つを選択し、Ctrl を押しながら他の図形を選択します。複数のシェイプを選択した状態で、
要素と同様に共通のプロパティを一括更新することができます。

シェイプ内にテキストを追加することができます。表現方法マクロはテキストでサポートされていますが、
時間をパラメータとした avg, last, min, max 関数のみです (例えば、{?avg(/host/key, 1h)})

テキストだけを表示するには、シェイプのボーダーを削除して非表示にすることができます。(Border フィールドで'None' を選択)

例えば、上のスクリーンショットで表示されている {MAP.NAME} マクロは、実際にはテキスト付きの矩形形状です。
マクロをクリックすると、テキスト付きの長方形のシェイプが表示されます。:



{MAP.NAME} は、マップを表示するときに設定されたマップ名に解決されます。

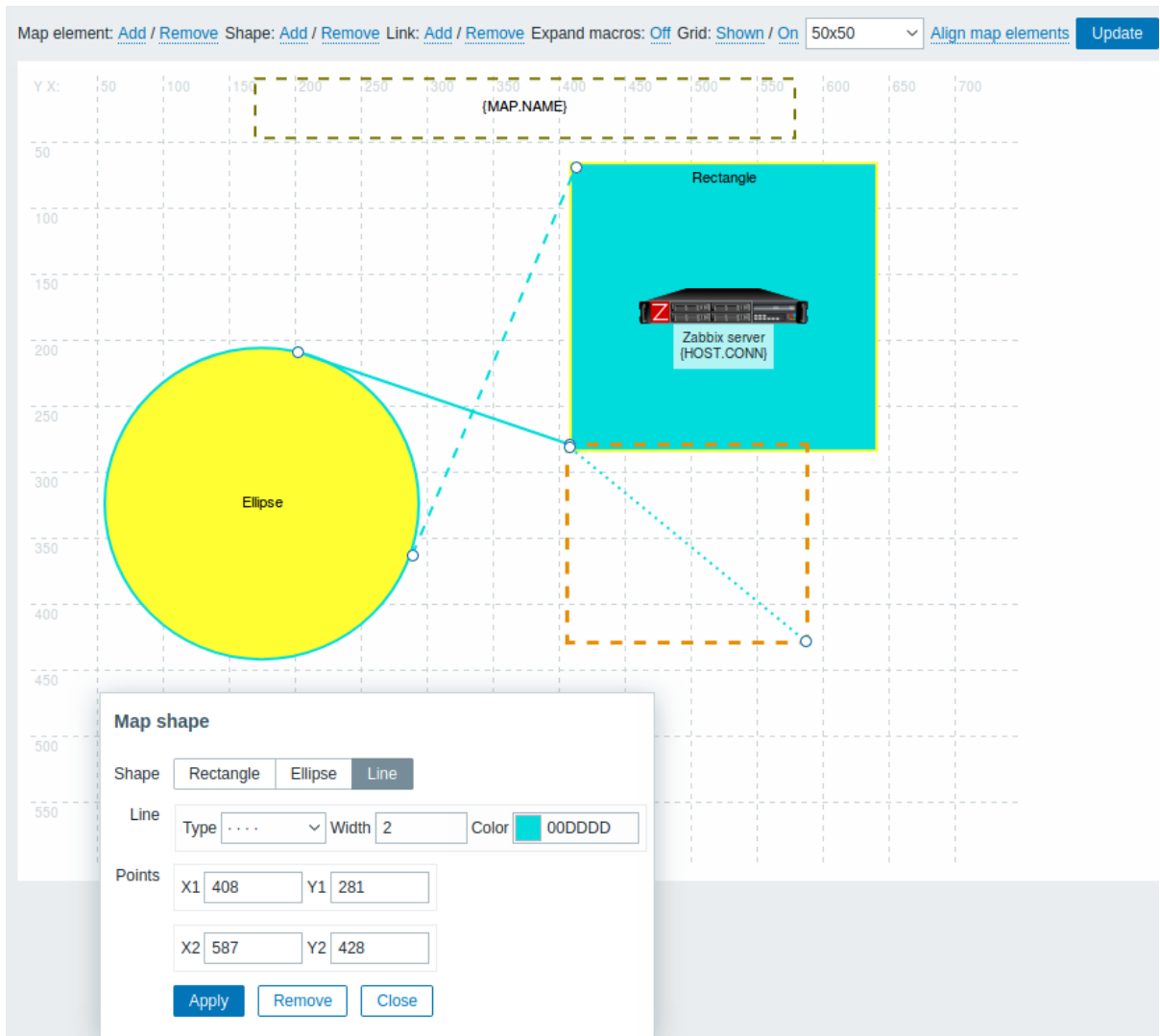
テキストでハイパーリンクが使用されている場合、マップを表示するときにクリックできるようになります。

図形内では、テキストの行の折り返しは常に"on"になっています。ただし、楕円の中では、楕円が矩形であるかのように
 線が折り返されます。単語の折り返しは実装されていないので、長い単語（シェイプに合わない単語）は折り返されず、
 マスク（コンストラクタ・ページ）またはクリップ（マップを含む他のページ）されます。

線の追加

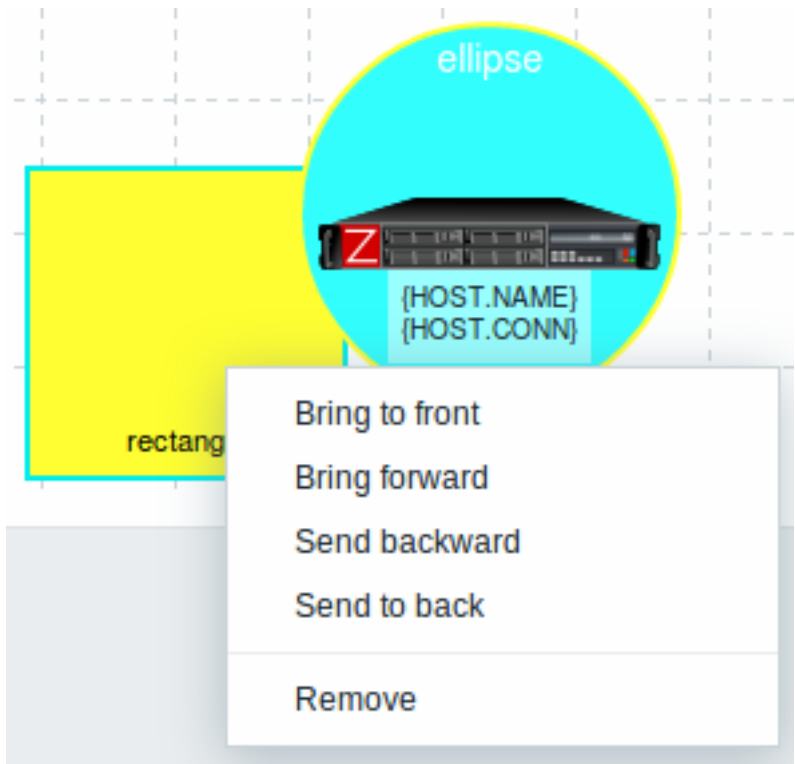
図形だけでなく、線を追加することも可能です。線は、地図上の要素や図形をつなぐために使用します。

線を追加するには、図形の隣にある Add をクリックします。新しい図形がマップの左上隅に表示されます。それを選択し、
 編集フォームの Line をクリックして、図形を線に変更します。次に、線の種類、幅、色など、線のプロパティを調整します。



図形と線の順序付け

ある図形を他の図形より前に（またはその逆に）持ってくるには、その図形をマウスの右ボタンでクリックし、地図の
 図形メニューを表示させます。



2 ホストグループの構成要素

概要

network map を設定する際に、“Host group” タイプの要素を追加する方法について説明します。

設定

Map element: [Add](#) / [Remove](#) Shape: [Add](#) / [Remove](#) Link: [Add](#) / [Remove](#) Expand macros: [Off](#) Grid: [Shown](#) / [On](#) 50x50 [Align map elements](#)

Y X: 50 100 150 200 250 Local network 2 400 450 500 550 600 650

50
100
150
200
250
300
350
400

Servers

[HOST.HOST]

Map element

Type

Show

Area type

Area size Width Height

Placing algorithm

Label

Label location

* Host group

Application

すべての必須入力フィールドには赤いアスタリスクが付いています。

この表は、ホストグループ要素タイプの典型的なパラメーターで構成されています。

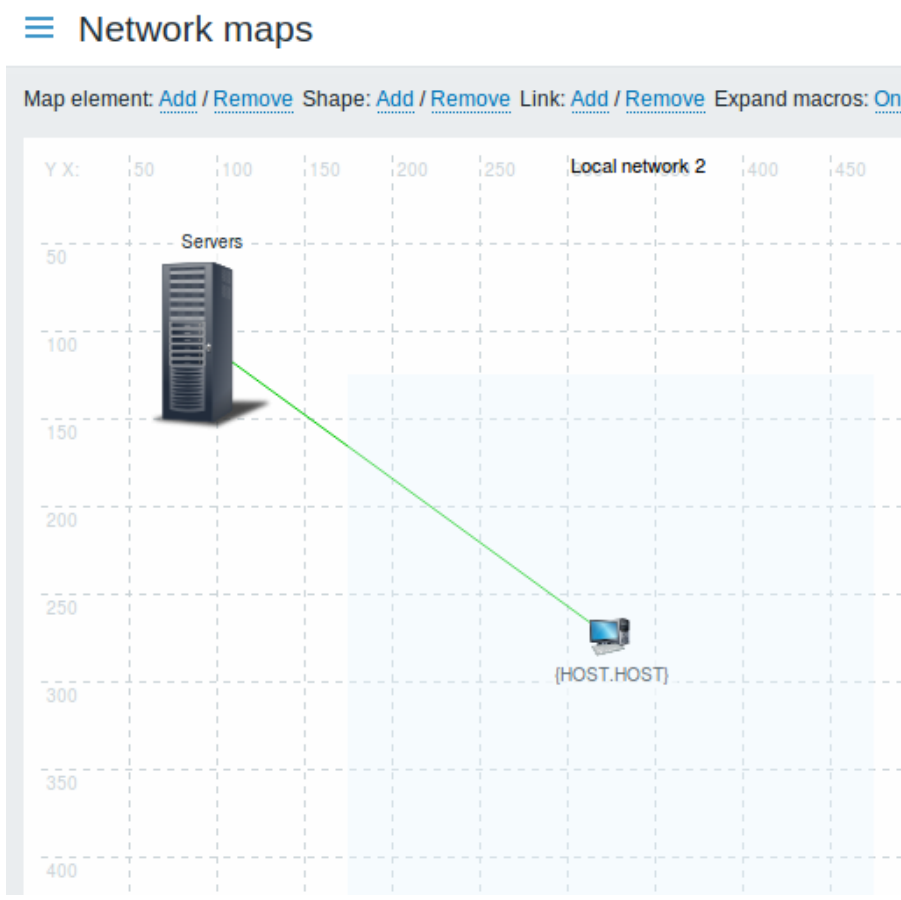
パラメータ	説明
タイプ	要素のタイプを選択: ホストグループ - 選択したグループに属するすべてのホストのすべてのトリガーのステータスを表示アイコン
表示	オプションの表示: ホストグループ - このオプションを選択すると、特定のホストグループに関する対応する情報が1つのアイコンとして表示されます ホストグループ内のホスト - このオプションを選択すると、特定のホストグループのすべてのホストに関する対応する情報が、複数のアイコンで表示されます

パラメータ	説明
エリアタイプ	この設定は" ホストグループ内のホスト" パラメーターが選択されている場合に使用できます。 マップに合わせる - すべてのホストグループ要素がマップ内に均等に配置されます カスタムサイズ - すべてのホストグループ要素を表示するためのマップエリアを手動設定します
エリアサイズ	この設定は" ホストグループ内のホスト" パラメータと" エリアタイプ" パラメータの" カスタムサイズ" が選択されている場合に使用できます。 幅 - マップエリアの幅を数値で指定 高さ - マップエリアの高さを数値で指定
配置アルゴリズム	グリッド - すべてのホストグループ要素を表示する唯一の利用可能なオプション
ラベル	アイコンラベル。任意の文字列。 マクロ および複数行の文字列をラベルで使用できます。 マップ要素のタイプが" ホストグループ" の場合、特定のマクロを指定すると、すべてのホストに関する対応する情報を表示するマップビューに影響を与えます。
たとえば、{HOST.IP} マクロが使用されている場合、編集マップビューにはマクロ {HOST.IP} のみが表示され、マップビューには各ホストの一意の IP アドレスが含まれて表示されます。	

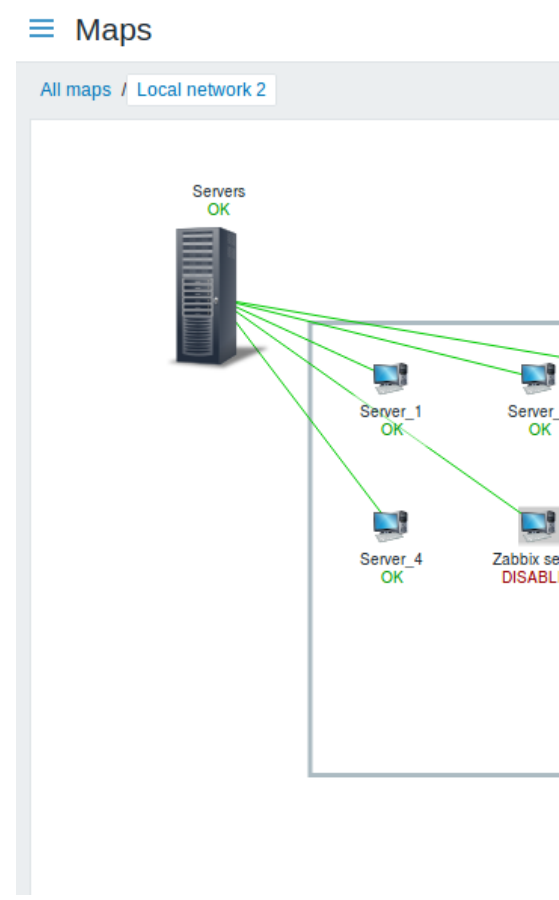
ホストグループ要素の表示

このオプションは、"Host group elements" の表示オプションが選択されている場合に使用できます。
 ホストグループの要素を show するを選択した場合、最初はホストグループのアイコンが 1 つだけ表示されます。
 しかしマップを保存してマップビューを表示すると、特定のホストグループに属するすべての要素 (ホスト) がマップに含まれて
 いることがわかります:

Map editing view



Map view



{HOST.NAME} マクロがどのように使用されているかに注目してください。マップ編集ではマクロ名は未解決ですが、
 マップビューではホストの固有名がすべて表示されます。

3 リンクインジケータ

概要

ネットワーク地図の要素間のlinkに、いくつかの trigger を
 割り当てることができます。これらの trigger が問題状態になると、リンクにそれを反映させることができます。

リンクを設定するときは、デフォルトのリンクの種類と色を設定します。リンクに trigger を割り当てるとき、これらの
 trigger で異なるリンクタイプと色を割り当てることができます。

これらの trigger のいずれかが問題状態になると、そのリンクスタイルと色がリンクに表示されます。
 つまり、デフォルトのリンクは緑の線だったとします。今、trigger が問題状態になると、リンクは赤の太字になるかもしれません。
 (そのように定義している場合)

設定

trigger をリンクインジケータとして割り当てるには、次のようにします。

- マップ要素を選択します。
- 適切なリンクの Links セクションで Edit をクリックします。
- Link indicators ブロックの Add をクリックし、1つまたは複数の trigger を選択します。

The screenshot displays the 'Network maps' interface. On the left, a network map shows a 'Local network' with a 'New element' (server icon) connected to a 'Zabbix server 127.0.0.1' (server icon). A link between them is labeled '100MBps'. Other elements include a 'Remote host group', a 'Firewall', and a 'Proxy'. On the right, the 'Map element' configuration panel is open. It shows the following settings:

- Type: Host
- Label: New element
- Label location: Default
- * Host: Server1
- Tags: And/Or, Or, Tag, Contains, f, Tag, Equals, fff
- Automatic icon selection:
- Icons: Default (Server_(64)), Problem (Default), Maintenance (Default), Disabled (Default)
- Coordinates: X 59, Y 44
- URLs: Name, URL
- Links: Element name (Zabbix server), Link indicators (New host (former tech name: Server4): Trap trigger)
- Label: 100MBps
- Connect to: Zabbix server
- Type (OK): Bold line
- Color (OK): 00CC00
- Link indicators: Trigger (New host (former tech name: Server4): Trap trigger), Type (Line), Color (DD0000)

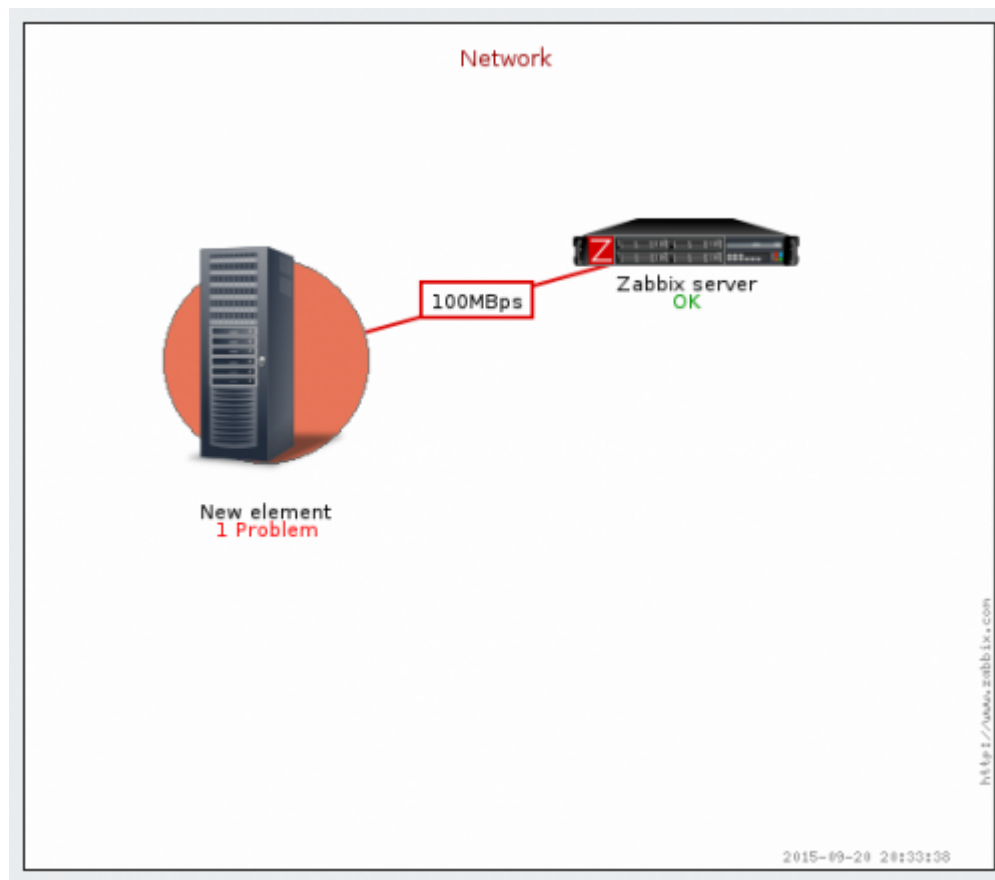
すべての必須入力項目には赤いアスタリスクが表示されます。

追加された trigger は、Link indicators のリストで確認することができます。

各 trigger のリンクタイプと色は、一覧から直接設定できます。完了したら、Apply をクリックし、フォームを閉じて、
 Update をクリックしてマップの変更を保存します。

表示

Monitoring → Maps で、trigger が問題状態になった場合、リンク上にそれぞれの色が表示されます。



Note:

複数のトリガーが問題状態になった場合、深刻度が最も高い問題がリンクのスタイルと色を決定します。
 同じ深刻度を持つ複数の trigger が同じマップリンクに割り当てられた場合、ID が最も低いものが優先されます。
 また、以下の点にも注意してください:

1. マップ構成の Minimum trigger severity および Show suppressed problem 設定は、どの問題が考慮されるかに影響します。

2. 複数の問題を持つ trigger (複数問題生成) の場合、各問題は trigger の (手動で変更された) 重要度と異なる重要度を持ち、
 異なるタグを持ち、抑制される可能性があります。

3 ダッシュボード

Dashboardsとそのウィジェットは、最新のグラフ、地図、スライドショーなど、強力な視覚化のプラットフォームを提供します。



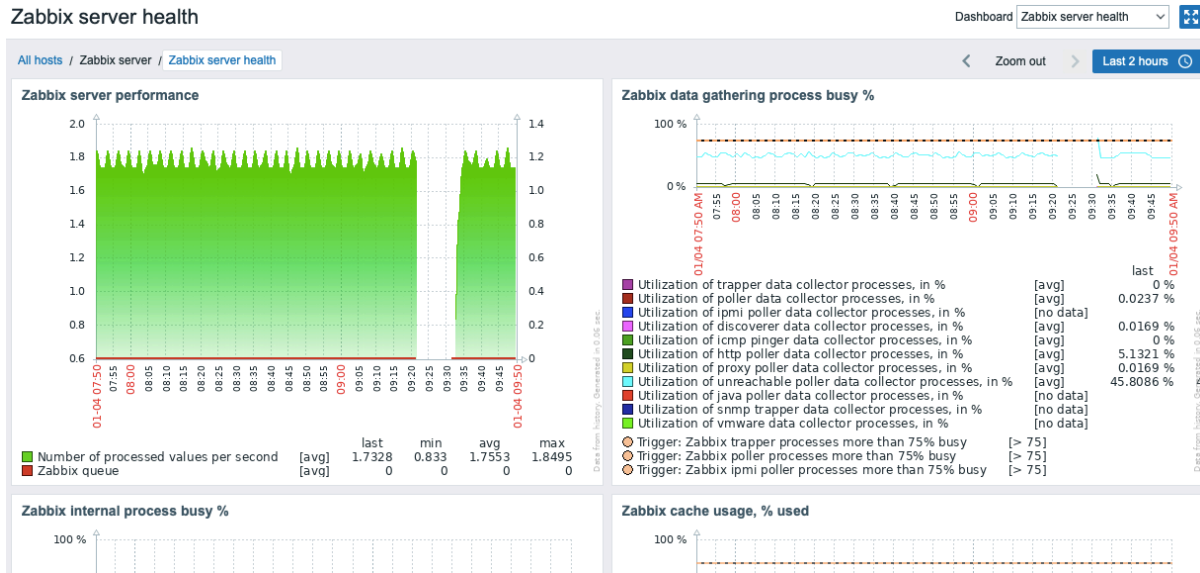
4 ホストダッシュボード

概要

ホストダッシュボードは、[global dashboards] (/manual/web_interface/frontend_sections/monitoring/dashboard) に似ていますが、ホストダッシュボードは
 ホストに関するデータのみを表示します。ホスト・ダッシュボードにはオーナーがいません。

ホスト・ダッシュボードは **template** レベルで設定され、
 テンプレートがホストにリンクされると、そのホスト用に生成されます。ホスト・ダッシュボードのウィジェットは、
 同じテンプレートのホスト・ダッシュボードにのみコピーすることができます。
 グローバル・ダッシュボードのウィジェットは、ホスト・ダッシュボードにコピーできません。

ホストダッシュボードは、グローバルダッシュボード用に予約されている **Monitoring** → **Dashboard** セクションで構成することも
 直接アクセスすることもできません。ホスト・ダッシュボードにアクセスする方法は、このセクションの下に記載されています。



ホストダッシュボードを表示する際、右上のドロップダウンを使用して、設定されたダッシュボードを切り替えることが
 できます。Monitoring→Hosts セクションに切り替えるには、左上隅のダッシュボード名の下にある
 All hosts ナビゲーションリンクをクリックします。

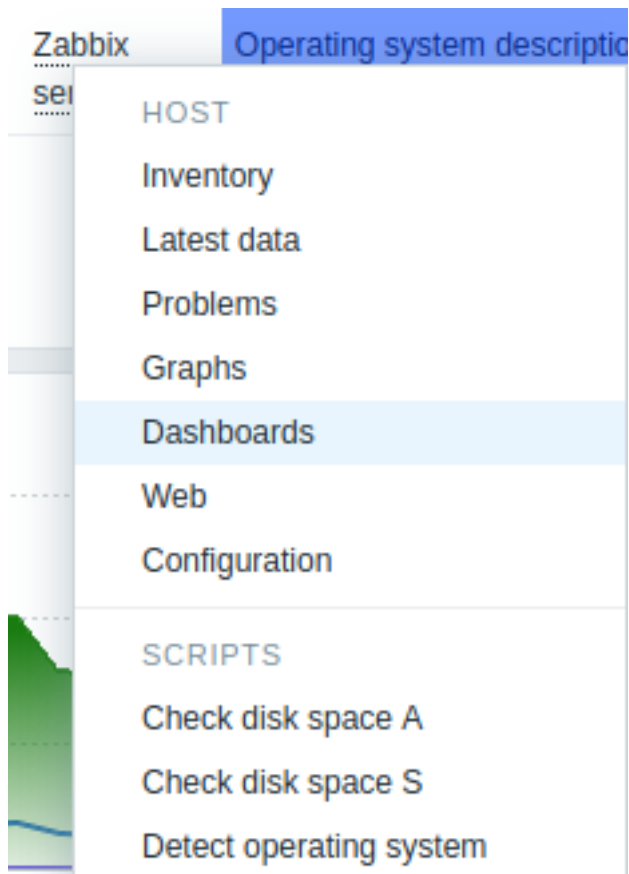
ホストダッシュボードのウィジェットを編集することはできません。

Zabbix 5.2 以前では、ホストダッシュボードはホスト画面であったことに注意してください。スクリーンを含む
 古いテンプレートをインポートする場合、スクリーンのインポートは無視されます。

ホストダッシュボードへのアクセス

ホストダッシュボードへのアクセスを提供します:

- フロントエンドの多くの場所にある **host menu** から
 - ホスト名をクリックし、ドロップダウンメニューから **Dashboards** を選択します。



- [global search](#)でホスト名を検索する場合
 - 検索結果に表示される Dashboards のリンクをクリックします。
- Inventory → [Inventory](#) でホスト名をクリックする場合
 - 提供されている Dashboards のリンクをクリックします。

8 テンプレート

概要

テンプレートは、複数のホストに便利に適用できるエンティティのセットです。

エンティティは以下のようなものがあります。

- item
- trigger
- グラフ
- ダッシュボード
- ローレベルディスクカバリルール
- ウェブシナリオ

現実の多くのホストは同一かかなり類似しているので、あるホスト用に作成したエンティティ (item, trigger, グラフなど) の `
` セットは、多くのホストにとって有用であることが自然に理解されます。もちろん、新しいホストにそれらをコピーすることもできますが、`
` それは多くの手作業が必要です。その代わりに、テンプレートを使えば、それらを 1 つのテンプレートにコピーし、そのテンプレートを `
` 必要なだけのホストに適用することができます。 `
`

テンプレートがホストにリンクされると、そのホストのすべてのエンティティ (item, trigger, グラフ, ...) がリンクされます。 `
` テンプレートは、個々のホストに直接割り当てられます。(ホストグループには割り当てられません)

テンプレートは、特定のサービスやアプリケーション (Apache, MySQL, PostgreSQL, Postfix など) のエンティティをグループ化し、 `
` それらのサービスを実行しているホストに適用するためによく使われます。

テンプレートを使用するもう一つの利点は、すべてのホストに対して何かを変更しなければならない場合です。テンプレートレベルで `
` 一度何かを変更すれば、リンクされているすべてのホストにその変更が伝搬されます。

このように、テンプレートの使用は作業負荷を軽減し、Zabbix の設定を合理化する優れた方法です。 `
`

テンプレートの作成と設定 [creating and configuring a template](#) へ進んでください。

9 すぐに使えるテンプレート (Templates out of the box)

概要

Zabbix は、すぐに使える便利な **templates** を提供するように努めています。すぐに使えるテンプレートは `
` あらかじめ設定されているため、監視ジョブのデプロイを高速化するのに便利な方法です。

テンプレートは利用可能です:

- 新しいインストールでは、Configuration → Templates にあります。
- 以前のバージョンからアップグレードする場合、これらのテンプレートはダウンロードした最新バージョンの Zabbix の `templates` ディレクトリにあります。Configuration → Templates で、このディレクトリからテンプレートを手動でインポートすることができます。
- [Zabbix git repository](#) から直接テンプレートをダウンロードすることも可能です。(テンプレートがお使いの Zabbix のバージョンと互換性があることを確認してください)

特定のテンプレートの種類と動作要件に関する情報にはサイドバーからアクセスしてください。

こちらもご参照ください。

- テンプレートのインポート [Template import](#)
- テンプレートのリンク [Linking a template](#)(/manual/config/templates/linking#linking_a_template)

HTTP テンプレートの操作

HTTP agent でメトリクスを収集するテンプレートが正しく動作するための手順です:

1. Zabbix でホストを作成し、メインインターフェイスとして監視対象の IP アドレスまたは DNS 名を指定します。 `
` これは、テンプレートアイテムの `{HOST.CONN}` マクロが正しく解決されるために必要です。 `
` 2. ステップ 1 で作成したホストにテンプレートを **Link** します。 `
` (テンプレートが Zabbix インストールにない場合、最初にテンプレートの `.xml` ファイルをインポートする必要があります。 `
` 手順については [Templates out-of-the-box](#) をご覧ください。) `
` 3. 必要に応じて、必須マクロの値を調整します。 `
` 4. Zabbix とのデータ共有を許可するように監視対象のインスタンスを設定します (Additional steps/comment 欄の手順を参照) `
`

Note:

このページでは、テンプレートが正しく動作するために必要な最低限のマクロと設定手順のみを掲載しています。 `
` テンプレートの詳細な説明 (マクロ、item、trigger の全リストを含む) は、テンプレートの `Readme.md` ファイル `
` (テンプレート名をクリックするとアクセスできます) に記載されています。 `
`

テンプレート	必須マクロ	Additional steps/comments
Apache by HTTP	{\$APACHE.STATUS.HOST} - the hostname or IP address of Apache status page (default: 127.0.0.1). {\$APACHE.STATUS.PATH} - the URL path (default: server-status?auto). {\$APACHE.STATUS.PORT} - the port of Apache status page (default: 80). {\$APACHE.STATUS.SCHEME} - the request scheme. Supported: http (default), https.	Apache module <code>mod_status</code> should be set (see Apache documentation for details). To check availability, run: <code>httpd -M 2>/dev/null \ grep status_module</code> Apache configuration example: <code><Location "/server-status"> SetHandler server-status Require host example.com </Location></code>
Asterisk by HTTP	{\$AMI.PORT} - AMI port number for checking service availability (default: 8088). {\$AMI.SECRET} - the Asterisk Manager secret (default: zabbix). {\$AMI.URL} - the Asterisk Manager API URL in the format <code><scheme>://<host>:<port>/<prefix>/rawman</code> (default: <code>http://asterisk:8088/asterisk/rawman</code>). {\$AMI.USERNAME} - the Asterisk Manager name.	1. Enable the mini-HTTP Server . 2. Add the option <code>webenabled=yes</code> to the general section of <code>manager.conf</code> file. 3. Create Asterisk Manager user in the Asterisk instance.

テンプレート	必須マクロ	Additional steps/comments
ClickHouse by HTTP	<p>`\${CLICKHOUSE.PORT}` - the port of ClickHouse HTTP endpoint (default: 8123).</p> <p>`\${CLICKHOUSE.SCHEME}` - the request scheme. Supported: http (default), https.</p> <p>`\${CLICKHOUSE.USER}`, `\${CLICKHOUSE.PASSWORD}` - ClickHouse login credentials (default username: zabbix, password: zabbix_pass).</p> <p>If you don't need authentication, remove headers from HTTP agent type items.</p>	<p>Create a ClickHouse user with a 'web' profile and permission to view databases (see ClickHouse documentation for details).</p> <p>See template's Readme.md file for a ready-to-use zabbix.xml file configuration.</p>
Cloudflare by HTTP	<p>`\${CLOUDFLARE.API.TOKEN}` - Cloudflare API token value (default: '<change>').</p> <p>`\${CLOUDFLARE.ZONE_ID}` - Cloudflare Site Zone ID (default: '<change>').</p>	<p>Cloudflare API Tokens are available in the Cloudflare account under My Profile→ API Tokens.</p> <p>Zone ID is available in the Cloudflare account under Account Home → Site.</p>
CockroachDB by HTTP	<p>`\${COCKROACHDB.API.PORT}` - the port of CockroachDB API and Prometheus endpoint. (default: 8080).</p> <p>`\${COCKROACHDB.API.SCHEME}` - the request scheme. Supported: http (default), https.</p>	<p>Internal node metrics are collected from Prometheus <code>/_status/vars</code> endpoint.</p> <p>Node health metrics are collected from <code>/health</code> and <code>/health?ready=1</code> endpoints.</p> <p>The template doesn't require usage of session token.</p> <p>Depending on your CockroachDB version and configuration, some metrics may not be collected.</p> <p>In the Dell iDRAC interface of your server:</p> <ol style="list-style-type: none"> 1. Enable Redfish API . 2. Create a user for monitoring with read-only permissions.
DELL PowerEdge R720 by HTTP , DELL PowerEdge R740 by HTTP , DELL PowerEdge R820 by HTTP , DELL PowerEdge R840 by HTTP	<p>`\${API.URL}` - Dell iDRAC Redfish API URL in the format <code><scheme>://<host>:<port></code> (default: <Put your URL here>)</p> <p>`\${API.USER}`, `\${API.PASSWORD}` - Dell iDRAC login credentials (default: not set).</p>	
Elasticsearch Cluster by HTTP	<p>`\${ELASTICSEARCH.PORT}` - the port of the Elasticsearch host (default: 9200).</p> <p>`\${ELASTICSEARCH.SCHEME}` - the request scheme. Supported: http (default), https.</p> <p>`\${ELASTICSEARCH.USERNAME}`, `\${ELASTICSEARCH.PASSWORD}` - login credentials, required only if used for Elasticsearch authentication.</p>	
Envoy Proxy by HTTP	<p>`\${ENVOY.METRICS.PATH}` - the path from where to retrieve metrics in the Prometheus format (default: <code>/stats/prometheus</code>).</p> <p>`\${ENVOY.URL}` - Envoy Proxy instance URL (default: <code>http://localhost:9901</code>)</p>	<p>Depending on your Envoy Proxy instance version and configuration, some metrics may not be collected.</p>

テンプレート	必須マクロ	Additional steps/comments
Etcd by HTTP	<p>{\$ETCD.PORT} - the port used by Etcd API endpoint (default: 2379).</p> <p>{\$ETCD.SCHEME} - the request scheme. Supported: http (default), https.</p> <p>{\$ETCD.USER}, {\$ETCD.PASSWORD} - login credentials, required only if used for Etcd authentication.</p>	<p>Metrics are collected from /metrics endpoint; to specify the endpoint's location use <code>--listen-metrics-urls</code> flag (see Etcd documentation for details).</p> <p>To verify, whether Etcd is configured to allow metric collection, run: <code>curl -L http://localhost:2379/metrics</code></p> <p>To check, if Etcd is accessible from Zabbix proxy or Zabbix server run: <code>curl -L http://<etcd_node_address>:2379/metrics%</code></p>
GitLab by HTTP	<p>{\$GITLAB.PORT} - the port of GitLab web endpoint (default: 80)</p> <p>{\$GITLAB.URL} - GitLab instance URL (default: localhost)</p>	<p>The template should be added to each node with Etcd. This template works with self-hosted GitLab instances; metrics are collected from the /metrics endpoint.</p> <p>To access the metrics, the client IP address must be explicitly allowed (see GitLab documentation for details).</p> <p>Note, that certain metrics may not be available for a particular GitLab instance version and configuration.</p>
Hadoop by HTTP	<p>{\$HADOOP.NAMENODE.HOST} - the Hadoop NameNode host IP address or FQDN (default: NameNode).</p> <p>{\$HADOOP.NAMENODE.PORT} - the Hadoop NameNode web-UI port (default: 9870).</p> <p>{\$HADOOP.RESOURCEMANAGER.HOST} - the Hadoop ResourceManager host IP address or FQDN (default: ResourceManager).</p> <p>{\$HADOOP.RESOURCEMANAGER.PORT} - the Hadoop ResourceManager web-UI port (default: 8088).</p>	<p>Metrics are collected by polling the Hadoop API remotely using an HTTP agent and JSONPath preprocessing. Zabbix server (or proxy) executes direct requests to ResourceManager, NodeManagers, NameNode, DataNodes APIs.</p>
HAProxy by HTTP	<p>{\$HAPROXY.STATS.PATH} - the path of HAProxy Stats page (default: stats).</p> <p>{\$HAPROXY.STATS.PORT} - the port of the HAProxy Stats host or container (default: 8404).</p> <p>{\$HAPROXY.STATS.SCHEME} - the request scheme. Supported: http (default), https.</p>	<p>HAProxy Stats page should be set up (see HAProxy blog post for details or template's Readme.md for configuration example).</p>
HashiCorp Consul Cluster by HTTP	<p>{\$CONSUL.API.PORT} - Consul API port, used in node LLD (default: 8500).</p> <p>{\$CONSUL.API.SCHEME} - Consul API scheme, used in node LLD (default: http).</p> <p>{\$CONSUL.CLUSTER.URL} - Consul cluster URL (default: http://localhost:8500).</p> <p>{\$CONSUL.TOKEN} - Consul authorization token (default: <PUT YOUR AUTH TOKEN>).</p>	
HashiCorp Consul Node by HTTP	<p>{\$CONSUL.NODE.API.URL} - Consul instance URL (default: http://localhost:8500).</p> <p>{\$CONSUL.TOKEN} - Consul authorization token (default: <PUT YOUR AUTH TOKEN>).</p>	<p>Internal service metrics are collected from <code>/v1/agent/metrics</code> endpoint.</p> <p>Prometheus format must be enabled for export metrics. See Consul documentation for details.</p>

テンプレート	必須マクロ	Additional steps/comments
HashiCorp Vault by HTTP	<p>`\${VAULT.API.PORT}` - the port on which the Vault listens for API requests (default: 8200).</p> <p>`\${VAULT.API.SCHEME}` - the API request scheme. Supported: http (default), https.</p> <p>`\${VAULT.HOST}` - Vault host name (default: <PUT YOUR VAULT HOST>).</p> <p>`\${VAULT.TOKEN}` - Vault authorization token (default: <PUT YOUR AUTH TOKEN>).</p>	<ol style="list-style-type: none"> 1. Configure the Vault API (see official documentation for details). 2. Create a Vault service token, then copy and paste it into `\${VAULT.TOKEN}` macro value in Zabbix.
Hikvision camera by HTTP	<p>`\${HIKVISION_ISAPI_PORT}` - ISAPI port on a device (default: 80).</p> <p>`\${USER}`, `\${PASSWORD}` - camera login credentials (default username: admin, password: 1234).</p>	
InfluxDB by HTTP	<p>`\${INFLUXDB.API.TOKEN}` - InfluxDB API authorization token (default: "").</p> <p>`\${INFLUXDB.URL}` - InfluxDB instance URL in the format <scheme>://<host>:<port> (default: http://localhost:8086).</p>	<p>This template collects internal service metrics from the InfluxDB /metrics endpoint of self-hosted InfluxDB instances.</p> <p>See InfluxDB documentation for details.</p>
HPE MSA 2040 Storage by HTTP/HPE MSA 2060 Storage by HTTP	<p>`\${HPE.MSA.API.PORT}` - connection port for API (default: 443)</p> <p>`\${HPE.MSA.API.SCHEME}` - connection scheme for API. Supported: http, https (default)</p> <p>`\${HPE.MSA.API.USERNAME}`, `\${HPE.MSA.API.PASSWORD}` - API credentials (default username: zabbix, password:"").</p>	<p>Create a separate Storage user (for example, zabbix) with monitor role and specify username and password in template macros.</p>
HPE Primera by HTTP	<p>`\${HPE.PRIMERA.API.PORT}` - connection port for API (default: 443)</p> <p>`\${HPE.PRIMERA.API.SCHEME}` - connection scheme for WSAPI. Supported: http, https (default)</p> <p>`\${HPE.PRIMERA.API.USERNAME}`, `\${HPE.PIMERA.API.PASSWORD}` - WSAPI credentials (default username: zabbix, password:"").</p>	<p>Create a separate Storage user (for example, zabbix) with browse role and enable it for all domains.</p> <p>To start WSAPI server, log in to the CLI as a user whose role has the <code>wsapi_set</code> right, then run:</p> <pre>startwsapi</pre> <p>To check WSAPI state, run:</p> <pre>showwsapi</pre>
InfluxDB by HTTP	<p>`\${INFLUXDB.API.TOKEN}` - InfluxDB API authorization token (default: "").</p> <p>`\${INFLUXDB.URL}` - InfluxDB instance URL in the format <scheme>://<host>:<port> (default: http://localhost:8086).</p>	<p>This template collects internal service metrics from the InfluxDB /metrics endpoint of self-hosted InfluxDB instances.</p> <p>See InfluxDB documentation for details.</p>
Jenkins by HTTP	<p>`\${JENKINS.API.KEY}` - API key to access Metrics Servlet; required for common metrics (default: "").</p> <p>`\${JENKINS.API.TOKEN}` - API token for HTTP BASIC authentication; required for monitoring computers and builds (default: "").</p> <p>`\${JENKINS.URL}` - Jenkins URL in the format <scheme>://<host>:<port>; required for monitoring computers and builds (default: "").</p> <p>`\${JENKINS.USER}` - username for HTTP BASIC authentication; required for monitoring computers and builds (default: zabbix).</p>	<p>Metrics are collected by requests to Metrics API.</p> <p>For common metrics: install and configure Metrics plugin parameters according to the official documentation. Issue an API key for access to the Metrics Servlet, then use it as `\${JENKINS.API.KEY}` macro value.</p> <p>For monitoring computers and builds: create an API token for the Jenkins user that will be used for monitoring, then use it as `\${JENKINS.API.TOKEN}` macro value. See Jenkins documentation for details.</p>

テンプレート	必須マクロ	Additional steps/comments
Kubernetes API server by HTTP	<p>`\${KUBE.API.SERVER.URL}` - instance URL (default: <code>http://localhost:8086/metrics</code>).</p> <p>`\${KUBE.API.TOKEN}` - API authorization token (default: "").</p>	<p>The template requires Zabbix Helm Chart to be installed in your Kubernetes cluster.</p> <p>Internal metrics are collected from the <code>/metrics</code> endpoint.</p> <p>Use bearer API token for authorization. See Kubernetes documentation for details.</p>
Kubernetes Controller manager by HTTP	<p>`\${KUBE.CONTROLLER.SERVER.URL}` - instance URL (default: <code>http://localhost:10252/metrics</code>).</p> <p>`\${KUBE.CONTROLLER.TOKEN}` - API authorization token (default: "").</p>	<p>The template requires Zabbix Helm Chart to be installed in your Kubernetes cluster.</p> <p>Internal metrics are collected from the <code>/metrics</code> endpoint.</p> <p>Use bearer API token for authorization. See Kubernetes documentation for details.</p>
Kubernetes kubelet by HTTP	<p>`\${KUBE.KUBELET.URL}` - instance URL (default: <code>https://localhost:10250</code>).</p> <p>`\${KUBE.API.TOKEN}` - API authorization token (default: "").</p>	<p>The template requires Zabbix Helm Chart to be installed in your Kubernetes cluster.</p> <p>Internal metrics are collected from the <code>/metrics</code> endpoint.</p> <p>Use bearer API token for authorization. See Kubernetes documentation for details.</p>
Kubernetes nodes by HTTP	<p>`\${KUBE.API.ENDPOINT}` - Kubernetes API endpoint in the format <code><scheme>://<host>:<port>/api</code> (default: not set).</p> <p>`\${KUBE.API.TOKEN}` - API authorization token (default: "").</p>	<p>The template requires Zabbix Helm Chart to be installed in your Kubernetes cluster.</p> <p>To generate a service account token, run:</p> <pre>kubectl get secret zabbix-service-account -n zabbix -o jsonpath={.data.token} base64 -d</pre> <p>See Kubernetes documentation for details.</p>
Kubernetes Scheduler by HTTP	<p>`\${KUBE.SCHEDULER.SERVER.URL}` - instance URL (default: <code>http://localhost:10251/metrics</code>).</p> <p>`\${KUBE.SCHEDULER.TOKEN}` - Scheduler API authorization token (default: "").</p>	<p>The template contains additional macros, which can be used to filter out certain metrics of discovered worker nodes.</p> <p>The template requires Zabbix Helm Chart to be installed in your Kubernetes cluster.</p> <p>Internal metrics are collected from the <code>/metrics</code> endpoint.</p> <p>Use bearer API token for authorization. See Kubernetes documentation for details.</p>
Kubernetes cluster state by HTTP	<p>`\${KUBE.API.HOST}` - Kubernetes API host (default: not set).</p> <p>`\${KUBE.API.PORT}` - Kubernetes API port (default: 6443).</p> <p>`\${KUBE.API.TOKEN}` - API authorization token (default: "").</p>	<p>The template requires Zabbix Helm Chart to be installed in your Kubernetes cluster.</p> <p>Internal service metrics are collected from the <code>kube-state-metrics</code> endpoint.</p> <p>Use bearer API token for authorization. See Kubernetes documentation for details.</p>
Microsoft SharePoint by HTTP	<p>`\${SHAREPOINT.URL}` - portal page URL, for example <code>http://sharepoint.companyname.local/</code> (default: "").</p> <p>`\${SHAREPOINT.ROOT}` - a root directory; only the specified directory and all its subfolders will be monitored (default: <code>/Shared Documents</code>).</p> <p>`\${SHAREPOINT.USER}`, `\${SHAREPOINT.PASSWORD}` - SharePoint login credentials (default: not set).</p>	<p>The template contains additional macros, which can be used to filter out certain metrics of discovered worker nodes.</p> <p>The template contains additional macros, which can be used to filter out certain dictionaries and types during LLD process (see template's Readme.md for the description of available filter macros).</p>

テンプレート	必須マクロ	Additional steps/comments
NetApp AFF A700 by HTTP	<p>{ \$URL } - AFF700 cluster URL address (default: ' ')</p> <p>{ \$USERNAME }, { PASSWORD } - AFF700 login credentials (default: not set).</p>	Create a host for AFF A700 with cluster management IP as the Zabbix agent interface.
NGINX by HTTP	<p>{ \$NGINX.STUB_STATUS.HOST } - the hostname or IP address of NGINX stub_status host or container (default: localhost).</p> <p>{ \$NGINX.STUB_STATUS.PATH } - the path of NGINX stub_status page (default: basic_status).</p> <p>{ \$NGINX.STUB_STATUS.PORT } - the port of NGINX stub_status host or container (default: 80).</p> <p>{ \$NGINX.STUB_STATUS.SCHEME } - the request scheme. Supported: http (default), https.</p>	<p>'ngx_http_stub_status_module' should be set up (see NGINX documentation for details or template's Readme.md for configuration example).</p> <p>To check availability, run: <pre>nginx -V 2>&1 \ grep -o with-http_stub_status_module</pre></p>
NGINX Plus by HTTP	<p>{ \$NGINX.API.ENDPOINT } - NGINX Plus API URL in the format <code><scheme>://<host>:<port>/<location></code> (default: ' ').</p>	<ol style="list-style-type: none"> 1. Enable NGINX Plus API (see NGINX documentation for details). 2. Set the macro <code>{ \$NGINX.API.ENDPOINT }</code> 3. If required, use other template macros to filter out discovery operations and discover only required zones and upstreams.
OpenWeatherMap by HTTP	<p>{ \$OPENWEATHERMAP.API.ENDPOINT } - OpenWeatherMap API endpoint (default: <code>api.openweathermap.org/data/2.5/weather</code>)</p> <p>{ \$OPENWEATHERMAP.API.TOKEN } - OpenWeatherMap API key (default: ' ')</p> <p>{ \$LOCATION } - locations for which to retrieve the metrics (default: Riga)</p>	<p>For instructions on getting the API key, see OpenWeatherMap documentation.</p> <p>{ \$LOCATION } macro supports the following formats: geo coordinates - for example, 56.95,24.0833 location name - for example, Chicago OpenWeatherMap location ID - download the ID list zip/postal code with a country code - for example, 94040,us To specify multiple locations, use the delimiter. Example: 43.81821,7.76115 Riga 2643743 94040,us</p>
PHP-FPM by HTTP	<p>{ \$PHP_FPM.HOST } - a hostname or an IP of PHP-FPM status host or container (default: localhost).</p> <p>{ \$PHP_FPM.PING.PAGE } - PHP-FPM ping page path (default: ping).</p> <p>{ \$PHP_FPM.PORT } - the port of PHP-FPM status host or container (default: 80).</p> <p>{ \$PHP_FPM.PROCESS_NAME } - PHP-FPM process name (default: php-fpm).</p> <p>{ \$PHP_FPM.SCHEME } - the request scheme. Supported: http (default), https.</p> <p>{ \$PHP_FPM.STATUS.PAGE } - PHP-FPM status page path (default: status).</p>	<ol style="list-style-type: none"> 1. Open the php-fpm configuration file and enable the status page: <pre>pm.status_path = /status ping.path = /ping</pre> 2. Validate the syntax: <code>\$ php-fpm7 -t</code> 3. Reload the php-fpm service. 4. In the Nginx Server Block (virtual host) configuration file, add (see template's Readme.md for an expanded example with comments): <pre>location ~ ^/(status ping)\$ { access_log off; fastcgi_param SCRIPT_FILENAME \$document_root\$fastcgi_script_name; fastcgi_index index.php; include fastcgi_params; fastcgi_pass 127.0.0.1:9000; }</pre> 5. Check the syntax: <code>\$ nginx -t</code> 6. Reload Nginx 7. Verify: <code>curl -L 127.0.0.1/status</code>

テンプレート	必須マクロ	Additional steps/comments
Proxmox VE by HTTP	<p>`\${PVE.TOKEN.ID}` - API token that allows stateless access to most parts of the REST API (default: not set).</p> <p>`\${PVE.TOKEN.SECRET}` - secret key (default: not set).</p> <p>`\${PVE.URL.PORT}` - the port the server listens to (default: 8006).</p>	<p>Create a separate user for monitoring, then generate an API token for this user.</p> <p>Grant the following access levels to the token and user:</p> <p>Check: <code>["perm", "/", ["Sys.Audit"]]</code></p> <p>Check: <code>["perm", "/nodes/{node}", ["Sys.Audit"]]</code></p> <p>Check: <code>["perm", "/vms/{vmid}", ["VM.Audit"]]</code></p>
RabbitMQ cluster by HTTP	<p>`\${RABBITMQ.API.CLUSTER_HOST}` - the hostname or IP address of RabbitMQ cluster API endpoint (default: 127.0.0.1).</p> <p>`\${RABBITMQ.API.SCHEME}` - the request scheme. Supported: http (default), https.</p> <p>`\${RABBITMQ.API.USER}`, `\${RABBITMQ.API.PASSWORD}` - RabbitMQ login credentials (default username: zbx_monitor, password: zabbix).</p>	<p>Enable RabbitMQ management plugin (see RabbitMQ documentation).</p> <p>To create a RabbitMQ user with necessary permissions for monitoring, run:</p> <pre>rabbitmqctl add_user zbx_monitor <PASSWORD> " rabbitmqctl set_permissions -p / zbx_monitor %% " " " .*"%% rabbitmqctl set_user_tags zbx_monitor monitoring</pre> <p>If the cluster consists of several nodes, it is recommended to assign the cluster template to a separate balancing host. In case of a single-node installation, the cluster template can be assigned to the host with a node template.</p>
TiDB by HTTP	<p>`\${TIDB.PORT}` - The port of TiDB server metrics web endpoint (default: 10080)</p> <p>`\${TIDB.URL}` - TiDB server URL (default: localhost).</p>	<p>This template works with TiDB server of PingCAP TiDB cluster.</p> <p>Internal service metrics are collected from TiDB /metrics endpoint and TiDB monitoring API.</p>
TiDB PD by HTTP	<p>`\${TIDB.PORT}` - The port of TiDB server metrics web endpoint (default: 2379)</p> <p>`\${TIDB.URL}` - TiDB server URL (default: localhost).</p>	<p>This template works with PD server of PingCAP TiDB cluster.</p> <p>Internal service metrics are collected from PD /metrics endpoint and TiDB monitoring API.</p>
TiDB TiKV by HTTP	<p>`\${TIDB.PORT}` - The port of TiDB server metrics web endpoint (default: 20180)</p> <p>`\${TIDB.URL}` - TiDB server URL (default: localhost).</p>	<p>This template works with TiKV server of PingCAP TiDB cluster.</p> <p>Internal service metrics are collected from TiKV /metrics endpoint.</p>
Travis CI by HTTP	<p>`\${TRAVIS.API.TOKEN}` - Travis API Token (default: not set)</p> <p>`\${TRAVIS.API.URL}` - Travis API URL (default: api.travis-ci.com).</p>	<p>Travis API authentication token can be found in the User → Settings → API authentication section.</p> <p><code>`\${TRAVIS.API.URL}`</code> format for a private project is <code>api.travis-ci.com</code>.</p> <p><code>`\${TRAVIS.API.URL}`</code> format for an enterprise project is <code>api.example.com</code> (replace <code>example.com</code> with the domain Travis CI is running on).</p>
VMWare SD-WAN VeloCloud by HTTP	<p>`\${VELOCLOUD.TOKEN}` - VMware SD-WAN Orchestrator API Token (default: "").</p> <p>`\${VELOCLOUD.URL}` - VMware SD-WAN Orchestrator URL, for example, <code>velocloud.net</code> (default: "").</p>	<p>API token should be created in the VMware SD-WAN Orchestrator (see VMware documentation for details).</p>
ZooKeeper by HTTP	<p>`\${ZOOKEEPER.COMMAND_URL}` - admin.commandURL; the URL for listing and issuing commands relative to the root URL (default: <code>commands</code>).</p> <p>`\${ZOOKEEPER.PORT}` - admin.serverPort; the port the embedded Jetty server listens on (default: 8080).</p> <p>`\${ZOOKEEPER.SCHEME}` - the request scheme. Supported: http (default), https.</p>	<p>Metrics are collected from each ZooKeeper node by requests to AdminServer (enabled by default). See ZooKeeper documentation to enable or configure AdminServer.</p>

IPMI テンプレートの操作

IPMI テンプレートは特別な設定を必要としません。監視を開始するには、[link](#) でテンプレートをターゲットホストに移動します
 (テンプレートが Zabbix インストール内がない場合、最初にテンプレートの.xml ファイルをインポートする必要があります
 - 手順については [Templates out-of-the-box](#) の項をご参照ください。)

Note:

このページでは、テンプレートが正しく動作するために必要な最低限のマクロと設定手順のみを掲載しています。
 テンプレートの詳細な説明 (マクロ、item、trigger の全リストを含む) は、テンプレートの Readme.md ファイル
 (テンプレート名をクリックするとアクセスできます) に記載されています。

テンプレート	必須マクロ	Additional steps/comments
Chassis by IPMI	{\$IPMI.USER} , {\$IPMI.PASSWORD} - credentials for access to BMC (default: none)	-

JMX テンプレートの操作

JMXによってメトリクスを収集するテンプレートを正しく運用するためのステップ:

1. Zabbix [Java gateway](#) が正しくインストールされ、設定されていることを確認します。

2. ターゲットホストにテンプレートを [Link](#) します。
 ホストには、JMX インターフェースが設定されている必要があります。テンプレートが Zabbix のインストール先がない場合、最初に
 テンプレートの.xml ファイルをインポートする必要があります。 [out-of-the-box](#) セクションを参照してください。

3. 必要に応じて、必須マクロの値を調整します。

4. Zabbix とのデータ共有を許可するように監視対象のインスタンスを設定します。

Note:

このページでは、テンプレートが正しく動作するために必要な最低限のマクロと設定手順のみを掲載しています。
 テンプレートの詳細な説明 (マクロ、item、trigger の全リストを含む) は、テンプレートの Readme.md ファイル
 (テンプレート名をクリックするとアクセスできます) に記載されています。

テンプレート	必須マクロ	Additional steps/comments
Apache ActiveMQ by JMX	{\$ACTIVEMQ.PORT} - port for JMX (default: 1099). {\$ACTIVEMQ.USERNAME} , {\$ACTIVEMQ.PASSWORD} - login credentials for JMX (default username: admin, password: activemq).	JMX access to Apache ActiveMQ should be enabled and configured per instructions in the official documentation .
Apache Cassandra by JMX	{\$CASSANDRA.USER} , {\$CASSANDRA.PASSWORD} - Apache Cassandra login credentials (default username: zabbix, password: zabbix)	JMX access to Apache Cassandra should be enabled and configured per instructions in the official documentation .
Apache Kafka by JMX	{\$KAFKA.USER} , {\$KAFKA.PASSWORD} - Apache Kafka login credentials (default username: zabbix, password: zabbix)	JMX access to Apache Kafka should be enabled and configured per instructions in the official documentation .
Apache Tomcat by JMX	{\$TOMCAT.USER} , {\$TOMCAT.PASSWORD} - Apache Tomcat login credentials; leave blank if Tomcat installation does not require authentication (default: not set).	JMX access to Apache Tomcat should be enabled and configured per instructions in the official documentation (choose the correct version).

テンプレート	必須マクロ	Additional steps/comments
GridGain by JMX	`\${GRIDGAIN.USER}` , `\${GRIDGAIN.PASSWORD}` - GridGain login credentials (default username: zabbix, password: <secret>).	JMX access to GridGain In-Memory Computing Platform should be enabled and configured per instructions in the documentation .
Ignite by JMX	`\${IGNITE.USER}` , `\${IGNITE.PASSWORD}` - Apache Ignite login credentials (default username: zabbix, password: <secret>).	<p>Enable and configure JMX access to Apache Ignite.</p> <p>JMX tree hierarchy contains ClassLoader by default. Adding the following Java Virtual Machine option <code>-DIGNITE_MBEAN_APPEND_CLASS_LOADER_ID=false</code> will exclude one level with ClassLoader name.</p> <p>Cache and Data Region metrics can be configured as needed - see Ignite documentation for details.</p>
WildFly Domain by JMX	`\${WILDFLY.JMX.PROTOCOL}` - JMX scheme (default: remote+http) `\${WILDFLY.USER}` , `\${WILDFLY.PASSWORD}` - WildFly login credentials (default username: zabbix, password: zabbix).	<p>See also: Monitoring and Management Using JMX Technology</p> <ol style="list-style-type: none"> 1. Enable and configure JMX access to WildFly according to instructions in the official documentation. 2. Copy <code>jboss-client.jar</code> from <code>/(wildfly,EAP,Jboss,AS)/bin/client</code> into directory <code>/usr/share/zabbix-java-gateway/lib</code>. 3. Restart Zabbix Java gateway.
WildFly Server by JMX	`\${WILDFLY.JMX.PROTOCOL}` - JMX scheme (default: remote+http) `\${WILDFLY.USER}` , `\${WILDFLY.PASSWORD}` - WildFly login credentials (default username: zabbix, password: zabbix).	<ol style="list-style-type: none"> 1. Enable and configure JMX access to WildFly according to instructions in the official documentation. 2. Copy <code>jboss-client.jar</code> from <code>/(wildfly,EAP,Jboss,AS)/bin/client</code> into directory <code>/usr/share/zabbix-java-gateway/lib</code>. 3. Restart Zabbix Java gateway.

ODBC テンプレート操作

ODBC モニタリングを介してメトリックを収集するテンプレートが正しく動作することを確認するための手順:

1. 必要な ODBC ドライバーが Zabbix サーバーまたはプロキシにインストールされていることを確認してください。
2. [リンク](#) テンプレートをターゲットホストに送信します (Zabbix インストールでテンプレートを使用できない場合は、最初にテンプレートの.xml ファイルをインポートする必要がある場合があります) - 手順については[すぐに使えるテンプレート](#) セクションを参照してください
3. 必要に応じて必須マクロの値を調整します。マクロ値に配置されたパスワードにセミコロン (;) が含まれている場合は、中かっここで囲む必要があります。詳細については、[ODBC モニタリング](#)を参照してください。
4. 監視対象のインスタンスを構成して、Zabbix とデータを共有できるようにします。追加の手順/コメント列の手順を参照してください。

Note:

このページには、適切なテンプレート操作に必要な最小限のマクロセットとセットアップ手順のみが含まれています。マクロ、項目、およびトリガーの完全なリストを含む、テンプレートの詳細な説明は、テンプレートの `Readme.md` ファイル (テンプレート名をクリックするとアクセスできます) にあります。

テンプレート	必須マクロ	追加の手順/コメント
MSSQL by ODBC	<p>{\$MSSQL.DSN} - システムデータソース名 (デフォルト: <Put your DSN here>)</p> <p>{\$MSSQL.PORT} - Microsoft SQL Server の TCP ポート (デフォルト: 1433)</p> <p>{\$MSSQL.USER},</p> <p>{\$MSSQL.PASSWORD} - Microsoft SQL ログイン資格情報 (デフォルト: 未設定)</p>	<p>監視用の Microsoft SQL ユーザーを作成し、そのユーザーに次の権限を付与します: サーバー状態表示、任意の定義表示 (詳細は Microsoft SQL ドキュメント 参照)</p> <p>"Service's TCP port state" 項目は、{HOST.CONN} および {\$MSSQL.PORT} マクロを使用して、Microsoft SQL インスタンスの可用性をチェックします。</p>
MySQL by ODBC	<p>{\$MYSQL.DSN} - システムデータソース名 (デフォルト: <Put your DSN here>)</p> <p>{\$MYSQL.USER},</p> <p>{\$MYSQL.PASSWORD} - MySQL ログイン資格情報。パスワードは空白にすることができます (デフォルト: 未設定)</p>	<p>監視に使用される MySQL ユーザーに必要な権限を付与するには次を実行します:</p> <pre>GRANT USAGE,REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON %% *.* TO '<username>'@'%' ;%%</pre> <p>詳細についてはMYSQL ドキュメント を参照してください。</p>

テンプレート	必須マクロ	追加の手順/コメント
Oracle by ODBC	<p>{\$ORACLE.DSN} - システムデータソース名 (デフォルト:<Put your DSN here>)</p> <p>{\$ORACLE.PORT} - Oracle DB の TCP ポート (デフォルト: 1521)</p> <p>{\$ORACLE.USER},</p> <p>{\$ORACLE.PASSWORD} - Oracle ログイン資格情報 (デフォルト: 未設定)</p>	<p>1. 監視用の Oracle ユーザーを作成するには次を実行します: <pre>CREATE USER zabbix_mon IDENTIFIED BY <PASSWORD>;</pre> -- zabbix_mon ユーザーにアクセス権を付与します <pre>GRANT CONNECT, CREATE SESSION TO zabbix_mon; GRANT SELECT ON V_\$instance TO zabbix_mon; GRANT SELECT ON V_\$database TO zabbix_mon; GRANT SELECT ON v_\$sysmetric TO zabbix_mon; GRANT SELECT ON v\$recovery_file_dest TO zabbix_mon; GRANT SELECT ON v\$active_session_history TO zabbix_mon; GRANT SELECT ON v\$osstat TO zabbix_mon; GRANT SELECT ON v\$restore_point TO zabbix_mon; GRANT SELECT ON v\$process TO zabbix_mon; GRANT SELECT ON v\$datafile TO zabbix_mon; GRANT SELECT ON v\$pgastat TO zabbix_mon; GRANT SELECT ON v\$sgastat TO zabbix_mon; GRANT SELECT ON v\$log TO zabbix_mon; GRANT SELECT ON v\$archive_dest TO zabbix_mon; GRANT SELECT ON v\$asm_diskgroup TO zabbix_mon; GRANT SELECT ON sys.dba_data_files TO zabbix_mon; GRANT SELECT ON DBA_TABLESPACES TO zabbix_mon; GRANT SELECT ON DBA_TABLESPACE_USAGE_METRICS TO zabbix_mon; GRANT SELECT ON DBA_USERS TO zabbix_mon;</pre> </p> <p>2. ODBC がセッションパラメータで Oracle に接続していることを確認してください:NLS_NUMERIC_CHARACTERS= '.,'</p> <p>3. odbc.ini に新しいレコードを追加します: <pre>[\$ORACLE.DSN] Driver = Oracle 19 ODBC driver Servername = \$ORACLE.DSN DSN = \$ORACLE.DSN</pre> </p> <p>4. isql 経由で接続を確認します: <pre>isql \$TNS_NAME \$DB_USER \$DB_PASSWORD</pre> </p> <p>5. Oracle ENV を使用するための Zabbix サーバーまたは Zabbix プロキシを設定します。新しいファイルを編集または追加: /etc/sysconfig/zabbix-server、プロキシの場合: /etc/sysconfig/zabbix-proxy。次にファイルに以下の行を追加します。 <pre>export ORACLE_HOME=/usr/lib/oracle/19.6/client64 export PATH=\$PATH:\$ORACLE_HOME/bin export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:/usr/lib64:/usr/lib:\$ORACLE_HOME/bin export TNS_ADMIN=\$ORACLE_HOME/network/admin</pre> </p> <p>6. Zabbix サーバーまたはプロキシを再起動します。</p>

Zabbix agent 2 テンプレートの運用

Zabbix agent 2 でメトリクスを収集するテンプレートが正しく動作するためのステップです:

1. ホスト上に agent 2 がインストールされていること、およびインストールされたバージョンに必要なプラグインが含まれていることを
 確認します。場合によっては、以下の作業が必要になることがあります。
 agent 2 の [upgrade](#) を先に行う必要があります。

2. ターゲットホストにテンプレートを [Link](#) します。
 (テンプレートが Zabbix インストールにない場合、最初にテンプレートのインポートファイルをインポートする必要があるかもしれません。

 - 手順については [Templates out-of-the-box](#) を参照してください)

3. 必要に応じて、必須マクロの値を調整します。ユーザーマクロは、設定パラメータを上書きするために使用することができることに
 注意してください。

4. Zabbix とのデータ共有を許可するように監視対象のインスタンスを設定します。(Additional steps/comments 欄の説明を参照)

Attention:

Zabbix agent 2 テンプレートはプラグインと連動して動作します。基本的な設定は、ユーザーマクロを調整することで可能ですが
 [configuring the plugin](#) 自体によって、より詳細なカスタマイズを行うことができます。
 例えば、プラグインが名前付きセッションをサポートしている場合、設定ファイルで各エンティティに独自の URI、ユーザー名、パスワードを
 持つ名前付きセッションを指定することで、同じ種類の複数のエンティティ (例えば、MySQL1 と MySQL2) を監視することが可能です。

Note:

このページでは、テンプレートが正しく動作するために必要な最低限のマクロと設定手順のみを掲載しています。
 テンプレートの詳細な説明 (マクロ、item、trigger の全リストを含む) は、テンプレートの Readme.md ファイル
 (テンプレート名をクリックするとアクセスできます) に記載されています。

テンプレート名	必須マクロ	Additional steps/comments
Ceph by Zabbix agent 2	<p>{\$CEPH.API.KEY} - the API key (default: zabbix_pass). Required, if</p> <p>{\$CEPH.CONNSTRING} is a URI. Must be empty, if</p> <p>{\$CEPH.CONNSTRING} is a session name.</p> <p>{\$CEPH.CONNSTRING} - connection string; can be a session name or a URI defined in the following format:
 <protocol(host:port)>. For URI only HTTPS schema is supported. Examples: Prod, https://localhost:8003 (default)</p> <p>{\$CEPH.USER} - user to be used for monitoring (default:zabbix). Required, if</p> <p>{\$CEPH.CONNSTRING} is a URI. Must be empty, if</p> <p>{\$CEPH.CONNSTRING} is a session name.</p>	<p>Works with Ceph plugin; named sessions are supported.</p> <ol style="list-style-type: none"> 1. Configure the Ceph RESTful Module according to documentation. 2. Make sure a RESTful API endpoint is available for connection.
Docker	-	<p>Works with Docker plugin; named sessions are not supported.</p> <p>To set path to Docker API endpoint edit Plugins.Docker.Endpoint parameter in the agent 2 configuration file (default: Plugins.Docker.Endpoint=unix:///var/run/docker.sock)</p> <p>To test availability, run: zabbix_get -s docker-host -k docker.info</p>

テンプレート名	必須マクロ	Additional steps/comments
Memcached	<p>{\$MEMCACHED.CONN.URI} - connection string in the URI format; port is optional; password is not used. If not set, the plugin's default value is used: tcp://localhost:11211. Examples: tcp://127.0.0.1:11211, tcp://localhost, unix:/var/run/memcached.sock.</p>	<p>Works with Memcached plugin; named sessions are supported.</p> <p>To test availability, run: zabbix_get -s memcached-host -k memcached.ping</p>
MongoDB cluster by Zabbix agent 2	<p>{\$MONGODB.CONNSTRING} - connection string in the URI format; password is not used (default: tcp://localhost:27017). Can be a session name or a URI defined in the following format: %% <protocol(host:port)>%% For URI only TCP scheme is supported. Examples: MongoDB1, tcp://172.16.0.10</p> <p>{\$MONGODB.USER}, {\$MONGODB.PASSWORD} - MongoDB credentials (default: none). If not set and {\$MONGODB.CONNSTRING} is a URI, parameters from the configuration file will be used. Must be empty, if {\$MONGODB.CONNSTRING} is a session name.</p>	<p>Works with MongoDB plugin; named sessions are supported. For MongoDB configuration instructions, see plugins. To test availability, run: zabbix_get -s mongos.node -k 'mongodb.ping["{\$MONGODB.CONNSTRING}","{\$MONGODB.USER}"]</p>
MongoDB node by Zabbix agent 2	<p>{\$MONGODB.CONNSTRING} - connection string in the URI format; password is not used (default: tcp://localhost:27017). Can be a session name or a URI defined in the following format: %% <protocol(host:port)>%% For URI only TCP scheme is supported. Examples: MongoDB1, tcp://172.16.0.10</p> <p>{\$MONGODB.USER}, {\$MONGODB.PASSWORD} - MongoDB credentials (default: none). If not set and {\$MONGODB.CONNSTRING} is a URI, parameters from the configuration file will be used. Must be empty, if {\$MONGODB.CONNSTRING} is a session name.</p>	<p>Works with MongoDB plugin; named sessions are supported. For MongoDB configuration instructions, see plugins. To test availability, run: zabbix_get -s mongodb.node -k 'mongodb.ping["{\$MONGODB.CONNSTRING}","{\$MONGODB.USER}"]</p>

テンプレート名	必須マクロ	Additional steps/comments
MySQL by Zabbix agent 2	<p>{\$MYSQL.DSN} - the system data source name of the MySQL instance (default: <Put your DSN>).</p> <p>Can be a session name or a URI defined in the following format: %% <protocol(host:port or /path/to/socket)/>%% For URI only TCP and Unix schemas are supported. Examples: MySQL1, tcp://localhost:3306, tcp://172.16.0.10, unix:/var/run/mysql.sock</p> <p>{\$MYSQL.USER}, {\$MYSQL.PASSWORD} - MySQL credentials (default: none). Required, if {\$MYSQL.DSN} is a URI. Must be empty, if {\$MYSQL.DSN} is a session name.</p>	<p>Works with MySQL plugin; named sessions are supported.</p> <p>To grant required privileges to a MySQL user that will be used for monitoring, run: GRANT USAGE,REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO '<username>'@'%';</p> <p>See MySQL documentation for information about user privileges and Unix sockets.</p>

テンプレート名	必須マクロ	Additional steps/comments
Oracle by Zabbix agent 2	<p>{\$ORACLE.CONNSTRING} - connection string; can be a session name or a URI defined in the following format: <protocol(host:port or /path/to/socket)/> For URI only TCP schema is supported. Examples: Oracle1, tcp://localhost:1521</p> <p>{\$ORACLE.SERVICE} - Oracle Service name (default: ORA). Required, if {\$ORACLE.CONNSTRING} is a URI. Must be empty, if {\$ORACLE.CONNSTRING} is a session name.</p> <p>{\$ORACLE.USER}, {\$ORACLE.PASSWORD} - Oracle credentials (default username: zabbix, password: zabbix_password). Required, if {\$ORACLE.CONNSTRING} is a URI. Must be empty, if {\$ORACLE.CONNSTRING} is a session name.</p>	<p>Works with Oracle plugin; named sessions are supported.</p> <p>Install Oracle Instant Client. To create Oracle user with required privileges, run: CREATE USER zabbix_mon IDENTIFIED BY <PASSWORD>; -- Grant access to the zabbix_mon user. GRANT CONNECT, CREATE SESSION TO zabbix_mon; GRANT SELECT ON DBA_TABLESPACE_USAGE_METRICS TO zabbix_mon; GRANT SELECT ON DBA_TABLESPACES TO zabbix_mon; GRANT SELECT ON DBA_USERS TO zabbix_mon; GRANT SELECT ON SYS.DBA_DATA_FILES TO zabbix_mon; GRANT SELECT ON V\$ACTIVE_SESSION_HISTORY TO zabbix_mon; GRANT SELECT ON V\$ARCHIVE_DEST TO zabbix_mon; GRANT SELECT ON V\$ASM_DISKGROUP TO zabbix_mon; GRANT SELECT ON V\$DATABASE TO zabbix_mon; GRANT SELECT ON V\$DATAFILE TO zabbix_mon; GRANT SELECT ON V\$INSTANCE TO zabbix_mon; GRANT SELECT ON V\$LOG TO zabbix_mon; GRANT SELECT ON V\$OSSTAT TO zabbix_mon; GRANT SELECT ON V\$PGASTAT TO zabbix_mon; GRANT SELECT ON V\$PROCESS TO zabbix_mon; GRANT SELECT ON V\$RECOVERY_FILE_DEST TO zabbix_mon; GRANT SELECT ON V\$RESTORE_POINT TO zabbix_mon; GRANT SELECT ON V\$SESSION TO zabbix_mon; GRANT SELECT ON V\$SGASTAT TO zabbix_mon; GRANT SELECT ON V\$SYSMETRIC TO zabbix_mon; GRANT SELECT ON V\$SYSTEM_PARAMETER TO zabbix_mon;</p>
PostgreSQL Agent 2	<p>{\$PG.URI} - connection string; can be a session name or a URI defined in the following format: %% <protocol(host:port or /path/to/socket)/>%%. For URI only TCP and Unix schemas are supported. Examples: Postgres1, tcp://localhost:5432, tcp://172.16.0.10</p> <p>{\$PG.USER}, {\$PG.PASSWORD} - PostgreSQL credentials (default username: postgres, password:postgres). Required, if {\$PG.URI} is a URI. Must be empty, if {\$PG.URI} is a session name.</p>	<p>Works with PostgreSQL plugin; named sessions are supported.</p> <p>To create a user with required privileges, for PostgreSQL 10 and newer, run: CREATE USER 'zbx_monitor' IDENTIFIED BY '<password>'; GRANT EXECUTE ON FUNCTION pg_catalog.pg_ls_dir(text) TO zbx_monitor;\GRANT EXECUTE ON FUNCTION pg_catalog.pg_stat_file(text) TO zbx_monitor;</p> <p>Edit pg_hba.conf to allow connections from Zabbix agent (see PostgreSQL documentation for details).</p>

テンプレート名	必須マクロ	Additional steps/comments
Redis	{\$REDIS.CONN.URI} -connection string in the URI format; port is optional; password is not used. If not set, the plugin's default value is used: tcp://localhost:6379	Works with Redis plugin; named sessions are supported. To test availability, run: <code>zabbix_get -s redis-master -k redis.ping</code>
SMART by Zabbix agent 2 / SMART by Zabbix agent 2 active	-	Sudo/root access rights to smartctl are required for the user executing Zabbix agent 2. The minimum required smartctl version is 7.1. Disk discovery LLD rule finds all HDD, SSD, NVMe disks with S.M.A.R.T. enabled. Attribute discovery LLD rule finds all Vendor Specific Attributes for each disk. To skip some attributes, set regular expressions with disk names in <code>{\$SMART.DISK.NAME.MATCHES}</code> and with attribute IDs in <code>{\$SMART.ATTRIBUTE.ID.MATCHES}</code> on the host level. No specific configuration is required.
Systemd by Zabbix agent 2 Website certificate by Zabbix agent 2	- ** {\$CERT.WEBSITE.HOSTNAME}** - the website's DNS name for the connection (default: <Put DNS name>).	Works with WebCertificate plugin; named sessions are not supported. To test availability, run: <code>zabbix_get -s <zabbix_agent_addr> -k web.certificate.get [<website_dns_name>]</code> Create a separate host for the TLS/SSL certificate with Zabbix agent interface and link the template to this host.

Zabbix エージェントテンプレートの操作

Zabbix エージェントでメトリクスを収集するテンプレートが正しく動作するための手順:

1. Zabbix エージェントがホストにインストールされていることを確認します。アクティブチェックの場合、エージェント **構成ファイル** の 'ServerActive' パラメーターにホストが追加されていることも確認します。
2. テンプレートをターゲットホストに **リンク** します (Zabbix インストールでテンプレートを使用できない場合は、最初にテンプレートの.xml ファイルをインポートする必要がある場合があります - 手順については **すぐに使えるテンプレート** セクションを参照してください。)
3. 必要に応じて必須マクロの値を調整します。
4. 監視対象のインスタンスを設定して、Zabbix とデータを共有できるようにします。追加手順/コメント列の手順を参照してください。

Note:

このページには、適切なテンプレート操作に必要な最小限のマクロセットとセットアップ手順のみが含まれています。マクロ、項目、およびトリガーの完全なリストを含む、テンプレートの詳細な説明は、テンプレートの README.md ファイル (テンプレート名をクリックするとアクセスできます) にあります。

テンプレート名	必須マクロ	追加手順/コメント
Apache by Zabbix agent	<p>{\$APACHE.STATUS.HOST} - Apache ステータスページのホスト名または IP アドレス (デフォルト: 127.0.0.1)</p> <p>{\$APACHE.STATUS.PATH} - URL パス (デフォルト: server-status?auto)</p> <p>{\$APACHE.STATUS.PORT} - Apache ステータスページのポート (デフォルト: 80)</p>	<p>Apache モジュール mod_status を設定する必要があります (詳細については Apache ドキュメント を参照してください)。</p> <p>利用可能かどうかを確認するには次のコマンドを実行します。</p> <pre>httpd -M 2>/dev/null \ grep status_module</pre> <p>Apache 設定例:</p> <pre><Location "/server-status"> SetHandler server-status Require host example.com </Location></pre>
HAProxy by Zabbix agent	<p>{\$HAPROXY.STATS.PATH} - HAProxy Stats ページのパス (デフォルト: stats)</p> <p>{\$HAPROXY.STATS.PORT} - HAProxy Stats ホストまたはコンテナのポート (デフォルト: 8404)</p> <p>{\$HAPROXY.STATS.SCHEME} - HAProxy Stats ページのスキーム。サポート: http (デフォルト), https</p>	<p>HAProxy Stats ページを設定する必要があります (詳細については HAProxy ブログ投稿 を、構成例についてはテンプレートの Readme.md を参照してください)。</p>
IIS by Zabbix agent / IIS by Zabbix agent active	<p>{\$IIS.PORT} - IIS サーバーがリスンするポート (デフォルト: 80)</p> <p>{\$IIS.SERVICE} - ポートチェックのサービス (デフォルト: http)。詳細については net.tcp.service セクションを参照してください。</p>	<p>サーバーには次の役割が必要です:</p> <p>Web Server</p> <p>IIS Management Scripts and Tools</p> <p>詳細については IIS のドキュメントを参照してください。</p> <p>注, テンプレートは Windows サービスの状態に関する情報を提供しないことに注意してください。OS Windows by Zabbix agent または OS Windows by Zabbix agent active テンプレートと組み合わせて使用することをお勧めします。</p>
Microsoft Exchange Server 2016 by Zabbix agent/Microsoft Exchange Server 2016 by Zabbix agent active	<p>{\$NGINX.STUB_STATUS.HOST} - Nginx stub_status ホストまたはコンテナのホスト名または IP アドレス (デフォルト: localhost)</p> <p>{\$NGINX.STUB_STATUS.PATH} - Nginx stub_status ページのパス (デフォルト: basic_status)</p> <p>{\$NGINX.STUB_STATUS.PORT} - Nginx stub_status ホストまたはコンテナのポート (デフォルト: 80)</p>	<p>ngx_http_stub_status_module を設定する必要があります (詳細については Nginx ドキュメント を参照し、設定例についてはテンプレートの Readme.md を参照してください)</p> <p>可用性を確認するには次を実行します:</p> <pre>nginx -V 2>&1 \ grep -o with-http_stub_status_module</pre>

テンプレート名	必須マクロ	追加手順/コメント
PHP-FPM by Zabbix agent	<p>{\$PHP_FPM.HOST} - ホスト名または PHP-FPM ステータスホストまたはコンテナの IP (デフォルト: localhost)</p> <p>{\$PHP_FPM.PING.PAGE} - PHP-FPM ping ページのパス (デフォルト: ping)</p> <p>{\$PHP_FPM.PORT} - PHP-FPM ステータスホストまたはコンテナのポート (デフォルト: 80)</p> <p>{\$PHP_FPM.PROCESS_NAME} - PHP-FPM プロセス名 (デフォルト: php-fpm)</p> <p>{\$PHP_FPM.STATUS.PAGE} - PHP-FPM ステータスページのパス (デフォルト: status)</p>	<ol style="list-style-type: none"> 1. php-fpm 設定ファイルを開き、ステータスページを有効にします。: <pre>pm.status_path = /status ping.path = /ping</pre> 2. 構文を検証します: <code>\$ php-fpm7 -t</code> 3. php-fpm サービスをリロードします。 4. Nginx サーバブロック (仮想ホスト) 設定ファイルに追加します (コメント付きの拡張例についてはテンプレートの Readme.md を参照してください): <pre>location ~ ^/(status ping)\$ { access_log off; fastcgi_param SCRIPT_FILENAME \$document_root\$fastcgi_script_name; fastcgi_index index.php; include fastcgi_params; fastcgi_pass 127.0.0.1:9000; }</pre> 5. 構文を確認します: <code>\$ nginx -t</code> 6. Nginx をリロードします 7. 次のコマンドで確認します: <code>curl -L 127.0.0.1/status</code>
RabbitMQ cluster by Zabbix agent	<p>{\$RABBITMQ.API.CLUSTER_HOST} - RabbitMQ クラスター API エンドポイントのホスト名または IP アドレス (デフォルト: 127.0.0.1)</p> <p>{\$RABBITMQ.API.USER}, {\$RABBITMQ.API.PASSWORD} - RabbitMQ ログイン資格情報 (デフォルトのユーザー名: zbx_monitor, パスワード: zabbix)</p>	<p>RabbitMQ 管理プラグインを有効にします (RabbitMQ ドキュメント を参照してください)。</p> <p>監視に必要な権限を持つ RabbitMQ ユーザーを作成するには、次のコマンドを実行します:</p> <pre>" rabbitmqctl add_user zbx_monitor <PASSWORD> " rabbitmqctl set_permissions -p / zbx_monitor %% "" "" ".*"%% rabbitmqctl set_user_tags zbx_monitor monitoring</pre> <p>クラスターが複数のノードで構成されている場合は、クラスターテンプレートを別の分散ホストに割り当てることをお勧めします。単一ノードインストールの場合、ノードテンプレートを使用してクラスターテンプレートをホストに割り当てることができます。</p>

テンプレート名	必須マクロ	追加手順/コメント
MySQL by Zabbix agent	<p>{\$MYSQL.HOST} - MySQL ホスト またはコンテナのホスト名または IP アドレス (デフォルト: 127.0.0.1(6.0.8 以 前)/localhost(6.0.8 以降))</p> <p>{\$MYSQL.PORT} - データベース サービスポート (デフォルト: 3306)</p>	<p>1. 必要に応じて、mysql および mysqladmin ユーティ リティへのパスをグローバル環境変数 PATH に追加しま す。</p> <p>2. <code>template_db_mysql.conf</code> ファイルを Zabbix の templates ディレクトリから Zabbix エージェント構成 のフォルダ (デフォルトで は/etc/zabbix/zabbix_agentd.d/) にコピーし、Zabbix エージェントを再起動します。</p> <p>3. MySQL ユーザー <code>zbx_monitor</code> を作成します。ユ ーザーに必要な権限を付与するには、次を実行します: <pre>GRANT USAGE,REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON %% *.* TO '<username>'@'%' ;%%</pre> (詳細はMYSQL ドキュメンテーションを参照してくださ い).</p> <p>4. Linux の場合は Zabbix エージェントのホームディレ クトリ (デフォルトでは /var/lib/zabbix) に <code>.my.cnf</code> を 作成し、Windows の場合は <code>c:\</code> に <code>my.cnf</code> を作成しま す。ファイルには次の 3 つの文字列が必要です: <pre>[client] "user='zbx_monitor' " "password='<password>' "</pre> </p>

テンプレート名	必須マクロ	追加手順/コメント
PostgreSQL	<p>{PG.DB} - サーバーに接続するためのデータベース名 (デフォルト: postgres)</p> <p>{PG.HOST} - データベースサーバーホストまたはソケットディレクトリ (デフォルト: 127.0.0.1)</p> <p>{PG.PORT} - データベースサーバーポート (デフォルト: 5432)</p> <p>{PG.USER} - データベースユーザー名 (デフォルト: zbx_monitor)</p>	<p>1. PostgreSQL サーバーへの適切なアクセス権を持つ読み取り専用ユーザー <code>zbx_monitor</code> を作成します。PostgreSQL 10 以降の場合は次を実行します: <pre>CREATE USER zbx_monitor WITH PASSWORD '<PASSWORD>' INHERIT; GRANT pg_monitor TO zbx_monitor;</pre> For older PostgreSQL versions, run: <pre>CREATE USER zbx_monitor WITH PASSWORD '<PASSWORD>'; GRANT SELECT ON pg_stat_database TO zbx_monitor;</pre></p> <p>2. <code>postgresql/</code> を Zabbix エージェントのホームディレクトリ (<code>/var/lib/zabbix/</code>) にコピー</p> <p>3. Zabbix の <code>templates</code> ディレクトリから Zabbix エージェントの設定ディレクトリ (<code>/etc/zabbix/zabbix_agentd.d/</code>) に <code>template_db_postgresql.conf</code> をコピーし、Zabbix エージェントを再起動します。</p> <p>4. <code>pg_hba.conf</code> を編集して、Zabbix エージェントからの接続を許可します (詳細については PostgreSQL ドキュメント を参照してください)。 行の例: <pre>host all zbx_monitor 127.0.0.1/32 trust host all zbx_monitor 0.0.0.0/0 md5 host all zbx_monitor ::0/0 md5</pre></p> <p>5. リモートサーバーを監視するには、Zabbix エージェントのホームディレクトリ (<code>/var/lib/zabbix/</code>) に <code>.pgpass</code> ファイルを作成し、インスタンス、ポート、データベース、ユーザー、およびパスワード情報を含む行を追加します。(詳細については、PostgreSQL ドキュメント を参照してください)。 行の例: <pre><REMOTE_HOST1>:5432:postgres:zbx_monitor:<PASSWORD> *:5432:postgres:zbx_monitor:<PASSWORD></pre></p>

ネットワークデバイス用の標準テンプレート

概要

スイッチやルータのようなネットワークデバイスの監視を行うために、ネットワークデバイス自体 (基本的にはシャーシ) と ネットワークインターフェイスの 2 つのいわゆるモデルを作成しました。

Zabbix3.4 以降、多くのネットワークデバイスのテンプレートが提供されています。すべてのテンプレートは、以下の項目をカバーしています。 (デバイスからこれらの item を取得できる場合)

- Chassis fault monitoring (power supplies, fans and temperature, overall status)
- Chassis performance monitoring (CPU and memory items)
- Chassis inventory collection (serial numbers, model name, firmware version)
- Network interface monitoring with IF-MIB and EtherLike-MIB (interface status, interface traffic load, duplex status for Ethernet)

これらのテンプレートは利用可能です:

- 新しいインストールでは、Configuration → Templates にあります。
- 旧バージョンからアップグレードする場合、これらのテンプレートはダウンロードした最新バージョンの Zabbix の `templates` ディレクトリにあります。Configuration → Templates で、このディレクトリから手動でインポートすることができます。

新しいテンプレートをインポートする場合、@Network interfaces for discovery グローバル正規表現を次のように更新するとよいでしょう:

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[1L]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
Result is FALSE: ^[1L]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

こうすれば、ほとんどのシステムでループバックと null インターフェースをフィルタリングできます。

デバイス

テンプレートが利用可能なデバイスファミリーの一覧です:

テンプレート名	ベンダ	デバイスファミリー	モデル	MIBs OS used	Tags
Alcatel Timetra TIMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TIMOS- SYSTEM- MIB,TIMETRA- CHASSIS- MIB	Certified
Brocade FC SNMP	Brocade	Brocade FC switches	Brocade 300 SAN Switch-	SW- MIB,ENTITY- MIB	Performance, Fault
Brocade_FourStackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F	FOUNDRY- SN- AGENT- MIB, FOUNDRY- SN- STACKING- MIB	Certified
Brocade_FourNonstackable SNMP	Brocade	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424	FOUNDRY- SN- AGENT- MIB	Performance, Fault
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP	CISCO- MEMORY- POOL- MIB, IF-MIB, EtherLike- MIB, SNMPv2- MIB, CISCO- PROCESS- MIB, CISCO- ENVMON- MIB, ENTITY- MIB	Certified

テンプレート名	ベンダ	デバイスファミリー	モデル	OS	MIBs used	Tags
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T and 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certified
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX switches	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-265C	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certified
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Performance Fault Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810	-	F10-S-SERIES-CHASSIS-MIB	Certified
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certified
Huawei VRP SNMP	Huawei	Huawei VRP	S2352P-EI	-	ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certified
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infiniband devices	Infiniband 12300	-	ICS-CHASSIS-MIB	Fault Inventory

テンプレート名	ベンダ	デバイスファミリー	モデル	MIBs OS used	Tags
Juniper SNMP	Juniper	MX,SRX,EX models	Juniper MX240, Juniper EX4200-24F	JunOS	Certified
Mellanox SNMP	Mellanox	Mellanox Infiniband devices	SX1036	MLNX-OS	Certified
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Routers (CCR series)	Separate dedicated templates are available for MikroTik CCR1009-7G-1C-1S+, MikroTik CCR1009-7G-1C-1S+PC, MikroTik CCR1009-7G-1C-PC, MikroTik CCR1016-12G, MikroTik CCR1016-12S-1S+, MikroTik CCR1036-12G-4S-EM, MikroTik CCR1036-12G-4S, MikroTik CCR1036-8G-2S+, MikroTik CCR1036-8G-2S+EM, MikroTik CCR1072-1G-8S+, MikroTik CCR2004-16G-2S+, MikroTik CCR2004-1G-12S+2XS	RouterOS	Certified

テンプレート名	ベンダ	デバイスファミリー	モデル	MIBs OS used	Tags
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (CRS series)	Separate dedicated templates are available for MikroTik CRS106-1C-5S, MikroTik CRS109-8G-1S-2HnD-IN, MikroTik CRS112-8G-4S-IN, MikroTik CRS112-8P-4S-IN, MikroTik CRS125-24G-1S-2HnD-IN, MikroTik CRS212-1G-10S-1S+IN, MikroTik CRS305-1G-4S+IN, MikroTik CRS309-1G-8S+IN, MikroTik CRS312-4C+8XG-RM, MikroTik CRS317-1G-16S+RM, MikroTik CRS326-24G-2S+IN, MikroTik CRS326-24G-2S+RM, MikroTik CRS326-24S+2Q+RM, MikroTik CRS328-24P-4S+RM, MikroTik CRS328-4C-20S-4S+RM, MikroTik	Router OS, Switch OS	MIB, HOST-RESOURCES-MIB

テンプレート名	ベンダ	デバイスファミリー	モデル	MIBs OS used	Tags
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (CSS series)	Separate dedicated templates are available for MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MikroTik- Certified MIB,HOST-RESOURCES-MIB
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MikroTik- Certified MIB,HOST-RESOURCES-MIB
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Separate dedicated templates are available for MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MikroTik- Certified MIB,HOST-RESOURCES-MIB
MikroTik netPower <device model> SNMP	MikroTik	MikroTik netPower	Separate dedicated templates are available for MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS, Mikrotik Lite	MikroTik- Certified MIB,HOST-RESOURCES-MIB

テンプレート名	ベンダ	デバイスファミリー	モデル	MIBs used	Tags
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik PowerBox	Separate dedicated templates are available for MikroTik Power-Box, MikroTik PowerBox Pro	RouteMIB, HOST-RESOURCES-MIB	MikroTik- Certified
MikroTik RB<device model> SNMP	MikroTik	MikroTik RB series routers	Separate dedicated templates are available for MikroTik RB1100AHx4, MikroTik RB1100AHx4 Dude Edition, MikroTik RB2011iL-IN, MikroTik RB2011iL-RM, MikroTik RB2011iLS-IN, MikroTik RB2011UiAS-IN, MikroTik RB2011UiAS-RM, MikroTik RB260GS, MikroTik RB3011UiAS-RM, MikroTik RB4011iGS+RM, MikroTik RB5009UG+S+IN	RouteMIB, HOST-RESOURCES-MIB	MikroTik- Certified

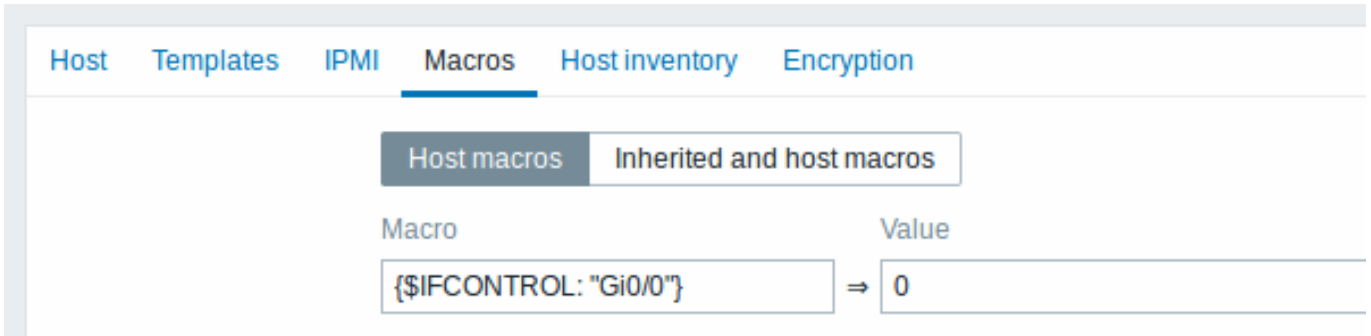
テンプレート名	ベンダ	デバイスファミリー	モデル	MIBs used	Tags
MikroTik SNMP	MikroTik	MikroTik RouterOS devices	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	ROUTEROS-MIB, HOST-RESOURCES-MIB	MikroTik- Certified
QTech QSW SNMP	QTech	Qtech devices	Qtech QSW-2800-28T	QTECH-MIB, ENTITY-MIB	Performance Inventory
Ubiquiti AirOS SNMP	Ubiquiti	Ubiquiti AirOS wireless devices	NanoBridge, NanoStation	UBNT-MIB, ENTITY-MIB, IEEE802dot11-MIB	Performance Inventory
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch	HH3C-ENTITY-EXT-MIB, ENTITY-MIB	Certified
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch	STATISTICS-MIB, NETSWITCH-MIB, HP-ICF-CHASSIS, ENTITY-MIB, SEMI-MIB	Certified
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0	TPLINK-SYSMONITOR-MIB, TPLINK-SYSINFO-MIB	Performance Inventory
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G	FASTPATH-SWITCHING-MIB, FASTPATH-BOXSERVICES-PRIVATE-MIB	Fault Inventory

テンプレートデザイン

テンプレートは、以下の点を考慮して設計されています:

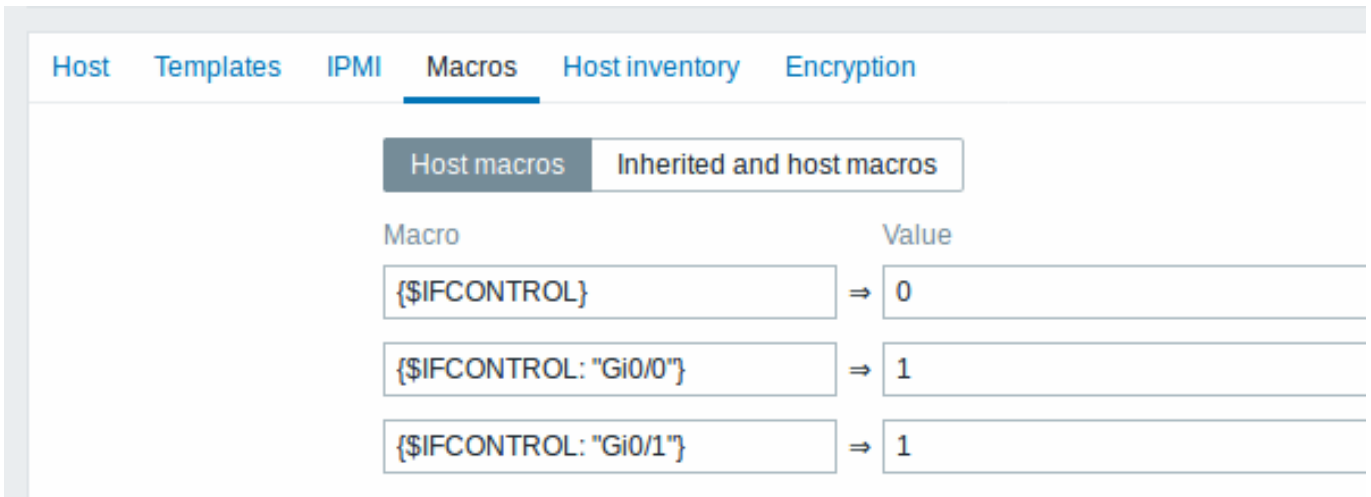
- トリガーをユーザーが調整できるように、ユーザーマクロを極力使用しています。
- ローレベルディスカバリーを可能な限り使用し、サポートされていない項目の数を最小限に抑えています。
- すべてのテンプレートは、テンプレート ICMP Ping に依存しているため、すべてのデバイスも ICMP でチェックされます。
- item は MIB を使用しない - SNMP OID は item とローレベルディスカバリーで使用されます。そのため、テンプレートが動作するために Zabbix に MIB を読み込む必要はありません。
- ループバックネットワークインタフェースは、ifAdminStatus = down(2) のインタフェースと同様に検出時にフィルタリングされます。

- 64 ビットカウンタは、可能な限り IF-MIB::ifXTable から使用されます。サポートされていない場合は、代わりにデフォルトの 32 ビットカウンタが使用されます。
- 検出されたすべてのネットワークインタフェースには、その動作状態（リンク）を制御するトリガがあります。
 - 特定のインタフェースでこの条件を監視しない場合は、値 0 のコンテキストを持つユーザーマクロを作成します。たとえば、以下のようになります：



ここで、Gi0/0 は {#IFNAME} である。このように、trigger はこの特定のインタフェースではそれ以上使用されません。

* 全ての trigger を発火させないデフォルトの動作を変更し、アップリンクのような限られた数のインタフェースのトリガーを有効にすることもできます。



タグ

- Performance - デバイスファミリーの MIB は、CPU とメモリ item を監視する方法を提供します。
- Fault - デバイスファミリーの MIB は、少なくとも 1 つの温度センサを監視する方法を提供します。
- Inventory - デバイスファミリーの MIB は、少なくともデバイスのシリアル番号とモデル名を収集する方法を提供します。
- Certified - 上記 3 つの主要カテゴリを全てカバーします。

10 イベント時の通知

概要

いくつかの item と trigger を設定し、trigger の状態が変化した場合、いくつかのイベントが発生したと仮定します。

そもそも、trigger やイベントのリストをずっと眺めていたいとは思いません。何か重要なこと（問題など）が起きたときに通知が来るほうがよっぽどです。またまた、問題が発生したときには、関係者全員に知らせてほしいものです。

そのため、通知の送信は Zabbix が提供する主要なアクションの 1 つとなっています。特定のイベント時に誰にいつ通知するかは定義することができます。

Zabbix から通知を送受信するためには、以下の設定が必要です。

- **define some media**
- **configure an action** によって定義されたメディアの 1 つにメッセージを送信する

アクションは、conditions と operations で構成される。基本的に、conditions が満たされると operations が実行される。
 主要な操作は、メッセージの送信（通知）とリモートコマンドの実行の 2 つである。

発見と自動登録が作成されたイベントには、いくつかの追加オペレーションが利用可能である。
それらは、ホストの追加や削除、テンプレートのリンクなどである。

1 メディアタイプ

概要

メディアとは、Zabbix からの通知やアラートの送信に使用される配信チャンネルです。

いくつかのメディアタイプを設定することができます:

- [E-mail](#)
- [SMS](#)
- [Custom alertscripts](#)
- [Webhook](#)

メディアタイプは、Administration → Media types で設定します。

Media types Create media type Import

<input type="checkbox"/>	Name ▲	Type	Status	Used in actions	Details	Action
<input type="checkbox"/>	E-mail	Email	Enabled	Report problems to Zabbix administrators	SMTP server: "mail.example.com", SMTP helo: "example.com", SMTP email: "zabbix@example.com"	Test
<input type="checkbox"/>	Jira	Webhook	Enabled			Test
<input type="checkbox"/>	Mattermost	Webhook	Enabled			Test
<input type="checkbox"/>	Opsgenie	Webhook	Enabled			Test
<input type="checkbox"/>	Pushover	Webhook	Enabled			Test
<input type="checkbox"/>	Slack	Webhook	Enabled			Test
<input type="checkbox"/>	SMS	SMS	Enabled		GSM modem: "/dev/ttyS0"	Test
<input type="checkbox"/>	Test THROW	Webhook	Enabled			Test
<input type="checkbox"/>	Zendesk	Webhook	Enabled			Test

メディアタイプによっては、デフォルトのデータセットにあらかじめ定義されているものもあります。それらを動作させるためには、パラメータを調整する必要があります。

設定したメディアタイプが動作するかどうかは、最後の列にある [Test](#) をクリックしてテストすることができます。(詳しくは、[Media type testing](#) を参照してください)

新しいメディアタイプを作成するには、[Create media type](#) ボタンをクリックします。メディアタイプの設定フォームが表示されます。

共通パラメーター

いくつかのパラメータは、すべてのメディアタイプで共通です。

Media type Message templates 5 Options

* Name SMS

Type SMS ▾

* GSM modem /dev/ttyS0

Description

Enabled

Add
Cancel

Media type タブでは、共通の一般属性があります：

パラメータ	説明
Name	Name of the media type.
Type	Select the type of media.
Description	Enter a description.
Enabled	Mark the checkbox to enable the media type.

メディア固有のパラメータについては、メディアタイプの各ページを参照してください。

Message templates タブでは、以下のイベントタイプのすべてまたは一部について、デフォルトの通知メッセージを設定できます：

- Problem
- Problem recovery
- Problem update
- Service
- Service recovery
- Service update
- Discovery
- Autoregistration
- Internal problem
- Internal problem recovery

Media types

Media type Message templates 6 Options

Message type	Template	Actions
Problem	Problem started at {EVENT.TIME} on {EVENT.DA...}	Edit Remove
Problem recovery	Problem has been resolved at {EVENT.RECOVE...}	Edit Remove
Problem update	{USER.FULLNAME} {EVENT.UPDATE.ACTION} prob...	Edit Remove
Service	Service problem started at {EVENT.TIME} on {EV...}	Edit Remove
Service recovery	Service "{SERVICE.NAME}" has been resolved a...	Edit Remove
Autoregistration	Host name: {HOST.HOST} Host IP: {...	Edit Remove

[Add](#)

メッセージテンプレートをカスタマイズするには:

- Message templates タブで [Add](#) をクリックすると、Message template ポップアップウィンドウが開きます。
- 必要な Message type を選択し、Subject と Message のテキストを編集します。
- Add をクリックして、メッセージテンプレートを保存します。

Message template ✕

Message type:

Subject:

Message:

メッセージテンプレートのパラメータ:

パラメータ	説明
Message type	Type of an event for which the default message should be used. Only one default message can be defined for each event type.

パラメータ	説明
Subject	Subject of the default message. The subject may contain macros. It is limited to 255 characters. Subject is not available for SMS media type.
Message	The default message. It is limited to certain amount of characters depending on the database type (see Sending messages for more information). The message may contain supported macros . In problem and problem update messages, expression macros are supported (for example, <code>{?avg(/host/key,1h)}</code>).

既存のメッセージテンプレートに変更を加えるには: Actions 列で [Edit](#) をクリックして、テンプレートを編集します。
または、[Remove](#) をクリックして、メッセージテンプレートを削除します。

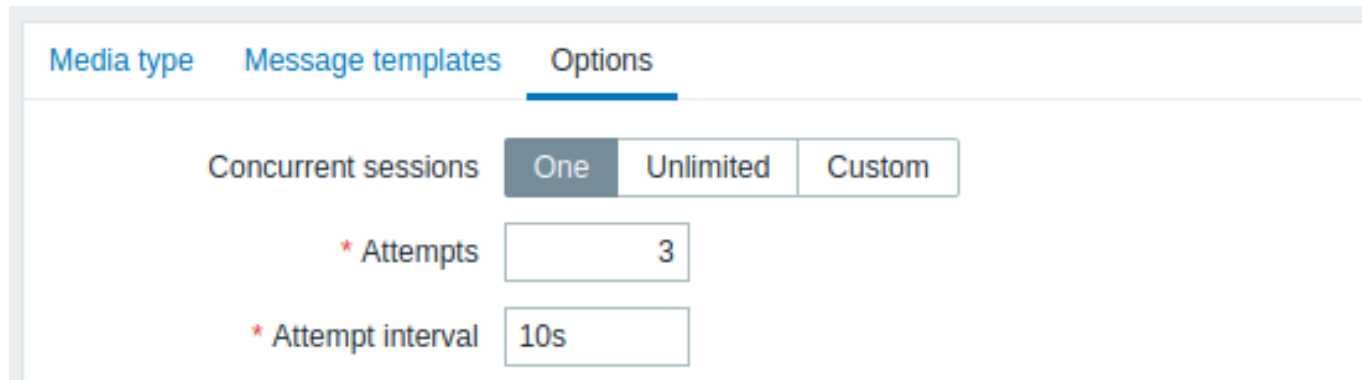
特定のアクションに対して、カスタムメッセージテンプレートを定義することができます。 (詳細は [action operations](#) を参照してください) アクションの設定で定義されたカスタムメッセージは、メディアタイプのデフォルトメッセージテンプレートをオーバーライドします。

Warning:

メッセージテンプレートの定義は、通知にデフォルトメッセージを使用しない Webhook やカスタムアラートスクリプトを含む、すべてのメディアタイプで必須です。例えば、“Send message to Pushover webhook” アクションは、Pushover webhook 用の Problem メッセージが定義されていない場合、問題通知の送信に失敗します。

Options タブには、アラート処理の設定が含まれています。各メディアタイプで同じオプションセットを設定できます。

すべてのメディアタイプは並行して処理されます。同時セッションの最大数はメディアタイプごとに設定できますが、サーバ上のアラートプロセスの総数は `StartAlertersparameter` によってのみ制限されます。1つのトリガーによって生成されたアラートは、順次処理されます。したがって、複数の通知は、複数のトリガーによって生成された場合のみ、同時に処理される可能性があります。



パラメータ	説明
Concurrent sessions	Select the number of parallel alerter sessions for the media type: One - one session Unlimited - unlimited number of sessions Custom - select a custom number of sessions Unlimited/high values mean more parallel sessions and increased capacity for sending notifications. Unlimited/high values should be used in large environments where lots of notifications may need to be sent simultaneously. If more notifications need to be sent than there are concurrent sessions, the remaining notifications will be queued; they will not be lost.
Attempts	Number of attempts for trying to send a notification. Up to 100 attempts can be specified; the default value is '3'. If '1' is specified Zabbix will send the notification only once and will not retry if the sending fails.

パラメータ	説明
Attempt interval	Frequency of trying to resend a notification in case the sending failed, in seconds (0-3600). If '0' is specified, Zabbix will retry immediately. Time suffixes are supported, e.g. 5s, 3m, 1h.

メディアタイプのテスト

設定したメディアタイプが動作するかどうかをテストすることが可能です。

E-mail

例えば、電子メールのメディアタイプをテストするには:

- メディアタイプのlistで、該当する電子メールを見つけます。
- リストの最後の列で Test をクリックします。(テストウィンドウが開きます)
- 受信者のアドレス、本文、およびオプションの件名を入力します。
- Test をクリックして、テストメッセージを送信します。

テストの成功または失敗のメッセージが同じウィンドウに表示されます。:

Test media type

✔ Media type test successful.

* Send to

Subject

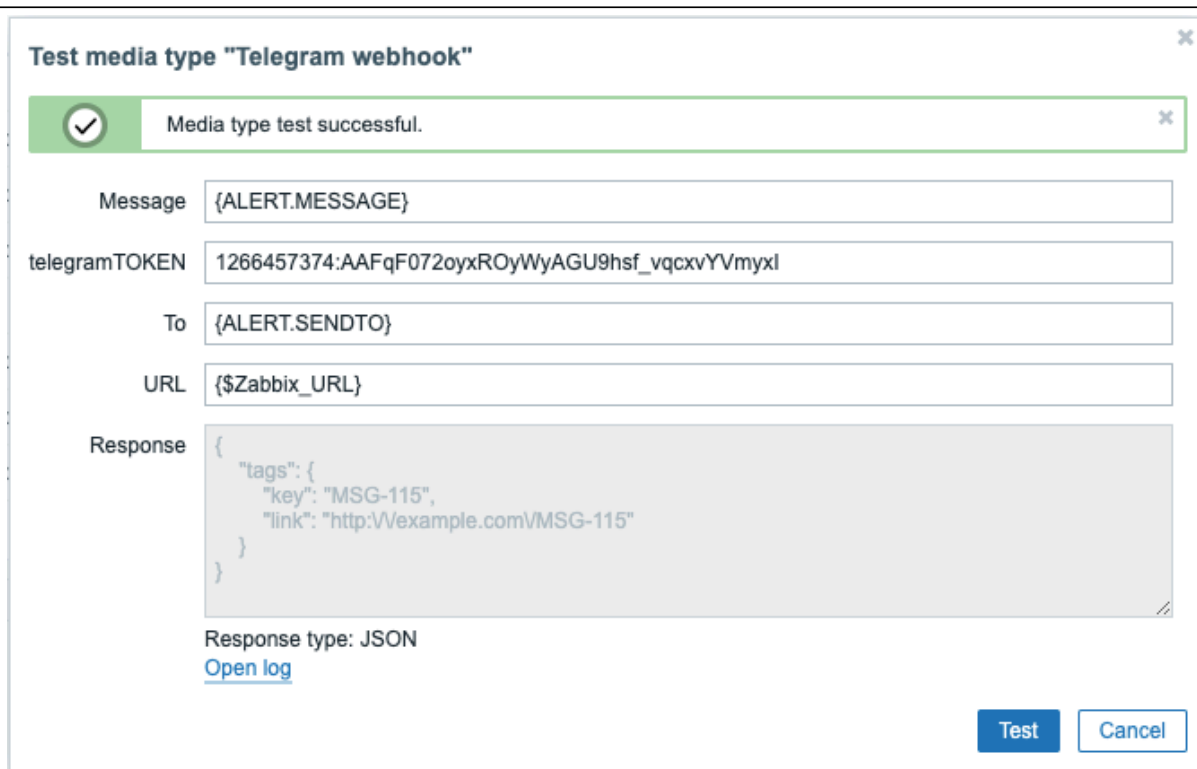
* Message

**Webhook **

webhook のメディアタイプをテストするには:

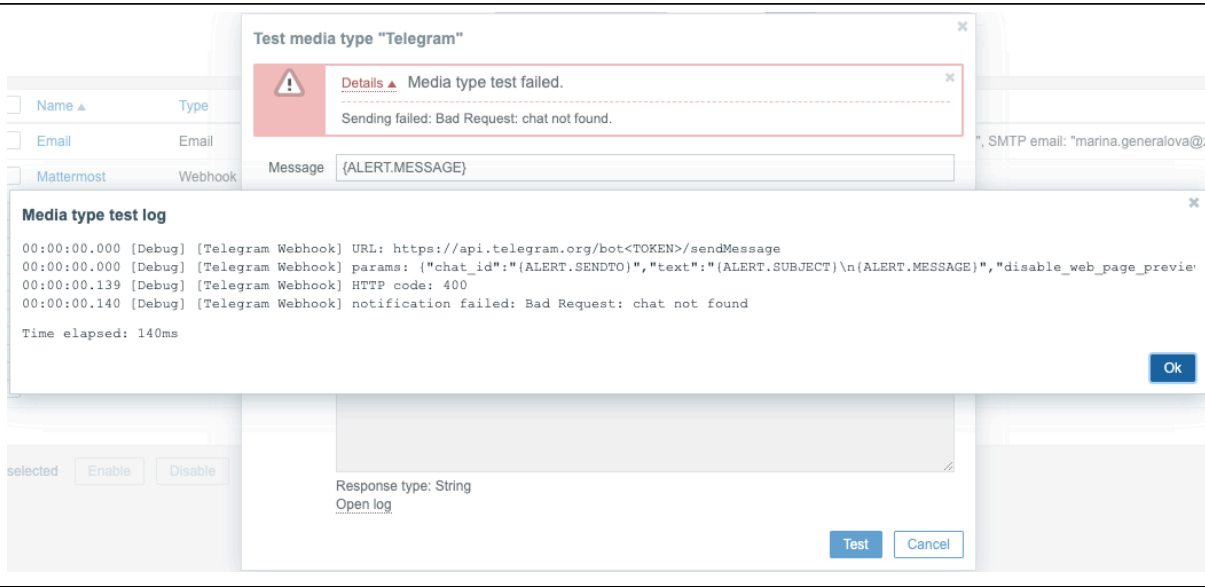
- メディアタイプのlistで、関連する webhook を見つけます。
- リストの最後の列で Test をクリックします。(テストウィンドウが開きます)
- 必要であれば、webhook パラメータ値を編集します。
- Test をクリックします。

デフォルトでは、Webhook のテストは設定時に入力されたパラメータで実行されます。しかし、テストのために属性値を変更することは
 可能です。テストウィンドウで値を交換または削除すると、テスト手順にのみ影響し、実際の Webhook 属性値は変更されません。



テストウィンドウを離れずにメディアタイプのテストログエントリを表示するには

- Open log をクリックします。(新しいポップアップウィンドウが開きます)



Webhook テストが成功した場合

- "Media type test successful. " のメッセージが表示されます。
- 灰色の Response フィールドにサーバーの応答が表示されます。
- レスポンスの種類 (JSON または String) は、Response フィールドの下に表示されます。

Webhook テストが失敗した場合

- "Media type test failed. " というメッセージが表示され、その後に失敗の詳細が表示されます。

ユーザーメディア

あるメディアタイプの通知を受け取るには、そのメディアタイプの媒体 (メールアドレス/電話番号/webhook ユーザー ID など) がユーザープロファイルに定義されている必要があります。例えば、webhook "X" を使ってユーザー "Admin" にメッセージを送るアクションは、webhook "X" メディアがユーザープロファイルに定義されていない場合、常に何も送信されないこととなります。

ユーザーメディアを定義するには:

- ユーザープロフィールにアクセスするか、または Administration → Users にアクセスし、ユーザーのプロパティフォームを開きます。
- メディアタブで以下をクリックします。 [Add](#)

Media ✕

Type Email ▼

* Send to example@company.com [Remove](#)

example recipient <example2@company.com> [Remove](#)

[Add](#)

* When active 1-7,00:00-24:00

Use if severity Not classified

Information

Warning

Average

High

Disaster

Enabled

Update
Cancel

ユーザーメディアの属性:

パラメータ	説明
Type	The drop-down list contains names of all configured media types.
Send to	Provide required contact information where to send messages. For an e-mail media type it is possible to add several addresses by clicking on Add below the address field. In this case, the notification will be sent to all e-mail addresses provided. It is also possible to specify recipient name in the Send to field of the e-mail recipient in a format 'Recipient name <address1@company.com>'. Note, that if a recipient name is provided, an e-mail address should be wrapped in angle brackets (<>). UTF-8 characters in the name are supported, quoted pairs and comments are not. For example: John Abercroft <manager@nycdatcenter.com> and manager@nycdatcenter.com are both valid formats. Incorrect examples: John Doe zabbix@company.com, %%"Zabbix\@\ <h(comment)q\>" %%.<br="" zabbix@company.com=""></h(comment)q\>"> You can limit the time when messages are sent, for example, set the working days only (1-5,09:00-18:00). Note that this limit is based on the user time zone . If the user time zone is changed and is different from the system time zone this limit may need to be adjusted accordingly so as not to miss important messages. See the Time period specification page for description of the format.
When active	

パラメータ	説明
Use if severity	Mark the checkboxes of trigger severities that you want to receive notifications for. Note that the default severity ('Not classified') must be checked if you want to receive notifications for non-trigger events . After saving, the selected trigger severities will be displayed in the corresponding severity colors, while unselected ones will be grayed out.
Status	Status of the user media. Enabled - is in use. Disabled - is not being used.

1 E-mail

概要

電子メールをメッセージの配信チャネルとして設定するには、メディアタイプとして電子メールを設定し、ユーザーに特定のアドレスを
 割り当てる必要があります。

Note:

1つのイベントに対する複数の通知は同じメールスレッドにまとめられます。

設定

メディアタイプに e-mail を設定する場合:

- Administration → Media types に移動します。
- Create media type をクリックします (または、定義済みメディアタイプの一覧から E-mail をクリックします)

Media type タブには、メディアタイプに関する一般的な属性が含まれています:

Media type Message templates 5 Options

* Name

Type ▾

* SMTP server

SMTP server port

* SMTP helo

* SMTP email

Connection security None STARTTLS SSL/TLS

Authentication None Username and password

Message format HTML Plain text

Description

Enabled

すべての必須入力項目には、赤いアスタリスクが付けられています。

以下のパラメータは、E-mail メディアタイプに固有のものです：

パラメータ	説明
SMTP server	Set an SMTP server to handle outgoing messages.
SMTP server port	Set the SMTP server port to handle outgoing messages. This option is supported starting with Zabbix 3.0.
SMTP helo	Set a correct SMTP helo value, normally a domain name.

パラメータ	説明
SMTP email	<p>The address entered here will be used as the From address for the messages sent.</p> <p>Adding a sender display name (like "Zabbix_info" in Zabbix_info <zabbix@company.com> in the screenshot above) with the actual e-mail address is supported since Zabbix 2.2 version.</p> <p>There are some restrictions on display names in Zabbix emails in comparison to what is allowed by RFC 5322, as illustrated by examples:</p> <p>Valid examples:</p> <p>zabbix@company.com (only email address, no need to use angle brackets)</p> <p>Zabbix_info <zabbix@company.com> (display name and email address in angle brackets)</p> <p>ΣΩ-monitoring <zabbix@company.com> (UTF-8 characters in display name)</p> <p>Invalid examples:</p> <p>Zabbix HQ zabbix@company.com (display name present but no angle brackets around email address)</p> <p>"Zabbix@\<H(comment)Q\>" <zabbix@company.com> (although valid by RFC 5322, quoted pairs and comments are not supported in Zabbix emails)</p>
Connection security	<p>Select the level of connection security:</p> <p>None - do not use the CURLOPT_USE_SSL option</p> <p>STARTTLS - use the CURLOPT_USE_SSL option with CURLUSESSL_ALL value</p> <p>SSL/TLS - use of CURLOPT_USE_SSL is optional</p> <p>This option is supported starting with Zabbix 3.0.</p>
SSL verify peer	<p>Mark the checkbox to verify the SSL certificate of the SMTP server. The value of "SSLCAlocation" server configuration directive should be put into CURLOPT_CAPATH for certificate validation. This sets cURL option CURLOPT_SSL_VERIFYPEER.</p>
SSL verify host	<p>This option is supported starting with Zabbix 3.0.</p> <p>Mark the checkbox to verify that the Common Name field or the Subject Alternate Name field of the SMTP server certificate matches. This sets cURL option CURLOPT_SSL_VERIFYHOST.</p>
Authentication	<p>This option is supported starting with Zabbix 3.0.</p> <p>Select the level of authentication:</p> <p>None - no cURL options are set (since 3.4.2) Username and password - implies "AUTH=*" leaving the choice of authentication mechanism to cURL (until 3.4.2) Normal password - CURLOPT_LOGIN_OPTIONS is set to "AUTH=PLAIN"</p>
Username	<p>This option is supported starting with Zabbix 3.0.</p> <p>User name to use in authentication. This sets the value of CURLOPT_USERNAME.</p>
Password	<p>This option is supported starting with Zabbix 3.0.</p> <p>Password to use in authentication. This sets the value of CURLOPT_PASSWORD.</p>
Message format	<p>This option is supported starting with Zabbix 3.0.</p> <p>Select message format:</p> <p>HTML - send as HTML</p> <p>Plain text - send as plain text</p>

Attention:

SMTP 認証オプションを利用するためには、Zabbix server を cURL 7.20.0 以上で --with-libcurl **compilation** オプションを付けて
 コンパイルする必要があります。

また、[common media type parameters](#) を参照してください。

ユーザーメディア

メールメディアの種類を設定したら、Administration → Users セクションで、ユーザープロファイルを編集して、ユーザーに
 メールメディアを割り当てます。ユーザーメディアの設定方法は、全メディアタイプ共通なので、Media types ページに記述されています。

2 SMS

概要

Zabbix は Zabbix server のシリアルポートに接続されたシリアル GSM モデムを使用した SMS メッセージの送信に対応しています。

以下の点を確認してください:

- シリアルデバイス (Linux では通常/dev/ttyS0) の速度は GSM モデムの速度に合わせます。Zabbix はシリアルリンクの速度を設定しません。デフォルトの設定を使用します。
- zabbix ユーザーは、シリアルデバイスへのリード/ライトアクセス権を持ちます。ls -l /dev/ttyS0 コマンドを実行し、シリアルデバイスの現在のパーミッションを確認します。
- GSM モデムには PIN が入力されており、パワーリセット後も PIN が保持されます。また、SIM カードの PIN を無効化することもできます。PIN は Unix の minicom や Windows の HyperTerminal などのターミナルソフトで AT+CPIN="NNNN" (NNNN は PIN 番号、引用符は必ず入れてください) コマンドを発行することで入力することができます。

Zabbix は以下の GSM モデムでテストされています:

- Siemens MC35
- Teltonika ModemCOM/G10

メッセージの配信チャネルとして SMS を設定するには、メディアタイプとして SMS を設定し、ユーザーのそれぞれの電話番号を入力する
 必要があります。

設定

メディアタイプとして SMS を設定するには:

- Administration → Media types に進みます。
- Create media type をクリックします (または、定義済みメディアタイプのリストから SMS をクリックします)

以下のパラメータは、SMS メディアタイプに固有のものです:

Parameter	Description
GSM modem	Set the serial device name of the GSM modem.

[common media type parameters](#)を参照してください。SMS 通知の送信は並列処理できないことに注意してください。

User media

ユーザーメディア

SMS メディアタイプを設定したら、Administration → Users セクションでユーザープロファイルを編集し、ユーザーに SMS メディアを
 割り当てます。ユーザーメディアの設定方法は、全てのメディアタイプで共通なので、Media types ページに記載されているとおりです。

3 カスタムアラートスクリプト

概要

警告を送信するための既存のメディアタイプに満足できない場合、別の方法があります。独自の方法で通知を処理するスクリプトを
 作成することができます。

アラートスクリプトは Zabbix server 上で実行されます。これらのスクリプトはサーバの **configuration file** の
 AlertScriptsPath 変数で定義されたディレクトリに配置されています。

以下はアラートスクリプトの例です。:

```
#####!/bin/bash

to=$1
subject=$2
body=$3

cat <<EOF | mail -s "$subject" "$to"
```

```
$body
EOF
```

Attention:

バージョン 3.4 以降、Zabbix は実行されたコマンドとスクリプトの終了コードをチェックします。終了コードが **** 0 **** 以外の終了コードは、**
command execution** エラーとみなされます。この場合、Zabbix は実行に失敗したコマンドを再度実行しようとします。

環境変数はスクリプトのために保存も作成もされないため、明示的に処理する必要があります。

設定

カスタムアラートスクリプトをメディアタイプとして設定するには:

- Administration → Media types に進みます。
- Create media type をクリックします。

Media type タブには、メディアタイプに関する一般的な属性が含まれています:

The screenshot shows the Zabbix configuration interface for a media type. It has three tabs: 'Media type' (selected), 'Message templates', and 'Options'. The 'Name' field is 'Notification script' with a red asterisk. The 'Type' dropdown is set to 'Script'. The 'Script name' field is 'notification.sh' with a red asterisk. Under 'Script parameters', there are three text boxes containing '{ALERT.SENDTO}', '{ALERT.SUBJECT}', and '{ALERT.MESSAGE}'. Below them is an 'Add' link. The 'Description' field is empty. At the bottom, the 'Enabled' checkbox is checked.

すべての必須入力フィールドには、赤いアスタリスクが表示されます。

以下のパラメータは、スクリプトメディアタイプに固有のものです:

パラメータ	説明
Script name	Enter the name of the script.

パラメータ	説明
Script parameters	Add command-line parameters to the script. {ALERT.SENDTO}, {ALERT.SUBJECT} and {ALERT.MESSAGE} macros are supported in script parameters. Customizing script parameters is supported since Zabbix 3.0.

common media type parameters でデフォルトのメッセージとアラート処理オプションを設定する方法の詳細について説明します。

Warning:

アラートスクリプトがデフォルトのメッセージを使用しない場合でも、このメディアタイプで使用される操作タイプ用の
 メッセージテンプレートが定義されていなければなりません。そうでなければ、通知は送られません。

Attention:

メディアタイプの並列処理は Zabbix 3.4.0 から実装されているため、複数のスクリプトメディアタイプが設定されている場合、これらの
 スクリプトはアラータプロセスによって並行して処理される可能性があることに注意することが重要です。アラータプロセスの総数は、
 StartAlerters parameter によって制限されます。

ユーザーメディア

メディアタイプを設定したら、Administration → Users セクションで、ユーザープロファイルを編集して、このタイプのメディアを
 ユーザーに割り当てます。ユーザーメディアの設定方法は、すべてのメディアタイプに共通なので、Media types に記述されています。

ユーザーメディアを定義する場合、Send to フィールドを空にすることはできないことに注意してください。このフィールドをアラートスクリプトで使用しない場合は、検証要件を回避するために、サポートされている文字の任意の組み合わせを入力してください。

4 Webhook

概要

webhook メディアタイプは、ヘルプデスクシステム、チャット、メッセージャーなどの外部ソフトウェアと簡単に統合するために、
 カスタム JavaScript コードを使用して HTTP コールを行うのに便利です。Zabbix が提供する統合機能をインポートするか、
 カスタム統合機能を一から作成することができます。

インテグレーション

以下のインテグレーションを利用することで、定義済みの Webhook メディアタイプを使用して Zabbix 通知をプッシュすることができます:

- [brevis.one](#)
- [Discord](#)
- [Express.ms messenger](#)
- [Github issues](#)
- [GLPi](#)
- [iLert](#)
- [iTop](#)
- [Jira](#)
- [Jira Service Desk](#)
- [ManageEngine ServiceDesk](#)
- [Mattermost](#)
- [Microsoft Teams](#)
- [Opsgenie](#)
- [OTRS](#)
- [Pagerduty](#)
- [Pushover](#)
- [Redmine](#)
- [Rocket.Chat](#)
- [ServiceNow](#)
- [SIGNL4](#)
- [Slack](#)
- [SolarWinds](#)
- [SysAid](#)
- [Telegram](#)

- [TOPdesk](#)
- [VictorOps](#)
- [Zammad](#)
- [Zendesk](#)

Note:

ここに挙げたサービスに加え、Zabbix は **Spiceworks** と連携することができます。(Webhook は不要です) Zabbix の通知を Spiceworks のチケットに変換するには、**email media type**を作成し、Spiceworks ヘルプデスクのメールアドレス
 (例: help@zabbix.on.spiceworks.com) を指定した Zabbix のプロファイル設定に入力する必要があります。

構成

Webhook 連携の利用を開始するには:

1. ダウンロードした Zabbix の templates/media ディレクトリにある必要な.xml ファイルを見つけるか、Zabbix [git repository](#) からダウンロードしてください。
2. ファイルを Zabbix のインストールに **Import** します。メディアタイプのリストに Webhook が表示されます。
3. Readme.md ファイルの指示に従って Webhook を設定します (上記の Webhook の名前をクリックすると、Readme.md にすぐにアクセスできます)

カスタム Webhook をゼロから作成する場合:

- Administration → Media types に進みます。
- Create media type をクリックします。

Media type タブには、そのメディアタイプに固有のさまざまな属性が含まれています:

* Name

Type

Parameters

Name	Value
<input type="text" value="event_source"/>	<input type="text" value="{EVENT.SOURCE}"/>
<input type="text" value="event_update_status"/>	<input type="text" value="{EVENT.UPDATE.STATUS}"/>
<input type="text" value="event_value"/>	<input type="text" value="{EVENT.VALUE}"/>
<input type="text" value="express_message"/>	<input type="text" value="{ALERT.MESSAGE}"/>
<input type="text" value="express_send_to"/>	<input type="text" value="{ALERT.SENDTO}"/>
<input type="text" value="express_tags"/>	<input type="text" value="{EVENT.TAGSJSON}"/>
<input type="text" value="express_token"/>	<input bot="" token>")="" type="text" value("<place=""/>
<input type="text" value="express_url"/>	<input instance="" type="text" url>")="" value("<place=""/>

[Add](#)

* Script

* Timeout

Process tags

Include event menu entry

* Menu entry name

* Menu entry URL

Description

Enabled

すべての必須入力フィールドには、赤いアスタリスクが表示されます。

以下のパラメータは、webhook メディアタイプに固有のもので:

パラメータ	説明
Parameters	<p>Specify the webhook variables as the attribute and value pairs. For preconfigured webhooks, a list of parameters varies, depending on the service. Check the webhook's Readme.md file for parameter description.</p> <p>For new webhooks, several common variables are included by default (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:%7BALERT.MESSAGE}), feel free to keep or remove them.</p> <p>All macros that are supported in problem notifications are supported in the parameters.</p> <p>If you specify an HTTP proxy, the field supports the same functionality as in the item configuration HTTP proxy field. The proxy string may be prefixed with <code>[scheme]://</code> to specify which kind of proxy is used (e.g. https, socks4, socks5; see documentation).</p>
Script	<p>Enter JavaScript code in the block that appears when clicking in the parameter field (or on the view/edit button next to it). This code will perform the webhook operation.</p> <p>The script is a function code that accepts parameter - value pairs. The values should be converted into JSON objects using <code>JSON.parse()</code> method, for example: <code>var params = JSON.parse(value);</code>.</p> <p>The code has access to all parameters, it may perform HTTP GET, POST, PUT and DELETE requests and has control over HTTP headers and request body.</p> <p>The script must contain a return operator, otherwise it will not be valid. It may return OK status along with an optional list of tags and tag values (see Process tags option) or an error string.</p> <p>Note, that the script is executed only after an alert is created. If the script is configured to return and process tags, these tags will not get resolved in <code>{EVENT.TAGS}</code> and <code>{EVENT.RECOVERY.TAGS}</code> macros in the initial problem message and recovery messages because the script has not had the time to run yet.</p> <p>See also: Webhook development guidelines, Webhook script examples, Additional JavaScript objects.</p>
Timeout	JavaScript execution timeout (1-60s, default 30s).
Process tags	<p>Time suffixes are supported, e.g. 30s, 1m.</p> <p>Mark the checkbox to process returned JSON property values as tags. These tags are added to the already existing (if any) problem event tags in Zabbix.</p> <p>If a webhook uses tags (the Process tags checkbox is marked), the webhook should always return a JSON object containing at least an empty object for tags:<code>var result = {tags: {}};</code></p> <p>Examples of tags that can be returned: Jira ID: PROD-1234, Responsible: John Smith, Processed:<no value>, etc.</p>
Include event menu entry	<p>Mark the checkbox to include an entry in the event menu linking to the created external ticket.</p> <p>If marked, the webhook should not be used to send notifications to different users (consider creating a dedicated user instead) or in several alert actions related to a single problem event.</p>
Menu entry name	<p>Specify the menu entry name.</p> <p><code>{EVENT.TAGS.<tag name>}</code> macro is supported.</p>
Menu entry URL	<p>This field is only mandatory if Include event menu entry is selected.</p> <p>Specify the underlying URL of the menu entry.</p> <p><code>{EVENT.TAGS.<tag name>}</code> macro is supported.</p> <p>This field is only mandatory if Include event menu entry is selected.</p>

デフォルトのメッセージとアラート処理オプションを設定する方法の詳細については [common media type parameters](#) を参照してください。

Warning:

Webhook がデフォルトメッセージを使用しない場合でも、この Webhook が使用する操作タイプ用のメッセージテンプレートは `
` 定義されている必要があります。

ユーザーメディア

メディアタイプを設定したら、Administration → Users セクションで、Webhook メディアを既存のユーザーに割り当てるか、`
` Webhook を表すユーザーを新規に作成します。既存のユーザーにユーザーメディアを設定する手順は、すべてのメディアタイプに `
` 共通なので、[Media types](#) ページで説明されています。

Webhook がタグを使用して `ticket \`メッセージ ID を格納する場合、同じ Webhook を異なるユーザーにメディアとして割り当てることは `
` 避けてください。エラーの原因となる可能性があります。(Include event menu entry option を利用するウェブフックの大部分に `
` 適用されます) この場合、ベストプラクティスは、Webhook を代表する専用のユーザを作成することです:

1. Webhook メディアタイプの設定後、Administration → Users セクションに移動し、Webhook 専用の Zabbix ユーザ (例: Slack Webhook 用のユーザ名 Slack) を作成します。このユーザは Zabbix にログインしないため、メディア以外の設定はすべてデフォルトのままでよいです。
2. ユーザプロファイルの Media タブで、必要な連絡先情報を入力し、[add a webhook](#)を行います。Webhook が Send to フィールドを使用しない場合、検証要件を回避するためにサポートされている文字の任意の組み合わせを入力します。
3. このユーザーに、アラートを送信するすべてのホストに対する少なくとも読み取り [permissions](#) を許可します。

アラートアクションの設定時に、操作の詳細の Send to users フィールドにこのユーザを追加することで、Zabbix にこのアクションからの `
` 通知に Webhook を使用するよう指示します。

アラートアクションを設定する際、操作の詳細の Send to users フィールドにこのユーザーを追加します。これにより、`
` このアクションからの通知に Webhook を使用するよう Zabbix に指示します。

アラートアクションの設定

アクションは、どの通知を Webhook 経由で送信するかを決定します。[configuring actions](#)の手順は以下の通りです。Webhook を含む他のすべてのメディアタイプの場合と同じですが、以下の例外があります:

- Webhook が `ticket \`message ID を保存し、`update \`resolve 操作でフォローアップするためにタグを使用する場合、この Webhook は単一の問題イベントに対する複数の警告アクションで使用されるべきではありません。{EVENT.TAGS.<name>} が既に存在し、Webhook で更新される場合、その結果の値は定義されません。このような場合、更新された値を保存するために、Webhook で新しいタグ名を使用する必要があります。これは、Zabbix が提供する Jira、Jira Service Desk、Mattermost、Opsgenie、OTRS、Redmine、ServiceNow、Slack、Zammad、Zendesk Webhook と、Include event menu entry オプションを利用する Webhook の大部分に適用されます。複数の操作で Webhook を使用することは可能です。これらの操作またはエスカレーションステップが同じアクションに属している場合、Webhook を複数の操作で使用することができます。また、フィルタ条件が異なるため、同じ問題イベントにアクションが適用されない場合、異なるアクションでこの Webhook を使用することもできます。
- [internal events](#) アクションで Webhook を使用する場合: アクションの動作設定で、Custom message チェックボックスをチェックし、カスタムメッセージを定義します。さもないと、通知が送信されません。

Webhook スクリプトの例

概要

Zabbix は多くの Webhook 統合機能を提供しています。しかし、独自の Webhook を作成することもできます。`
` このセクションでは、カスタム Webhook スクリプトの例を示します。(Script パラメータで使用します)`
` [webhook](#) セクションでその他の Webhook パラメータについて説明しています。

Jira webhook (カスタム)

* Name Jira webhook

Type Webhook

Parameters	Name	Value
	HTTPProxy	
	Message	{ALERT.MESSAGE}
	Subject	{ALERT.SUBJECT}
	To	{ALERT.SENDTO}
	URL	
	Add	

* Script try {...

* Timeout 30s

Process tags Include event menu entry

* Menu entry name {EVENT.tags.issue_key}

* Menu entry URL https://tsupport.zabbix.lan/browse/{EVENT.tags.issue_key}

Description Creating a JIRA issue.

Enabled

このスクリプトは、JIRA 課題を作成し、作成された課題に関する情報を返します。

```
try {
  Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

  var result = {
    'tags': {
      'endpoint': 'jira'
    }
  },
  params = JSON.parse(value),
  req = new HttpRequest(),
  fields = {},
  resp;

  if (params.HTTPProxy) {
    req.setProxy(params.HTTPProxy);
  }
}
```

```

req.addHeader('Content-Type: application/json');
req.addHeader('Authorization: Basic ' + params.authentication);

fields.summary = params.summary;
fields.description = params.description;
fields.project = {key: params.project_key};
fields.issuetype = {id: params.issue_id};

resp = req.post('https://tsupport.zabbix.lan/rest/api/2/issue/',
    JSON.stringify({"fields": fields})
);

if (req.getStatus() != 201) {
    throw 'Response code: ' + req.getStatus();
}

resp = JSON.parse(resp);
result.tags.issue_id = resp.id;
result.tags.issue_key = resp.key;

return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}

```

Slack webhook (カスタム)

この Webhook は、Zabbix からの通知を Slack チャンネルに転送します。

Media type	Message templates	Options														
<p>* Name <input type="text" value="Slack chat bot"/></p>																
<p>Type <input type="text" value="Webhook"/></p>																
<p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td><input type="text" value="URL"/></td> <td><input type="text"/></td> </tr> <tr> <td><input type="text" value="HTTPProxy"/></td> <td><input type="text"/></td> </tr> <tr> <td><input type="text" value="channel"/></td> <td><input data-bbox="927 1563 1308 1608" type="text" value="{ALERT.SENDTO}"/></td> </tr> <tr> <td><input type="text" value="text"/></td> <td><input data-bbox="927 1630 1308 1675" type="text" value="{ALERT.SUBJECT}"/></td> </tr> <tr> <td><input type="text" value="username"/></td> <td><input data-bbox="927 1697 1308 1742" type="text" value="bot"/></td> </tr> <tr> <td colspan="2"> <p>Add</p> </td> </tr> </tbody> </table>			Name	Value	<input type="text" value="URL"/>	<input type="text"/>	<input type="text" value="HTTPProxy"/>	<input type="text"/>	<input type="text" value="channel"/>	<input data-bbox="927 1563 1308 1608" type="text" value="{ALERT.SENDTO}"/>	<input type="text" value="text"/>	<input data-bbox="927 1630 1308 1675" type="text" value="{ALERT.SUBJECT}"/>	<input type="text" value="username"/>	<input data-bbox="927 1697 1308 1742" type="text" value="bot"/>	<p>Add</p>	
Name	Value															
<input type="text" value="URL"/>	<input type="text"/>															
<input type="text" value="HTTPProxy"/>	<input type="text"/>															
<input type="text" value="channel"/>	<input data-bbox="927 1563 1308 1608" type="text" value="{ALERT.SENDTO}"/>															
<input type="text" value="text"/>	<input data-bbox="927 1630 1308 1675" type="text" value="{ALERT.SUBJECT}"/>															
<input type="text" value="username"/>	<input data-bbox="927 1697 1308 1742" type="text" value="bot"/>															
<p>Add</p>																
<p>* Script <input type="text" value="try {..."/></p>																

```

try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {

```

```

    req.setProxy(params.HTTPProxy);
}

req.addHeader('Content-Type: application/x-www-form-urlencoded');

Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);

response = req.post(params.hook_url, 'payload=' + encodeURIComponent(value));
Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.Status() + '. Response: ' + response);

try {
    response = JSON.parse(response);
}
catch (error) {
    if (req.getStatus() < 200 || req.getStatus() >= 300) {
        throw 'Request failed with status code ' + req.getStatus();
    }
    else {
        throw 'Request success, but response parsing failed.';
    }
}

if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
    throw response.error;
}

return 'OK!';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
}

```

2 アクション

概要

イベント (通知の送信など) の結果として何らかの操作を実行する場合は、アクションを設定する必要があります。

アクションは、サポートされているすべてのタイプのイベントに応答して定義できます。

- トリガーアクション - トリガーのステータスが OK から PROBLEM に変化または戻ったとき
- サービスアクション - サービスステータスが OK から PROBLEM に変化または戻ったとき
- ディスカバリアクション - ネットワークディスカバリが行われるとき
- 自動登録アクション - 新しいアクティブなエージェントが自動登録されたとき (または登録済エージェントのホストメタデータが更新された)
- 内部アクション - アイテムがサポートされなくなった、またはトリガーが不明な状態になったとき

アクションの構成

アクションを構成するには、次の手順を実行します。

- 設定 -> アクションに移動し、サブメニューから登録したいアクションタイプを選択します (移動後にタイトルドロップダウンで別のタイプに切り替えることもできます)
- アクションの作成をクリックします
- アクションに名前を付けます
- 操作の**実行条件**を設定します
- **実行内容**を設定します

Note サービスアクションは**サービスアクション** セクションで構成できます。

一般的なアクション属性:

* Name

Type of calculation A and B

Conditions	Label	Name
	A	Trigger severity is greater than or equals <i>Not classified</i>
	B	Trigger severity does not equal <i>Information</i>
	Add	

Enabled

必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	一意なアクション名
計算のタイプ	複数のアクション条件がある場合の評価オプションを選択します: And - すべての条件が満たされている必要があります ** Or** - 1つの条件が満たされれば可 And/Or - 2つの組み合わせ: 異なる条件タイプの AND と同じ条件タイプの OR カスタム式 - ユーザーアクション条件を評価するための定義された計算式。
実行条件	アクション条件リスト 追加をクリックして新しい 実行条件 を追加します。
有効	チェックボックスをマークしてアクションを有効にします。しない場合は無効になります。

1 アクション

概要

イベントが定義済みの一連の条件に一致する場合にのみアクションが実行されるように定義できます。アクションの設定時に条件を設定します。

条件一致では大文字と小文字が区別されます。

トリガーアクション

トリガーベースのアクションでは、次の条件を使用できます。

タイプ	オペレータ	説明
ホストグループ	等しい 等しくない	除外するホストグループまたは除外しないホストグループを指定します。 等しい - イベントはこのホストグループに属します。 等しくない - イベントはこのホストグループに属しません。 親ホストグループは、ネストされたすべてのホストグループを暗黙的に選択します。親グループのみを指定するには、ネストされたすべてのグループを等しくないで追加設定する必要があります。
テンプレート	等しい 等しくない	除外するテンプレートまたは除外しないテンプレートを指定。 等しい - イベントはこのテンプレートから継承されたトリガーに属している。 等しくない - イベントはこのテンプレートから継承されたトリガーに属していない。
ホスト	等しい 等しくない	除外するホストを指定します。 等しい - イベントはこのホストに属します。 等しくない - イベントはこのホストに属しません。

タイプ	オペレータ	説明
タグ名	等しい 等しくない 含む 含まない	除外するイベントタグまたは除外しないイベントタグを指定。 等しい - イベントにこのタグがある 等しくない - イベントにこのタグが無い 含む - イベントにこの文字列を含むタグがある 含まない - イベントにこの文字列を含むタグが無い
タグの値	等しい 等しくない 含む 含まない	除外するイベントタグと値の組み合わせ、またはタグと値の組み合わせを指定します。 等しい - イベントのこのタグと値が等しい 等しくない - イベントのこのタグと値が等しくない 含む - イベントにこれらの文字列を含むタグと値がある 含まない - イベントにこれらの文字列を含むタグと値が無い
トリガー	等しい 等しくない	トリガーまたは除外するトリガーを指定します。 等しい - イベントはこのトリガーによって生成されます。 等しくない - イベントは、このトリガー以外の他のトリガーによって生成されます。
トリガー名	含む 含まない	トリガー名に含まれる文字列を指定するか、除外する文字列を指定します。 含む - イベントはトリガーによって生成され、名前にこの文字列が含まれます。 含まない - この文字列は使用できません 注: 入力された値は、すべてのマクロが展開されたトリガー名と比較されます。
トリガーの深刻度	等しい 等しくない 以上 以下	トリガーの深刻度を指定します。 等しい - トリガーの深刻度と等しい 等しくない - トリガーの深刻度と等しくない 以上 - トリガーの深刻度以上 以下 - トリガーの深刻度以下
期間	期間中 期間外	期間または除外する期間を指定します。 期間中 - イベント時間が期間内です。 期間外 - イベント時間が期間内ではありません。 フォーマットの説明については 期間指定 のページを参照してください。 ユーザーマクロ は、Zabbix 3.4.0 からサポートされています。
メンテナンス期間中	いいえ はい	ホストのメンテナンスのために障害が抑制されている (表示されていない)かどうかを指定します。 いいえ - 障害は抑制されていません。 はい - 障害は抑制されています。

ディスクバリアクション

次の条件は、ディスクバリエースのイベントで使用できます。

タイプ	オペレータ	説明
ホストの IP アドレス	等しい 等しくない	検出されたホストから除外するまたは除外しない IP アドレス範囲を指定します。 等しい - ホスト IP は範囲内 等しくない - ホスト IP は範囲外 次のフォーマットを受け付けます: 単体 IP アドレス: 192.168.1.33 IP アドレス範囲: 192.168.1-10.1-254 IP マスク: 192.168.4.0/24 List: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 リスト形式でのスペースは Zabbix 3.0.0 以降でサポートされています。

タイプ	オペレータ	説明
サービスのタイプ	等しい 等しくない	検出されたサービスの除外するまたは除外しないサービスタイプを指定します。 等しい - 検出されたサービスと一致する 等しくない - 検出されたサービスと一致しない。 使用可能なサービスの種類: SSH, LDAP, SMTP, FTP, HTTP, HTTPS (Zabbix 2.2 バージョン以降で使用可能), POP, NNTP, IMAP, TCP, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping, telnet (Zabbix 2.2 バージョン以降で使用可能)。
サービスのポート	等しい 等しくない	検出されたサービスの除外するまたは除外しない TCP ポート範囲を指定します。 等しい - サービスのポートは範囲内 等しくない - サービスのポートは範囲外
ディスカバリルール	等しい 等しくない	除外するまたは除外しないディスカバリルールを指定します。 等しい - ディスカバリルールを使用する 等しくない - 他のディスカバリルールを使用する
ディスカバリチェック	等しい 等しくない	除外するまたは除外しないディスカバリチェックを指定します。 等しい - ディスカバリチェックを使用する does not equal - 他のディスカバリチェックを使用する
ディスカバリオブジェクト	等しい	検出されたオブジェクトを指定 等しい - 発見されたオブジェクト (デバイスまたはサービス) に等しい
ディスカバリのステータス	等しい	Up - 'Host Up' および 'Service Up' イベントに一致 Down - 'Host Down' および 'Service Down' イベントに一致 Discovered - 'Host Discovered' および 'Service Discovered' イベントに一致 Lost - 'Host Lost' および 'Service Lost' イベントに一致
アップタイム/ダウンタイム	以上 以下	'ホストアップ' および 'サービスアップ' イベントのアップタイム。'ホストダウン' および 'サービスダウン' イベントのダウンタイム。 以上 - 以上です。パラメータは秒単位で指定されます。 以下 - 以下です。パラメータは秒単位で指定されます。
受信した値	等しい 等しくない 以上 以下 含む 含まない	ディスカバリルールでエージェント (Zabbix, SNMP) チェックから受け取った値を指定します。文字列比較。1 つのルールに対して複数の Zabbix エージェントまたは SNMP チェックが設定されている場合、それぞれの受信値がチェックされます (各チェックは、すべての条件に一致する新しいイベントを生成します)。 等しい - 値が等しい 等しくない - 値は等しくない 以上 - 値は指定値以上 以下 - 値は指定値以下 含む - 部分文字列が含まれます。パラメータは文字列として与えられません。 含まない - 部分文字列が含まれていません。パラメータは文字列として与えられます。
プロキシ	等しい 等しくない	除外するまたは除外しないプロキシを指定します。 等しい - このプロキシを使用 等しくない - 他のプロキシを使用

Note:

ディスカバリイベントを発生させるディスカバリルールのサービスチェックは、同時には実行されません。したがって、アクションのサービスのタイプ、サービスのポートまたは受信した値に複数の値が設定されている場合、それらは一度に1つのディスカバリイベントと比較されますが、同時に複数のイベントと比較されることはありません。その結果、同じチェックタイプに対して複数の値を持つアクションが正しく実行されない場合があります。

自動登録アクション

次の条件は、アクティブなエージェントの自動登録に基づくアクションで使用できます。

タイプ	オペレータ	説明
ホストメタデータ	含む 含まない 一致する 一致しない	ホストのメタデータまたは除外するホストのメタデータを指定します。 含む - ホストメタデータには指定文字列が含まれる。 含まない - ホストメタデータに指定文字列が含まれない。 ホストのメタデータは エージェント設定ファイル で指定できます。 一致する - ホストメタデータは正規表現と一致する。 一致しない - ホストメタデータは正規表現と一致しない。
ホスト名	含む 含まない 一致する 一致しない	ホスト名または除外するホスト名を指定します。 含む - ホスト名に文字列が含まれる。 含まない - ホスト名に文字列が含まれない。 **一致する** - ホスト名が正規表現に一致する。 一致しない - ホスト名が正規表現に一致しない。
プロキシ	等しい 等しくない	プロキシまたは除外するプロキシを指定します。 等しい - このプロキシを使用します。 等しくない - このプロキシ以外の他のプロキシを使用します。

内部イベントアクション

内部イベントに基づくアクションには、次の条件を設定できます。

タイプ	オペレータ	説明
イベントタイプ	等しい	アイテムが "取得不可" の状態 - アイテムが '通常' から 'サポートされていない' に移行するイベントに一致します ローレベルディスクバリアが "取得不可" の状態 - ローレベルディスクバリアルールが '通常' から 'サポートされていない' に移行するイベントに一致します トリガーが "不明" の状態 - トリガーが '通常' から '不明' に移行するイベントに一致します
ホストグループ	等しい 等しくない	ホストグループまたは除外するホストグループを指定。 等しい - イベントはこのホストグループに属します。 等しくない - イベントはこのホストグループに属しません。
タグ名	等しい 等しくない 含む 含まない	イベントタグまたは除外するイベントタグを指定。 等しい - イベントにこのタグがある 等しくない - イベントにこのタグが無い 含む - イベントにこの文字列を含むタグがある **含まない** - イベントにこの文字列を含むタグがない
タグの値	等しい 等しくない 含む 含まない	タグと値の組み合わせ、または除外するイベントタグと値の組み合わせを指定します。 等しい - イベントにこのタグと値がある 等しくない - イベントにこのタグと値がない 含む - イベントにこれらの文字列を含むタグと値がある 含まない - イベントにこれらの文字列を含むタグと値がない
テンプレート	等しい 等しくない	テンプレートまたは除外するテンプレートを指定します。 等しい - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスクバリアルールに属する 等しくない - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスクバリアルールに属さない
ホスト	等しい 等しくない	ホストまたは除外するホストを指定します。 等しい - イベントはこのホストに属する 等しくない - イベントはこのホストに属さない

計算のタイプ

条件の計算には、次のオプションを使用できます。

- **And** - すべての条件を満たす

Trigger= 条件として複数のトリガーが選択されている場合、複数のトリガー間で**"And"**を使用することはできません。アクションは1つのトリガーのイベントに基づいてのみ実行できます。

- **Or** - 1つの条件が満たされる
- **And/Or** - 2つの組み合わせ: 異なる条件タイプの AND と同じ条件タイプの OR 例:

ホストグループは Oracle servers と等しい
ホストグループは MySQL servers と等しい
トリガー名に 'Database is down' を含む
トリガー名に 'Database is unavailable' を含む

上記として評価される式は

(ホストグループが Oracle servers と等しい **or** ホストグループが MySQL servers と等しい) **and** (トリガー名に 'Database is down' を含む **or** トリガー名に 'Database is unavailable' を含む)

- カスタム条件式 - アクション条件を評価するためのユーザー定義の計算式。すべての条件 (大文字の A、B、C など) で表される) を含める必要があり、スペース、タブ、角かっこ (), **and** (大文字と小文字を区別)、**or** (大文字と小文字を区別)、**not** (大文字と小文字を区別) を含めることができます。

And/Or を使用した前の例は (A or B) and (C or D) として表されますが、カスタム式では、他の複数の計算方法を使用することもできます。

(A and B) and (C or D)
(A and B) or (C and D)
((A or B) and C) or D
(not (A or B) and C) or not D
等

オブジェクト削除によるアクション無効化

アクション条件/操作で使用されている特定のオブジェクト (ホスト、テンプレート、トリガーなど) が削除されると、その条件/操作が削除され、アクションが無効になり、アクションが正しく実行されなくなります。

ユーザーはアクションを再度有効にすることができます。

この動作は、以下を削除するときに発生します。

- ホストグループ ("ホストグループ" 条件、特定のホストグループでの "リモートコマンド" 操作)
- ホスト ("ホスト" 条件、特定のホストでの "リモートコマンド" 操作)
- テンプレート ("テンプレート" 条件, "テンプレートへのリンク" および "テンプレートからのリンク解除" 操作)
- トリガー ("トリガー" 条件)
- ディスカバリールール ("ディスカバリールール" および "ディスカバリチェック" 条件)

注: リモートコマンドに多くのターゲットホストがあり、そのうちの 1 つを削除すると、このホストのみがターゲットリストから削除され、操作自体は残ります。ただし、それが唯一のホストである場合、操作も削除されます。"テンプレートへのリンク" および "テンプレートからのリンク解除" 操作についても同様です。

"メッセージの送信" 操作で使用するユーザーまたはユーザーグループを削除する場合、アクションは無効になりません。

2 実行内容

概要

すべてのイベントに対して次の操作を定義できます:

- メッセージの送信
- リモートコマンドの実行

Attention:

アクション操作の受信者として定義されたユーザーに対してホストへのアクセスが明示的に "拒否" された場合、またはユーザーがホストに対して定義された権限をまったく持っていない場合、Zabbix サーバーはアラートを作成しません。

ディスカバリおよび自動登録イベントについては、追加の操作を使用できます。

- ホストの追加
- ホストの削除
- ホストの有効化
- ホストの無効化
- ホストグループの追加
- ホストグループの削除
- テンプレートへのリンク
- テンプレートへのリンク解除
- ホストインベントリモードの設定

実行内容の設定

実行内容を設定するには、アクション設定の実行内容タブに移動します。

Action
Operations 2

* Default operation step duration

Operations

Steps	Details
1	Send message to user groups: Zabbix administrators vi
	Add

Recovery operations

Details	Action
Notify all involved	Edit
	Add

Update operations

Details	Action
	Add

Pause operations for suppressed problems

Notify about canceled escalations

* At least one operation must exist.

Add
Cancel

一般的な実行属性:

パラメータ	説明
デフォルトのアクション実行ステップの間隔	デフォルトでの 1 つのアクションステップの間隔 (60 秒から 1 週間)。たとえば 1 時間のステップ継続時間アクションを実行すると、次のステップまでに 1 時間経過します。Zabbix 3.4.0 以降で タイムサフィックス がサポートされています。例:60s, 1m, 2h, 1d
実行内容	Zabbix 3.4.0 以降で ユーザーマクロ がサポートされています。 アクションオペレーション (存在する場合) が次の詳細とともに表示されます。 ステップ - オペレーションが割り当てられているエスカレーションステップ 詳細 - オペレーションのタイプとその受信者/ターゲット アクションリストには、使用されているメディアタイプ (電子メール、SMS、またはスクリプト) と、通知受信者の名前と姓 (ユーザー名の後の括弧内) も表示されます。 開始時刻 - イベント後に操作が実行される時間 継続期間 - ステップ期間が表示されます。ステップでデフォルトの期間が使用されている場合は Default が表示され、カスタムの期間が使用されている場合は時間が表示されます。 アクション - 実行内容を編集および削除するためのリンクが表示されます。
復旧時の実行内容	アクション操作 (存在する場合) が次の詳細と共に表示されます: 詳細 - 操作のタイプとその受信者/ターゲット。 操作リストには、メディアの種類 (電子メール、SMS、またはスクリプト) も表示されます。)、および通知受信者の名前と姓 (ユーザー名の後の括弧内) が使用されます。
更新時の実行内容	アクション - 実行内容を編集および削除するためのリンクが表示されます。 アクション操作 (存在する場合) が次の詳細と共に表示されます: 詳細 - 操作のタイプとその受信者/ターゲット。 操作リストには、メディアの種類 (電子メール、SMS、またはスクリプト) も表示されます。)、および通知受信者の名前と姓 (ユーザー名の後の括弧内) が使用されます。 アクション - 実行内容を編集および削除するためのリンクが表示されます。

パラメータ	説明
メンテナンス中の場合に実行を保留	メンテナンス期間中、運用開始を遅らせるには、このチェックボックスをオンにします。運用開始時、メンテナンス後は、メンテナンス中のイベントも含め全ての運用を実施します。復旧および更新操作には影響しません。 このチェックボックスをオフにすると、メンテナンス期間中も遅延なく操作が実行されます。 このオプションはサービスアクションには使用できません。
キャンセルされたエスカレーションの通知	キャンセルされたエスカレーションに関する通知を無効にするには、このチェックボックスをオフにします (ホスト、アイテム、トリガー、またはアクションが無効になっている場合)。

すべての必須入力フィールドには赤いアスタリスクが付いています。

新しい実行内容の詳細を設定するには、操作ブロックの [Add](#) をクリック、既存の実行内容を編集するには、操作の横にある [Edit](#) をクリックすると、アクションステップの詳細を編集できるポップアップウィンドウが開きます。

実行内容の詳細

Operation details ✕

Operation Send message ▼

Steps 1 - 1 (0 - infinitely)

Step duration 0 (0 - use action default)

*** At least one user or user group must be selected.**

Send to user groups

User group	Action
Zabbix administrators	Remove
Add	

Send to users

User	Action
Add	

Send only to Email ▼

Custom message

Conditions

Label	Name	Action
A	Event is not acknowledged	Remove
Add		

Update
Cancel

パラ
メ
ー
タ

説明

処
理
内
容
ス
テ
ッ
プ
ス
テ
ッ
プ
の
間
隔
処
理
内
容:
メ
ッ
セ
ー
ジ
の
送
信

アクションを選択します。
メッセージを送信 - ユーザーにメッセージを送信します
< リモートコマンド名 > - リモートコマンドを実行します。 **グローバルスクリプト** でアクション処理がスコープとして選択されている場合、コマンドを実行できます。(上記参照)

ユーザーグループに送信

[追加] をクリックして、メッセージを送信するユーザーグループを選択します。
通知を受け取るには、ユーザーグループがホストに対して少なくとも”読み取り”**権限**を持っている必要があります。

ユーザーに送信

[追加] をクリックして、メッセージを送信するユーザーを選択します。
通知を受け取るには、ユーザーがホストに対して少なくとも”読み取り” **権限**を持っている必要があります。

次のメディアのみ使用

定義されたすべてのメディアタイプまたは選択したメディアタイプのみメッセージを送信します。

メッセージのカスタマイズ
件名

カスタムメッセージを設定できます。
webhook 経由の内部イベントに関する通知の場合、カスタムメッセージは必須です。
カスタムメッセージの件名。件名にはマクロが含まれている場合があります。255 文字までに制限されています。

メッセージ

カスタムメッセージ本文。メッセージにはマクロが含まれている場合があります。データベースのタイプに応じて特定の文字数に制限されます (詳細については**メッセージの送信** を参照してください)

処
理
内
容:
<
リ
モ
ー
ト
コ
マ
ン
ド
名
>(/man-

ual/config/notifications/action/operation/remote_command)

パラメータ	説明
ターゲットリスト	<p>Select targets to execute the command on: 現在のホスト - コマンドは、問題イベントを引き起こしたトリガーのホストで実行されます。トリガーに複数のホストがある場合、このオプションは機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。よってリモートコマンドはネストされたグループのホストでも実行されます。 ホストが複数回一致した場合でも、ホスト上のコマンドは 1 回だけ実行されます (たとえば、複数のホストグループから、個別に、およびホストグループから一致した場合など)。 Zabbix サーバー上でカスタムスクリプトを実行すると、ターゲットリストは無効になります。この場合により多くのターゲットを選択すると、スクリプトがサーバー上でより多くの回数実行されるだけです。 グローバルスクリプトの場合、ターゲットの選択は、グローバルスクリプトの設定で可能です。ターゲットリストオプションはサービスアクションには使用できません。この場合リモートコマンドは常に Zabbix サーバーで実行されるためです。</p>
実行条件	<p>アクションを実行するための条件: 確認済みがいいえ - イベント未確認の場合実行 確認済みがはい - イベント確認済みの場合実行 実行条件オプションは サービスアクションでは使用できません。</p>

完了したら [add] をクリックして、アクションを [実行内容] リストに追加します。

1 メッセージの送信

概要

メッセージの送信は、障害について人々に通知する最良の方法の 1 つです。これが、Zabbix が提供する主要なアクションの 1 つである理由です。

設定

Zabbix から通知を送受信できるようにするには、次のことを行う必要があります。

- **定義メディア** にメッセージを送信する

Warning:

ディスカバリ、アクティブエージェントの自動登録、または内部イベントなどの非トリガーイベントの通知を受け取りたい場合は、**ユーザーメディア設定**でデフォルトのトリガー深刻度 ('未分類') を必ずチェックします。

- **アクションの実行内容設定**で定義されたメディアの 1 つにメッセージを送信する

Attention:

Zabbix は、イベントを生成したホストに対して少なくとも '読み取り' 権限を持つユーザーにのみ通知を送信します。トリガー一式の少なくとも 1 つのホストにアクセスできる必要があります。

エスケレーション を使用して、メッセージを送信するためのカスタムシナリオを設定できます。

Zabbix は UTF-8 データを送信するため、Zabbix から電子メールを正常に受信して読み取るには、電子メールサーバー/クライアントが標準の 'SMTP/MIME e-mail' フォーマットをサポートしている必要があります。(件名が ASCII 文字のみの場合は UTF-8 エンコードされません) メッセージの件名と本文は 'SMTP/MIME e-mail' フォーマットの標準に従って base64 でエンコードされています。

全マクロ展開後のメッセージ制限は、**リモートコマンド**のメッセージ制限と同じです。

メッセージの追跡

監視データ → 障害で送信されたメッセージのステータスを表示できます。

アクション列には、実行されたアクションに関する要約情報が表示されます。緑の数字は送信されたメッセージを表し、赤の数字は失敗したメッセージを表します。進行中は、アクションが開始されていることを示します。Failed は、正常に実行されたアクションがないことを示します。

イベント時間をクリックしてイベントの詳細を表示すると、イベントによって送信された（または送信されなかった）メッセージの詳細を含むメッセージアクションブロックも表示されます。

レポート → アクションログでは、アクションが設定されているイベントに対して実行されたすべてのアクションの詳細が表示されません。

2 リモートコマンド

概要

リモートコマンドを使用すると、特定の定義済みコマンドが監視対象ホストで何らかの条件で自動的に実行されるように定義できます。

リモートコマンドはスマートプロアクティブモニタリングの強力なメカニズムです。

この機能の最も明白な使用方法で、次のことを試みることができます。

- 一部のアプリケーション (Web サーバー、ミドルウェア、CRM) が応答しない場合、自動的に再起動する
- 要求に応答しない場合、IPMI の 'reboot' コマンドを使用してリモートサーバーを再起動する
- ディスク容量が不足した場合に自動でディスク容量を解放する (古いファイルの削除、/tmp のクリーニング)
- CPU 負荷に応じて、ある物理ボックスから別の物理ボックスに VM を移行する
- 不十分な CPU (またはディスク、メモリなど) リソースに新しいノードをクラウド環境に追加する

リモートコマンドのアクションの設定は、メッセージを送信する場合と似ていますが、Zabbix がメッセージを送信する代わりにコマンドを実行する点だけが異なります。

リモートコマンドは、Zabbix サーバー、プロキシ、またはエージェントによって実行できます。Zabbix エージェントのリモートコマンドは、Zabbix サーバーまたは Zabbix プロキシ経由で直接実行できます。Zabbix エージェントと Zabbix プロキシの両方で、リモートコマンドはデフォルトで無効になっています。それらは次の方法で有効にできます。

- エージェント設定に AllowKey=system.run[*] パラメータを追加します。
- プロキシ設定で EnableRemoteCommands パラメータを '1' に設定します。

Zabbix サーバーによって実行されるリモートコマンドは、**コマンドの実行** で説明されているように、終了コードのチェックを含めて実行されます。

対象ホストがメンテナンス中であってもリモートコマンドを実行します。

リモートコマンドの制限

すべてのマクロを解決した後のリモートコマンドの制限は、データベースの種類と文字セットによって異なります (ASCII 以外の文字を格納するには、1 バイト以上が必要です)。

データベース	文字数制限	バイト数制限
MySQL	65535	65535
Oracle Database	2048	4000
PostgreSQL	65535	not limited
SQLite (only Zabbix proxy)	65535	not limited

次のチュートリアルでは、リモートコマンドを設定する方法について順を追って説明します。

設定

Zabbix エージェント (カスタムスクリプト) で実行されるリモートコマンドは、最初にエージェントの**設定** で有効にする必要があります。

エージェント設定で許可されている各コマンドに AllowKey=system.run[<command>,*] パラメータが追加されていることを確認して、nowait モードで特定のコマンドを許可します。このパラメータを変更する場合は、エージェントデーモンを再起動します。

Attention:

リモートコマンドは、アクティブな Zabbix エージェントでは機能しません。

次に設定 → アクションで新しいアクションを設定:

- 適切な条件を定義します。この例ではいずれかの Apache アプリケーションで障害が発生した場合にアクションがアクティブになるように設定します。:

Action
Operations

*** Name**

Type of calculation And A and B and C

Conditions	Label	Name
	A	Problem is not suppressed
	B	Application contains Apache
	C	Trigger severity is greater than or equals Disaster
Add		

Enabled

- 実行内容** タブで、実行内容/復旧時の実行内容/更新時の実行内容ブロックの [追加] をクリックします。
- [処理内容] ドロップダウンフィールドから、定義済みスクリプトの 1 つを選択します。

Operation details

Operation Restart webserver

Steps

Send message

Restart webserver

- スクリプトのターゲットリストを選択します

定義済みスクリプト

アクション操作に使用できるすべてのスクリプト (Webhook、スクリプト、SSH、Telnet、IPMI) は、**グローバルスクリプト** で定義されています。

例:

```
sudo /etc/init.d/apache restart
```

この場合、Zabbix は Apache プロセスの再起動を試みます。このコマンドを使用して、コマンドが Zabbix エージェントで実行されることを確認します (Zabbix agent の実行ボタンをクリックします)。

Attention:

sudo の使用に注意してください。Zabbix ユーザーには、デフォルトでシステムサービスを再起動する権限がありません。**sudo** の設定方法に関するヒントについては、以下を参照してください。

Note:

Zabbix エージェントはリモートホストで実行され、着信接続を受け入れる必要があります。Zabbix エージェントはバックグラウンドでコマンドを実行します。

Zabbix エージェントのリモートコマンドは、system.run[,nowait] キーによってタイムアウトなしで実行され、実行結果はチェックされません。Zabbix サーバーと Zabbix プロキシでは、リモートコマンドは、zabbix_server.conf または zabbix_proxy.conf ファイルの TrapperTimeout パラメータで設定されたタイムアウトで実行され、実行結果について**チェック** されます。

アクセス権限

'zabbix' ユーザーが、設定されたコマンドの実行権限を持っていることを確認してください。sudo を使用して、特権コマンドへのアクセスを許可する必要があるかもしれません。アクセスを設定するには root として以下を実行します。

```
# visudo
```

sudoers ファイルで使用できる行の例:

```
# allows 'zabbix' user to run all commands without password.  
zabbix ALL=NOPASSWD: ALL
```

```
# allows 'zabbix' user to restart apache without password.  
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:

一部のシステムでは、sudoers ファイルにより、ローカル以外のユーザーがコマンドを実行できなくなります。これを変更するには、/etc/sudoers の **requiretty** オプションをコメントアウトします。

複数のインターフェースによるリモートコマンド

ターゲットシステムに選択したタイプ (Zabbix エージェントまたは IPMI) の複数のインターフェースがある場合、リモートコマンドはデフォルトのインターフェースで実行されます。

Zabbix エージェント以外のインターフェースを使用して、SSH および Telnet 経由でリモートコマンドを実行することができます。使用可能なインターフェースは、次の順序で選択されます。

- * Zabbix エージェントのデフォルトインターフェース
- * SNMP デフォルトインターフェース
- * JMX デフォルトインターフェース
- * IPMI デフォルトインターフェース

IPMI リモートコマンド

IPMI リモートコマンドの場合、次の構文を使用する必要があります。

```
<command> [<value>]
```

内容

- <command> - スペースなしの 1 つの IPMI コマンド
- <value> - 'on', 'off' または任意の符号なし整数。<value> はオプションパラメーターです。

例

アクションの実行内容でリモートコマンドとして使用できるグローバルスクリプトの例。

例 1

特定の条件で Windows を再起動します。

Zabbix によって障害が検出されたときに Windows を自動的に再起動するには、次のスクリプトを定義します。

スクリプトパラメータ	値
Scope	'Action operation'
Type	'Script'
Command	c:\windows\system32\shutdown.exe -r -f

例 2

IPMI コントロールを使用してホストを再起動します。

スクリプトパラメータ	値
Scope	'Action operation'
Type	'IPMI'
Command	reset

例 3

IPMI コントロールを使用して、ホストの電源をオフにします。

スクリプトパラメータ	値
Scope	'Action operation'
Type	'IPMI'
Command	power off

3 追加操作

概要

このセクションでは、検出/自動登録イベントの**追加操作**の詳細を確認できます。

ホストの追加

ホストが検出されると検出プロセスの最後まで待たず、検出プロセス中すぐに追加されます。

Note:

多くの利用できないホスト/サービスが原因で、ネットワークの検出に時間がかかることがあるため、忍耐力と妥当な IP 範囲を使用することをお勧めします。

ホストを追加するとき、その名前は標準の `gethostbyname` 関数によって決定されます。ホストを解決できる場合は、解決された名前が使用されます。そうでない場合は、IP アドレスが使用されます。また、ホスト名に IPv6 アドレスを使用する必要がある場合は、ホスト名にコロンを使用できないため、すべての ":"(コロン) が "_"(アンダースコア) に置き換えられます。

Attention:

プロキシによる検出を実行する場合でも、ホスト名のルックアップは引き続き Zabbix サーバーで行われます。

Attention:

If a host already exists in Zabbix configuration with the same name as a newly discovered one, versions of Zabbix prior to 1.8 would add another host with the same name. Zabbix 1.8.1 and later adds `_N` to the hostname, where **N** is increasing number, starting with 2.

4 メッセージでのマクロ使用

概要

メッセージの件名とメッセージテキストでは、マクロを使用して障害をより効率的に報告できます。

多くの組み込みマクロに加えて、**ユーザーマクロ** と **式マクロ** もサポートされています。Zabbix でサポートされている **マクロの完全なリスト** が利用可能です。

例

以下の例は、メッセージでマクロを使用する方法を示しています。

例 1

メッセージの件名:

```
Problem: {TRIGGER.NAME}
```

メッセージを受信すると、メッセージの件名は次のように置き換えられます。

```
Problem: Processor load is too high on Zabbix server
```

例 2

メッセージ本文:

```
Processor load is: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

メッセージは受信時に次のように置き換えられます。

```
Processor load is: 1.45
```

例 3

メッセージ本文:

```
Latest value: {?last(/{HOST.HOST}/{ITEM.KEY})}  
MAX for 15 minutes: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}  
MIN for 15 minutes: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

メッセージは受信時に次のように置き換えられます。

```
Latest value: 1.45  
MAX for 15 minutes: 2.33  
MIN for 15 minutes: 1.01
```

例 4

メッセージ本文:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

メッセージを受信すると、本文にイベントの詳細ページへのリンクが含まれます。このページには、イベント、そのトリガー、および同じトリガーによって生成された最新のイベントのリストに関する情報が表示されます。

例 5

トリガー式で複数のホストからの値を通知します。

メッセージ本文:

```
Problem name: {TRIGGER.NAME}  
Trigger expression: {TRIGGER.EXPRESSION}
```

1. Item value on {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Item value on {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})

メッセージを受信すると、メッセージは次のように置き換えられます:

```
Problem name: Processor load is too high on a local host  
Trigger expression: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[percpu,avg1])>5
```

1. Item value on Myhost: 0.83 (Processor load (1 min average per core))
2. Item value on Myotherhost: 5.125 (Processor load (1 min average per core))

例 6

recovery メッセージで障害イベントと復旧イベントの両方の詳細を受信する:

メッセージ本文:

Problem:

```
Event ID: {EVENT.ID}  
Event value: {EVENT.VALUE}  
Event status: {EVENT.STATUS}  
Event time: {EVENT.TIME}  
Event date: {EVENT.DATE}  
Event age: {EVENT.AGE}  
Event acknowledgment: {EVENT.ACK.STATUS}  
Event update history: {EVENT.UPDATE.HISTORY}
```

Recovery:

```
Event ID: {EVENT.RECOVERY.ID}  
Event value: {EVENT.RECOVERY.VALUE}  
Event status: {EVENT.RECOVERY.STATUS}  
Event time: {EVENT.RECOVERY.TIME}  
Event date: {EVENT.RECOVERY.DATE}  
Operational data: {EVENT.OPDATA}
```

メッセージを受信すると、次のように置き換えられます:

Problem:

Event ID: 21874
Event value: 1
Event status: PROBLEM
Event time: 13:04:30
Event date: 2018.01.02
Event age: 5m
Event acknowledgment: Yes
Event update history: 2018.01.02 13:05:51 "John Smith (Admin)"
Actions: acknowledged.

Recovery:

Event ID: 21896
Event value: 0
Event status: OK
Event time: 13:10:07
Event date: 2018.01.02
Operational data: Current value is 0.83

Attention:

Zabbix 2.2.0 以降、元の障害イベントと復旧イベントの個別通知マクロがサポートされています。

3 復旧時実行

概要

復旧操作により、障害が解決されたときに通知を受けることができます。

復旧操作では、メッセージとリモートコマンドの両方がサポートされています。複数の操作を追加できますが、エスカレーションはサポートされていません。すべての操作が1つのステップに割り当てられるため、同時に実行されます。

ユースケース

復旧時実行内容の使用例は次のとおりです。

1. 障害について通知されたすべてのユーザーに通知する

- * 操作タイプとして'リカバリメッセージを送信する'を選択します
- 回復時に複数の操作を行う: 通知の送信とリモート コマンドの実行
 - * メッセージの送信とコマンドの実行の操作タイプを追加
- 外部のヘルプデスク/チケット システムでチケットを開き、障害が解決したらクローズします
 - * ヘルプデスク システムと通信する外部スクリプトを作成する
 - * このスクリプトを実行してチケットを開く操作を持つアクションを作成します
 - * このスクリプトを他のパラメーターで実行し、チケットをクローズする復旧操作を行う
 - * {EVENT.ID} マクロを使用して、元の障害を参照します

復旧時実行内容の設定

Action **Operations 2**

* Default operation step duration

Pause operations for suppressed problems

Operations
 1
[Add](#)

Recovery operations
Notify a
[Add](#)

Update operations
[Add](#)

* At least one oper

復旧時実行内容を設定するにはアクション設定の実行内容タブに移動します。

新しい復旧時実行の詳細を設定するには、[復旧時の実行内容]ブロックで[Add](#)をクリック、既存の復旧時実行を編集するには、操作の横にある[Edit](#)をクリックすると、ステップの詳細を編集できるポップアップウィンドウが開きます。

復旧時実行の詳細

Operation details ✕

Operation

* At least one user or user group must be selected.

Send to user groups

User group	Action
Zabbix administrators	Remove
Add	

Send to users

User	Action
Add	

Send only to

Custom message

復旧イベントには3つのタイプがあります: - メッセージの送信 - 指定したユーザーに復旧メッセージを送信する - 障害通知送信済み

のユーザー全てにメッセージを送信 - 障害イベントで通知されたすべてのユーザーに復旧メッセージを送信する - < リモートコマンド名 > - リモートコマンドを実行します。グローバルスクリプト で事前に定義されていて、アクション処理がスコープとして選択されている場合、コマンドを実行できます。

各タイプのパラメータを次に示します。すべての必須入力フィールドには赤いアスタリスクが付いています。入力が完了したら追加をクリックして、復旧時の実行内容リストに追加します。

:::note classic メッセージのカスタマイズを指定せずに同じ受信者が複数の操作タイプで定義されている場合、重複した通知は送信されません。:::

処理内容: **メッセージの送信**

パラメータ	説明
ユーザーグループに送信	追加をクリックして、復旧メッセージを送信するユーザーグループを選択します。通知を受け取るには、ユーザーグループがホストに対して少なくとも”読み取り”権限を持っている必要があります。
ユーザーに送信	追加をクリックして、復旧メッセージを送信するユーザーを選択します。ユーザーは、通知を受け取るために、ホストに対して少なくとも”読み取り”権限を持っている必要があります。
次のメディアのみ使用	デフォルトの復旧メッセージを、定義されたすべてのメディアタイプまたは選択したメディアタイプのみに送信します。
メッセージのカスタマイズ	選択すると、カスタムメッセージを設定できます。
件名	カスタムメッセージの件名。マクロが含まれている場合があります。
メッセージ	カスタムメッセージ本文。マクロが含まれている場合があります。

処理内容: **リモートコマンド**

パラメータ	説明
ターゲットリスト	<p>コマンドを実行するターゲットを選択:</p> <p>現在のホスト - コマンドは障害イベントを引き起こしたトリガーのホストで実行されます。トリガーに複数のホストがある場合、このオプションは機能しません。</p> <p>ホスト - コマンドを実行するホストを選択します。</p> <p>ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。したがって、リモートコマンドは、ネストされたグループのホストでも実行されます。</p> <p>ホストが複数回一致した場合でも、ホスト上のコマンドは1回だけ実行されます(たとえば複数のホストグループで一致、個別に一致およびホストグループでも一致等)</p> <p>Zabbix サーバーでコマンドを実行した場合、ターゲットリストは意味がありません。この場合、より多くのターゲットを選択しても、サーバー上でコマンドが実行される回数が増えるだけです。</p>

グローバルスクリプトの場合、ターゲットの選択は、グローバルスクリプトの**設定**にあるホストグループにも依存することに注意してください。

処理内容: **障害通知送信済のユーザー全てにメッセージを送信**

パラメータ	説明
メッセージのカスタマイズ	選択すると、カスタムメッセージを設定できます。
件名	カスタムメッセージ件名。件名にはマクロが含まれている場合があります。
メッセージ	カスタムメッセージ本文。メッセージにはマクロが含まれている場合があります。

4 更新時実行

概要

更新時の実行は次のイベントソースを使用したアクションで使用できます:

- トリガー - 障害が他のユーザーによって更新されたとき、すなわち、コメントされた、承認された、深刻度を変更された、(手動で)クローズされたとき。
- サービス - サービスの深刻度を変更されたが、サービスがまだ復旧されていない場合。

更新時の実行では、メッセージとリモートコマンドの両方がサポートされています。複数の操作を追加できますが、エスカレーションはサポートされていません。すべての操作が1つのステップに割り当てられるため、同時に実行されます。

更新時の実行内容設定

更新時実行内容を設定するには、アクション設定の実行内容タブに移動します。

Action
Operations 2

* Default operation step duration

Pause operations for suppressed problems

	Steps	Details	Start in	Duratio
Operations	Add			
Recovery operations	Add	Details		Action
Update operations	Add	Details Notify all involved Send message to user groups: Zabbix administrators via SMS		

新しい更新時の実行内容の詳細を設定するには、更新操作ブロックの [Add](#) を、既存の操作を編集するには、操作の横にある [Edit](#) をクリックします。ステップの実行内容の詳細を編集できるポップアップウィンドウが開きます。

更新時の実行内容は復旧時の実行内容と同じパラメーターセットを使用します。

Operation details ✕

Operation Send message ▾

* At least one user or user group must be selected.

Send to user groups

User group	Action
Zabbix administrators	Remove
Add	

Send to users

User	Action
Add	

Send only to SMS ▾

Custom message

Add
Cancel

5 エスカレーション

概要

エスカレーションを使用すると、通知を送信したり、リモートコマンドを実行したりするためのカスタムシナリオを作成できます。

具体的には、次のようなことです。

- ユーザーは新しい問題をすぐに通知できます
- 問題が解決するまで通知を繰り返すことができます
- 通知の送信を遅らせることができます
- 通知は、別の”より権限が上の”ユーザーグループにエスカレートできます
- リモートコマンドは、障害発生時すぐに、または障害が長期間解決されない場合に実行できます

アクションは、エスカレーションステップに基づいてエスカレートされます。各ステップには期間があります。

個々のステップのデフォルト期間とカスタム期間の両方を定義できます。1つのエスカレーションステップの最小所要時間は60秒です。

通知の送信やコマンドの実行などのアクションは、どのステップからでも開始できます。ステップ1は即時のアクションです。アクションを遅らせたい場合は、後のステップに割り当てることができます。ステップごとに、いくつかのアクションを定義できます。

エスカレーションのステップ数に制限はありません。

エスカレーションは、**オペレーションの設定**時に定義されます。エスカレーションは障害のある操作に対してのみサポートされており、復旧はサポートされていません。

エスカレーション動作その他の例

アクションに複数のエスカレーション手順が含まれている場合、さまざまな状況で何が起るかを考えてみましょう。

状況	動作
障害が発生したホストは、最初の障害通知が送信された後にメンテナンスに入る場合	アクション 設定 のメンテナンス中の場合に実行を保留の設定に応じて、残りのエスカレーション手順はすべて、メンテナンス期間中遅延するか、遅延なしで即時実行されます。メンテナンス期間によって操作がキャンセルされることはありません。

状況	動作
* 期間アクション条件で定義された期間は、最初の通知が送信された後、即時終了する場合 *	残りのすべてのエスカレーション手順が実行されます。期間条件では操作を停止できません。操作ではなく、アクションが開始されるかされないかに関して効果があります。
メンテナンス中に不具合が発生し、メンテナンス終了後も継続する (解消されない) 場合	アクション設定のメンテナンス中の場合に実行を保留の設定に応じて、すべてのエスカレーション手順は、メンテナンスが終了した瞬間、または即時実行されます。
データなしのメンテナンス中に障害が発生し、メンテナンス終了後も継続する (解決されない) 場合	すべてのエスカレーション手順が実行される前に、トリガーが起動するのを待つ必要があります。
さまざまなエスカレーションが連続して続き、重なる場合	新しい各エスカレーションの実行は前のエスカレーションに取って代わりますが、前のエスカレーションで常に行われるエスカレーション手順が少なくとも 1 つ必要です。
進行中のエスカレーション中 (メッセージの送信など)、あらゆるタイプのイベントに基づいて: - アクションは無効 トリガーイベントに基づいて: - トリガーは無効 - ホストまたはアイテムは無効 トリガーに関する内部イベントに基づいて: - トリガーは無効 アイテム/ローレベルディスクバリエーションに関する内部イベントに基づいて: - アイテムは無効 - ホストは無効	この動作は、トリガーのすべての障害評価で作成されたイベントに対するアクションに関連しています。進行中のメッセージが送信され、エスカレーションに関するメッセージがもう 1 つ送信されます。フォローアップメッセージには、メッセージ本文の先頭にキャンセルテキスト (注: エスカレーションがキャンセルされました) があり、理由が示されます (例: 注: エスカレーションがキャンセルされました: アクション '<アクション名>' が無効化されました)。このように、エスカレーションがキャンセルされ、それ以上の手順が実行されないことが受信者に通知されます。このメッセージは、以前に通知を受け取ったすべての人に送信されます。キャンセルの理由は、サーバーのログファイルにも記録されます (デバッグレベル 3=Warning 以上)。
エスカレーションの進行中 (メッセージの送信など) にアクションが削除される	エスカレーションがキャンセルされましたメッセージもオペレーションが終了したが、リカバリオペレーションが設定されていて、まだ実行されていない場合に送信されません。これ以上メッセージは送信されません。情報はサーバーログファイルに記録されます (デバッグレベル 3= 警告以上)

エスカレーション例

例 1

"MySQL 管理者" グループへ 30 分ごとに 1 回 (合計 5 回) 繰り返し通知を送信します。設定するには:

- [実行内容] タブで、デフォルトのアクション実行ステップの間隔を '30m' (30 分) に設定します。
- エスカレーションステップを From '1' To '5' に設定します。
- メッセージ受信者として "MySQL 管理者" グループを選択します。

通知は、障害が発生してから 0:00,0:30,1:00,1:30,2:00 時に送信されます (もちろん問題がより早く解決されない限り)。

障害が解決され、回復メッセージが設定されている場合、このエスカレーションシナリオ内で少なくとも 1 つの障害メッセージを受信したユーザーに送信されます。

Note:

アクティブなエスカレーションを生成したトリガーが無効になっている場合、Zabbix はそれに関するメッセージを、既に通知を受け取っているすべての人に送信します。

例 2

長期障害について遅延通知を送信する。設定するには:

- [実行内容] タブで、デフォルトのアクション実行ステップの間隔を '10h' (10 時間) に設定します。
- エスカレーション手順を From '2' To '2' に設定します

Steps	Details	Start in	Duration
2	Send message to user groups: Managers via SMS	10:00:00	Default

通知は、エスカレーションシナリオのステップ 2、または障害が発生してから 10 時間後にのみ送信されます。

メッセージテキストは '障害が発生して 10 時間以上経過しています' などにカスタマイズできます。

例 3

上司に障害をエスカレーションします。

上記の最初の例では、MySQL 管理者に定期的にメッセージを送信するように設定しました。上記設定では管理者は障害がデータベースマネージャーにエスカレーションされる前に 4 回メッセージを受け取ります。マネージャーは障害が管理者に確認されていない場合にのみメッセージを受け取ることに注意してください。この時点ではおそらく誰も障害に対応していません。

Steps	Details	Start in	Duration
1 - 0	Send message to user groups: MySQL administrators via Email	Immediately	Default
5	Send message to users: Database manager (J S) via all media	02:00:00	Default

操作の詳細 2:

Operation details ✕

Operation type: Send message ▾

Steps: 5 - 5 (0 - Infinitely)

Step duration: 0 (0 - use action default)

*** At least one user or user group must be selected.**

Send to user groups

User group	Action
Add	

Send to users

User	Action
Database manager	Remove
Add	

Send only to: - All - ▾

Custom message:

Subject: Unacknowledged problem: {EVENT.NAME}

Message:
 Problem started at {EVENT.TIME} on {EVENT.DATE}
 Problem name: {EVENT.NAME}
 Host: {HOST.NAME}
 Severity: {EVENT.SEVERITY}

 Original problem ID: {EVENT.ID}
 {TRIGGER.URL}
 {ESC.HISTORY}

Conditions

Label	Name	Action
A	Event is not acknowledged	Remove
Add		

Update
Cancel

カスタマイズされたメッセージで {ESC.HISTORY} マクロが使用されていることに注意してください。マクロには、送信された通知や実行されたコマンドなど、このエスカレーションで以前に実行されたすべてのステップに関する情報が含まれます。

例 4

より複雑なシナリオ。MySQL 管理者への複数のメッセージとマネージャーへのエスカレーションの後、Zabbix は MySQL データベースの再起動を試みます。障害が 2:30 時間存在し、確認されていない場合に発生します。

それでも障害が解決しない場合は、さらに 30 分後に Zabbix がすべてのゲストユーザーにメッセージを送信します。

これで障害が解決しない場合は、さらに 1 時間後に、Zabbix は IPMI コマンドを使用して MySQL データベース (2 番目のリモートコマンド) でサーバーを再起動します。

Action Operations 5

* Default operation step duration

Pause operations for suppressed problems

Operations	Steps	Details	Start in	Duration
	1 - 0	Send message to user groups: MySQL Administrators via Email	Immediately	Default
	5	Send message to users: Database Manager (J S) via all media	02:00:00	Default
	6	Run script "Restart MySQL" on current host	02:30:00	Default
	7	Send message to user groups: Guests via all media	03:00:00	Default
	9	Run script "Restart server" on current host	04:00:00	Default

[Add](#)

例 5

複数の操作が 1 つのステップに割り当てられ、カスタム間隔が使用されるエスカレーション。デフォルトの操作ステップの所要時間は 30 分です。

Action Operations 4

* Default operation step duration

Pause operations for suppressed problems

Operations	Steps	Details	Start in	Duration
	1 - 4	Send message to user groups: MySQL Administrators via Email	Immediately	Default
	5 - 6	Send message to users: Database Manager (J S) via all media	02:00:00	1h
	5 - 7	Send message to user groups: Zabbix administrators via Email	02:00:00	10m
	11	Send message to user groups: Guests via Email	04:00:00	Default

[Add](#)

通知は次のように送信されます:

- 問題が発生してから 0:00、0:30、1:00、1:30 に MySQL 管理者に
- 2:00 と 2:10 にデータベースマネージャーに (3:00 ではありません。ステップ 5 と 6 が次の操作と重複しているため、次の操作の 10 分の短いカスタムステップ期間は、ここで設定しようとした 1 時間の長いステップ期間をオーバーライドします)
- 問題が発生した後の 2:00、2:10、2:20 に Zabbix 管理者に (10 分間のカスタムステップ継続時間)
- 問題が発生してから 4:00 時間後にゲストユーザーに (ステップ 8 と 11 の間に戻る 30 分のデフォルトのステップ継続時間)

サポート対象外通知の受信 {#manual-config-notifications-unsupported_item}

概要

Zabbix 2.2 以降、サポートされていないアイテムに関する通知の受信がサポートされています。

これは、Zabbix の内部イベントの概念の一部であり、受信時にユーザーに通知できるようにします。**内部イベント** は状態の変化を反映します:

- アイテムが '通常' から 'サポート対象外' になったとき (および元に戻ったとき)
- トリガーが '通常' から '不明' になったとき (および元に戻ったとき)
- ローレベルディスカバリルールが '通常' から 'サポート対象外' (およびその逆) になるとき

このセクションでは、アイテムがサポート対象外になったときに通知を受け取る方法を示します。

設定

おおまかには通知を設定するプロセスは以前に Zabbix でアラートを設定したことがある人には馴染みのあるものです。

ステップ 1

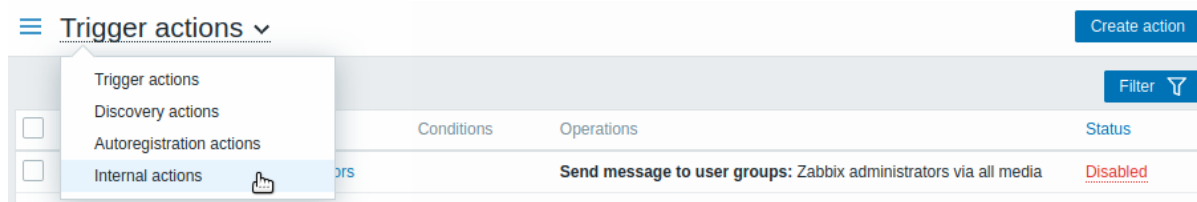
通知に使用する電子メール、SMS、スクリプトなどの**何れかのメディア**を構成します。このタスクを実行するには、マニュアルの対応するセクションを参照してください。

Attention:

内部イベントの通知にはデフォルトの深刻度('未分類')が使用されるため、内部イベントの通知を受け取りたい場合は、**ユーザーメディア**を構成するときにチェックしたままにします。

ステップ 2

設定 → アクションに移動し、第 3 レベルのメニュー (またはページタイトルのドロップダウン) から 内部イベントアクションを選択します。

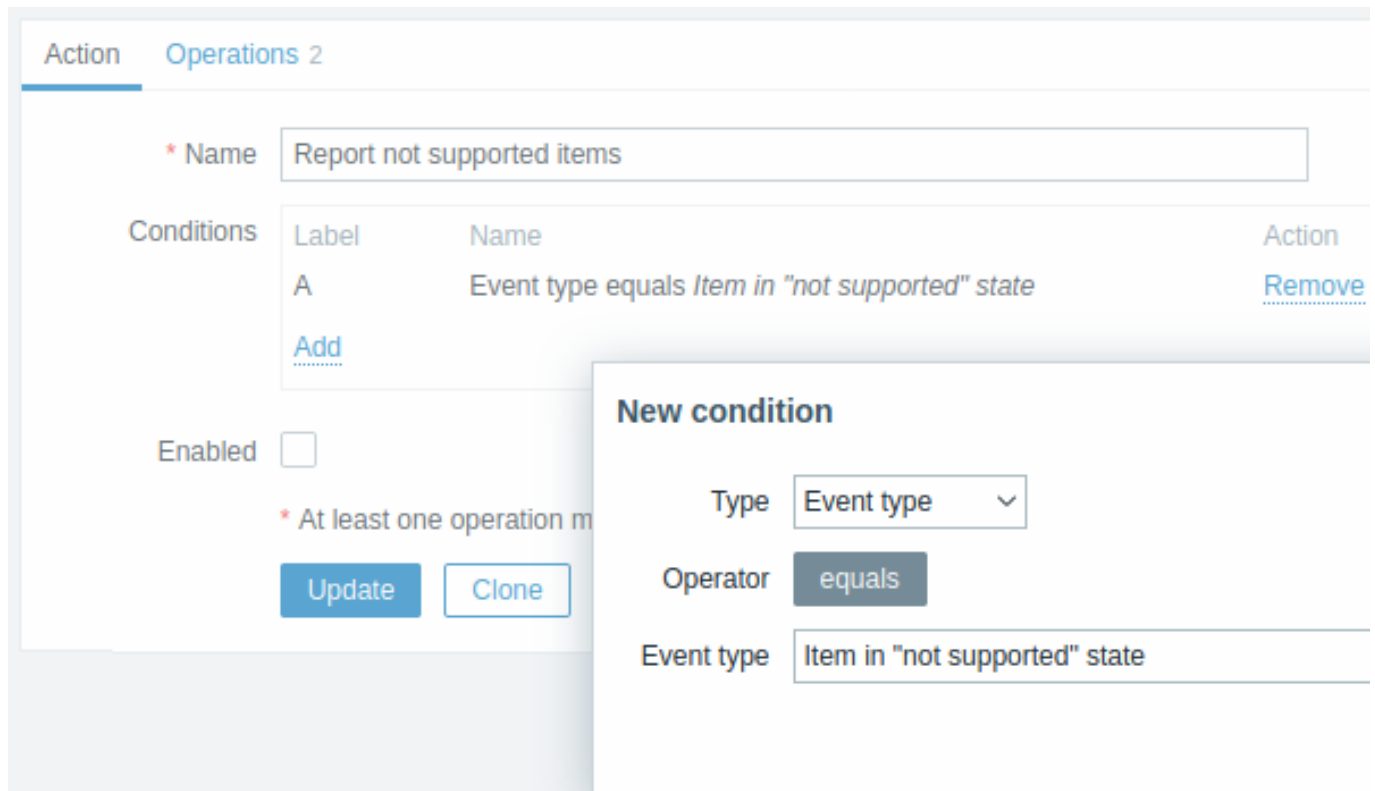


右上のアクションの作成をクリックして、アクション設定フォームを開きます。

ステップ 3

アクションタブで、アクションの名前を入力します。次に実行条件ブロックで 追加をクリックして、新しい条件を追加します。

新しい条件ポップアップウィンドウで、条件タイプとして イベントタイプを選択し、イベントタイプの値として アイテムが" 取得不可" の状態を選択します。



追加をクリックして 実行条件ブロックに条件を追加してください。

ステップ 4

In the **Operations** tab, click on Add in the Operations block and select some recipients of the message (user groups/users) and the media types (or 'All') to use for delivery. 実行内容タブで、実行内容ブロックの 追加をクリックし、配信に使用するメッセージの受信者 (ユーザーグループ/ユーザー) とメディアタイプ (または「すべて」) を選択します。

障害メッセージの件名/内容をカスタマイズする場合は、メッセージのカスタマイズチェックボックスをオンにします。

* Default operation step duration

Operations	Steps	Details
	1	Send message to user groups: Zabbix administrators via all media
		Add

Recovery operations	Details	Action
	Notify all involved	Edit Remove
	Add	

Operation details

Operation type **Send message**

Steps - (0 - infinitely)

Step duration (0 - use action default)

* At least one user or user group must be selected.

Send to user groups	User group	Action
	Zabbix administrators	Remove
	Add	

Send to users	User	Action
	Add	

Send only to

Custom message

Subject

Message

Click on Add to actually list the operation in the Operations block. 追加をクリックして 実行内容ブロックに実行内容を追加します。

複数の通知を受け取りたい場合は、ステップの間隔 (メッセージの送信間隔) を設定し、別のステップを追加します。

ステップ 5

The **Recovery operations** block allows to configure a recovery notification when an item goes back to the normal state. Click on Add in the Recovery operations block, select the operation type, the recipients of the message (user groups/users) and the media types (or 'All') to use for delivery. 復旧時の実行内容ブロックを使用すると、アイテムが通常の状態に戻ったときの回復通知を設定できます。復旧時の実行内容ブロックで追加をクリックし、処理内容、メッセージの受信者 (ユーザーグループ/ユーザー)、および配信に使用するメディアタイプ (または「すべて」) を選択します。

障害メッセージの件名/内容をカスタマイズする場合は、メッセージのカスタマイズチェックボックスをオンにします。

The screenshot shows the Zabbix configuration interface for Recovery operations. The main window is titled 'Operations 2' and has a 'Default operation step duration' of '1h'. Below this, there is a table of operations with one entry: '1 Send m...'. A modal window titled 'Operation details' is open, showing 'Operation type' set to 'Notify all involved', 'Custom message' checked, and a subject line '{ITEM.STATE}: {HOST.NAME}:{ITEM.NAME}'. The message body contains: 'Host: {HOST.NAME}', 'Item: {ITEM.NAME}', 'Key: {ITEM.KEY}', and 'State: {ITEM.STATE}'.

実行内容の詳細ポップアップウィンドウで追加をクリックして、復旧時の実行内容ブロックに実行内容を追加します。

ステップ 6

完了後、フォームの下部にある追加ボタンをクリックします。

これで完了です。一部のアイテムがサポートされなくなった場合、Zabbix から最初の通知を受け取ることができるようになりました。

11 マクロ

概要

Zabbix は、さまざまな状況で使用できる多数の組み込みマクロをサポートしています。これらのマクロは変数であり、特定の構文で識別されます:

{MACRO}

マクロは、コンテキストに応じて特定の値に解決されます。

マクロを効果的に使用することで、時間を節約し、Zabbix の設定をより透過的にすることができます。

典型的な用途の 1 つとして、テンプレートでマクロを使用することがあげられます。例えばテンプレートのトリガーは、"{HOST.NAME}のプロセッサ負荷が高すぎます" という名前となります。テンプレートが Zabbix サーバーなどのホストに適用され、トリガーが監視セクションに表示されると、この名前は "Zabbix サーバーのプロセッサ負荷が高すぎます" に解決されます。

マクロは、アイテムキーパラメータで使用できます。たとえば `item.key[server_{HOST.HOST}_local]` のようにパラメータの一部のみとしても可能です。解決されたマクロにあいまいな特殊記号存在する場合、Zabbix が処理するのでパラメータをダブルクオートで囲む必要はありません。

Zabbix には他の種類のマクロもあります。

Zabbix は次のマクロをサポートしています。

- {MACRO} - 組み込みマクロ (全リストを参照)

- `{<macro>.<func>(<params>)}` - マクロ関数
- `{$MACRO}` - ユーザー定義のマクロ, オプションでコンテキスト付き
- `{#MACRO}` - ローレベルディスカバリマクロ
- `{?EXPRESSION}` - 式マクロ

1 マクロ関数

概要

マクロ関数は、`macro`の値をカスタマイズする機能を提供します。

マクロは、必ずしも簡単に扱えない値を解決することがあります。値が長かったり、抽出したい特定の部分文字列が `
` 含まれていたりする場合があります。このような場合に、マクロ関数が役に立ちます。

マクロ関数の書式は:

```
{<macro>.<func>(<params>)}
```

ここで:

- `<macro>` - カスタマイズするマクロ (例 `{ITEM.VALUE}` や `{L#LDMACRO}` など) です。
- `<func>` - 適用する関数です。
- `<params>` - カンマで区切られた関数パラメータのリストです。パラメータが" " (スペース) で始まる場合、" で始まる場合、あるいは), , を含む場合は、引用符で囲む必要があります。

例:

```
{{TIME}}.fmttime(format,time_shift)}
{{ITEM.VALUE}}.regsub(pattern,output)}
{#LLDMACRO}.regsub(pattern,output)}
```

サポートされるマクロ関数

関数	説明	パラメーター	サポート対象
fmtnum (<digits>)	小数点以下に表示される桁数を制御する数値の書式設定	digits - 小数点以下の桁数。末尾のゼロは生成されません。	{ITEM.VALUE} {ITEM.LASTVALUE} 式マクロ
fmttime (<format>,<time_shift>)	時間のフォーマット	format - <code>strftime</code> 関数のフォーマットと互換性のある必須のフォーマット文字列	

{TIME}

time_shift - フォーマット前の時間に適用されるタイムシフト。
 -<N><time_unit> または +<N><time_unit> で開始する必要があります。

N - 加算または減算する時間単位の数。

time_unit - h (時間)、d (日)、w (週)、M (月) または y (年)

Zabbix 5.4 以降、time_shift パラメータは複数ステップの時間操作をサポートし、時間単位の先頭にシフトするために /<time_unit> を含めることができます (/d - 真夜中、/w - 初日) 曜日 (月曜日)、/M - 月の最初の日など) 例:

-1w - ちょうど 7 日前

-1w/w - 前の週の月曜日

-1w/w+1d - 前週の火曜日

時間操作は優先度なしで左から右に計算されることに注意してください。たとえば、-1M/d+1h/w は ((-1M/d)+1h)/w として解析されます。

iregsub (<pattern>,<output>)

正規表現	pattern	{ITEM.VALUE}
一致 (大文字と小文字を区別しない) による部分文字列の抽出	- 一致する正規表現	{ITEM.LASTVALUE}
	output -	ローレベルディスクカバリマクロ (ローレベルディスクカバリルーアルフィルタを除く)
	出力オプション。	
	\1 - \9	ディスクカバリルーアルフィルタを除く)
	プレースホルダーはグループをキャプチャするためにサポートされています。 \0	
	一致したテキストを返します。	

regsub (<pattern>,<output>)

正規表現	pattern	{ITEM.VALUE}
一致 (大文字と小文字を区別する) による部分文字列の抽出	- 一致する正規表現	{ITEM.LASTVALUE}
	output -	ローレベルディスクカバリマクロ (ローレベルディスクカバリルーアルフィルタを除く)
	出力オプション。	
	\1 - \9	ディスクカバリルーアルフィルタを除く)
	プレースホルダーはグループをキャプチャするためにサポートされています。 \0	
	一致したテキストを返します。	

関数がサポートされている場所で使用され、マクロ関数をサポートしていないマクロに適用されると、そのマクロは 'UNKNOWN' と評価されます。

pattern が正しい正規表現でない場合、マクロは'UNKNOWN' と評価されます (ローレベルディスカバリマクロを除く。この場合、関数は無視され、マクロは展開されません)。

マクロ関数がサポートされていない場所でマクロに適用された場合、その関数は無視されます。

例

マクロ関数を使用してマクロ値をカスタマイズする方法を、受信値に関する以下の例で説明します:

受信値	マクロ	出力
24.3413523	{{ITEM.VALUE}.fmtnum(2)}	24.34
24.3413523	{{ITEM.VALUE}.fmtnum(0)}	24
12:36:01	{{TIME}.fmttime(%B)}	October
12:36:01	{{TIME}.fmttime(%d %B,-1M/M)}	1 September
123Log line	{{ITEM.VALUE}.regsub("[0-9]+", Problem))	Problem
123 Log line	{{ITEM.VALUE}.regsub("[0-9]+", "Problem Problem"))	Problem
123 Log line	{{ITEM.VALUE}.regsub("[0-9]+", "Problem ID: 123 Problem ID: \1")}	Problem ID: 123
Log line	{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}	"Problem ID: "
MySQL crashed errno 123	{{ITEM.VALUE}.regsub("[0-9]+", "Problem ID: MySQL_123 " " Problem ID: \1_2 ")}	Problem ID: MySQL_123
123 Log line	{{ITEM.VALUE}.regsub("[1-9]+", *UNKNOWN* (invalid regular expression) "Problem ID: \1")}	*UNKNOWN*
customername_1	{{#IFALIAS}.regsub(".*"_[0-9]+", \1)}	customername
customername_1	{{#IFALIAS}.regsub(".*"_[0-9]+", \2)}	1
customername_1	{{#IFALIAS}.regsub(".*"_[0-9]+", {{#IFALIAS}.regsub(".*"_[0-9]+", \1)}) (invalid regular expression)	customername
customername_1	MACRO:"{{#IFALIAS}.regsub("[0-9]+", "customername \1")}"	customername
customername_1	MACRO:"{{#IFALIAS}.regsub("[0-9]+", \2)}"	1
customername_1	MACRO:"{{#IFALIAS}.regsub("[0-9]+", {{#IFALIAS}.regsub("[0-9]+", \1)})" (invalid regular expression)	customername
customername_1	MACRO:"{{#IFALIAS}.regsub("[0-9]+", "customername \1")}"	customername
customername_1	MACRO:"{{#IFALIAS}.regsub("[0-9]+", \2)}"	1
customername_1	MACRO:"{{#IFALIAS}.regsub("[0-9]+", {{#IFALIAS}.regsub("[0-9]+", \1)})" (invalid regular expression)	customername

全アイテム値の表示

{ITEM.VALUE} および {ITEM.LASTVALUE} マクロが解決された時に長い値になる場合は 20 文字に切り捨てられます。これらのマクロの完全な値を表示するには、次のようなマクロ関数を使用します。

```
{{ITEM.VALUE}.regsub(".*", \1)}<br> {{ITEM.LASTVALUE}.regsub(".*", \1)}
```

2 ユーザーマクロ

概要

Zabbix では、supportedマクロに加え、より柔軟性の高いユーザーマクロがサポートされています。

ユーザーマクロは、グローバル、テンプレート、ホストレベルで定義することができます。これらのマクロは特別な構文を持っています:

```
{$MACRO}
```

Zabbix は以下の優先順位に従ってマクロを解決します:

1. ホストレベルのマクロ (最初にチェックされます)
2. ホストの第 1 階層のテンプレート (ホストに直接リンクされたテンプレート) に定義されたマクロ (テンプレート ID でソートされます)
3. ホストの第 2 階層のテンプレートに対して定義されたマクロ (テンプレート ID でソートされます)
4. ホストの第 3 階層のテンプレートに対して定義されたマクロで、テンプレート ID 順に並べられます。
5. グローバルマクロ (最後にチェック)

つまり、あるホストにマクロが存在しない場合、Zabbix はホストテンプレートから徐々にマクロを探そうとします。
 それでも見つからない場合、グローバルマクロが存在すれば、それを使用します。

Warning:

同じレベルのリンクされた複数のテンプレートに同じ名前のマクロが存在する場合、ID が最も小さいテンプレートの
 マクロが使用されます。したがって、複数のテンプレートに同じ名前のマクロが存在することは、設定上のリスクとなります。

Zabbix がマクロを見つけることができない場合、マクロは解決されません。

Attention:

マクロ (ユーザーマクロを含む) は、複雑な設定をより分かりやすくするため、設定セクション (トリガーリストなど) で
 未解決のままになっています。

以下の箇所でユーザーマクロを使用することができます:

- item キーパラメーター
- item の更新間隔とフレキシブル間隔
- トリガー名と説明
- トリガ式パラメータと定数 (参照: [examples](#))
- 他の多くの場所 - [full list](#)を参照してください

グローバルマクロとホストマクロの一般的な使用例

- グローバルマクロを複数の場所で使用し、マクロの値を変更すると、ワンクリックですべての場所に設定変更が適用されます。
- ホスト固有の属性 (パスワード、ポート番号、ファイル名、正規表現など) を持つテンプレートを活用します。

設定

ユーザーマクロを定義するには、フロントエンドの対応する場所に移動します。

- グローバルマクロの場合、Administration → General → Macros をご覧ください。
- ホストとテンプレートレベルのマクロは、ホストまたはテンプレートのプロパティを開き、Macros タブを探します。




Note:

ユーザー・マクロがテンプレート内の item またはトリガーで使用されている場合、テンプレート内の item や
 トリガーでユーザーマクロが使用される場合、たとえそれがグローバルレベルで定義されていても、そのマクロを
 テンプレートに追加することが推奨されます。そうすれば、マクロのタイプが text の場合、テンプレートを
 XML にエクスポートして別のシステムでインポートしても、期待通りに動作します。
 シークレットマクロの値は **exported** されません。

ユーザーマクロは以下の属性を持ちます:

Macro	Value		Description
{MYSQL_PASSWORD}		description
{MYSQL_USERNAME}		description
{SECRET_PASSWORD}	path/to/secret:password		description
{SECRET_USERNAME}	path/to/secret:username		description
{SNMP_COMMUNITY}	public		description
{WORKING_HOURS}	1-5,09:00-18:00		description

[Add](#)

パラメータ	説明
Macro	Macro name. The name must be wrapped in curly brackets and start with a dollar sign. Example: <code>{\$FRONTEND_URL}</code> . The following characters are allowed in the macro names: A-Z (uppercase only) , 0-9 , <code>_</code> , <code>.</code>
Value	Macro value. Three value types are supported: Text (default) - plain-text value Secret text - the value is masked with asterisks, which could be useful to protect sensitive information such as passwords or shared keys. Vault secret - the value contains a reference path (as 'path:key', for example "secret/zabbix:password") to a Vault secret Note that while the value of a secret macro is hidden from sight, the value can be revealed through the use in items. For example, in an external script an 'echo' statement referencing a secret macro may be used to reveal the macro value to the frontend because Zabbix server has access to the real macro value. To select the value type click on the button at the end of the value input field:  icon indicates a text macro;  icon indicates a secret text macro. Upon hovering, the value field transforms into a button, which allows to enter a new value of the macro (to exit without saving a new value, click the backwards arrow (↶)).  icon indicates a secret Vault macro. Maximum length of a user macro value is 2048 characters (255 characters in versions before 5.2.0).
Description	Text field used to provide more information about this macro.

Note:

シークレットマクロを含む URL は、マクロが"*****"として解決されるため、動作しないことがあります。

Attention:

トリガ式では、ユーザマクロはパラメータや定数を参照する場合に解決されます。
パラメータまたは定数を参照している場合、解決されます。ホスト、item キー、関数、演算子、または別のトリガ式を
参照している場合は、解決されません。シークレットマクロは、トリガ式では使用できません。

例

例 1

"Status of SSH daemon" item キーにホストレベルマクロを使用する。

```
net.tcp.service[ssh,,$SSH_PORT]
```

この項目は、複数のホストで **{\$SSH_PORT}** の値が定義されている場合に、複数のホストに割り当てることができます。

例 2

"CPU load is too high" トリガーでのホストレベルマクロの使用:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

このようなトリガーは、個々のホストで編集するのではなく、テンプレート上で作成されることになります。

Note:

値の量に関数のパラメーターとして使用する場合（例えば、**max(/host/key,#3)**）、`
 SOME_PERIOD => #3
` というようにマクロ定義にハッシュ記号を入れてください。

例 3

“CPU load is too high” トリガーで 2 つのマクロを使用する:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

マクロはトリガー関数（この例では関数 **min()**）のパラメータとして使用することができることに注意。

例 4

agent の利用不可条件と item の更新間隔を同期させる。:

- {INTERVAL} マクロを定義し、item の更新間隔に使用する。
- {INTERVAL} を agent 使用不可トリガーのパラメータとして使用する。

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

例 5

稼働時間の設定を一元化する。

- グローバルな {WORKING_HOURS} マクロを作成し、1-5,09:00-18:00 とします。
- Administration → General → GUI の Working time フィールドで使用します。
- Administration → User → Media の When active フィールドで使用します。
- 稼働時間中に、より頻繁に item のポーリングを設定するために使用します。

Update interval

Custom intervals	Type	Interval	Period
	Flexible	Scheduling	<input type="text" value="{ \$SHORT_INTERVAL }"/>
			<input type="text" value="{ \$WORKING_HOURS }"/>

- Time period の動作条件で使用します。
- 必要に応じて、Administration → General → Macros で稼働時間を調整します。

例 6

ホストプロトタイプマクロを使用して、検出されたホストの item を設定します:

ホストプロトタイプで、{ \$SNMPVALUE } を { #SNMPVALUE } **low-level discovery** マクロ値として定義します:

Host prototype macros

Macro	Value
<input type="text" value="{ \$SNMPVALUE }"/>	<input type="text" value="{ #SNMPVALUE }"/> <input type="button" value="T"/>

[Add](#)

- ホストプロトタイプに Generic SNMPv2 テンプレートを割り当てます。
- {SNMPVALUE} を使用します。汎用 SNMPv2 テンプレート項目の SNMP OID* フィールドに { \$SNMPVALUE } を使用します。

ユーザーマクロのコンテキスト

[user macros with context](#) を参照してください。

3 コンテキストユーザーマクロ

概要

ユーザーマクロ でオプションのコンテキストを使用して、デフォルト値をコンテキスト固有の値で上書きすることができます。

マクロ名にコンテキストが追加されます。構文は、コンテキストが静的テキスト値であるかどうかによって異なります。

```
{${MACRO:"static text"}}
```

または正規表現:

```
{${MACRO:regex:"regular expression"}}
```

正規表現コンテキストを持つマクロは、ユーザーマクロ構成でのみ定義できることに注意してください。regex: プレフィックスがトリガー式のようにユーザーマクロコンテキストとして他の場所で使用されている場合、静的コンテキストとして扱われます。

コンテキストの引用はオプションです (**重要な注意事項** も参照してください)

マクロコンテキストの例:

例	説明
<code>{\${LOW_SPACE_LIMIT}}</code>	コンテキストのないユーザーマクロ
<code>{\${LOW_SPACE_LIMIT:/tmp}}</code>	コンテキストを持つユーザーマクロ (静的文字列)
<code>{\${LOW_SPACE_LIMIT:regex:"~/tmp\$"}}</code>	コンテキストを持つユーザーマクロ (正規表現)。 <code>{\${LOW_SPACE_LIMIT:/tmp}}</code> と同じ。
<code>{\${LOW_SPACE_LIMIT:regex:"~/var/log/.*\$"}}</code>	コンテキストを持つユーザーマクロ (正規表現)。/var/log/ で始まるすべての文字列に一致。

ユースケース

コンテキストを持つユーザーマクロを定義して、トリガー式で (ローレベルディスクバリによって取得された値に基づく) より柔軟なしきい値を実現できます。たとえば、次のマクロを定義できます。- `{${LOW_SPACE_LIMIT} = 10 - ${LOW_SPACE_LIMIT:/home} = 20 - ${LOW_SPACE_LIMIT:regex:"^[a-z]+$"} = 30`

次に、ローレベルディスクバリマクロを、マウントされたファイルシステム検出のトリガープロトタイプでマクロコンテキストとして使用できます:

```
last(/host/vfs.fs.size[#{FSNAME},pfree])<{${LOW_SPACE_LIMIT:"#{FSNAME}"}
```

検出後、検出されたマウントポイントまたはファイルシステムの種類に応じて、さまざまな低容量しきい値がトリガーに適用されます。障害イベントは、次の場合に生成されます。

- /home フォルダの空きディスク容量が 20% 未満 - 正規表現パターン (/etc, /tmp, /var など) に一致するフォルダの空きディスク容量が 30% 未満
- 正規表現パターンに一致せず、/home ではないフォルダの空きディスク容量が 10% 未満

重要事項

- コンテキストを持つ複数のユーザーマクロが存在する場合、Zabbix は最初に単純なコンテキストマクロを照合し、次にコンテキストマクロを未定義の順序で正規表現と照合しようとします。

Warning:

未定義の動作を避けるために、同じ文字列に一致する別のコンテキストマクロを作成しないでください。

- コンテキストを持つマクロがホスト、リンクされたテンプレート、またはグローバルで見つからない場合、コンテキストのないマクロが検索されます。
- コンテキストでは、ローレベルディスクバリマクロのみがサポートされています。その他のマクロはすべて無視され、プレーンテキストとして扱われます。

技術的には、マクロコンテキストは **item key** パラメータと同様のルールを使用して指定されますが、文字がある場合、マクロコンテキストは複数のパラメータとして解析されません。

- コンテキストに } 文字が含まれている場合、または " 文字で始まる場合、マクロコンテキストは " で引用する必要があります。引用されたコンテキスト内の引用符は、\ 文字でエスケープする必要があります。
- \ 文字自体はエスケープされません。つまり、\ 文字で終わる引用符付きコンテキストを持つことは不可能です - マクロ `{${MACRO:"a:\b\c"}}` は無効です。
- コンテキスト内の先頭のスペースは無視されますが、末尾のスペースは無視されません:
 - たとえば、`{${MACRO:A}` は `{${MACRO: A}` と同じですが、`{${MACRO:A }` とは異なります。
- 先頭の引用符の前と末尾の引用符の後のすべてのスペースは無視されますが、引用符内のすべてのスペースは無視されません。
 - マクロ `{${MACRO:"A"}`、`{${MACRO: "A"}`、`{${MACRO:"A" }`、および `{${MACRO: "A" }` は同じですが、マクロ `{${MACRO:"A" }` および `{${MACRO:" A "}` ではありません。

{`MACRO:A`}, {`MACRO: A`}, および {`MACRO:"A"`} のマクロは、同じコンテキストを持っているため、すべて同等です。これは、'`key[a]`'、'`key[a]`'、および'`key["a"]`' が意味的に同じである項目キーとは対照的ですが、一意性のために異なります。

4 ローレベルディスカバリマクロ

概要

これは **low-level discovery** (LLD) 関数の中で使用されるマクロの一種です。:

{`MACRO`}

LLD ルールの中で使用されるマクロで、ファイルシステム名、ネットワークインターフェース、SNMP OID などの実数値を返します。

これらのマクロは、アイテム、トリガー、グラフの prototypes の作成に使用することができます。そして、実際の `
` ファイルシステムやネットワーク・インターフェイスなどを発見する際に、これらのマクロを実際の値で置き換え、`
` 実際の item やトリガー、グラフを作成するためのベースとします。

これらのマクロは、仮想マシン **discovery** でホストとホストグループの `
` prototypes を作成する際にも使用されます。

いくつかの低レベルの発見マクロは Zabbix の LLD 関数と一緒に "pre-packaged" として提供されています。 `
` - {`FSNAME`}, {`FSTYPE`}, {`IFNAME`}, {`SNMPINDEX`}, {`SNMPVALUE`} `
` ただし、**custom** ローレベルディスカバリルールを `
` 作成する場合は、これらの名前に準拠することは必須ではありません。その場合は、他の任意の LLD マクロ名を使用し、`
` その名前を参照することができます。

サポートされる箇所

以下の箇所で LLD マクロを使用することができます:

- in the low-level discovery rule filter
- for item prototypes in
 - name
 - key parameters
 - unit
 - update interval¹
 - history storage period¹
 - trend storage period¹
 - item value preprocessing steps
 - SNMP OID
 - IPMI sensor field
 - calculated item formula
 - SSH script and Telnet script
 - database monitoring SQL query
 - JMX item endpoint field
 - description
 - HTTP agent URL field
 - HTTP agent HTTP query fields field
 - HTTP agent request body field
 - HTTP agent required status codes field
 - HTTP agent headers field key and value
 - HTTP agent HTTP authentication username field
 - HTTP agent HTTP authentication password field
 - HTTP agent HTTP proxy field
 - HTTP agent HTTP SSL certificate file field
 - HTTP agent HTTP SSL key file field
 - HTTP agent HTTP SSL key password field
 - HTTP agent HTTP timeout¹ field
 - tags
- for trigger prototypes in
 - name
 - operational data
 - expression (only in constants and function parameters)
 - URL
 - description
 - tags
- for graph prototypes in
 - name
- for host prototypes in

- name
- visible name
- custom interface fields: IP, DNS, port, SNMP v1/v2 community, SNMP v3 context name, SNMP v3 security name, SNMP v3 authentication passphrase, SNMP v3 privacy passphrase
- host group prototype name
- host tag value
- host macro value
- (full list を参照してください。)

これらのすべての場所で、LLD マクロは静的なユーザー [macro context] (/manual/config/macros/user_macros_context) の中で使用することができます。

マクロ関数の使用

マクロ関数は、ローレベルディスカバリマクロでサポートされており (ローレベルディスカバリルールフィルターを除く)、
 正規表現を使用してマクロ値の特定の部分を抽出することが可能です。

たとえば、イベントタグ付けのために、次の LLD マクロから顧客名とインターフェイス番号を抽出したい場合があります。

```
{#IFALIAS}=customername_1
```

これを行うには、regsub マクロ関数をトリガープロトタイプのイベントタグの値フィールドのマクロと一緒に
 使用することができます:

Tags		
Customer	{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}	Remove
Interface	{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}	Remove

引用符で囲まれていない item [key parameters] (/manual/config/items/item/key#key_parameters) にはカンマが使えないので
 マクロ関数を含むパラメータは引用符で囲む必要があることに注意しましょう。パラメータ内の二重引用符をエスケープするには
 バックスラッシュ (\) 文字を使用する必要があります。

例:

```
net.if.in["{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)"] , bytes]
```

マクロ関数の構文の詳細については、[Macro functions](#) を参照してください。

Zabbix 4.0 からローレベルディスカバリマクロでマクロ関数がサポートされています。

脚注

¹ でマークされたフィールドでは、1 つのマクロでフィールド全体を埋める必要があります。
 フィールド内の複数のマクロや、テキストと混在するマクロはサポートされていません。

5 式マクロ

概要

式マクロは、式の計算に役立ちます。内部のすべてのマクロを展開し、結果の式を評価することによって計算されます。

式マクロには特別な構文があります:

```
{?EXPRESSION}
```

{HOST.HOST<1-9>} および {ITEM.KEY<1-9>} マクロは、式マクロ内でサポートされています。{ITEM.KEY<1-9>} マクロは、Zabbix 6.0.9 以降、式マクロでサポートされています。

使用法

次の場所で使用可能です。

- グラフ名
- マップ要素のラベル
- マップシェーブラベル
- マッピングラベル

次のセットから **1** つの関数のみ式マクロとして使用できます: avg,last,max,min 例:

```
{?avg(/{HOST.HOST}/{ITEM.KEY},1h)}
```


{?last(/host/item1)/last(/host/item2)}, {?count(/host/item1,5m)}, {?last(/host/item1)*10} 等の式は前述の場所で機能しません。

ただし、

- トリガーイベント名
- トリガーベースの通知とコマンド
- 障害の更新通知とコマンド

では複雑な式を使用できます。例:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

参照:

- 式マクロのサポートされている場所のリストについては[サポートされているマクロを参照](#)
- **例** イベント名に式マクロを使用する

12 ユーザーとユーザーグループ

概要

Zabbix のすべてのユーザは、Web ベースのフロントエンドから Zabbix アプリケーションにアクセスします。各ユーザーには
固有のログイン名とパスワードが割り当てられます。

全てのユーザパスワードは暗号化され、Zabbix のデータベースに保存されます。ユーザーは、UNIX に応じた設定をしていない限り、
ユーザー ID とパスワードを使用して UNIX サーバに直接ログインすることはできません。Web サーバとユーザブラウザ間の通信は SSL を使用して
保護することができます。

柔軟な **user permission schema**により、以下の権限を制限し差別化することができます:

- Zabbix フロントエンドの管理用機能へのアクセス
- フロントエンドでの特定のアクションの実行
- ホストグループ内の監視対象ホストへのアクセス
- 特定の API メソッドの使用

1 ユーザの設定

概要

Zabbix の初期インストールでは、以下の 2 つのユーザが事前に設定されています:

- Admin - 完全なパーミッションを持つ Zabbix **superuser**です。
- guest* - 特別な Zabbix **user**です。'guest' ユーザはデフォルトでは無効になっています。
Guests ユーザーグループに追加すると、ログインしていなくても Zabbix の監視ページにアクセスすることができます。
デフォルトでは、guest は Zabbix オブジェクトに対するパーミッションを持っていないことに注意してください。

新しいユーザーを設定するには:

- Administration → Users に行きます。
- Create user をクリックします。(または、既に登録されているユーザの名前をクリックします。)
- フォームのユーザー属性を編集します。

一般的な属性

User タブには、一般的なユーザー属性が表示されます:

User Media 1 Permissions

* Username

Name

Last name

* Groups
type here to search

Password

Language ▼

Time zone ▼

Theme ▼

Auto-login

Auto-logout

* Refresh

* Rows per page

URL (after login)

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
Username	Unique username, used as the login name.
Name	User first name (optional). If not empty, visible in acknowledgment information and notification recipient information.
Last name	User last name (optional). If not empty, visible in acknowledgment information and notification recipient information.
Groups	Select user groups the user belongs to. Starting with Zabbix 3.4.3 this field is auto-complete so starting to type the name of a user group will offer a dropdown of matching groups. Scroll down to select. Alternatively, click on Select to add groups. Click on 'x' to remove the selected. Adherence to user groups determines what host groups and hosts the user will have access to .
Password	Two fields for entering the user password. With an existing password, contains a Password button, clicking on which opens the password fields.
Language	Note that passwords longer than 72 characters will be truncated. Language of the Zabbix frontend.
Time zone	The php gettext extension is required for the translations to work. Select the time zone to override global time zone on user level or select System default to use global time zone settings.

パラメータ	説明
Theme	Defines how the frontend looks like: System default - use default system settings Blue - standard blue theme Dark - alternative dark theme High-contrast light - light theme with high contrast High-contrast dark - dark theme with high contrast
Auto-login	Mark this checkbox to make Zabbix remember the user and log the user in automatically for 30 days. Browser cookies are used for this.
Auto-logout	With this checkbox marked the user will be logged out automatically, after the set amount of seconds (minimum 90 seconds, maximum 1 day). Time suffixes are supported, e.g. 90s, 5m, 2h, 1d. Note that this option will not work: * If the "Show warning if Zabbix server is down" global configuration option is enabled and Zabbix frontend is kept open; * When Monitoring menu pages perform background information refreshes; * If logging in with the Remember me for 30 days option checked.
Refresh	Set the refresh rate used for graphs, plain text data, etc. Can be set to 0 to disable.
Rows per page	You can determine how many rows per page will be displayed in lists.
URL (after login)	You can make Zabbix transfer the user to a specific URL after successful login, for example, to Problems page.

ユーザーメディア

Media タブには、ユーザーに対して定義されたすべてのメディアのリストが含まれています。メディアは、通知を送信するために使用されます。メディアを割り当てるには、Add をクリックします。

ユーザーメディアの設定については、**Media types** の項を参照してください。

アクセス権

パーミッション * タブには、以下の情報が含まれています:

- ユーザーロール。ユーザーは自分のロールを変更することはできません。
- ロール設定で定義されたユーザータイプ (User、Admin、Super Admin)
- そのユーザーがアクセスできるホストグループ。タイプ 'User' および 'Admin' のユーザーは、デフォルトではどのホストグループおよびホストへのアクセス権も持っていません。アクセス権を得るには、それぞれのホストグループおよびホストへのアクセス権を持つユーザグループに含まれる必要があります。
- Zabbix フロントエンド、モジュール、API メソッドのセクションとエレメントへのアクセス権。アクセスが許可されているエレメントは緑色で表示されます。薄い灰色は、その要素へのアクセスが拒否されていることを意味します。
- 特定のアクションを実行する権利。許可されているアクションは、緑色で表示されます。薄い灰色は、ユーザーがそのアクションを実行する権利を持っていないことを意味します。

詳しくは、**User permissions** のページをご覧ください。

2 ユーザー権限

概要

Zabbix では、それぞれのユーザーロールを定義することで、ユーザーパーミッションを区別することができます。非特権ユーザーはホストグループのデータにアクセスできるユーザグループに含まれる必要があります。

ユーザーロール

ユーザーロールは、ユーザがどの UI 部分、どの API メソッド、どのアクションを利用できるかを定義します。Zabbix には以下のロールがあらかじめ定義されています:

ユーザタイプ	説明
Guest role	ユーザーは、監視データ、インベントリ、およびレポートメニューセクションにアクセスできますが、アクションを実行する権限はありません。
User role	ユーザーは、監視データ、インベントリ、およびレポートメニューセクションにアクセスできます。デフォルトでは、ユーザーはどのリソースにもアクセスできません。ホストグループへアクセスする権限は、明示的に付与する必要があります。
Admin role	ユーザーは、監視データ、インベントリ、レポート、および設定メニューセクションにアクセスできます。デフォルトでは、ユーザーはどのホストグループにもアクセスできません。ホストグループへアクセスする権限は、明示的に付与する必要があります。
Super Admin role	ユーザーは、すべてのメニューセクションにアクセスできます。ユーザーには、すべてのホストグループへの読み取り/書き込みアクセス権があります。特定のホストグループへのアクセスを拒否して権限を取り消すことはできません。

ユーザーロール は 管理 → ユーザーの役割セクションで設定します。Super Admins は、事前に定義されたロールを修正または削除し、権限のカスタムセットを持つより多くのロールを作成することができます。

ユーザーにロールを割り当てるには、ユーザー設定フォームの権限タブで、ユーザーの役割フィールドから設定したいロールを選択します。ロールが選択されると、関連する権限のリストが以下に表示されます。

User Media **Permissions**

* Role

User type

Permissions	Host group	Permissions
	All groups	None

Access to UI elements

Monitoring

Inventory

Reports

Configuration

Access to modules
No enabled modules found.

Access to API

Access to actions

ホストグループへのパーミッション

Zabbix のあらゆるホストデータへのアクセスは、ホストグループレベルで **user groups** にのみ許可されます。

つまり、個々のユーザが直接ホスト（またはホストグループ）へのアクセスを許可されることはありません。ホストを含むホストグループへの
 アクセスを許可されているユーザーグループの一部であることによつてのみ、ホストへのアクセスを許可されます。

3 ユーザーグループ

概要

ユーザーグループは、組織的な目的とデータへのアクセス許可を割り当てるために、ユーザーをグループ化することができます。

 ホストグループのデータを監視するための権限は、個々のユーザーではなく、ユーザーグループに割り当てられます。

あるユーザーグループにはどのような情報を、別のユーザーグループにはどのような情報を提供するかを分けることは、しばしば意味を
 持つ場合があります。これは、ユーザーをグループ化し、ホスト・グループに様々な権限を割り当てることで実現できます。

ユーザーはいくつでもグループに所属することができます。

設定

ユーザーグループを設定するには:

- Administration → User groups に移動します。
- Create user group をクリックします。(既存のグループを編集する場合はグループ名をクリックします)
- フォームでグループ属性を編集します。

User group タブには、グループの一般的な属性が含まれています:

The screenshot shows the 'User group' configuration interface. It has three tabs: 'User group', 'Permissions', and 'Tag filter'. The 'User group' tab is selected. The form contains the following fields and controls:

- Group name:** A text input field containing 'Security specialists'. A red asterisk (*) is placed to the left of the label, indicating it is a required field.
- Users:** A multi-select dropdown menu showing 'Admin (Zabbix Administrator)' and 'user (New User)'. Below the dropdown is a search prompt 'type here to search'.
- Frontend access:** A dropdown menu set to 'System default'.
- Enabled:** A checked checkbox.
- Debug mode:** An unchecked checkbox.
- Buttons:** 'Add' and 'Cancel' buttons at the bottom.

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
Group name	Unique group name.
Users	To add users to the group start typing the name of an existing user. When the dropdown with matching user names appears, scroll down to select. Alternatively you may click the Select button to select users in a popup.
Frontend access	How the users of the group are authenticated. System default - use default authentication method (set globally) Internal - use Zabbix internal authentication (even if LDAP authentication is used globally). Ignored if HTTP authentication is the global default. LDAP - use LDAP authentication (even if internal authentication is used globally). Ignored if HTTP authentication is the global default.
Enabled	Disabled - access to Zabbix frontend is forbidden for this group Status of user group and group members. Checked - user group and users are enabled Unchecked - user group and users are disabled
Debug mode	Mark this checkbox to activate debug mode for the users.

Permissions タブでは、ホストグループ (およびホストデータ) へのユーザーグループアクセスを指定できます。
グループ (およびホスト) のデータへのアクセスを指定します:

User group Permissions ● Tag filter

Permissions	Host group	Permissions
	All groups	None
	Discovered hosts	Read-write Read Deny None
	Hypervisors	Read-write Read Deny None
	Linux servers	Read-write Read Deny None
	Templates (including subgroups)	Read-write Read Deny None
	Templates/Server hardware	Read-write Read Deny None
	Templates/Virtualization	Read-write Read Deny None

type here to search Select Read-write Read Deny None

Include subgroups

[Add](#)

ホストグループに対する現在のパーミッションは、Permissions ブロックに表示されます。

ホストグループの現在のパーミッションがネストしたすべてのホストグループに継承される場合、ホストグループ名の後の括弧内の
 including subgroups テキストで示されます。

ホストグループへのアクセスレベルは変更することができます。

- **Read-write** - ホストグループへの読み書き可能なアクセス。
- **Read** - ホストグループへの読み取り専用アクセス。
- **Deny** - ホストグループへのアクセスは拒否されます。
- **None** - 権限は設定されません。

以下の選択フィールドを使用して、ホストグループとそのアクセスレベルを選択します。(None を選択すると、そのグループが
すでにリストに含まれている場合、ホストグループがリストから削除されることに注意してください) もし、ネストしたホストグループも
含めたい場合は、Include subgroups チェックボックスをマークします。このフィールドはオートコンプリートで、ホストグループの
名前を入力し始めると、一致するグループのドロップダウンが表示されます。すべてのホストグループを表示したい場合は、Select を
クリックします。

ホストグループ設定の Super Admin ユーザーは、ネストしたホストグループに
に対して親ホストグループと同じレベルの権限を強制することが可能であることに注意してください。

Tag filter タブでは、タグ名とその値でフィルタリングされた問題を見るために、ユーザーグループにタグベースのパーミッションを
設定することができます:

User group Permissions Tag filter ●

Permissions	Host group	Tags	Action
	Templates/Databases	Service: MySQL	Remove

type here to search Select tag

Include subgroups

[Add](#)

タグフィルターを適用するホストグループを選択するには、Select をクリックして既存のホストグループの完全なリストを取得するか、
一致するグループのドロップダウンを取得するためにホストグループの名前を入力し始めます。もし、ネストしたホストグ

ループに
 タグフィルタをネストしたホストグループに適用する場合は、Include subgroups チェックボックスをオンにします。

タグフィルターによって、ホストグループへのアクセスと問題を見る可能性を分離することができます。

例えば、データベース管理者が“MySQL” データベースの問題のみを参照する必要がある場合、まずデータベース管理者用のユーザーグループを
 作成し、“Service” タグ名と“MySQL” 値を指定する必要があります。

Templates/Databases X

Select
Service
MySQL

“Service” タグ名が指定され、値フィールドが空白のままである場合、対応するユーザーグループには、タグ名が“Service” である選択された
 ホストグループに関するすべての問題が表示されます。タグ名と値の両方のフィールドが空白のまま、ホストグループが選択されて
 いる場合、対応するユーザーグループには、選択されたホストグループのすべての問題が表示されます。
 タグ名とタグ値が正しく指定されていることを確認してください。そうでなければ、対応するユーザーグループには問題が表示されません。

あるユーザーが複数のユーザーグループのメンバーとして選択されている場合の例を見てみましょう。
 この場合のフィルタリングは、タグに OR 条件を使用します。

User group A			User group B			Visible result for a user (member) of both groups
Host group	Tag name	Tag value	Host group	Tag name	Tag value	
Templates/Databases	Service	MySQL	Templates/Databases	Service	Oracle	Service: MySQL or Oracle problems visible
Templates/Databases	blank	blank	Templates/Databases	Service	Oracle	All problems visible
not selected	blank	blank	Templates/Databases	Service	Oracle	Service:Oracle problems visible

Attention:

フィルター（例えば、特定のホストグループ” テンプレート/データベース” のすべてのタグ）を追加すると、他のホストグループの問題が
 表示されなくなります。

複数のユーザーグループからのホストアクセス

ユーザーは、任意の数のユーザーグループに所属することができます。これらのグループは、ホストに対する異なるアクセス権限を
 持つことができます。

したがって、非特権ユーザーが結果的にどのホストにアクセスできるようになるかを知ることは重要です。例えば、ユーザーグループ
 A と B に属するユーザーにとって、ホスト X (ホストグループ 1) へのアクセスが様々な状況でどのように影響されるかを考えてみましょう。

- グループ A が Hostgroup 1 への Read アクセスしか持っておらず、グループ B が Hostgroup 1 への Read-Write アクセスを持っている場合、ユーザーは‘X’ への **Read-Write** アクセスを取得することになります。

Attention:

Zabbix2.2 以降、“Read-write” パーミッションは“Read” パーミッションより優先されます。

- 上記と同じシナリオで、‘X’ が同時にグループ A または B に **denied** されているホストグループ 2 にもいる場合、ホストグループ 1 への **Read-write** アクセス権にもかかわらず、‘X’ へのアクセスは **unavailable** となります。
- グループ A に何の権限も定義されておらず、グループ B に Hostgroup 1 への Read-write アクセスがある場合、そのユーザーは‘X’ への **Read-write** アクセスを得ることができます。
- グループ A が Hostgroup 1 への Deny アクセス権を持っており、グループ B が Hostgroup 1 への Read-write アクセス権を持っている場合、そのユーザーは‘X’ への **Denied** アクセス権を得ます。

その他の詳細

- ホストへの Read-write アクセス権を持つ Admin レベルのユーザーは、Templates グループへのアクセス権を持っていない場合、テンプレートのリンク/アンリンクを解除することはできません。Templates グループへの Read アクセス権を持つユーザーは、ホストへのテンプレートのリンク/アンリンクを解除できますが、テンプレートリスト内のテンプレートは表示されず、他の場所のテンプレートで操作することはできません。
- ホストへの Read アクセス権を持つ Admin レベルのユーザーは、設定セクションのホストリストでホストを見ることはできませんが、IT サービス設定でホストトリガーにアクセスできます。
- Super Admin 以外のユーザー ('guest' を含む) は、マップが空であるかイメージのみである限り、ネットワークマップを見ることができます。ホスト、ホストグループ、トリガーがマップに追加されると、パーミッションが尊重されます。
- Zabbix server は、アクション操作の受信者として定義されたユーザーに対して、当該ホストへのアクセスが明示的に "denied" されている場合、通知を送信しないようになりました。

13 secrets の保存

概要

HashiCorp Vault KV Secrets Engine - Version 2 では、いくつかの機密情報を秘密裏に保存することが可能です。
 シークレットは、以下の目的で保存することができます：

- ユーザーマクロの値
- データベースアクセスクレデンシャル

Zabbix は Vault 内のシークレットを他の人が管理していると仮定して、読み取り専用アクセスを提供します。

ユーザーマクロ値

ユーザーマクロの値を Vault に秘密裏に保存することが可能です。

"Vault secret" というユーザーマクロの値には参照パス ('path:key', 例えば "secret/zabbix:password") が含まれます。

以下のコマンドを使用すると、例で述べたパスの値を設定することができます。

```
# Enable "secret/" mount point if not already enabled, note that "kv-v2" must be used
$ vault secrets enable -path=secret/ kv-v2
```

```
# Put new secret with key password under mount point "secret/" and path "secret/zabbix"
$ vault kv put secret/zabbix password=<password>
```

```
# Test that secret is successfully added
$ vault kv get secret/zabbix
```

```
# Finally test with Curl, note that "data" need to be manually added after mount point and "/v1" before the path
$ curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix
```

秘密値は設定データを更新するたびに Zabbix server によって取得され、設定キャッシュに保存されます。
 参照パスに読み取り専用でアクセスするための認証トークンをサーバ設定 ('VaultToken' パラメータ) で指定する必要があります。
 マクロの値が正常に取得できない場合、その値を使用する対応する項目はサポートされなくなります。

また、Vault から秘密の値を更新することも可能です。
 'secrets_reload' コマンドライン option を使用することで、Vault のシークレット値を更新することも可能です。

Zabbix proxy は Vault と通信を行い、データベース認証情報以外のシークレットを取得することはありません。
 Zabbix proxy のシークレット値は、設定変更のたびに Zabbix server から取得され、Zabbix server と同じように
 設定キャッシュに保存されます。

つまり、Zabbix proxy は再起動後、Zabbix server から初めて設定データの更新を受け取るまでデータ収集を開始することが
 できません。Zabbix server と proxy 間の暗号化を有効にする必要があります、有効にしない場合、サーバ警告メッセージが
 記録されます。

データベースクレデンシャル

Zabbix server、proxy、フロントエンドで使用するデータベースのクレデンシャルを Vault に密かに保存することが
 サポートされています：

- データベース認証情報を取得するための Vault 関連パラメータは、フロントエンドの **installation wizard** で任意に入力することができます。

Vault から取得したデータベース証明書は、フロントエンドでキャッシュされます。ファイルシステムの一時ファイル
 ディレクトリが、フロントエンドでのデータベースクレデンシャルのキャッシュに使われることに注意してください。
 ZBX_DATA_CACHE_TTL **constant** を使用して、データキャッシュの更新/無効化の
 頻度を制御することができます。

- server / proxy の場合、VaultDBPath 設定パラメータを使用して、キー 'password' と 'username' によるデータベースの認証情報を取得するパスを指定できます (例: secret/zabbix/database)。

例で述べたパスの値を設定するために、次のコマンドを使用することができます:

```
# Enable "secret/" mount point if not already enabled, note that "kv-v2" must be used
$ vault secrets enable -path=secret/ kv-v2

# Put new secrets with keys username and password under mount point "secret/" and path "secret/zabbix/data"
$ vault kv put secret/zabbix/database username=zabbix password=<password>

# Test that secret is successfully added
$ vault kv get secret/zabbix/database

# Finally test with Curl, note that "data" need to be manually added after mount point and "/v1" before the path
$ curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

設定パラメータ

Zabbix server / proxy に、Vault 認証とデータベース認証情報の取得のための新しい設定パラメータが追加されました。:

- VaultToken - Vault 認証トークン (詳細は Zabbix server / proxy 設定ファイルを参照してください)
- VaultURL - Vault サーバの HTTP[S] URL
- VaultDBPath - データベースの認証情報を取得するための Vault のパス (例: secret/zabbix/database)

Zabbix server と Zabbix proxy は、起動時に zabbix_server.conf と zabbix_proxy.conf から Vault 関連の設定パラメータを
 読み取ります。

Zabbix server と Zabbix proxy は、起動時に VAULT_TOKEN 環境変数を読み込み、フォークされたスクリプトで
 使用できないように設定を解除します。

Note:

フォワードスラッシュとコロンは予約済みシンボルです。フォワードスラッシュは、マウントポイントとパスの区切りにのみ
 使用できます (例: secret/zabbix はマウントポイントが secret、zabbix がパス)
 Vault マクロの場合、パスとキーの区切りにはコロンのみを使用できます。マウントポイントをフォワードスラッシュで
 区切った名前で作成する必要がある場合 (例: マウントポイントが "foo/bar"、パスが "zabbix" の場合は "foo%2Fbar/zabbix")、マウントポイント名またはパスにコロンが必要な場合は "/" ":" を URL エンコードすることが可能です。

TLS の設定

証明機関 (CA) によって署名された証明書は、デフォルトの CA ストアに追加する必要があります。または、
 SSLCALocation 構成パラメータを使用して、カスタム CA ストアの場所を指定することもできます。
 この場合、証明書ディレクトリは openssl c_rehash ユーティリティを使用して準備する必要があります。
 例えば、SSLCALocation を設定し、そのディレクトリ内に "ca.pem" をコピーして、次のコマンドを実行します。


```
$ c_rehash .
```

14 スケジュールレポート

概要

このセクションでは、スケジュールレポートの設定について説明します。

前提条件:

- スケジュールレポートを作成するためには、Zabbix Web サービスがインストールされ、正しく設定されている必要があります。Setting up scheduled reports に手順が記載されています。
- ユーザーは、Admin または Super admin タイプの user role を持ち、以下の権限を持つ必要があります。:
 - *Access to UI elements* ブロックの *Scheduled reports*(レポートを表示する)
 - *Access to actions* ブロックの *Manage scheduled reports*(レポートの作成/編集) をクリックします。

Attention:

現在、スケジュールレポートのサポートは実験的なものです。

Zabbix フロントエンドでスケジュールレポートを作成するには、以下の手順で行います:

- Reports → Scheduled reports にアクセスします。
- 画面右上の Create report をクリックします。
- レポートのパラメータをフォームに入力します。

また、既存のレポートを開いて Clone ボタンを押し、別の名前で保存することでレポートを作成することも可能です。
設定

スケジュールレポートタブには、一般的なレポート属性が含まれています。

*** Owner**

*** Name**

*** Dashboard**

Period Previous day Previous week Previous month Previous year

Cycle Daily Weekly Monthly Yearly

Start time :

Start date

End date

Subject

Message

Select

Select

*** Subscriptions**

Recipient	Generate report by	Status	Action
Admin (Zabbix Administra...	Admin (Zabbix Administra...	Include	Remove
Add user Add user group			

Description

Enabled

Add
Test
Cancel

必須入力項目には、赤いアスタリスクが表示されています。






パラメータ	説明
Owner	User that creates a report. Super admin level users are allowed to change the owner. For Admin level users, this field is read-only.
Name	Name of the report; must be unique.

パラメータ	説明
Dashboard	Dashboard on which the report is based; only one dashboard can be selected at a time. To select a dashboard, start typing the name - a list of matching dashboards will appear; scroll down to select. Alternatively, you may click on Select next to the field and select a dashboard from the list in a popup window. If a dashboard contains multiple pages, only the first page will be sent as a report.
Period	Period for which the report will be prepared. Select one of the available options: Previous day, Previous week, Previous month, Previous year.
Cycle	Report generation frequency. The reports can be sent daily, weekly, monthly, or yearly. Weekly mode allows to select days of the week when the report will be sent.
Start time	Time of the day in the format hh:mm when the report will be prepared.
Repeat on	Days of the week when the report will be sent. This field is available only if Cycle is set to weekly.
Start date	The date when regular report generation should be started
End date	The date when regular report generation should be stopped.
Subject	Subject of the report email. Supports {TIME} macro.
Message	Body of the report email. Supports {TIME} macro.
Subscriptions	List of report recipients. By default, includes only the report owner. Any Zabbix user with configured email media may be specified as a report recipient. Press Add user or Add user group to add more recipients. Press on the username to edit settings: Generate report by - whether the report should be generated on behalf of the report owner or the recipient. Status - select Include to send the report to user or Exclude to prevent sending the report to this user. At least one user must have Include status. Exclude status can be used to exclude specific users from a user group that is included. Note that users with insufficient permissions***** will see Inaccessible user or Inaccessible user group instead of the actual names in the fields Recipient and Generate report by; the fields Status and Action will be displayed as read-only.
Enabled	Report status. Clearing this checkbox will disable the report.
Description	An optional description of the report. This description is for internal use and will not be sent to report recipients.

* 権限不足のユーザーとは、Admin ユーザータイプに基づくロールを持ち、受信者またはレポート所有者が所属する
 ユーザーグループのメンバーでないユーザーのことを指します。

フォームのボタン

フォームの下部にあるボタンで、いくつかの操作を行うことができます。

	Add a report. This button is only available for new reports.
	Update the properties of a report.
	Create another report based on the properties of the current report.
	Test if report configuration is correct by sending a report to the current user.
	Delete the report.

Cancel

Cancel the editing of report properties.

テスト

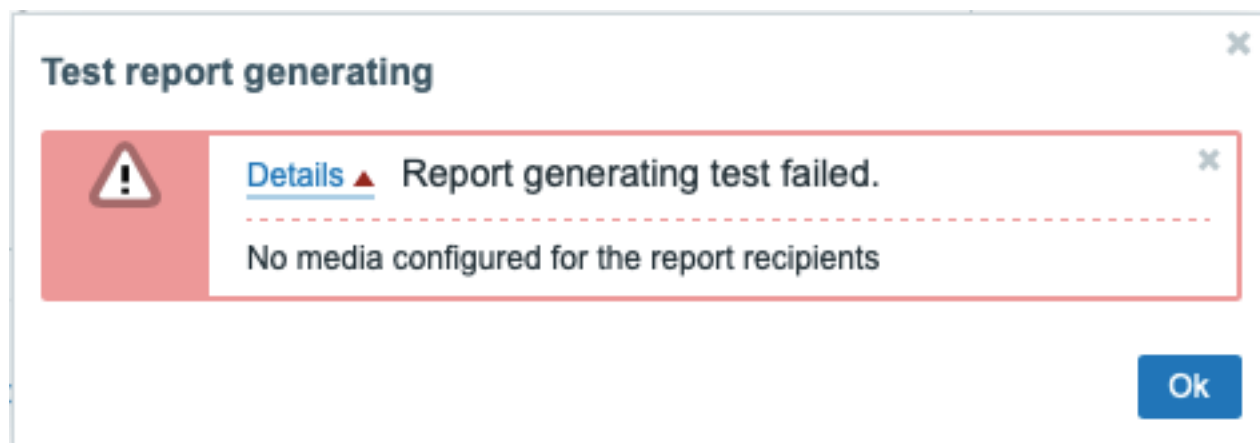
レポートをテストするには、レポート設定フォームの下部にある Test ボタンをクリックします。

Note:

ダッシュボード **action menu** からレポート設定フォームが開かれている場合、テストボタンは使用できません。

設定が正しければ、テストレポートは現在のユーザーに直ちに送信されます。テストレポートでは、購読者と 'generated by'
 ユーザーの設定は無視されます。

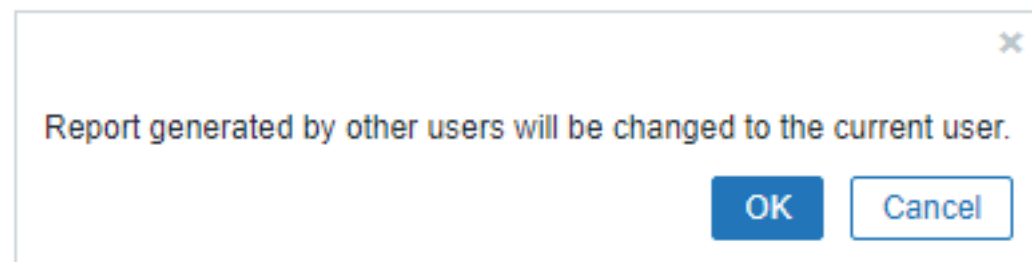
設定が正しくない場合、考えられる原因を説明するエラーメッセージが表示されます。



レポートの更新

既存のレポートを更新するには、レポート名を押して、必要な設定変更を行い、更新ボタンを押します。

既存のレポートを他のユーザーが更新し、そのユーザーがダッシュボードを変更した場合、更新ボタンを押すと、
 「他のユーザーによって生成されたレポートは、現在のユーザーに変更されます」という警告メッセージが表示されます。



このステップで OK を押すと、以下のように変更されます：

- Generated by 設定は、レポートを最後に編集したユーザーを表示するように更新されます（Generated by が Recipient に設定されている場合を除く）。
- アクセスできないユーザーまたはアクセスできないユーザーグループとして表示されていたユーザーは、レポートサブスクライバーのリストから削除されます。

キャンセルを押すと、ポップアップウィンドウが閉じられ、レポートの更新がキャンセルされます。

レポートのクローン

既存のレポートを複製するには、既存のレポート設定フォームの下にある Clone ボタンをクリックします。
 他のユーザーによって作成されたレポートを複製する場合、現在のユーザーは新しいレポートの所有者になります。

レポート設定は、ユーザー権限に従って新しいレポート設定フォームにコピーされます。

- レポートのクローンを作成するユーザーにダッシュボードに対するパーミッションがない場合、Dashboard フィールドはクリアされます。
- レポートのクローンを作成したユーザーが、Subscriptions リストの一部のユーザーまたはユーザーグループに対する権限を持っていない場合、アクセスできない受信者はクローン作成されません。

- Generated by の設定は、現在のユーザーを表示するように更新されます（Generated by が受信者に設定されている場合を除く）

必要な設定とレポート名を変更し、Add を押します。

8. サービス監視

概要 サービス監視は、IT インフラストラクチャサービスのツリー全体の概要を取得し、インフラストラクチャの弱点を特定し、さまざまな IT サービスの SLA を計算し、その他の情報をより高いレベルで確認するために使用できるビジネスレベルの監視です。サービスの監視は、ディスク容量の不足、プロセッサの高負荷などの低レベルの詳細ではなく、サービスの全体的な可用性に焦点を当てています。Zabbix 6.0 以降、サービス監視は、サービスが期待どおりに実行されていない場合に問題の根本原因を見つける機能も提供します。

サービス監視により、監視対象データの階層表現を作成できます。

非常に単純なサービスの構造は次のようになります。

```
サービス
|
|--ワークステーション
| |
| |--ワークステーション1
| |
| |--ワークステーション2
|
|--サーバー
```

構造の各ノードには属性ステータスがあります。ステータスは、選択したアルゴリズムに従って計算され、上位レベルに伝達されます。個々のノードのステータスは、マップされた問題のステータスの影響を受けます。問題のマッピングは、**タグ付け**によって実現されます。

Zabbix は、サービスステータスの変化が検出された場合に、通知を送信したり、Zabbix サーバーでスクリプトを自動的に実行したりできます。子サービスのステータスに基づいて、親サービスが「障害状態」になるかどうかの柔軟なルールを定義することができます。その後、サービスの問題データを使用して SLA を計算し、柔軟な一連の条件に基づいて SLA レポートを送信できます。

サービス監視は、次のセクションで説明されるサービスのメニューで設定されます：

- **サービス**

サービスのセクションでは、親サービスを追加し、次に子サービスを親サービスに追加して、監視対象のインフラストラクチャの階層を構築できます。

このセクションでは、サービスツリーの構成に加えて、インフラストラクチャ全体の概要を提供し、サービスステータスの変更につながった問題をすばやく特定できます。

- **サービスのアクション**

このセクションでは、サービスのアクションを設定できます。サービスのアクションはオプションであり、以下を行うことができます：

- サービスがダウンしているという通知を送信します。
- サービスステータスの変更時に Zabbix サーバーでリモートコマンドを実行します。
- サービスが再開したときに回復通知を送信します。

- **SLA**

このセクションでは、サービスレベルアグリーメントを定義し、特定のサービスのサービスレベル目標を設定できます。

- **SLA レポート**

このセクションでは、SLA レポートを表示できます。

参考：

-- SLA 監視設定 [サンプル](#) -- Zabbix 6.0 以前のバージョンから **サービスをアップグレード** する際の注意事項

1 サービスツリー

サービスツリーは、Services->Services メニューセクションで設定できます。右上隅で、viewモードから Edit モードに切り替えます。

Services ? Create service View Edit

Name	Status	Root cause	Created at	Tags	
<input type="checkbox"/> Load balancer 5	OK		2000-01-01	SLA-1	+ ↙ ×
<input type="checkbox"/> Video surveillance 2	Warning	Hikvision camera: Error receiving data	2000-01-01	SLA-2	+ ↙ ×

新しいサービスを**設定**するには、右上隅にある Create service ボタンをクリックします。

子サービスをすばやく追加するには、親サービスの横にあるプラスアイコンを押すこともできます。これにより親サービスパラメーターが事前に入力されたサービス構成フォームを開くことができます。

サービスの設定 **Service** タブで必要なサービスパラメータを設定します。

Service ×

Service **Tags 2** Child services

* Name

Parent services
type here to search

Name	Operation	Value	Action
<input type="text" value="Type"/>	<input type="text" value="Equals"/> ▾	<input type="text" value="Connection"/>	Remove

[Add](#)

* Sort order (0->999)

Status calculation rule ⓘ ▾

Description

Created at

Advanced configuration

すべての必須入力フィールドは、赤いアスタリスクでマークされています。

パラメーター	説明
名前	サービス名
親サービス	サービスが属する親サービス。 一番レベルの高いサービスを追加する場合は、このフィールドを空のままにします。 1つのサービスに複数の親サービスがある場合があります。その場合は各親サービスのサービスツリーに表示されます。
障害タグ	問題をサービスにマップするタグを指定します： 等しい - 指定されたタグ名が値と一致するとき対象とします（大文字と小文字を区別します） 含む - 指定されたタグ名が値に入力された文字列が含まれる時に対象とします（部分文字列の一致、大文字と小文字は区別されません） タグ名の一致では常に大文字と小文字が区別されます。
ソート順	表示の並べ替え順序。最も低いものが最初になります。

パラメーター	説明
ステータス計算ルール	サービスステータスの計算ルール すべての子が障害であったときにもっとも重要 (Zabbix バージョン 6.0 以前の "Problem, if all children have problems" と同じ) -すべての子に問題がある場合、子サービスの最も深刻な障害の色でサービスステータスが表示されます 子サービスで最も重要 (Zabbix バージョン 6.0 以前の "Problem, if one child has a problem" と同じ) -子ノードの最も深刻な障害の色でサービスステータスが表示されます ステータスを正常に設定-サービスステータスを計算しません 下の高度な設定チェックボックスをオンにして、追加のステータス計算ルールを設定します。
説明 高度な設定	サービスの説明 高度な設定 オプションにアクセスするには、チェックボックスをオンにします。

詳細設定

Advanced configuration

Additional rules

Name	Action
Average - If at least 4 child services have Average status or above	Edit Remove
Disaster - If at least 3 child services have High status or above	Edit Remove
Add	

Status propagation rule: As is

Weight: 0

パラメータ	説明
Additional rules	追加のステータス計算ルールを定義するには Add をクリックします。
Set status to	条件が一致する場合は、サービスステータスを OK (default)、未分類、情報、警告、軽度の障害、重度の障害、致命的な障害のいずれかに設定します。
Condition	直接の子サービスの条件を選択します。 少なくとも (N) 個の子サービスが (Status) ステータス以上の場合 ** 少なくとも (N%) の子サービスが (Status) ステータス以上の場合 (N) 個未満の子サービスが (Status) ステータス以下の場合 (N) ステータス以下の子サービスが (N%) 未満の場合 (N) ステータス以上の子サービスの重みが (W) 以上の場合 (N) ステータス以上の子サービスの重みが (N%) 以上の場合 ** (Status) ステータス以下の子サービスの重みが (W) 未満の場合 (Status) ステータス以下の子サービスの重みが (N%) 未満の場合
N (W)	複数の条件が指定され、状況が複数の条件に一致する場合、最も高い重大度が設定されます。
Status	条件で N または W (1-100000)、または N% (1-100) の値を設定します。 条件となるステータスの値を選択します：OK (default)、未分類、情報、警告、軽度の障害、重度の障害、致命的な障害
Status propagation rule	サービスステータスを親サービスに伝達するためのルール: 現状のまま - ステータスは変更されずに伝達されます。 増加 - 伝播されたステータスの深刻度を 1 ~ 5 増加させて伝達されます 減少 - 伝播されたステータスの深刻度を 1 ~ 5 現象させて伝達されます このサービスを無視 - ステータスは親サービスにまったく伝達されません 固定ステータス - ステータスは静的に伝達されます。つまり、常に同じステータスです。
Weight	サービスの重み (0 (デフォルト) から 1000000 の範囲の整数)

Note:

追加のステータス計算ルールは、メインの Status calculation rule パラメーターに従って計算されたレベルよりも重大度レベルを上げるためのみ使用できます。Additional rules では警告だが、Status calculation rule では致命的な障害である場合、サービスのステータスは致命的な障害になります。

Tags タブには **service-level tags** が含まれています。サービスレベルのタグは、サービスを識別するために使用されます。このタイプのタグは問題をサービスにマップするためには使用されません。(そのためには、最初のタブから **Problem tags(#problem-tags)** を使用します)

Child services タブでは、依存サービスを指定できます。Add をクリックして、既存のサービスのリストからサービスを追加します。新しい子サービスを追加する場合は、最初にこのサービスを保存してから、作成したサービスの横にあるプラスアイコンをクリックします。

タグ サービスには 2 つの異なるタイプのタグがあります。

- サービスタグ
- 問題タグ

サービスタグ

サービスタグは、サービスを **service actions** および **SLA** と照合するために使用されます。これらのタグは [Tabs] サービス設定タブで指定されます。SLA のマッピングには、OR ロジックが使用されます。サービスに一致するタグが 1 つでもあればサービスは SLA にマッピングされます。マッピングルールはサービスアクションで設定可能であり、AND、* OR、または AND /OR* ロジックのいずれかを使用できます。

Service **Tags 1** Child services

Name	Value
internal	monitoring
tag	value

Add

問題タグ

問題タグは、問題とサービスを照合するために使用されます。これらのタグは、プライマリサービス設定タブで指定されます。

Problem tags

Name	Operation	Value	Action
Database	Equals	MySQL	Remove
Type	Contains	Server	Remove

Add

Zabbix は問題タグをチェックして、いずれかのサービスの問題タグ一致ルールに一致するかどうかを確認します。サービスには、いくつかの問題タグルールが定義されている場合があります。この場合、問題タグはすべての条件を満たす必要があります。

Note:

Zabbix の問題は、テンプレート、ホスト、アイテム、Web シナリオ、およびトリガーのチェーン全体からタグを継承します。これらのタグはいずれも、問題をサービスに一致させるために使用できます。

問題タグが一致すると、サービスステータスは問題と同じステータスに変わります。いくつかの問題が発生した場合、サービスは最もレベルの高いステータスになります。

最下位の階層レベルの子サービスのみが問題タグを定義し、問題に直接関連付けることができます。次に、親サービスのステータスは、ステータス計算ルールに従って子サービスのステータスに基づいて計算されます。

例：

問題 Web カメラ 3 がダウンしていますタグ `type:video surveillance,floor:1st,name:webcam 3 status Warning`

サービス **Web camera 3** には、問題タグが一つだけ指定されています：`:name:webcam 3`

Problem tags	Name	Operation	Value	Action
	<input type="text" value="name"/>	Equals <input type="button" value="v"/>	<input type="text" value="webcam 3"/>	<input type="button" value="Remove"/>
	<input type="button" value="Add"/>			

この問題が検出されると、サービスステータスが OK から Warning に変わります。

サービス **Web** カメラ **3** に問題タグ name:webcam 3 と floor:2nd が含まれている場合、条件が部分的にしか満たされていないため、問題が検出されてもステータスは変更されません。

Note:

以下で説明するボタンは、Services セクションが Edit モードの場合にのみ表示されます。

既存サービスの変更

既存のサービスを編集するには、サービスの横にある鉛筆アイコンを押します。

既存のサービスのクローンを作成するには、鉛筆アイコンを押してその構成を開き、[Clone] ボタンを押します。サービスが複製されると、その親リンクは保持されますが、子リンクは保持されません。

サービスを削除するには、サービスの横にある x アイコンを押します。親サービスを削除すると、その子サービスは削除されず、サービスツリーの 1 つ上のレベルに移動します (第 1 レベルの子は、削除された親サービスと同じレベルになります)。

サービスのリストの下にある 2 つのボタンは、いくつかの一括編集オプションを提供します。

Mass update - サービスのプロパティを一括更新

Delete - サービスを削除

これらのオプションを使用するには、サービスのチェックボックスをオンにしてから、必要なボタンをクリックします。

Note:

以下で説明するボタンは、Services セクションが編集モードの場合にのみ表示されます。

既存のサービスの変更

既存のサービスを編集するには、サービスの横にある鉛筆アイコンを押します。

既存のサービスを複製するには、鉛筆アイコンを押して設定を開き、[複製] ボタンを押します。サービスが複製されると、親リンクは保持されますが子リンクは保持されません。

サービスを削除するには、サービスの横にある x アイコンを押します。親サービスを削除しても、その子サービスは削除されず、サービスツリーで 1 レベル上に移動します (第 1 レベルの子は、削除された親サービスと同じレベルになります)。

サービスのリストの下にある 2 つのボタンには、一括編集オプションがあります。

- 一括更新 - サービスのプロパティを一括更新
- 削除 - サービスの削除

これらのオプションを使用するには、それぞれのサービスの前にあるチェックボックスをマークしてから、必要なボタンをクリックします。

2 サービスアクション

概要 このセクションでは、サービスアクションを表示および構成できます。

サービスアクションは、サービスステータスの変更 (OK⇄問題) の結果としていくつかの操作を実行する場合に役立ちます。例としては

- メッセージを送る
- Web サーバーを再起動する

サービスアクションは、Zabbix の他のアクションタイプ (トリガーアクションなど) と機能的に似ています。

設定 新しいサービスアクションを作成するには、Services メニューの Service actions サブセクションに移動し、右上隅にある Create action をクリックします。

サービスアクションは、Zabbix の他のタイプのアクションと同じ方法で設定されます。詳細については[actions](#)を参照してください。

主な違いは次のとおりです。

- サービスアクションへのユーザーアクセスは、ユーザー **role**によって付与されたサービスへのアクセス権に依存します。
- サービスアクションは、さまざまな**条件**のセットをサポートします。

条件 サービスアクションで使用できる条件は次の通りです。

条件タイプ	サポートされる演算子	説明
Service	<p>equals</p> <p>does not equal</p>	<p>含むサービスまたは除外するサービスを指定します。</p> <p>equals - イベントはこのサービスに属します。</p> <p>does not equal - イベントはこのサービスに属していません。</p> <p>親サービスを指定すると、すべての子サービスが暗黙的に選択されます。親サービスのみを指定するには、ネストされたすべての子サービスにdoes not equalを追加で設定する必要があります。</p>
Service name	<p>contains</p> <p>does not contain</p>	<p>サービス名に含まれる文字列または除外する文字列を指定します。</p> <p>contains - イベントはサービスによって生成され、名前にこの文字列が含まれます。</p> <p>does not contain - この文字列はサービス名に見つかりません。</p>
Service tag name	<p>equals</p> <p>does not equal</p> <p>contains</p> <p>does not contain</p>	<p>含まれるイベントタグまたは除外するイベントタグを指定します。サービスイベントタグは、サービス構成セクションタグで定義できます。</p> <p>equals - イベントにはこのタグがあります</p> <p>does not equal-イベントにはこのタグがありません</p> <p>contains - イベントにはこの文字列を含むタグがあります</p> <p>does not contain - イベントにはこの文字列を含むタグがありません。</p>

条件タイプ	サポートされる演算子	説明
Service tag value	equals does not equal contains does not contain	<p>含まれるイベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。サービスイベントタグは、サービス構成セクションタグで定義できます。</p> <p>equal - イベントにはこのタグと値があります</p> <p>does not equal-イベントにはこのタグと値がありません</p> <p>contains-イベントにはこれらの文字列を含むタグと値があります</p> <p>does not contain-イベントにはこれらの文字列を含むタグと値がありません。</p>

Attention:

Administration->Media types メニューで、サービスアクションのmessage templatesを必ず定義してください。未定義の場合、通知は送信されません。

3 SLA

概要 servicesが作成されると、サービスレベルアグリーメント (SLA) でパフォーマンスに問題がないかどうか監視できます。

Services->SLA メニューセクションでは、さまざまなサービスの SLA を構成できます。Zabbix の SLA は、サービスレベル目標 (SLO)、予想される稼働時間スケジュール、および計画されたダウンタイムを定義します。

SLA とサービスはサービスタグによって照合されます。同じ SLA を複数のサービスに適用できます。パフォーマンスは、一致するサービスごとに個別に測定されます。1つのサービスに複数の SLA が割り当てられている場合があります。各 SLA のデータは個別に表示されます。

SLA レポートでは、Zabbix は実際のサービスの可用性を測定するサービスレベルインジケータ (SLI) データを提供します。サービスが SLA 目標を満たしているかどうかは、SLO (期待される可用性 (%)) と SLI (実際の可用性 (%)) を比較することによって決定されます。

設定 新しい SLA を作成するには Create SLA ボタンをクリックします。

SLA タブでは、一般的な SLA パラメータを設定できます。

New SLA ✕

SLA
Excluded downtimes

* Name

* SLO %

Reporting period Daily Weekly Monthly Quarterly Annually

Time zone

Schedule 24x7 Custom

* Effective date ⋮

* Service tags

Name	Operation	Value	Action
<input style="width: 90%;" type="text" value="SLA"/>	Equals ▼	<input style="width: 80%;" type="text" value="1"/>	Remove
Add			

Description

Enabled

Add
Cancel

パラメータ	説明
Name	SLA 名を入力します。
SLO	サービスレベル目標 (SLO) をパーセンテージで入力します。
Reporting period	SLA レポートで使用される期間を設定します - daily,weekly,monthly,quarterly,annually
Time zone	SLA タイムゾーンを選択します。
Schedule	SLA スケジュールを選択します-24 時間 365 日またはカスタム。
Effective date	SLA 計算を開始する日付を選択します。
Service tags	サービスタグを追加して、この SLA を適用する必要があるサービスを識別します。 Name - サービスタグ名。完全一致。大文字と小文字が区別されます。 Operation - タグ値が完全に一致する必要がある場合は Equals (大文字と小文字を区別)、タグ値の一部が一致する必要がある場合は Contains (大文字と小文字を区別しない) を選択します。 Value - 選択した操作に従って検索するサービスタグの値。少なくとも 1 つのサービスタグが一致する場合、SLA がサービスに適用されます。
Description	SLA の説明を追加します。
Enabled	チェックボックスをオンにして、SLA 計算を有効にします。

Excluded downtimes タブでは、SLA 計算から除外されるダウンタイムを指定できます。

New SLA ✕

SLA Excluded downtimes 1

Excluded downtimes	Start time	Duration	Name	Action
	2022-02-01 02:00	3h	Maintenance	Edit Remove
Add				

Add
Cancel

[Add] をクリックして除外されたダウンタイムを構成し、期間名、開始日、および期間を入力します。

#####SLA レポート

SLA と比較したサービスのパフォーマンスは**SLA レポート**に表示されます。SLA レポートを表示するには

- SLA セクションから、SLA レポートのハイパーリンクをクリックする
- Services セクションから、info タブ SLA 名をクリックする
- ダッシュボード **widget** の SLA レポート

SLA を構成すると、Services セクションの info タブにも、サービスのパフォーマンスに関する情報が表示されます。

4 セットアップ例

概要 このセクションでは、Zabbix 高可用性クラスターをサービスとして監視するためのセットアップについて説明します。

前提条件 サービス監視を設定する前に、ホストを設定する必要があります。

- HA node 1 と少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）component : HA node 1
- HA node 2 と少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）component : HA node 2

サービスツリー 次のステップは、サービスツリーを構築することです。この例でのインフラストラクチャは非常に基本的であり、Zabbix cluster（親）と 2 つの子サービス Zabbix server node 1 および Zabbix server node 2 の 3 つのサービスで構成されています。

```
Zabbix cluster
|
|-Zabbix server node 1
|-Zabbix server node 2
```

[Services] ページで Edit モードをオンにして、[Create service] を押します。

Create service
View
Edit

サービス設定ウィンドウで、Name に Zabbix cluster を入力し、チェックボックス Advanced configuration をチェックします。

New service

? X

Service **Tags** Child services

* Name

Parent services

Name	Operation	Value	Action
<input type="text" value="tag"/>	Equals	<input type="text" value="value"/>	Remove
Add			

* Sort order (0->999)

Status calculation rule 

Description

Advanced configuration

Name	Action
Add	

Status propagation rule

Weight

追加のルールを構成します。

New additional rule

X

Set status to

Condition

N

Status

Zabbix cluster には 2 つの子サービスがあります。(HA node ごとに 1 つ) 両方の HA ノードに少なくとも Warning ステータスの問題がある場合は、親サービスのステータスを Disaster に設定する必要があります。これを実現するには、追加のルールを次のように構成する必要があります。

- Set status to: Disaster
- Condition: If at least N child services have Status status or above
- N: 2
- Status: Warning

Tags タブに切り替えて、タグ Zabbix:server を追加します。このタグは後でサービスアクションと SLA レポートに使用されます。

New service

? X

Service **Tags 1** Child services

Tags	Name	Value	Action
	Zabbix	server	Remove

[Add](#)

Add

Cancel

新しいサービスを保存します。

子サービスを追加するには、Zabbix cluster service の横にあるプラスアイコンを押します。(アイコンは Edit モードでのみ表示されます)

<input type="checkbox"/>	Name	Status	Root cause	Created at	Tags	
<input type="checkbox"/>	Zabbix cluster	OK		2022-05-10	Zabbix: server	+ ↙ ×

Displaying 1 of 1 found

サービス設定ウィンドウで、Name に Zabbix server node 1 を入力します。親サービスパラメータには、Zabbix cluster がすでに入力されていることに注意してください。

このサービスの可用性は component:HA node 1 問題タグでマークされたホスト HA node 1 で検出します。Problem tags パラメータに次のように入力します。

- Name: component
- Operation: Equals
- Value: HA node 1

New service

? X

Service **Tags** Child services

* Name

Parent services
type here to search

Problem tags	Name	Operation	Value	Action
	component	Equals	HA node 1	Remove

[Add](#)

* Sort order (0->999)

Status calculation rule ⓘ

Description

Advanced configuration

Add

Cancel

Tags タブに切り替えて、サービスタグ Zabbix server:node 1 を追加します。このタグは後でサービスアクションと SLA レポートに使用されます。

Service **Tags 1** Child services

Tags	Name	Value	Action
	Zabbix server	node 1	Remove
	Add		

Add

Cancel

新しいサービスを保存します。

“Zabbix server node 2” という名前で Zabbix cluster の別の子サービスを作成します。

問題タグを次のように設定します。

- Name: component
- Operation: Equals
- Value: HA node 2

Tags タブに切り替えて、サービスタグ Zabbix server:node 2 を追加します。

新しいサービスを保存します。

SLA この例では、半年ごとの 1 時間のメンテナンス期間を除き、期待される Zabbix クラスターのパフォーマンスは 100% です。

まず、新しい SLA を追加する必要があります。

services->SLA メニューセクションに移動し、[Create SLA] を押します。Name に Zabbix cluster performance と入力し、SLO を 100% に設定します。

Zabbix clustre サービスにはサービスタグ Zabbix:server があります。Zabbix cluster のパフォーマンスを測定するためにこの SLA を使用するには、Service tags パラメーターで次のように設定します。

- Name: Zabbix
- Operation: Equals
- Value: server

New SLA

? X

SLA Excluded downtimes

* Name

* SLO %

Reporting period Daily Weekly Monthly Quarterly Annually

Time zone ▼

Schedule 24x7 Custom

* Effective date 📅

* Service tags

Name	Operation	Value	Action
<input type="text" value="Zabbix"/>	<input type="text" value="Equals"/> ▼	<input type="text" value="server"/>	Remove
Add			

Description

実際の設定では、必要なレポート期間、タイムゾーン、開始日を更新したり、スケジュールを 24 時間 365 日からカスタムに変更したりすることもできます。ですがこの例ではデフォルト設定で十分です。

Excluded downtimes タブに切り替え、スケジュールされたメンテナンス期間のダウンタイムを追加して、これらの期間を SLA 計算から除外します。Excluded downtimes セクションで [Add] ボタンを押し、ダウンタイム名、開始時間、期間を入力します。

New SLA

? X

SLA Excluded downtimes 2

Excluded downtimes

Start time	Duration	Name	Action
2022-01-03 08:00	1h	Maintenance Jan	Edit Remove
2022-07-06 16:00	1h	Maintenance Jul	Edit Remove
Add			

[Add] を押して、新しい SLA を保存します。

[SLA reports] セクションに切り替えて、Zabbix cluster の SLA レポートを表示します。

Year	SLO	SLI	Uptime	Downtime	Error budget
2022	100%	100	36m 53s	0	0

SLA の情報は、Services セクションでも確認できます。

Zabbix cluster

Parent services:

Status: OKSLA: Zabbix cluster performance: 100 ?Tags: Zabbix: server

Name	Status	Rc
Zabbix server node 1	OK	
Zabbix server node 2	OK	

9. Web 監視

概要 Zabbix を使用して、Web サイトの稼働状況をさまざまな観点で確認できます。

Web 監視を実行するには、Zabbix サーバーを cURL (libcurl) サポートで構成する必要があります。

Web 監視を有効にするには Web シナリオを定義する必要があります。Web シナリオは 1 つまたは複数の HTTP リクエストまたは「ステップ」で構成されます。ステップは Zabbix サーバーによって事前に定義された順序で定期的に行われます。ホストがプロキシによって監視されている場合、ステップはプロキシによって実行されます。

Web シナリオは、アイテムやトリガーなどと同じ方法でホスト/テンプレートにアタッチされます。つまり、Web シナリオをテンプレートレベルで作成し、1 回の移動で複数のホストに適用することもできます。

次の情報は、すべての Web シナリオで収集されます。

- シナリオ全体のすべてのステップの 1 秒あたりの平均ダウンロード速度
- 失敗したステップの数
- 最後のエラーメッセージ

次の情報は、Web シナリオのステップで収集されます。

- 1 秒あたりのダウンロード速度
- 応答時間
- レスポンスコード

詳細については [Web 監視アイテム](#) を参照してください。

Web シナリオの実行から収集されたデータは、データベースに保持されます。データは、グラフ、トリガー、および通知に自動的に使用されます。

Zabbix は取得した HTML ページに事前定義された文字列が含まれているかどうかを確認することもできます。シミュレートされたログインを実行し、ページ上でシミュレートされたマウスクリックのパスをたどることができます。

Web 監視は HTTP と HTTPS の両方をサポートしています。Web シナリオを実行する場合、Zabbix はオプションでリダイレクトに従います。(以下のリダイレクトに従うオプションを参照) リダイレクトの最大数は 10 にハードコーディングされています。(cURL オプション `CURLOPT_MAXREDIRS` を使用) すべての Cookie は単一のシナリオの実行中に保持されます。

HTTPS プロトコルを使用した Web 監視の [既知の問題](#) も参照してください。

Web シナリオの設定 Web シナリオを設定するには：

- 設定 → ホスト (またはテンプレート) に移動します
- ホスト/テンプレートの行で Web をクリックします
- 右側の Web シナリオの作成をクリックします (またはシナリオ名をクリックして既存のシナリオを編集します)
- フォームにシナリオのパラメータを入力します

シナリオタブでは Web シナリオの一般的なパラメータを設定できます。

Scenario
Steps
Tags
Authentication

* Name

* Update interval

* Attempts

Agent

HTTP proxy

Variables

Name	⇒	Value	
<input style="width: 95%;" type="text" value="name"/>		<input style="width: 95%;" type="text" value="value"/>	Remove
Add			

Headers

Name	⇒	Value	
<input style="width: 95%;" type="text" value="name"/>		<input style="width: 95%;" type="text" value="value"/>	Remove
Add			

Enabled

すべての必須入力フィールドは、赤いアスタリスクでマークされています。

シナリオパラメータ：

パラメーター	説明
ホスト名前	シナリオが属するホスト/テンプレートの名前
監視間隔	一意のシナリオ名
試行回数	シナリオが実行される頻度。 タイムサフィックス が使用可能。例：30s, 1m, 2h, 1d。 ユーザーマクロ が使用可能。ユーザーマクロで値が変更された場合（例：5m→30s）次のチェックは前の値に従って実行されることに注意してください（将来的にはサンプル値を使用）。 Web シナリオステップ実行の試行回数。ネットワークに問題がある場合（タイムアウト、接続がないなど）、Zabbix はステップの実行を数回繰り返すことができます。フィギュアセットは、シナリオの各ステップに等しく影響します。最大 10 回試行でき、デフォルト値 1 回です。 注：応答コードが間違っているか、必要な文字列が一致していない場合、Zabbix はステップを繰り返しません。
エージェント	クライアントエージェントを選択します。 Zabbix は選択されたブラウザのふりをします。これは、Web サイトがブラウザごとに異なるコンテンツを返す場合に役立ちます。 このフィールドでは、ユーザーマクロを使用できます。

パラメーター	説明
HTTP プロキシ	<p>[protocol://] [username[:password]@]proxy.example.com[:port] の形式を使用して、使用する HTTP プロキシを指定できます。</p> <p>これにより CURLOPT_PROXY cURL オプションが設定されます。</p> <p>オプションの protocol://プレフィックスを使用して、代替プロキシプロトコルを指定できます。(プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました) プロトコルが指定されていない場合、プロキシは HTTP プロキシとして扱われます。</p> <p>デフォルトでは、1080 ポートが使用されます。</p> <p>指定されている場合、プロキシは http_proxy、HTTPS_PROXY などのプロキシ関連の環境変数を上書きします。指定しない場合、プロキシはプロキシ関連の環境変数を上書きしません。入力した値は「そのまま」渡され、健全性チェックは行われません。</p> <p>SOCKS プロキシアドレスを入力することもできます。間違ったプロトコルを指定すると、接続が失敗し、アイテムがサポートされなくなります。</p> <p>注 HTTP プロキシでは単純な認証のみがサポートされます。</p> <p>このフィールドではユーザーマクロを使用できます。</p>
変数	<p>シナリオステップで使用できる変数。(URL、投稿変数)</p> <p>次の形式になります：</p> <pre>{macro1}=value1 {macro2}=value2 {macro3}=regex:< 正規表現 ></pre> <p>例：</p> <pre>{username}=Alexei {password}=kj3h5kj34bd {hostid}=regex :hostid is ([0-9]+)</pre> <p>マクロは、ステップで {username}、{password}、および {hostid} として参照できます。Zabbix はそれらを実際の値に自動的に置き換えます。regex: を持つ変数は、正規表現の値を取得するために 1 つのステップが必要であるため、抽出された値は後のステップにのみ適用できることに注意してください。</p> <p>値の部分が regex: で始まる場合は、その後の部分は、Web ページを検索し、見つかった場合は変数に一致を格納する正規表現として扱われます。一致する値を抽出できるように、少なくとも 1 つのサブグループが存在する必要があります。</p> <p>ユーザーマクロと {HOST.*}macro がサポートされています。</p> <p>変数は、クエリフィールドまたは投稿変数のフォームデータで 사용되는場合は自動的に URL エンコードされますが、生の投稿または直接 URL で使用される場合は手動で URL エンコードする必要があります。</p>
ヘッダー	<p>リクエストの実行時に送信されるカスタム HTTP ヘッダー</p> <p>ヘッダーは、HTTP プロトコルに表示されるのと同じ構文を使用してリストする必要があります。オプションで CURLOPT_HTTPHEADER の cURL オプションでサポートされているいくつかの追加機能を使用します。</p> <p>例：</p> <pre>Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8</pre> <p>ユーザーマクロと {HOST.*}マクロがサポートされています。</p>
有効	<p>このボックスがチェックされている場合、シナリオはアクティブです。チェックされていない場合、シナリオは無効になります。</p>

既存のシナリオを編集する場合、次の 2 つの追加ボタンを使用できることに注意してください。

Clone

既存のシナリオのプロパティに基づいて別のシナリオを作成します。

Clear history and trends

シナリオのヒストリとトレンドデータを削除します。サーバーはデータを削除した直後にシナリオを

Note:

HTTP プロキシフィールドを未設定で HTTP プロキシを使用する方法は、プロキシ関連の環境変数を設定することです。HTTP チェックの場合-Zabbix サーバーユーザーの **http_proxy** 環境変数を設定します。例：
http_proxy=http://proxy_ip:proxy_port
HTTPS チェックの場合-**HTTPS_PROXY** 環境変数を設定します。例：HTTPS_PROXY=http://proxy_ip:proxy_port 詳細についてはシェルコマンドで #man curl を実行してください。

ステップタブでは Web シナリオのステップを構成できます。Web シナリオのステップを追加するには、ステップブロックの追加をクリックします。

Scenario	Steps 2	Tags	Authentication		
* Steps					
	Name	Timeout	URL	Required	Stat
	1: Site availability	15s	http://www.example.com		200
	2: About	15s	http://www.example.com/about		200
	Add				

Note:

シークレットユーザーマクロは「*****」に解決されるため、URL で使用しないでください。

Step of web scenario ✕

*** Name**

*** URL**

Query fields

Name	Value	
<input type="text" value="name"/>	⇒ <input type="text" value="value"/>	Remove

[Add](#)

Post type

Post fields

Name	Value	
<input type="text" value="name"/>	⇒ <input type="text" value="value"/>	Remove

[Add](#)

Variables

Name	Value	
<input type="text" value="name"/>	⇒ <input type="text" value="value"/>	Remove

[Add](#)

Headers

Name	Value	
<input type="text" value="name"/>	⇒ <input type="text" value="value"/>	Remove

[Add](#)

Follow redirects

Retrieve mode

*** Timeout**

Required string

Required status codes

ステップの設定

ステップパラメータ：

パラメーター

説明

名前

一意のステップ名

パラメーター	説明
URL	<p>接続してデータ取得するための URL。</p> <p>例： https://www.example.com http://www.example.com/download</p> <p>ドメイン名は Unicode 文字で指定できます。これらは、Web シナリオステップの実行時に自動的に Punycode 変換されて ASCII に変換されます。</p> <p>[URL 解析] ボタンを使用して、オプションのクエリフィールド (?name=Admin&password=myspassword など) を URL から分離し、属性と値を URL エンコード用のクエリフィールドに移動できます。</p> <p>{macro} 構文を使用して URL で変数を使用できます。変数は {{macro}.urlencode()} 構文を使用して URL エンコードできます。ユーザーマクロと {HOST.*}macros がサポートされています。2048 文字までの制限があります。</p>
クエリフィールド	<p>URL の HTTP GET 変数</p> <p>属性と値をペアで指定します。</p> <p>値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、または {HOST.*} マクロの値は解決され、URL エンコードされます。{{macro}.urlencode()} 構文を使用すると、URL エンコードが 2 倍になります。</p>
POST データの形式	<p>ユーザーマクロと {HOST.*}マクロがサポートされています。</p> <p>HTTP POST 変数</p> <p>フォームデータモードでは、属性と値をペアで指定します。</p> <p>値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、または {HOST.*} マクロの値は解決され、URL エンコードされます。
 Raw データモードでは、属性/値が 1 行に表示され &記号で連結されます。</p> <p>未加工の値は {{macro}.urlencode()} または {{macro}.urldecode()} 構文を使用して URL エンコード/デコードできます。</p> <p>例：id=2345&userid={user}</p> <p>{user} が Web シナリオの変数として定義されている場合、ステップの実行時に値が置き換えられます。変数を URL エンコードする場合は、{user} を {{user}.urlencode()} に置き換えます。</p> <p>ユーザーマクロと {HOST.*}マクロがサポートされています。</p>
変数	<p>GET および POST 関数に使用できるステップレベルの変数。</p> <p>属性と値をペアで指定します。</p> <p>ステップレベルの変数は、シナリオレベルの変数または前のステップの変数をオーバーライドします。ただし、ステップレベル変数の値は、後のステップにのみ影響します (現在のステップには影響しません)。</p> <p>形式は次の通りです： {macro}=value {macro}=regex : < 通常の式 ></p> <p>詳細については シナリオレベルの変数の説明 を参照してください。</p> <p>変数は、クエリフィールドまたは Post 変数のフォームデータで使用される場合は自動的に URL エンコードされますが、raw Post または直接 URL で使用される場合は手動で URL エンコードする必要があります。変数はクエリフィールドまたはフォームデータで使用されると自動的に URL エンコードされます。投稿変数。ただし、生の投稿で使用する場合、または URL で直接使用する場合は、手動で URL エンコードする必要があります。</p>
ヘッダー	<p>リクエストの実行時に送信されるカスタム HTTP ヘッダー</p> <p>属性と値をペアで指定します。</p> <p>ステップレベルのヘッダーは、シナリオに指定されたヘッダーを上書きします。</p> <p>たとえば、値を指定せずに「User-Agent」属性を設定すると、シナリオレベルで設定された User-Agent 値が削除されます。</p> <p>ユーザーマクロと {HOST.*} マクロがサポートされます。</p> <p>CURLOPT_HTTPHEADER cURL オプションが設定されます。</p>
リダイレクトをたどる	<p>チェックボックスをマークすると HTTP リダイレクトを追跡します。</p> <p>CURLOPT_FOLLOWLOCATION cURL オプションが設定されます。</p>

パラメーター	説明
取得モード	取得モードを選択します Body - HTTP の応答から本文のみを取得します Headers - HTTP の応答からヘッダーのみを取得します Body and headers - HTTP の応答から本文とヘッダーを取得します
タイムアウト	Zabbix は URL の処理に設定された時間 (1 秒から最大 1 時間) を超えて URL を処理することはありません。このパラメーターは URL に接続するための最大時間と HTTP 要求を実行するための最大時間を定義します。したがって Zabbix はステップに Timeout の 2 倍の秒数以上の時間をかけません。 Timesuffixes がサポートされています。例. 30s, 1m, 1h ユーザーマクロ がサポートされています。
要求文字列	要求される正規表現パターン 取得したコンテンツ (HTML) が必要なパターンと一致しない場合、ステップは失敗します。空の場合、チェックは実行されません。 例： Homepage of Zabbix Welcome.*admin 注：Zabbix フロントエンドで作成された 正規表現 は、このフィールドで参照されません。 ユーザーマクロ と {HOST.*}マクロ がサポートされています。
要求ステータスコード	予想される HTTP ステータスコードのリスト。Zabbix がリストにないコードを取得した場合、ステップは失敗します。 空の場合、ステータスコードのチェックは実行されません。 例：200,201,210-299 ユーザーマクロ がサポートされています。

Note:

Web シナリオのステップの変更は、シナリオ全体が保存されたときにのみ保存されます。

Web 監視ステップを構成する方法の **実例** も参照してください。

タグの構成 **[Tags]** タブでは、シナリオレベルの **タグ** を定義できます。

タグ付けにより、Web シナリオと Web モニタリング項目 (/manual/web_monitoring/items) をフィルタリングできます。

認証設定 認証タブでは、シナリオの認証オプションを設定できます。タブ名の横にある緑色の点は、あるタイプの HTTP 認証が有効になっていることを示します。

HTTP authentication

SSL verify peer

SSL verify host

SSL certificate file

SSL key file

SSL key password

認証パラメータ :

パラメーター	説明
HTTP 認証	<p>認証オプション なし - 認証は使用されません。 Basic - basic 認証が使用されます。 NTLM - NTLM (Windows NT LAN Manager) 認証が使用されます。 Kerberos - Kerberos 認証が使用されます。参照：Zabbix を使用した Kerberos の設定 Digest-ダイジェスト 認証が使用されます。 認証方法を選択すると、ユーザー名とパスワードを入力するための 2 つのフィールドが追加されます。 ユーザーマクロは、ユーザー名フィールドとパスワードフィールドで使用できます。</p>
SSL ピア検証	<p>チェックボックスをオンにすると Web サーバーの SSL 証明書を検証します。 サーバー証明書は、システム全体の認証局 (CA) の場所から自動的に取得されます。 Zabbix サーバーまたはプロキシ設定パラメータ SSLCALocation を使用して CA ファイルの場所を上書きできます。 CURLOPT_SSL_VERIFYPEER cURL オプションが設定されます。</p>
SSL ホスト検証	<p>チェックボックスをオンにすると Web サーバー証明書の Common Name フィールドまたは Subject Alternate Name フィールドが一致することを確認します。 CURLOPT_SSL_VERIFYHOST cURL オプションが設定されます。</p>
SSL 証明書ファイル	<p>クライアント認証に使用される SSL 証明書ファイル名。証明書ファイルは PEM¹ 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は SSL key file フィールドを空のままにします。キーが暗号化されている場合は SSL key password フィールドにパスワードを指定します。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシ設定パラメータ SSLCertLocation によって指定されます。 HOST.* マクロとユーザーマクロをこのフィールドで使用できます。</p>
SSL 秘密鍵ファイル	<p>CURLOPT_SSLCERT cURL オプションが設定されます。 クライアント認証に使用される SSL 秘密鍵ファイルの名前 秘密鍵ファイルは PEM¹ 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシ設定パラメータ SSLKeyLocation によって指定されます。 HOST.* マクロとユーザーマクロをこのフィールドで使用できます。 CURLOPT_SSLKEY cURL オプションが設定されます。</p>

パラメーター	説明
SSL 秘密鍵パスワード	SSL 秘密鍵ファイルパスワード このフィールドではユーザーマクロを使用できます。 CURLOPT_KEYPASSWD cURL オプションが設定されます。

[1]Zabbix は PEM 形式の証明書と秘密鍵ファイルのみをサポートします。証明書と秘密鍵データが PKCS # 12 形式のファイル (通常は拡張子.p12 または.pfx) である場合は、次のコマンドを使用してそこから PEM ファイルを生成できます。

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes -out ssl-cert.key
```

Note:

Zabbix サーバーは再起動せずに証明書の変更を取得します。

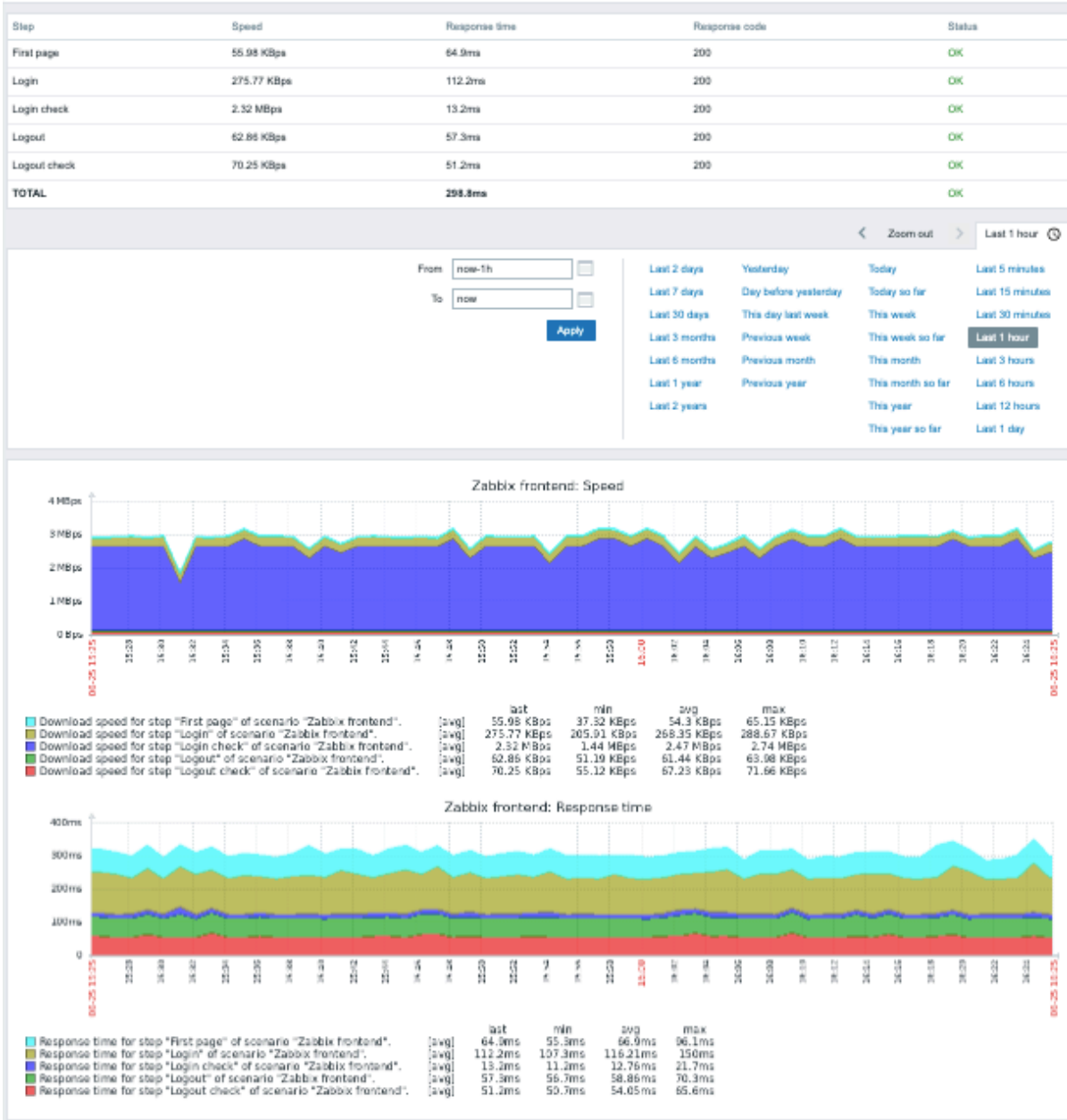
Note:

1 つのファイルにクライアント証明書と秘密鍵が含まれている場合は“SSL certificate file” フィールドで指定し、“SSL key file” フィールドは空のままにします。証明書とキーは引き続き PEM 形式である必要があります。証明書とキーの組み合わせは簡単です

```
cat client.crt client.key > client.pem
```

画面 ホスト用に構成された Web シナリオを表示するには Monitoring→Hosts に移動し、リストからホストを見つけて、最後の列の Web ハイパーリンクをクリックします。シナリオ名をクリックすると詳細情報が表示されます。

Details of web scenario: Zabbix frontend



Web シナリオの概要は、Web モニタリングウィジェットによって Monitoring→Dashboard に表示することもできます。

Web シナリオ実行の最近の結果は、Monitoring → Latest data セクションで確認できます。

拡張モニタリング 受信した HTML ページのコンテンツをログに記録する必要がある場合があります。これは一部の Web シナリオ ステップが失敗した場合に特に役立ちます。デバッグレベル 5 (trace) で確認することができます。サーバーおよびプロキシの構成 ファイルで設定するか、ランタイム制御オプション (-R log_level_increase="http poller,N" N はプロセス番号) を使用して設定できます。次の例は、デバッグレベル 4 がすでに設定されている場合に、拡張モニタリングを開始する方法です。

すべての http pollers のログレベルを上げます。

```
shell> zabbix_server -R log_level_increase="http poller"
```

2 番目の http pollers のログレベルを上げます。

```
shell> zabbix_server -R log_level_increase="http poller,2"
```

拡張 Web モニタリングが必要ない場合は -R log_level_decrease オプションを使用して停止できます。

1 Web 監視アイテム

概要

Web シナリオが作成されると、モニタリング用にいくつかの新しいアイテムが自動的に追加されます。

すべてのアイテムは、Web シナリオからタグを継承します

シナリオアイテム

シナリオが作成されると、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for scenario <Scenario>	このアイテムはシナリオ全体のダウンロード速度（1秒あたりのバイト数）つまりすべてのステップの平均速度を収集します。 Item key: web.test.in[Scenario,,bps] Type: Numeric(float)
Failed step of scenario <Scenario>	このアイテムはシナリオで失敗したステップの番号を表示します。すべてのステップが正常に実行されると、0が返されます。 Item key: web.test.fail[Scenario] Type: Numeric(unsigned)
Last error message of scenario <Scenario>	このアイテムはシナリオの最後のエラーメッセージテキストを返します。新しい値は、シナリオに失敗したステップがある場合にのみ保存されます。すべての手順に問題がない場合、新しい値は収集されません。 Item key: web.test.error[Scenario] Type: Character

“Scenario” の代わりに実際のシナリオ名が使用されます。

Note:

Web 監視アイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

Note:

シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

これらのアイテムを使用して、トリガーを作成し、通知条件を定義できます。

例 1

「Web シナリオに失敗しました」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.fail[Scenario])>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 2

トリガー名に問題の説明を含む「Web scenario failed」トリガーを作成するには、次の名前でもトリガーを定義します。

```
Web scenario "Scenario" failed: {ITEM.VALUE}
```

さらに次のようにトリガー式を定義します

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 3

「Web アプリケーションが遅い」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.in[Scenario,,bps])<10000
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

シナリオのステップアイテム

ステップが作成されるとすぐに、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for step <Step> of scenario <Scenario>	このアイテムはステップのダウンロード速度 (バイト/秒) を取得します。 Item key: web.test.in[Scenario,Step,bps] Type: Numeric(float)
Response time for step <Step> of scenario <Scenario>	このアイテムはステップの応答時間を秒単位で取得します。リクエストの開始からすべての情報が転送されるまでがカウントされます。 Item key: web.test.time[Scenario,Step,resp] Type: Numeric(float)
Response code for step <Step> of scenario <Scenario>	このアイテムはステップの応答コードを取得します。 Item key: web.test.rspcode[Scenario,Step] Type: Numeric(unsigned)

“Scenario” と “Step” の代わりに、実際のシナリオとステップ名がそれぞれ使用されます。

Note:

Web モニタリングアイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

Note:

シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

これらのアイテムを使用してトリガーを作成し、通知条件を定義できます。たとえば「ZabbixGUI ログインが遅すぎる」トリガーを作成するには次のようにトリガー式を定義します。

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 シナリオの実例

概要

このセクションでは、Web モニタリングの実例を段階的に提示します。

Zabbix Web モニタリングを使用して、Zabbix の Web インターフェイスをモニタリングしてみましょう。それが利用可能かどうか、適切なコンテンツを提供するかどうか、そしてそれがどれだけ速く機能するかを調査します。そのためにはユーザー名とパスワードを使用してログインする必要があります。

シナリオ

ステップ 1

新しい Web シナリオを追加します。

Zabbix の Web インターフェイスを監視するシナリオを追加します。シナリオはいくつかのステップを実行します。

Configuration→Hosts に移動し、ホストを選択して、そのホストの行にある Web をクリックします。次に Create web scenario をクリックします。

Scenario **Steps** Tags Authentication

* Name

* Update interval

* Attempts

Agent

HTTP proxy

Variables

Name	Value	
<input style="width: 150px;" type="text" value="{password}"/>	⇒ <input style="width: 150px;" type="text" value="zabbix"/>	Remove
<input style="width: 150px;" type="text" value="{user}"/>	⇒ <input style="width: 150px;" type="text" value="Admin"/>	Remove
Add		

Headers

Name	Value	
<input style="width: 150px;" type="text" value="name"/>	⇒ <input style="width: 150px;" type="text" value="value"/>	Remove
Add		

Enabled

すべての必須入力フィールドは、赤いアスタリスクでマークされています。

新しいシナリオフォームで、シナリオに Zabbix frontend という名前を設定します。また、{user} と {password} の2つの変数を作成します。

Tags タブに新しい Application:Zabbix frontend タグを追加することもできます。

ステップ 2

シナリオのステップを定義します。

Steps タブの Add ボタンをクリックして、個々のステップを追加します。

Web シナリオのステップ 1

最初のページが正しく応答していることを確認するため、HTTP が応答コード 200 で返され、テキスト「ZabbixSIA」が含まれていることをチェックします。

Step of web scenario ✕

*** Name**

*** URL**

Query fields

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove

[Add](#)

Post type Form data Raw data

Post fields

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove

[Add](#)

Variables

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove

[Add](#)

Headers

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove

[Add](#)

Follow redirects

Retrieve mode Body Headers Body and headers

*** Timeout**

Required string

Required status codes

ステップの構成が完了したら、Add をクリックします。

Web シナリオのステップ 2

Zabbix フロントエンドにログインするため、シナリオレベルで定義したマクロの変数（{user} と {password}）を利用します。

Step of web scenario ✕

*** Name**

*** URL**

Query fields

Name	Value	
<input type="text" value="name"/>	⇒ <input type="text" value="value"/>	Remove
Add		

Post type

Post fields

Name	Value	
<input type="text" value="name"/>	⇒ <input style="border: 1px solid #ccc;" type="text" value="{user}"/>	Remove
<input type="text" value="password"/>	⇒ <input style="border: 1px solid #ccc;" type="text" value="{password}"/>	Remove
<input type="text" value="enter"/>	⇒ <input style="border: 1px solid #ccc;" type="text" value="Sign in"/>	Remove
Add		

Variables

Name	Value	
<input style="border: 1px solid #ccc;" type="text" value="{sid}"/>	⇒ <input]="" style="border: 1px solid #ccc;" type="text" value="regex:name='csrf-token' content='([0-"/>	Remove
Add		

Headers

Name	Value	
<input type="text" value="name"/>	⇒ <input type="text" value="value"/>	Remove
Add		

Follow redirects

Retrieve mode

*** Timeout**

Required string

Required status codes

Attention:

Zabbix フロントエンドはログイン時に JavaScript リダイレクトを使用するため、最初にログインする必要があります、その後のステップでのみログイン機能を確認できます。さらにログイン手順では **index.php** ファイルへの完全な URL を使用する必要があります。

正規表現を使用した変数構文を使用して {sid} 変数 (セッション ID) を取得していることにも注意してください。
 regex:name="csrf-token" content="([0-9a-z]{16})" この変数はステップ 4 で必要になります。

Web シナリオのステップ 3

ログインできたかを確認する必要があります。そのためにログイン時にのみ表示される文字列、例えば **Administration** などを確認します。

Step of web scenario ✕

* Name

* URL

Query fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Remove](#)
[Add](#)

Post type Form data Raw data

Post fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Remove](#)
[Add](#)

Variables

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Remove](#)
[Add](#)

Headers

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Remove](#)
[Add](#)

Follow redirects

Retrieve mode Body Headers Body and headers

* Timeout

Required string

Required status codes

Web シナリオのステップ 4

フロントエンドにアクセスでき、ログインして、ログインしたコンテンツを取得できることを確認したので、ログアウトする必要があります。そうしないと、Zabbix データベースが大量のオープンセッションレコードで汚染されてしまいます。

Step of web scenario ✕

*** Name**

*** URL**

Query fields

Name	⇒	Value	
<input type="text" value="sid"/>	⇒	<input style="border: 1px solid #ccc; width: 100%;" type="text" value="{sid}"/>	Remove
<input type="text" value="reconnect"/>	⇒	<input style="border: 1px solid #ccc; width: 100%;" type="text" value="1"/>	Remove
Add			

Post type

Post fields

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input style="border: 1px solid #ccc; width: 100%;" type="text" value="value"/>	Remove
Add			

Variables

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input style="border: 1px solid #ccc; width: 100%;" type="text" value="value"/>	Remove
Add			

Headers

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input style="border: 1px solid #ccc; width: 100%;" type="text" value="value"/>	Remove
Add			

Follow redirects

Retrieve mode

*** Timeout**

Required string

Required status codes

Web シナリオのステップ 5

Username 文字列を探すことで、ログアウトしたことを確認することもできます。

Step of web scenario ✕

*** Name**

*** URL**

Query fields

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove
Add			

Post type Form data Raw data

Post fields

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove
Add			

Variables

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove
Add			

Headers

Name	⇒	Value	
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>	Remove
Add			

Follow redirects

Retrieve mode Body Headers Body and headers

*** Timeout**

Required string

Required status codes

構成完了したステップ

構成完了した Web シナリオのステップは次のようになります。

Scenario Steps 5 Tags 1 Authentication

* Steps

Name	Timeout	URL	Required	Status
1: First page	15s	http://localhost/zabbix/index.php	Zabbix SIA	200
2: Log in	15s	http://localhost/zabbix/index.php		200
3: Login check	15s	http://localhost/zabbix/index.php	Administration	200
4: Log out	15s	http://localhost/zabbix/index.php		200
5: Logout check	15s	http://localhost/zabbix/index.php	Username	200

[Add](#)

ステップ 3

完成した Web モニタリングシナリオを保存します。

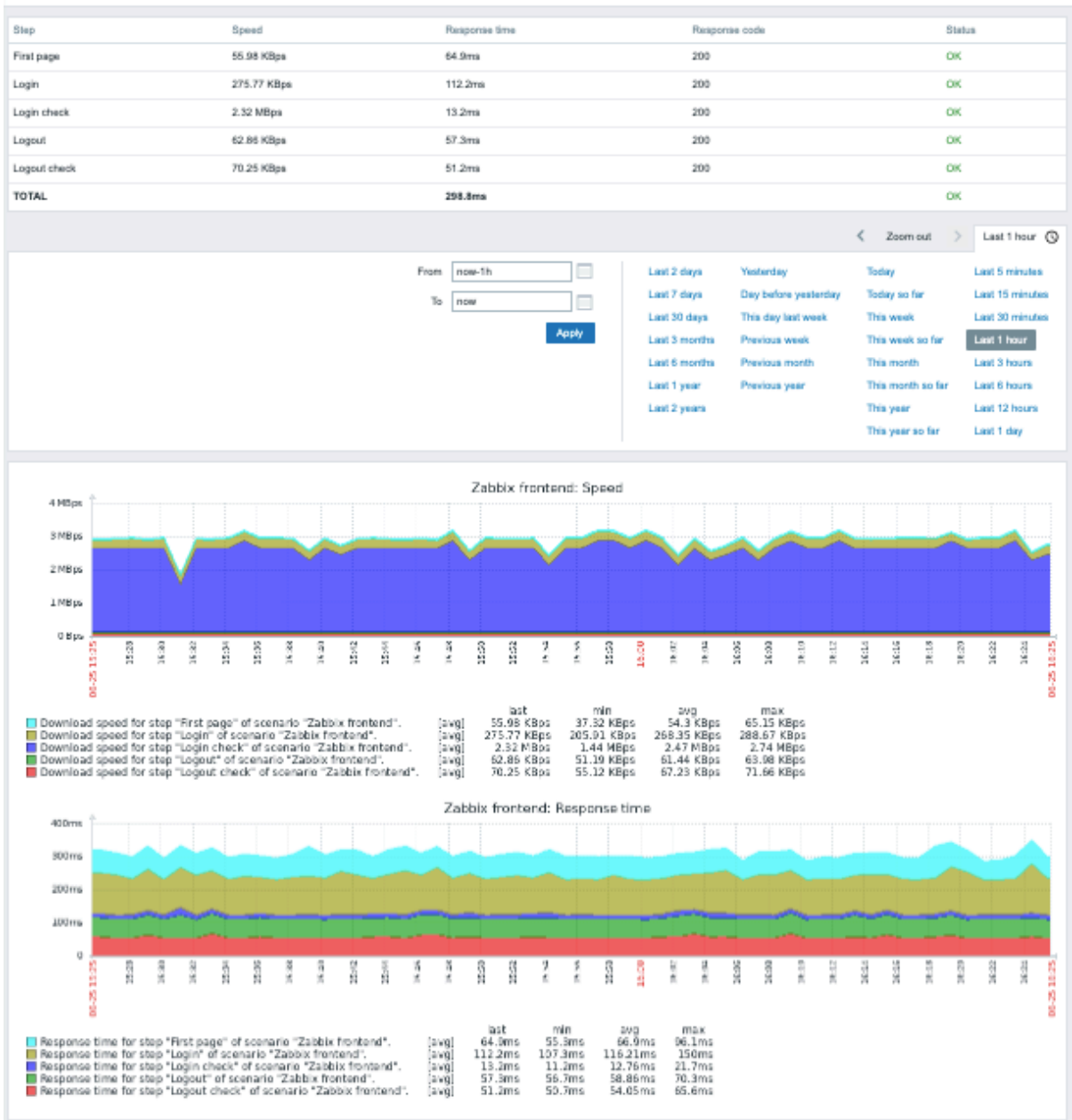
シナリオがホストに追加されます。Web シナリオ情報を表示するには Monitoring → Hosts に移動し、リストでホストを見つけて、最後の列のハイパーリンクをクリックします。

Web monitoring Filter

Host	Name	Number of steps	Last check	Status	Tags
New host	Zabbix frontend	5	46s	OK	Application: Zabbix fro...

Displaying 1 of 1 found

シナリオ名をクリックすると、より詳細な統計が表示されます。



10. 仮想マシン監視

概要 Zabbix では、バージョン 2.2.0 から VMware 環境の監視に対応しています。

Zabbix は事前に定義されたホストプロトタイプに基づき、ローレベルディスクバリエーションを使用して、
 VMware ハイパーバイザーと仮想マシンを自動的に検出し、監視するためのホストを作成することができます。

Zabbix のデフォルトデータセットには、VMware vCenter や ESX ハイパーバイザーの監視に適した
 テンプレートが用意されています。

VMware vCenter または vSphere の最低必要バージョンは 5.1 です。

詳細 仮想マシンの監視は、2つのステップで行われます。
 まず、vmware collector Zabbix プロセスによって、仮想マシンのデータが収集されます。
 これらのプロセスは、VMware 社の Web サービスから必要な情報を SOAP プロトコルで取得し、プリプロセスを行い、
 Zabbix server の共有メモリに格納します。
 そして、このデータはポーラーによって Zabbix のシンプルチェック **VMware keys** を使用して取得されます。

Zabbix バージョン 2.4.4 から、収集されたデータは2つのタイプに分けられます。
 VMware の設定データと VMware のパフォーマンスカウンタデータです。
 両データはそれぞれ独立した vmware collectors によって収集されます。
 このため、監視する VMware サービス数より多くのコレクタを有効にすることをお勧めします。
 そうしないと、VMware パフォーマンス・カ

カウンターの統計の取得が、
 VMware 構成データの取得（大規模なインストールでは時間がかかる）によって遅延する可能性があります。

現在、データストア、ネットワーク・インターフェイス、ディスク・デバイスの統計情報
 およびカスタム・パフォーマンス・カウンターの項目のみが、VMware のパフォーマンス・カウンターの情報に基づいています。

コンフィギュレーション 仮想マシンの監視を行うには、Zabbix が --with-libxml2 and --with-libcurl のオプションで
 compiled されている必要があります。

次の設定ファイルオプションを使用して、仮想マシンの監視を調整することができます。

- **StartVMwareCollectors** - フォークされた VMware コレクターインスタンスの数です。
 この値は、監視する VMware サービスの数に依存します。
 $\text{Servicenum} < \text{StartVMwareCollectors} < (\text{servicenum} * 2)$
 ここで、servicenum は VMware サービスの数です。例：監視する VMware サービスが 1 つの場合
 監視する VMware サービスが 1 つの場合、StartVMwareCollectors を 2 に設定し、
 3 つの VMware サービスがある場合、StartVMwareCollectors を 5 を設定します。
 ほとんどの場合、この値は 2 未満であってはならず、監視する VMware サービスの数の 2 倍を超えないようにしてください。
 またこの値は、VMware 環境のサイズと VMwareFrequency および VMwarePerfFrequency
 構成パラメータ（以下を参照）にも依存することに留意してください。

- **VMwareCacheSize**
- **VMwareFrequency**
- **VMwarePerfFrequency**
- **VMwareTimeout**

詳細は Zabbixserver , proxy の設定ファイルのページを参照してください。

Attention:

データストアのキャパシティメトリクスをサポートするために、Zabbix は
 VMware の設定 vpxd.stats.maxQueryMetrics パラメータに 64 以上の値を設定する必要があります。
 VMware ナレッジベース [article](#) を参照してください。

ディスクバリアー Zabbix はローレベルディスクバリアールールを使用して、VMware ハイパーバイザーと
 仮想マシンを自動的に検出することができます。

Discovery rule Preprocessing LLD macros Filters Overrides

* Name Discover VMware hypervisors

Type Simple check

* Key vmware.hv.discovery[{\$VMWARE.URL}]

* Host interface 192.0.2.255:10050

User name {\$VMWARE.USERNAME}

Password {\$VMWARE.PASSWORD}

* Update interval 1h

Custom intervals

Type	Interval	Period
Flexible Scheduling	50s	1-7,00:00

Add

* Keep lost resources period 30d

Description Discovery of hypervisors.

Enabled

Add Test Cancel

すべての必須入力フィールドには、赤いアスタリスクが表示されます。

上記のスクリーンショットのディスカバリールールのキーは次のとおりです。vmware.hv.discovery[{\$VMWARE.URL}]。

ホストプロトタイプ ホストプロトタイプは、ローレベルディスカバリールールで作成することができます。
 仮想マシンが検出されると、このプロトタイプは実際のホストになります。
 プロトタイプは、検出される前に、リンクされたテンプレートの item および trigger 以外の
 独自の item および trigger を持つことができません。
 発見されたホストは、既存のホストに属します。

Discovery rules

All templates / VMware Items 3 Triggers Graphs Dashboards Discovery rules 4 Web scenarios

Template	Name ▲	Items	Triggers	Graphs	Hosts
<input type="checkbox"/>	VMware Discover VMware clusters	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes
<input type="checkbox"/>	VMware Discover VMware datastores	Item prototypes 4	Trigger prototypes	Graph prototypes	Host prototypes
<input type="checkbox"/>	VMware Discover VMware hypervisors	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes 1
<input type="checkbox"/>	VMware Discover VMware VMs	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes 1

プロトタイプから作成されたホストが一意なホスト名を持つためには、Host name フィールドに少なくとも 1 つの low-level discovery macro を含む必要があります。

Zabbix5.2以降、検出されたホストはカスタムインターフェイスで設定することができます。
 またはディスカバリルールが属するホストのIPを継承することができます(デフォルト)
 カスタムインターフェイスを1つ以上追加するには、インターフェイスセレクトを
 InheritからCustomモードに切り替えて、次に [Add](#) を押して、
 表示されるドロップダウンメニューから必要なインターフェイスの種類を選択します。
 すべてのホストプロトタイプには、サポートされている全ての種類のインターフェイスを定義することができます。
 Zabbix agent, SNMP, JMX, IPMI. Interface fields support low-level discovery macros and **user macros**
 インタフェースフィールドは、ローレベルディスカバリマクロとユーザマクロ (/manual/config/macros/user_macros)をサポートします。
 複数のカスタム・インターフェイスが指定されている場合、Default列を使用してプライマリ・インターフェイスを指定します。

注意事項

- Customを選択し、インターフェイスを指定しなかった場合、ホストはインターフェイスなしで作成されます。

- テンプレートに属するホストのプロトタイプでInheritが選択されている場合、検出されたホストは、
 テンプレートに属するホストのインターフェイスを継承します。

Warning:

ホストインターフェイスに不正なデータが含まれている場合、ホストは作成されません。

Host
IPMI
Tags
Macros
Inventory
Encryption

* Host name

Visible name

Templates

* Groups

Group prototypes

[Add](#)

Interfaces

Inherit

Custom

	Type	IP address	DNS name
	Agent	<input style="width: 90%;" type="text" value="198.51.100.0"/>	<input style="width: 90%;" type="text"/>
	Agent	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text" value="{#LLDMACRO}"/>

[Add](#)

Monitored by proxy (no proxy)

Create enabled

Discover

LLD マクロは、可視名、ホスト・グループ・プロトタイプ・フィールド、タグ値、ホスト・
 プロトタイプ・ユーザー・マクロの値にも使用できます。

その他、ホストプロトタイプに指定可能なオプションは以下の通り:

- 既存ホストへのリンク
- テンプレートへのリンク
- 暗号化

Create enabled がチェックされている場合、ホストは有効な状態で追加されます。
 チェックされていない場合、ホストは追加されますが、無効な状態になります。

Discover がチェックされている場合 (デフォルト)、ホストが作成されます。
 もしチェックされていない場合、**discovery rule**でこの設定が上書きされない限り、
 ホストは作成されません。この機能により、検出ルール作成時の柔軟性がさらに高まります。

検出されたホストは、ホスト一覧の中で、それらを作成したディスカバリルールの名前の前に表示されます。
 検出されたホストは、手動で削除することができます。
 検出されたホストは、ディスカバリルールの Keep lost resources period (in days) の値に基づいて
 自動的に削除されることもあります。
 ホストおよびホストインベントリの有効化/無効化を除いて、ほとんどの設定オプションは読み取り専用です。
 検出されたホストは、それ自身のホストプロトタイプを持つことはできません。

すぐに使えるテンプレート Zabbix のデフォルトデータセットには、VMware venter や直接 ESX ハイパーバイザーを監視するために
 すぐに使えるテンプレートがいくつか用意されています。
 これらのテンプレートには事前に設定された LLD ルールと、仮想インスタレーション監視のための
 多くのビルトインチェックが含まれています。

VMware vCenter および ESX ハイパーバイザーの監視用テンプレートでは、

- VMware - 対応するマクロに UUID データを使用します。
- VMware FQDN - 対応するマクロに FQDN データを使用します。

Note:

VMware FQDN テンプレートが正しく機能するためには、以下のことが必要です。
 監視する各 VM は、FQDN ルールに準拠した一意の OS 名を持つ必要があります。
 VMware Tools が各マシンにインストールされている必要があります。これらの条件を満たす場合
 VMware FQDN* テンプレートを使用することをお勧めします。
 Zabbix 5.2. でカスタムインターフェイスを持つホストを作成する機能が追加されたことにより、
 VMware FQDN テンプレートの作成が可能になりました。

 FQDN の要件を満たせない場合には、従来の VMware テンプレートも利用可能です。
 ただし、VMware テンプレートには既知の問題があることに留意してください。
 検出された仮想マシンのホストは、vCenter に保存された名前で作成されます (たとえば、VM1、VM2 など)。
 これらのホストに Zabbix エージェントアクティブをインストールし、自動登録を有効にした場合
 自動再登録プロセスで起動時に登録されたホスト名 (例、vm1.example.com、vm2.example.com など) に一致するホストがないため、
 新しいホストを作成します。
 その結果、各マシンに異なる名前の 2 つの重複したホストが存在することになります。

ディスカバリーで使用されるテンプレート (通常、これらのテンプレートはホストに手動でリンクされるべきではありません)。

- VMware Hypervisor;
- VMware Guest

≡ Templates

<input type="checkbox"/> Name ▲	Hosts	Items	Triggers	Graphs	Dashboards	Discovery	Web
<input type="checkbox"/> VMware	Hosts	Items 3	Triggers	Graphs	Dashboards	Discovery 4	Web
<input type="checkbox"/> VMware FQDN	Hosts	Items 3	Triggers	Graphs	Dashboards	Discovery 4	Web
<input type="checkbox"/> VMware Guest	Hosts	Items 27	Triggers 1	Graphs	Dashboards	Discovery 3	Web
<input type="checkbox"/> VMware Hypervisor	Hosts	Items 26	Triggers 4	Graphs	Dashboards	Discovery 2	Web
<input type="checkbox"/> VMWare SD-WAN VeloCloud by HTTP	Hosts	Items 7	Triggers 5	Graphs	Dashboards	Discovery 5	Web

ホストの構成 VMware シンプルチェックを使用するには、ホストに以下のユーザーマクロが定義されている必要があります。

- **{ \$VMWARE.URL }** - VMware サービス (vCenter または ESX ハイパーバイザー) SDK URL (<https://servername/sdk>)
- **{ \$VMWARE.USERNAME }** - VMware サービスのユーザー名
- **{ \$VMWARE.PASSWORD }** - VMware サービス { \$VMWARE.USERNAME } ユーザー名パスワード

例 以下の例は、Zabbix 上で VMware の監視を素早く設定する方法です：

- オプション (--with-libxml2 と --with-libcurl) を指定して Zabbix サーバーをコンパイルします
- Zabbix 設定ファイル内の StartVMwareCollectors オプションを 1 以上にします
- 新しいホストを作成します
- VMware の認証のためのマクロを設定します：

Macro	Value	
{PASSWORD}	🔒
{URL}	<url>	T
{USERNAME}	<username>	T

[Add](#)

- ホストに VMware のサービステンプレートをリンクします：

Linked templates	Name	Action
Link new templates	<input type="text" value="VMware X"/> <input type="text" value="type here to search"/>	

- 追加ボタンをクリックしてホストを保存します。

拡張ログ VMware コレクターによって収集されたデータは、デバッグレベル 5 を使用して
 詳細なデバッグのためのログを記録することができます。
 このレベルは **server** と **proxy** 設定ファイルまたは
 ランタイムコントロールオプションを使用して設定します。
 実行時制御オプションの使用 (-R log_level_increase="vmware collector,N" (N はプロセス番号)) を使用します。
 次の例は、デバッグレベル 4 がすでに設定されている場合に、拡張ロギングを開始する方法を示しています。

vmware collectors 全体のログレベルの加算：

```
shell> zabbix_server -R log_level_increase="vmware collector"
```

second vmware collector のログレベルの加算: shell> zabbix_server -R log_level_increase="vmware collector,2"

VMware コレクターデータの拡張ログが必要ない場合は、-R log_level_decrease オプションを使用して、
 ログを停止することができます。

トラブルシューティング

- メトリクスが利用できない場合、最近の VMware vSphere のバージョンで利用できなくなったり、
 デフォルトでオフになったりしていないか、また、パフォーマンスメトリックのデータベースクエリに
 何らかの制限がかかっているかどうかを確認してください。
 [ZBX-12094](#) を参照してください。

- 'config.vpxd.stats.maxQueryMetrics' is invalid or exceeds the maximum number of characters permitted** エラーが発生した場合、
 vCenter Server の設定に config.vpxd.stats.maxQueryMetrics パラメータを追加します。
 このパラメータの値は、VMware の web.xml にある maxQuerysize の値と同じにする必要があります。
 この VMware のナレッジベース [article](#) を参照してください。

1 仮想マシンディスクカバリキーフィールド

次の表は、仮想マシン関連のディスクカバリキーが返すフィールドの一覧です。

Item key**Description****Field****Retrieved content**

vmware.cluster.discovery
Performs cluster discovery.

{#CLUSTER.ID} Cluster identifier.
{#CLUSTER.NAME} name.

vmware.datastore.discovery
Performs datastore discovery.

{#DATASTORE.ID} Datastore name.
{#DATASTORE.PATH} object with an array of {instance-Name:partitionId}.

vmware.dc.discovery
Performs datacenter discovery.

{#DATACENTER.ID} Datacenter name.
{#DATACENTER.ID} Datacenter ID.

vmware.hv.discovery
Performs hypervisor discovery.

{#HV.UUID} Unique hypervisor identifier.
{#HV.ID} Hypervisor identifier (Host-System managed object name).
{#HV.NAME} Hypervisor name.
{#HV.NETNAME} Hypervisor network host name.

Item key

{#HV.IP} Hypervisor IP address, might be empty. In case of an HA configuration with multiple net interfaces, the following selection priority for interface is observed:
- prefer the IP which shares the IP-subnet with the vCenter IP
- prefer the IP from IP-subnet with default gateway
- prefer the IP from interface with the lowest ID
This field is supported since Zabbix 5.2.2.

{#CLUSTER.NAME} name, might be empty.

{#DATACENTER.NAME} name.

Item key

		<p>{#PARENT.NAME} of container that stores the hypervisor. Supported since Zabbix 4.0.3.</p> <p>{#PARENT.TYPE} of container in which the hypervisor is stored. The values could be Datacenter, Folder, ClusterComputerRes VMware, where 'VMware' stands for unknown container type. Supported since Zabbix 4.0.3.</p>
vmware.hv.datastore.discovery	Performs hypervisor datastore discovery. Note that multiple hypervisors can use the same datastore.	<p>{#DATASTORE} Datastore name.</p> <p>{#MULTIPATH} Registered number of datastore paths.</p> <p>{#MULTIPATH.PARTITION.COUNT} Number of available disk partitions.</p>
vmware.vm.discovery	Performs virtual machine discovery.	<p>{#VM.UUID} Unique virtual machine identifier.</p>

Item key

{#VM.ID} Virtual machine identifier (Virtual-Machine managed object name).

{#VM.NAME} Virtual machine name.

{#HV.NAME} Hypervisor name.

{#DATACENTER.NAME} Datacenter name.

{#CLUSTER.NAME} Cluster name, might be empty.

{#VM.IP} Virtual machine IP address, might be empty. Supported since Zabbix 5.2.2.

{#VM.DNS} Virtual machine DNS name, might be empty. Supported since Zabbix 5.2.2.

{#VM.GUEST.FAMILY} virtual machine OS family, might be empty. Supported since Zabbix 5.2.2.

Item key

{#VM.GUESTFULLNAME}

guest
virtual
machine
OS
name,
might be
empty.
Supported
since
Zabbix
5.2.2.

{#VM.FOLDER}

chain of
virtual
machine
parent
folders,
can be
used as
value for
nested
groups;
folder
names
are com-
bined
with "/".
Might be
empty.
Supported
since
Zabbix
5.4.2.

vmware.vm.net.if.discovery

Performs virtual machine network interface discovery.

{#IFNAME} Network
interface
name.

vmware.vm.vfs.dev.discovery

Performs virtual machine disk device discovery.

{#DISKNAME} Disk
device
name.

vmware.vm.vfs.fs.discovery

Performs virtual machine file system discovery.

{#FSNAME} File
system
name.

11. メンテナンス

概要 Zabbix では、ホストグループ、ホスト、および特定のトリガー/サービスのメンテナンス期間を定義できます。

メンテナンスには、データ収集ありとデータ収集なしの2種類があります。

"データ収集あり"のメンテナンス中、トリガーは通常どおり処理され、必要に応じてイベントが作成されます。ただし、アクション構成でメンテナンス中の場合に実行を保留オプションがチェックされている場合、メンテナンス中のホスト/トリガーの障害のエスカレーションは一時停止されます。この場合、通知の送信やリモートコマンドを含む可能性のあるエスカレーション手順は、メンテナンス期間が続く限り無視されます。保守中は、問題のリカバリーおよび更新操作は抑止されず、エスカレーションのみが抑止されることに注意してください。

たとえば、障害の発生から 0 分、30 分、60 分にエスカレーション・ステップをスケジュールしていて、実際に障害が発生した時に 10 分から 40 分までの 30 分間のメンテナンスがある場合、ステップ 2 とステップ 3 は 30 分後、または 60 分後と 90 分後に実行されます (問題がまだ存在する場合)。同様にメンテナンス中に問題が発生した場合、エスカレーションはメンテナンス終了後に開始されます。

メンテナンス中に障害の通知を正常に (遅滞なく) 受け取るには、アクション構成でメンテナンス中の場合に実行を保留オプションのチェックを外す必要があります。

Note:

少なくとも 1 つのホスト (トリガー式で使用) がメンテナンスモードでない場合、Zabbix は障害通知を送信します。

メンテナンス中は Zabbix サーバーが稼働している必要があります。タイマープロセスは、毎分の 0 秒でホストステータスをメンテナンスとの間で切り替える役割を果たします。ホストがメンテナンスに入ると、Zabbix サーバーのタイマープロセスは未解決の障害をすべて読み取り、それらの障害を抑制する必要があるかどうかを確認します。未解決の障害が多数ある場合、これはパフォーマンスに影響を与える可能性があります。また、Zabbix サーバーは、その時点でメンテナンスが構成されていなくても、起動時に未解決の障害をすべて読み取ります。

プロキシは、メンテナンスの種類 ("データなし" のメンテナンスを含む) に関係なく、常にデータを収集します。'データ収集なし' が設定されている場合、データは後でサーバーによって無視されます。

"データなし" のメンテナンスが終了すると、nodata() 関数を使用したトリガーは、チェックしている期間中、次のチェックの前に実行しません。

ホストのメンテナンス中にログ項目が追加され、メンテナンスが終了した場合、メンテナンス終了以降の新しいログファイルエントリのみが収集されます。

"データなし" メンテナンスタイプのホストにタイムスタンプ付きの値が送信された場合 (Zabbix sender などを使用)、この値は削除されますが、期限切れのメンテナンス期間にタイムスタンプ付きの値を送信することは可能であり、受け入れられます。

メンテナンス期間、ホスト、グループ、またはタグがユーザーによって変更された場合、変更は構成キャッシュの同期後にのみ有効になります。

Attention:

メンテナンス期間の作成時には、作成したユーザーのタイムゾーンが使用されます。ただし、定期的なメンテナンス期間 (毎日、毎週、毎月) がスケジュールされた場合は、Zabbix サーバーのタイムゾーンが使用されます。定期的なメンテナンス期間の動作を保証するには、Zabbix のすべての部分で共通のタイムゾーンを使用する必要があります。

設定 メンテナンス期間を設定するには:

- 設定 → メンテナンスに移動します
- メンテナンス期間の作成 (または既存のメンテナンス期間の名前) をクリックします。
- フォームにメンテナンスパラメータを入力します

* Name

Maintenance type With data collection No data collection

* Active since

* Active till

* Periods

Period type	Schedule	Period	Action
One time only	2020-04-17 11:33	1y	Edit Remove
Add			

Host groups

Hosts

* At least one host group or host must be selected.

Tags

And/Or Or

Contains Equals [Remove](#)

[Add](#)

Description

すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
名前	メンテナンス期間名
メンテナンスタイプ	2種類のメンテナンスを設定できます。 データ収集あり - メンテナンス中でもサーバーによってデータが収集され、トリガーが処理されます データ収集なし - メンテナンス中、サーバーはデータ収集しません
設定有効期間の開始日時	メンテナンス期間の設定を有効にします。 注: この時間を設定するだけでは、メンテナンス期間はアクティブになりません。メンテナンス期間をアクティブにするには監視対象のメンテナンス期間タブに移動します。
設定有効期間の終了日時	メンテナンス期間が非アクティブになる日時。
監視対象のメンテナンス期間	このブロックにより、メンテナンスが行われる正確な日時を定義できます。 Add をクリックすると、柔軟なメンテナンス期間フォームを含むポップアップウィンドウが開き、メンテナンススケジュールを定義できます。詳細な説明については、 メンテナンス期間 を参照してください。

パラメータ	説明
ホストグループ	メンテナンスを有効にするホストグループを選択します。指定したホストグループのすべてのホストに対してメンテナンスが有効になります。このフィールドはオートコンプリートなので、入力を開始すると、使用可能なすべてのホストグループがドロップダウン表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。したがって、ネストされたグループのホストでもメンテナンスが有効になります。
ホスト	メンテナンスを有効にするホストを選択します。このフィールドはオートコンプリートなので、入力を開始すると、利用可能なすべてのホストがドロップダウン表示されます。
タグ	メンテナンスタグが指定されている場合、選択したホストのメンテナンスが有効になった際に、タグが一致する問題は抑制されます (つまりアクションは実行されません)。タグが複数ある場合は、次のように計算されます: And/Or - すべてのタグが対応している必要があります。ただし、同じタグ名を持つタグは Or 条件で計算されます Or - 少なくとも 1 つのタグが対応している必要があります タグ値を一致させる方法は 2 つあります: 含む - case - 部分文字列を区別して一致 (タグ値に入力された文字列が含まれる) 等しい - 大文字と小文字を区別して文字列一致 (タグ値が入力された文字列と等しい) データ収集ありモードが選択された場合にのみ、タグを指定できます。
説明	メンテナンス期間の説明

メンテナンス期間

メンテナンス期間ウィンドウは、定期的または 1 回限りのメンテナンスの時間をスケジュールするためのものです。フォームは動的で、選択した 期間のタイプに基づいて使用可能なフィールドが変化します。

Maintenance period ✕

Period type

* Date

* Maintenance period length Days Hours Minutes

期間のタイプ	説明
一度限り	日付と時刻、およびメンテナンス期間の長さを定義します。
毎日	毎日 - 繰返し間隔 (日): 1 (デフォルト) - 毎日、2 - 隔日など 開始時刻 (時: 分) - メンテナンスの開始時刻
毎週	メンテナンス期間 - メンテナンスがアクティブになる期間 毎週 - 繰返し間隔 (週): 1 (デフォルト) - 毎日、2 - 隔日など 曜日 - メンテナンスを行う曜日 開始時刻 (時: 分) - メンテナンスの開始時刻
毎月	メンテナンス期間 - メンテナンスがアクティブになる期間 毎月 - 定期メンテナンスが実施されるすべての月を選択します。 日付: 日 - メンテナンスが毎月同じ日に行われる場合は、このオプションを選択します。(たとえば、毎月 1 日)。次に、表示される新しいフィールドで必要な日を選択します。 日付: 曜日 - メンテナンスが特定の日 (毎月の最初の月曜日など) にのみ行われる場合は、このオプションを選択します) 次に、ドロップダウンで月の必要な週 (第 1、第 2、第 3、第 4、または最終) を選択し、メンテナンス日のチェックボックスをオンにします。 時刻 (時: 分) - メンテナンスの開始時刻 メンテナンス期間 - メンテナンスがアクティブになる期間

設定が完了したら追加を押して、メンテナンス期間を 監視対象のメンテナンス期間ブロックに追加します。

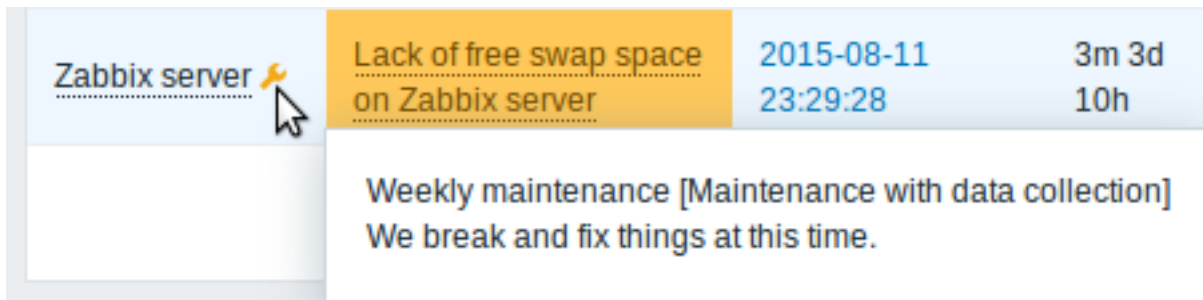
注意事項

- 毎日/毎週パラメータが 1 より大きい場合、開始日または開始週は設定有効期間の開始日時時刻が該当する日/週です。例えば：
 - 設定有効期間の開始日時を 1 月 1 日の 12:00 に設定し、2 日ごとの午後 11 時に 1 時間のメンテナンスを設定すると、最初のメンテナンス期間は 1 月 1 日午後 11 時に始まり、2 回目のメンテナンス期間は 1 月 3 日午後 11 時に始まります。
 - 同じ設定有効期間の開始日時時刻と 2 日ごとの午前 1 時に 1 時間のメンテナンスが設定されている場合、最初のメンテナンス期間は 1 月 3 日の午前 1 時に開始され、2 回目のメンテナンス期間は 1 月 5 日の午前 1 時に開始されます。
- 夏時間 (DST) の変更は、メンテナンス期間に影響しません
 - 例えば、通常では午前 1 時に開始して午前 3 時に終了する 2 時間のメンテナンスがあるとします。
 - 1 時間のメンテナンス後 (午前 2 時) に DST の変更が発生し、現在の時刻が 2:00 から 3:00 に変更された場合、メンテナンスはさらに 1 時間、4:00 まで続きます。
 - メンテナンスの 2 時間後 (午前 3 時) に夏時間の変更され、現在時刻が 3 時から 2 時になった場合、メンテナンスは 2 時間経過したため停止します。
 - メンテナンス期間が 1 日に設定されている場合、通常は午前 12 時に開始され、翌日の午前 12 時に終了します。
 - Zabbix は日数を時間単位で計算するため、メンテナンスの実際の期間は 24 時間です。
 - 現在時刻を 1 時間進めると、翌日の午前 1 時にメンテナンスが終了します。
 - 現在時刻を 1 時間戻すと、その日の 23 時にメンテナンスを終了します。
 - DST の変更によってスキップされた時間帯にメンテナンス期間が開始された場合：
 - メンテナンスは開始されません。

表示 メンテナンス中のホストの表示

オレンジ色のレンチアイコン (.../assets/en/manual/web_interface/frontend_sections/configuration/maintenance_wrench_icon.png) は、このホストがメンテナンス中であることを示します。

- Monitoring → Dashboard に表示されます。
- Monitoring → Problems
- Inventory → Hosts → Host inventory details
- Host inventory details ('Status' 列を参照)



マウスポインターをアイコンの上に合わせると、メンテナンスの詳細が表示されます。

また、メンテナンス中のホストは Monitoring → Maps にオレンジ色の背景が表示されます。

"通知なし" とされた問題の表示

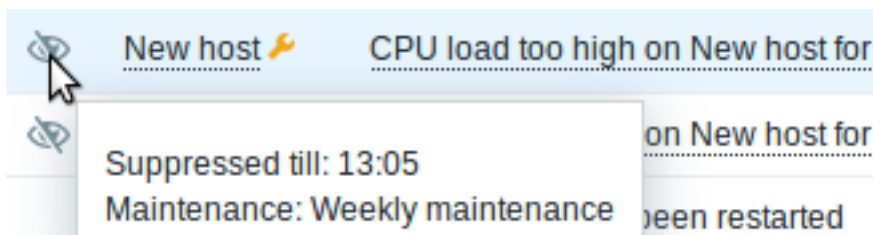
通常、メンテナンス中のホストの問題は"通知なし" とされ、フロントエンドに表示されません。
しかし、Show suppressed problems オプションを選択することで、"通知なし" とされた問題を表示するよう設定することもできます。

- Monitoring → Dashboard (Problem hosts , Problems , Problems by severity , Trigger overview ウィジェット設定)
- Monitoring → Problems (フィルター内)
- Monitoring → Maps (マップ設定内)
- Global notifications (ユーザープロファイル設定内)

"通知なし" とされた問題が表示されている場合、以下のアイコンが表示されます。



アイコンにマウスを乗せると、詳細が表示されます。



12. 正規表現

概要 [Perl 互換の正規表現](#) (PCRE、PCRE2) が Zabbix でサポートされています。

Zabbix で正規表現を使用するには 2 つの方法があります。

- 正規表現を手動で入力する
- Zabbix で作成したグローバルな正規表現を使用する。

正規表現 サポートされている場所では、正規表現を手動で入力することができます。式は @ で始まらないことに注意してください。Zabbix ではグローバルな正規表現を参照するために、この記号が使用されるためです。

Warning:

正規表現を使用する際、スタック不足になる可能性があります。詳細については、[pcrestack manpage](#)を参照してください。

複数行の一致では、アンカー ^ と \$ は、文字列全体の先頭/末尾ではなく、各行の先頭/末尾でそれぞれ一致することに注意してください。

グローバルな正規表現 Zabbix フロントエンドで複雑な正規表現を作成およびテストするための高度なエディターがあります。

このようにして作成した正規表現は、@mycustomregexp のように、先頭に @ を付けたその名前を参照することで、フロントエンドのさまざまな場所で使用できます。

グローバルな正規表現を作成するには:

- 管理 → 一般設定に移動。
- ドロップダウンから正規表現を選択。
- 正規表現の作成をクリック。

条件式タブでは、正規表現名を設定し、部分式を追加できます。

* Name	Expression type	Expression	Delimiter	Case s
Network interfaces for discovery	Result is FALSE	^Software Loopback Interface		<input checked="" type="checkbox"/>
	Result is FALSE	^(In)?[Ll]oop[Bb]ack[0-9._]*\$		<input checked="" type="checkbox"/>
	Result is FALSE	^NULL[0-9.]*\$		<input checked="" type="checkbox"/>
	Result is FALSE	^[Ll]o[0-9.]*\$		<input checked="" type="checkbox"/>
	Result is FALSE	^[Ss]ystem\$		<input checked="" type="checkbox"/>
	Result is FALSE	^Nu[0-9.]*\$		<input checked="" type="checkbox"/>

[Add](#)

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
名前	正規表現名を設定します。すべての Unicode 文字を使用できます。

パラメータ	説明
条件式	条件式ブロックの追加をクリックして、新しい部分式を追加します。 条件式の形式を選択します: 文字列が含まれる - 部分文字列と一致します いずれかの文字列が含まれる - 区切りリストの任意の部分文字列と一致します。区切りリストには、コンマ (,)、ドット (.), またはスラッシュ (/) が含まれます。 文字列が含まれない - 部分文字列以外の任意の文字列に一致します 結果が真 - 正規表現に一致する 結果が偽 - 正規表現に一致しない 条件式部分文字列/正規表現を入力します。
区切り文字	正規表現でテキスト文字列を区切るには、コンマ (,)、ドット (.), またはスラッシュ (/) を使用します。このパラメーターは条件式の形式が "いずれかの文字列が含まれる" を選択している場合にのみ有効です。
大文字と小文字を区別	正規表現が文字の大文字と小文字を区別するかどうかを指定するチェックボックス

式内のスラッシュ (/) は、区切り文字ではなく文字どおりに扱われるので、スラッシュを含む式はエラーなしで保存できます。

Attention:

Zabbix のカスタム正規表現名には、カンマやスペースなどを含むことができます。そのため、参照時に誤解を招く可能性がある場合 (例えば、アイテムキーのパラメータにカンマがある場合)、参照全体を次のように引用符で囲むことができます。
"@My custom regexp for purpose1, purpose2".

正規表現名は、他の場所 (たとえば LLD ルールのプロパティ) では引用符で囲まないでください。

テストタブでは、テスト文字列を指定することで、正規表現とその部分式をテストできます。

Expression type	Expression	Result
Result is FALSE	^Software Loopback Interface	TRUE
Result is FALSE	^\n)?[Ll]oop[Bb]ack[0-9.]*\$	TRUE
Result is FALSE	^NULL[0-9.]*\$	TRUE
Result is FALSE	^Ll[o0-9.]*\$	FALSE
Result is FALSE	^[Ss]ystem\$	TRUE
Result is FALSE	^Nu[0-9.]*\$	TRUE
Combined result		FALSE

結果には、各部分式のステータスとカスタム式の合計ステータスが表示されます。

カスタム式の合計ステータスは、結合結果として定義されます。複数のサブ式が定義されている場合、Zabbix は AND 論理演算子を使用して結合結果を計算します。これは、少なくとも 1 つの結果が False の場合、結合された結果も False ステータスになることを

意味します。

デフォルトのグローバル正規表現 Zabbix には、デフォルトのデータセットにいくつかのグローバル正規表現が含まれています。

名前	式	一致
File systems for discovery	^(btrfs ext2 ext3 ext4 jfs reiser xfs ufs vxfs zfs fat32 ntfs refs apfs)\$	または"jfs"または"reiser"または"xfs"または"ffs"または"ufs"または"jfs"または"jfs2"または"vxfs"または"hfs"または"refs"または"apfs"または"ntfs"または"fat32"または"zfs"
Network interfaces for discovery	^Software Loopback Interface ^lo\$ ^(In)?[Ll]oop[Bb]ack[0-9._]*\$ ^NULL[0-9.]*\$ ^[Ll]o[0-9.]*\$ ^[Ss]ystem\$ ^Nu[0-9.]*\$	"Software Loopback Interface" で始まる文字列。 "lo" オプションで"ln"で始まり、"l"または"l"、"oop"、"B"または"b"、"ack"の順に続く文字列。任意の数の数字、ドットまたはアンダースコアが続く場合があります。 "NULL"で始まる文字列。オプションでその後に任意の数の数字またはドットが続きます。 "Lo"または"lo"で始まり、任意の数の数字またはドットが続く文字列。 "System"または"system" "Nu"で始まる文字列。オプションでその後に任意の数の数字またはドットが続きます。
Storage devices for SNMP discovery	^(Physical memory Virtual memory Memory buffers Cached memory Swap space)\$	"Physical memory"または"Virtual memory"または"Memory buffers"または"Cached memory"または"Swap space"
Windows service names for discovery	^(MMCSS\ gupdate\ SysmonLog\ clr_optimization_v2.0.50727_32\ clr_optimization_v4.0.30319_32\ gupdate\ SysmonLog\ clr_optimization_v2.0.50727_32\ clr_optimization_v4.0.30319_32)\$	は"clr_optimization_v2.0.50727_32"または"clr_optimization_v4.0.30319_32"などの文字列は、ドットの代わりに改行以外の任意の文字を入れることができます。
Windows service startup states for discovery	^(automatic automatic delayed)\$	"automatic"または"automatic delayed"

例 例 1

低レベルの検出で次の式を使用すると、特定の名前のデータベース以外を検出できます。

^TESTDATABASE\$

Test string

TESTDATABASE

Test expressions

Result	Expression type	Expression	Result
	Result is FALSE	^TESTDATABASE	FALSE
	Combined result		FALSE

選択された Expression type : "結果は FALSE"です。"TESTDATABASE"を含んだ名前と一致しません。

インライン型正規表現修飾子を含む例

インラインモディファイアを含む次の正規表現を使用します。
 (?i) を使用して、"error" という文字にマッチさせます。
 (?i)error

Test string `Sometexthere1345Error1357`

Test expressions

Result	Expression type	Expression	Result
Result is TRUE		(?i)error	TRUE
Combined result			TRUE

選択された Expression type : "結果は TRUE" です。文字"error" にマッチする。

インライン正規表現修飾子を含む別の例

複数のインライン修飾子を含む次の正規表現を使用すると、特定の行以降の文字にマッチします。

(?<=match (?i)everything(?-i) after this line\n)(?sx).*# we add s modifier to allow .match newline character

Test string

```
Some text here for your consideration
1235kfd345
match eveRything after this line
Continuation
```

Test expressions

Result	Expression type	Expression	Result
Result is TRUE		(?<=match (?i)everything(?-i) after this line\n)(?sx).*# we add s modifier to allow .match newline characters	TRUE
Combined result			TRUE

選択された Expression type : "結果は TRUE" です。
 特定の行以降の文字にマッチします。

Attention:

g 修飾子は、行内で指定することはできません。利用可能な修飾子の一覧は
 利用可能な修飾子のリストは、[pcresyntax manpage](#)にあります。
 PCRE の構文に関するより詳しい情報は [PCRE HTML documentation](#) を参照してください。

場所による正規表現のサポート

場所	正規表現	グローバル正規表現	複数行一致	コメント	
エージェントアイテム	eventlog[] log[] log.count[] logrt[] logrt.count[] proc.cpu.util[] proc.mem[] proc.num[] sensor[]	Yes	Yes Yes/No No	Yes No	regexp, severity, source, even regexp パラメータ regexp パラメータは両方をサポート cmdline パラメータ Linux 2.4 の device および sens

場所	正規表現	グローバル正規表現	複数行一致	コメント	
	system.hw.macaddr[]			interface パラメータ	
	system.sw.packages[]			package パラメータ	
	vfs.dir.count[]			regex_incl, regex_excl, regex	
	vfs.dir.size[]			regex_incl, regex_excl, regex	
	vfs.file.regexp[]		Yes	regexp パラメータ	
	vfs.file.regmatch[]				
	web.page.regexp[]				
SNMP トラップ					
	snmptrap[]	Yes	Yes	No	regexp パラメータ
Item value preprocessing トリガー・計算項目の関数		Yes	No	No	pattern パラメータ
	count()	Yes	Yes	Yes	operator パラメータが regexp ま
	countunique()	Yes	Yes		
	find()	Yes	Yes		
	logeventid()	Yes	Yes	No	pattern パラメータ
	logsource()				
ローレベルディスカバリ					
	Filters	Yes	Yes	No	正規表現フィールド
	Overrides	Yes	No		一致、一致しないの 実行内容条件の
アクション条件		Yes	No	No	In 一致、ホスト名およびホストメタ
Web 監視		Yes	No	Yes	regex: 接頭辞が付いた 変数 必須文字列フィールド
ユーザーマクロコンテキスト マクロ関数		Yes	No	No	regex: プレフィックスを使用したマ
	regsub()	Yes	No	No	pattern パラメータ
	iregsub()				
アイコンマッピング		Yes	Yes	No	式フィールド
値マッピング		Yes	No	No	マッピングタイプが regexp の場合

13. 障害イベントの確認

概要 Zabbix の障害イベントはユーザーが確認することができます。

ユーザーは障害イベントの通知を受けると、Zabbix フロントエンドに移動し、以下のいずれかの方法で障害の更新ポップアップウィンドウを開き、障害を確認することができます。確認する際に、障害に対するコメント (対応中など) を入力することができます。

これにより、他のシステムユーザーが同じ問題を発見した場合、すぐに、それが確認済かどうかや、これまでのコメントを見ることができます。

このように、複数のシステムユーザーとの問題解決のワークフローを連携して行うことができます。

確認状況は、**アクション**操作を定義する際にも使用されます。例えば、あるイベントがしばらく確認されなかった場合にのみ、上位のマネージャに通知を送信するように定義できます。

イベントを確認し、それについてコメントするには、ユーザーは対応するトリガーに対して最低限の読み取り権限を持っている必要があります。障害の深刻度を変更したり、障害をクローズしたりするには、ユーザーは対応するトリガーに対して読み取り/書き込みの権限を持っている必要があります。

障害を確認するための、障害更新ポップアップウィンドウにアクセスするには、いくつかの方法があります。

- 監視 → 障害で障害を選択した状態で、リストの下の一括更新をクリックします。
- 障害の確認状況を示す 確認済列をクリックし、
 - 監視 → ダッシュボード (障害と 障害・深刻度ウィジェット)
 - 監視 → 障害
 - 監視 → 障害 → イベント詳細

確認済列には 'はい' または 'いいえ' のリンクがあり、それぞれ確認済かどうかを示しています。リンクをクリックすると、障害更新ポップアップウィンドウが表示されます。

- 以下の手順で未解決の問題のセルをクリックすることもできます。
 - 監視 → ダッシュボード (トリガー概要ウィジェット)

ポップアップメニューには、確認オプションがあり、障害更新ウィンドウに移動します。

障害の更新 障害更新ポップアップで可能です。

- 障害に対するコメント
- これまでのコメントとアクションを表示
- 障害の深刻度を変更
- 障害の確認/未確認の切り替え
- 手動で障害をクローズする

Update problem
✕

Problem **/: Disk space is critically low (>90% used)**

Message

History	Time	User	User action	Message
	2020-05-07 11:27:50	Admin (Zabbix Administrator)	✕	
	2020-05-07 11:27:43	Admin (Zabbix Administrator)	✓ ⋮	Ok

Scope

Only selected problem

Selected and all other problems of related triggers 1 event

Change severity Not classified Information Warning Average High Disaster

Acknowledge

Close problem

* At least one update operation or message must exist.

Update
Cancel

必須入力項目には、赤いアスタリスクが表示されています。

パラメータ	説明
障害	障害が1つだけ選択されている場合は障害名が表示されます。複数の障害が選択されている場合は、N個の障害が選択されていますと表示されます。
メッセージ	障害に対するコメントをテキストで入力してください（最大 2048 文字）
履歴	障害に対するこれまでの活動やコメントが、時間やユーザーの詳細とともに一覧表示されます。ユーザーの操作を表すアイコンの意味については、 イベント詳細 のページをご覧ください。なお、履歴は、更新対象の障害が1つだけ選択されている場合にのみ表示されます。
スコープ	障害の深刻度の変更、確認、手動によるクローズなどのアクションの範囲を定義する。 選択した障害のみ - このイベントのみに影響します。 選択した障害と関連するトリガーのすべての障害 - 確認/クローズされた障害がある場合、このイベントと、これまでに確認/クローズされていない他のすべての障害に影響します。スコープにすでに確認またはクローズされた障害が含まれている場合、これらの障害が繰り返し確認/クローズされることはありません。一方、メッセージと深刻度の変更操作の数は制限されていません。

パラメータ	説明
深刻度の変更	<p>チェックボックスをマークし、深刻度ボタンをクリックすると、障害の深刻度が更新されます。</p> <p>深刻度を変更するチェックボックスは、選択した障害のうち少なくとも1つに読み取り/書き込み権限がある場合に使用できます。更新をクリックすると、読み取り/書き込み可能な障害のみが更新されます。選択したトリガーのいずれにも読み取り/書き込み権限が存在しない場合、チェックボックスは無効になります。</p>
確認	<p>チェックボックスをマークして、障害を確認します。</p> <p>このチェックボックスは、選択した障害の中に未確認のものが1つでもある場合に有効です。</p> <p>すでに確認されている障害に対して、さらに確認を追加することはできません（ただし、別のコメントを追加することは可能です）。</p>
未確認	<p>チェックボックスをマークして、障害を未確認状態にします。</p> <p>このチェックボックスは、選択された中に少なくとも1つの確認済の障害がある場合に利用できます。</p>
障害のクローズ	<p>選択した障害を手動でクローズする場合は、チェックボックスをオンにします。</p> <p>障害をクローズするためのチェックボックスは、選択した障害の少なくとも1つについて、トリガー設定で手動クローズを許可するオプションがチェックされている場合に利用可能です。更新をクリックすると、クローズが許可されている障害のみがクローズされます。手動でクローズできる障害がない場合、チェックボックスは無効になります。</p> <p>すでにクローズした障害を、繰り返しクローズすることはできません。</p>

表示 障害の確認状況に基づいて、ダッシュボードやマップでの障害数の表示方法を設定できます。この設定を行うには、**マップ設定**、または障害・深刻度**ダッシュボードウィジェット**から、障害表示オプションで選択する必要があります。すべての障害数、未確認の障害数を分離して表示することが可能です。

障害の更新情報（確認状況など）をもとに、メッセージの送信やリモートコマンドの実行といった更新操作を設定することができます。

14. 設定のエクスポート/インポート

概要 Zabbix のエクスポート/インポート機能により、Zabbix システム間で様々な設定情報を交換することができます。

この機能の典型的な使用例:

- テンプレートやネットワークマップの共有 - Zabbix ユーザによる設定パラメータの共有
- ウェブシナリオを share.zabbix.com で共有する - Web シナリオを含むテンプレートをエクスポートし、share.zabbix.com にアップロードします。その後、他のユーザはテンプレートをダウンロードし、そのファイルを Zabbix にインポートすることができます。
- サードパーティツールとの統合 - 汎用的な YAML、XML、JSON 形式を使用することで、データのインポート/エクスポートを行ってサードパーティのツールやアプリケーションと統合することができます。

エクスポート/インポートできるもの

エクスポート/インポートできるオブジェクトは次のとおりです。

- **ホストグループ** (Zabbix API 経由のみ)
- **テンプレート**
- **ホスト**
- **ネットワークマップ**
- **メディアタイプ**
- images

エクスポート形式

Zabbix の Web インターフェースまたは**Zabbix API**を使用してデータをエクスポートすることができます。
 サポートされているエクスポート形式は YAML、XML、JSON です。

エクスポートに関する詳細

- サポートされているすべての要素が1つのファイルにエクスポートされます。

- ホストおよびテンプレートの要素（アイテム、トリガー、グラフ、ディスカバリールール）は、
 リンクされたテンプレートから継承されるため、エクスポートされません。
 ホストレベルでこれらのエンティティに加えられたすべての変更（アイテムの監視間隔の変更、正規表現の変更、
 ローレベルディスカバリールールへのプロトタイプの追加など）は、エクスポート時に失われます。
 インポートする場合、リンクされたテンプレートのすべての要素は、元のリンクされたテンプレートと同じように再作成されます。

- ローレベルディスカバリによって作成された要素およびそれに依存する要素は、エクスポートされません。
 例えば、LLDルールが生成された項目に対して作成されたトリガーはエクスポートされません。

インポートに関する詳細

- インポートは、最初のエラーで停止します。
- 画像のインポート時に、既存の画像を更新する場合、“imagetype” フィールドは無視されます。つまり、インポートを介してイメージタイプを変更することはできません。
- ホスト/テンプレートを”存在しない場合に削除” オプションでインポートする場合、インポートファイルに存在しないホスト/テンプレートマクロはインポート後にホスト/テンプレートから削除されます。
- アイテム、トリガー、グラフ、ホスト/テンプレート、アプリケーション、ディスカバリールール、アイテムプロトタイプ、トリガープロトタイプ、グラフプロトタイプでは、空のタグには意味がありません。つまり、存在しないのと同じです。他のタグ、例えば、アイテムアプリケーションのタグは意味があります。タグが空の場合はアイテムにアプリケーションが無いこと、タグが無い場合はアプリケーションを更新しないことを意味します。
- インポートは、YAML、XML、JSON をサポートしています。YAML は.yaml と.yml、XML は.xml、JSON は.json と、正しいファイル拡張子が必要です。サポートされている XML のバージョンに関しては、[バージョン間の互換性](#)を参照してください。
- インポートは、UTF-8 エンコーディング (BOM 付き/なしいずれも対応) の設定ファイルのみをサポートしています。他のエンコーディング (UTF16LE, UTF16BE, UTF32LE, UTF32BE 等) を使用した場合、変換エラーが生じます。

```
zabbix_export:  
  version: '6.0'  
  date: '2020-04-22T06:20:11Z'
```

YAML ペースフォーマット

```
zabbix_export:
```

Zabbix で YAML 形式でエクスポートした際のルートです。

```
version: '6.0'
```

エクスポートしたバージョンです。

```
date: '2020-04-22T06:20:11Z'
```

日付は、ISO 8601 ロングフォーマットで、エクスポートが作成された日付です。

その他のノードは、エクスポートしたオブジェクトに依存します。

```
<?xml version="1.0" encoding="UTF-8"?>  
<zabbix_export>  
  <version>6.0</version>  
  <date>2020-04-22T06:20:11Z</date>  
</zabbix_export>
```

XML フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
```

XML ドキュメントのデフォルトヘッダです。

```
<zabbix_export>
```

Zabbix で XML 形式でエクスポートした際のルートの要素です。

```
<version>6.0</version>
```

エクスポートしたバージョンです。

```
<date>2020-04-22T06:20:11Z</date>
```

日付は ISO 8601 ロングフォーマットでエクスポートが作成された日付です。

その他のタグは、エクスポートしたオブジェクトに依存します。

```
{
  "zabbix_export": {
    "version": "6.0",
    "date": "2020-04-22T06:20:11Z"
  }
}
```

JSON フォーマット

```
"zabbix_export":
```

Zabbix で JSON 形式でエクスポートした際のルートです。

```
"version": "6.0"
```

エクスポートしたバージョンです。

```
"date": "2020-04-22T06:20:11Z"
```

日付は ISO 8601 ロングフォーマットでエクスポートが作成された日付です。

その他のノードは、エクスポートしたオブジェクトに依存します。

1 ホストグループ

フロントエンドのホストグループは、ホストかテンプレートと一緒に **export** することができます。
 ホストやテンプレートがエクスポートされる時それが属しているすべてのグループは自動的に export されます。

API を使用すると、ホストやテンプレートから独立してホストグループを export することができます。

エクスポートフォーマット

```
groups:
- name: 'Zabbix servers'
```

groups/group

パラメータ	型	説明	詳細
name	文字列	グループ名	

2 テンプレート

概要

テンプレートは、多くの関連するオブジェクトと一緒に **エクスポート** されます。

テンプレートのエクスポートには以下の要素が含まれます:

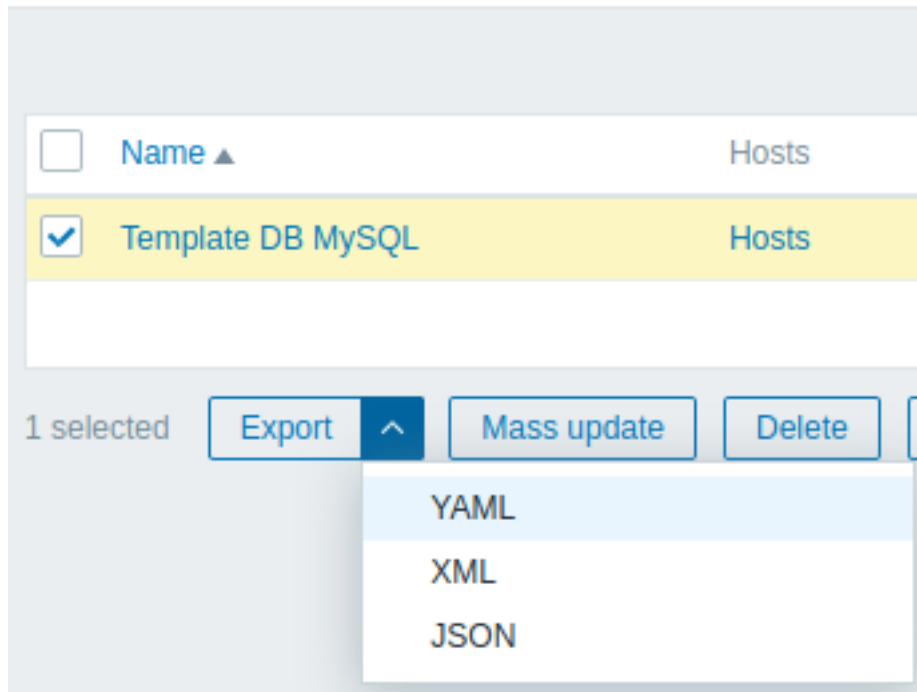
- リンクされたホストグループ
- テンプレートデータ
- 他のテンプレートとのリンク
- ホストグループへのリンク
- 直接リンクされたアイテム
- 直接リンクされたトリガー
- 直接リンクされたグラフ
- 直接リンクされたダッシュボード
- すべてのプロトタイプと直接リンクされたディスクパルルール
- 直接リンクされた Web シナリオ
- バリュemap

エクスポート

テンプレートをエクスポートするには、次の操作を行います。

- 設定 → テンプレート
- エクスポートするテンプレートのチェックボックスにマークを付けます。
- リストの下にあるエクスポートをクリックします。

≡ Templates



選択した形式に応じて、テンプレートはローカルにデフォルトファイル名でエクスポートされます。

- zabbix_export_templates.yaml - YAML エクスポートの場合 (エクスポートのデフォルトオプション)
- zabbix_export_templates.xml - XML エクスポートの場合
- zabbix_export_templates.json - JSON エクスポートの場合

インポート

テンプレートをインポートするには、次の操作を行います。

- 設定 → テンプレートに移動します。
- 右側のインポートをクリックします。
- インポートファイルを選択します。
- インポートルールで必要なオプションをマークします。
- インポートをクリックします。

Import ✕

* Import file Choose file template_power_apc_ups_snmp.xml

Rules	Update existing	Create new	Delete missing
Groups	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Templates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Value mappings	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Template dashboards	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Template linkage		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Items	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Discovery rules	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Triggers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Graphs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Web scenarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Import
Cancel

すべての必須入力項目には赤いアスタリスクが表示されます。

インポートルール:

ルール	説明
Update existing	既存の要素は、インポートファイルから取得したデータで更新されます。そうでない場合は、更新されません。
Create new	インポートファイルのデータを使用して新しい要素を追加します。そうでない場合は、追加されません。
Delete missing	インポートファイルに存在しない既存の要素は削除されます。そうでない場合は、削除されません。テンプレート・リンクに Delete missing が指定されている場合、インポート・ファイルに存在しない既存のテンプレート・リンクは、リンクされていない可能性のあるテンプレートから継承されたすべてのエンティティ（アイテム、トリガーなど）と共にテンプレートから削除されます。

次の画面では、インポートされたテンプレートの内容を確認することができます。新しいテンプレートであれば、すべての要素が緑色で表示されます。既存のテンプレートを更新する場合、新しいテンプレート要素は緑色でハイライトされ、削除されたテンプレート要素は赤色で、変更のない要素は灰色の背景で表示されます。

Templates

The screenshot shows a web interface on the left and a code editor on the right. The web interface has a sidebar menu with 'Updated' selected, containing 'Templates' and 'VMware'. The code editor shows a JSON snippet for a template named 'VMware' with various fields like 'name', 'description', 'groups', 'tags', and 'macros'.

```
templates:
  template: VMware
- name: VMware
+ name: 'VMware alternative'
- description: "You can discuss this template or leave feedback on our forum"
+ description: "You can discuss this fabulous template or leave feedback on our forum"
  groups:
    - name: Templates/Applications
  tags:
    - tag: class
      value: software
    - tag: target
      value: vmware
  macros:
    - macro: '{$VMWARE.PASSWORD}'
      description: 'VMware service {USERNAME} user password'
    - macro: '{$VMWARE.URL}'
      description: 'VMware service (vCenter or ESX hypervisor) SDK URL (https://...)'
    - macro: '{$VMWARE.USERNAME}'
      description: 'VMware service user name'
```

左のメニューから、変更点の一覧を表示することができます。セクション Updated は、既存のテンプレート要素に加えられたすべての変更点をハイライト表示します。セクション Added は、新しいテンプレート要素をリストアップします。各セクションの要素は、要素の種類によってグループ化されています。グレーの矢印を押して、要素のグループを展開または折りたたみます。

The screenshot shows the 'Templates' web interface. The 'Updated' section is expanded, showing a list of items: 'Templates', 'Items', 'Triggers', 'Discovery rules', 'Item prototypes', 'Dashboards', 'UPS Summary', 'Graphs', and 'Capacity of the UPS batteries'. The 'Added' section is also expanded, showing a list of items: 'Items', 'Battery capacity', and 'System name'.

テンプレートの変更を確認し、インポートを押してテンプレートのインポートを実行します。フロントエンドにインポートの成功または失敗のメッセージが表示されます。

エクスポートフォーマット

YAML でのエクスポートフォーマット:

```
zabbix_export:
  version: '6.0'
  date: '2021-08-31T12:40:55Z'
  groups:
    -
      uuid: a571c0d144b14fd4a87a9d9b2aa9fcd6
      name: Templates/Applications
  templates:
    -
      uuid: 56079badd056419383cc26e6a4fcc7e0
      template: VMware
      name: VMware
      description: |
        You can discuss this template or leave feedback on our forum https://www.zabbix.com/forum/zabbix-s

        Template tooling version used: 0.38
  templates:
    -
      name: 'VMware macros'
  groups:
    -
      name: Templates/Applications
  items:
    -
      uuid: 5ce209f4d94f460488a74a92a52d92b1
      name: 'VMware: Event log'
      type: SIMPLE
      key: 'vmware.eventlog[{$VMWARE.URL},skip]'
      history: 7d
      trends: '0'
      value_type: LOG
      username: '{$VMWARE.USERNAME}'
      password: '{$VMWARE.PASSWORD}'
      description: 'Collect VMware event log. See also: https://www.zabbix.com/documentation/6.0/manual'
      tags:
        -
          tag: Application
          value: VMware
    -
      uuid: ee2edad88ce943ef81d25dbbba8667a4
      name: 'VMware: Full name'
      type: SIMPLE
      key: 'vmware.fullname[{$VMWARE.URL}]'
      delay: 1h
      history: 7d
      trends: '0'
      value_type: CHAR
      username: '{$VMWARE.USERNAME}'
      password: '{$VMWARE.PASSWORD}'
      description: 'VMware service full name.'
      preprocessing:
        -
          type: DISCARD_UNCHANGED_HEARTBEAT
          parameters:
            - 1d
      tags:
        -
          tag: Application
          value: VMware
```

```

    uuid: a0ec9145f2234fbea79a28c57ebdb44d
    name: 'VMware: Version'
    type: SIMPLE
    key: 'vmware.version[{$VMWARE.URL}]'
    delay: 1h
    history: 7d
    trends: '0'
    value_type: CHAR
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'VMware service version.'
    preprocessing:
      -
        type: DISCARD_UNCHANGED_HEARTBEAT
        parameters:
          - 1d
    tags:
      -
        tag: Application
        value: VMware
discovery_rules:
  -
    uuid: 16ffc933cce74cf28a6edf306aa99782
    name: 'Discover VMware clusters'
    type: SIMPLE
    key: 'vmware.cluster.discovery[{$VMWARE.URL}]'
    delay: 1h
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'Discovery of clusters'
    item_prototypes:
      -
        uuid: 46111f91dd564a459dbc1d396e2e6c76
        name: 'VMware: Status of "{#CLUSTER.NAME}" cluster'
        type: SIMPLE
        key: 'vmware.cluster.status[{$VMWARE.URL},{#CLUSTER.NAME}]'
        history: 7d
        username: '{$VMWARE.USERNAME}'
        password: '{$VMWARE.PASSWORD}'
        description: 'VMware cluster status.'
        valuemap:
          name: 'VMware status'
        tags:
          -
            tag: Application
            value: VMware
  -
    uuid: 8fb6a45cbe074b0cb6df53758e2c6623
    name: 'Discover VMware datastores'
    type: SIMPLE
    key: 'vmware.datastore.discovery[{$VMWARE.URL}]'
    delay: 1h
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    item_prototypes:
      -
        uuid: 4b61838ba4c34e709b25081ae5b059b5
        name: 'VMware: Average read latency of the datastore {#DATASTORE}'
        type: SIMPLE
        key: 'vmware.datastore.read[{$VMWARE.URL},{#DATASTORE},latency]'
        history: 7d
        username: '{$VMWARE.USERNAME}'

```

```

password: '{$VMWARE.PASSWORD}'
description: 'Amount of time for a read operation from the datastore (milliseconds).'
```

- - tag: Application
 - value: VMware

```

uuid: 5355c401dc244bc588ccd18767577c93
name: 'VMware: Free space on datastore {#DATASTORE} (percentage)'
```

- - type: SIMPLE
 - key: 'vmware.datastore.size[{\$VMWARE.URL},{#DATASTORE},pfree]'
 - delay: 5m
 - history: 7d
 - value_type: FLOAT
 - units: '%'
 - username: '{\$VMWARE.USERNAME}'
 - password: '{\$VMWARE.PASSWORD}'
 - description: 'VMware datastore space in percentage from total.'
 - tags:
 - - tag: Application
 - value: VMware

```

uuid: 84f13c4fde2d4a17baaf0c8c1eb4f2c0
name: 'VMware: Total size of datastore {#DATASTORE}'
```

- - type: SIMPLE
 - key: 'vmware.datastore.size[{\$VMWARE.URL},{#DATASTORE}]'
 - delay: 5m
 - history: 7d
 - units: B
 - username: '{\$VMWARE.USERNAME}'
 - password: '{\$VMWARE.PASSWORD}'
 - description: 'VMware datastore space in bytes.'
 - tags:
 - - tag: Application
 - value: VMware

```

uuid: 540cd0fbc56c4b8ea19f2ff5839ce00d
name: 'VMware: Average write latency of the datastore {#DATASTORE}'
```

- - type: SIMPLE
 - key: 'vmware.datastore.write[{\$VMWARE.URL},{#DATASTORE},latency]'
 - history: 7d
 - username: '{\$VMWARE.USERNAME}'
 - password: '{\$VMWARE.PASSWORD}'
 - description: 'Amount of time for a write operation to the datastore (milliseconds).'
 - tags:
 - - tag: Application
 - value: VMware

```

uuid: a5bc075e89f248e7b411d8f960897a08
name: 'Discover VMware hypervisors'
```

- - type: SIMPLE
 - key: 'vmware.hv.discovery[{\$VMWARE.URL}]'
 - delay: 1h
 - username: '{\$VMWARE.USERNAME}'
 - password: '{\$VMWARE.PASSWORD}'
 - description: 'Discovery of hypervisors.'
 - host_prototypes:
 - - uuid: 051a1469d4d045cbbf818fcc843a352e

```

    host: '#{HV.UUID}'
    name: '#{HV.NAME}'
    group_links:
    -
      group:
        name: Templates/Applications
    group_prototypes:
    -
      name: '#{CLUSTER.NAME}'
    -
      name: '#{DATACENTER.NAME}'
    templates:
    -
      name: 'VMware Hypervisor'
    macros:
    -
      macro: '{$VMWARE.HV.UUID}'
      value: '#{HV.UUID}'
      description: 'UUID of hypervisor.'
    custom_interfaces: 'YES'
    interfaces:
    -
      ip: '#{HV.IP}'
  -
    uuid: 9fd559f4e88c4677a1b874634dd686f5
    name: 'Discover VMware VMs'
    type: SIMPLE
    key: 'vmware.vm.discovery[{$VMWARE.URL}]'
    delay: 1h
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'Discovery of guest virtual machines.'
    host_prototypes:
    -
      uuid: 23b9ae9d6f33414880db1cb107115810
      host: '#{VM.UUID}'
      name: '#{VM.NAME}'
      group_links:
      -
        group:
          name: Templates/Applications
      group_prototypes:
      -
        name: '#{CLUSTER.NAME} (vm)'
      -
        name: '#{DATACENTER.NAME}/#{VM.FOLDER} (vm)'
      -
        name: '#{HV.NAME}'
      templates:
      -
        name: 'VMware Guest'
      macros:
      -
        macro: '{$VMWARE.VM.UUID}'
        value: '#{VM.UUID}'
        description: 'UUID of guest virtual machine.'
      custom_interfaces: 'YES'
      interfaces:
      -
        ip: '#{VM.IP}'
  valuemaps:
  -

```

```

uuid: 3c59c22905054d42ac4ee8b72fe5f270
name: 'VMware status'
mappings:
-
  value: '0'
  newvalue: gray
-
  value: '1'
  newvalue: green
-
  value: '2'
  newvalue: yellow
-
  value: '3'
  newvalue: red

```

Element タグ

Element タグの値については、以下の表で説明する。

Template タグ

Element	Element プロパティ	必須?	型	レンジ	説明
templates	uuid	x	string		Root element for templates. Unique identifier for this template.
	template	x	string		Unique template name.
	name	-	string		Visible template name.
	description	-	text		Template description.
groups		x			Root element for template host groups.
	uuid	x	string		Unique identifier for this host group.
templates	name	x	string		Host group name.
		-			Root element for linked templates.
tags	name	x	string		Template name.
		-			Root element for template tags.
macros	tag	x	string		Tag name.
	value	-	string		Tag value.
		-			Root element for template user macros.
macros	macro	x	string		User macro name.
	type	-	string	0 - TEXT (default) 1 - SE-CRET_TEXT 2 - VAULT	Type of the macro.
	value	-	string		User macro value.
valuemaps	description	-	string		User macro description.
		-			Root element for template value maps.
	uuid	x	string		Unique identifier for this value map.
valuemaps	name	x	string		Value map name.
	mapping	-			Root element for mappings.
	value	x	string		Value of a mapping.

Element	Element プロパティ	必須?	型	レンジ	説明
	newvalue	x	string		New value of a mapping.

Template item タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
items		-			Root element for items.
	uuid	x	string		Unique identifier for the item.
	name	x	string		Item name.
	type	-	string	0 - ZABBIX_PASSIVE (default) 2 - TRAP 3 - SIMPLE 5 - INTERNAL 7 - ZABBIX_ACTIVE 10 - EXTERNAL 11 - ODBC 12 - IPMI 13 - SSH 14 - TELNET 15 - CALCULATED 16 - JMX 17 - SNMP_TRAP 18 - DEPENDENT 19 - HTTP_AGENT 20 - SNMP_AGENT 21 - ITEM_TYPE_SCRIPT	Item type.
	snmp_oid	-	string		SNMP object ID.
	key	x	string		Required by SNMP items. Item key.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	delay	-	string	Default: 1m	Update interval of the item. Accepts seconds or a time unit with suffix (30s, 1m, 2h, 1d). Optionally one or more custom intervals can be specified either as flexible intervals or scheduling. Multiple intervals are separated by a semicolon. User macros may be used. A single macro has to fill the whole field. Multiple macros in a field or macros mixed with text are not supported. Flexible intervals may be written as two macros separated by a forward slash (e.g. <code>{\$FLEX_INTERVAL}/{\$FLEX_INTERVAL}</code>).
	history	-	string	Default: 90d	A time unit of how long the history data should be stored. Time unit with suffix, user macro or LLD macro.
	trends	-	string	Default: 365d	A time unit of how long the trends data should be stored. Time unit with suffix, user macro or LLD macro.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Item status.
	value_type	-	string	0 - FLOAT 1 - CHAR 2 - LOG 3 - UNSIGNED (default) 4 - TEXT	Received value type.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	allowed_hosts	-	string		List of IP addresses (comma delimited) of hosts allowed sending data for the item.
	units	-	string		Used by trapper and HTTP agent items. Units of returned values (bps, B, etc).
	params	-	text		Additional parameters depending on the type of the item: - executed script for Script, SSH and Telnet items; - SQL query for database monitor items; - formula for calculated items.
	ipmi_sensor	-	string		IPMI sensor.
	authtype	-	string	Authentication type for SSH agent items: 0 - PASSWORD (default) 1 - PUBLIC_KEY Authentication type for HTTP agent items: 0 - NONE (default) 1 - BASIC 2 - NTLM	Used only by IPMI items. Authentication type. Used only by SSH and HTTP agent items.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	username	-	string		Username for authentication. Used by simple check, SSH, Telnet, database monitor, JMX and HTTP agent items. Required by SSH and Telnet items. When used by JMX agent, password should also be specified together with the username or both properties should be left blank.
	password	-	string		Password for authentication. Used by simple check, SSH, Telnet, database monitor, JMX and HTTP agent items. When used by JMX agent, username should also be specified together with the password or both properties should be left blank.
	publickey	-	string		Name of the public key file. Required for SSH agent items.
	privatekey	-	string		Name of the private key file. Required for SSH agent items.
	port	-	string		Custom port monitored by the item. Can contain user macros.
	description	-	text		Used only by SNMP items. Item description.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	inventory_link	-	string	0 - NONE Capitalized host inventory field name. For example: 4 - ALIAS 6 - OS_FULL 14 - HARDWARE etc.	Host inventory field that is populated by the item. Refer to the host inventory page for a list of supported host inventory fields and their IDs.
	logtimefmt	-	string		Format of the time in log entries. Used only by log items.
	jmx_endpoint	-	string		JMX endpoint. Used only by JMX agent items.
	url	-	string		URL string. Required only for HTTP agent items.
	allow_traps	-	string	0 - NO (default) 1 - YES	Allow to populate value as in a trapper item. Used only by HTTP agent items.
	follow_redirects	-	string	0 - NO 1 - YES (default)	Follow HTTP response redirects while pooling data. Used only by HTTP agent items.
headers		-			Root element for HTTP(S) request headers, where header name is used as key and header value as value. Used only by HTTP agent items.
	name	x	string		Header name.
	value	x	string		Header value.
	http_proxy	-	string		HTTP(S) proxy connection string. Used only by HTTP agent items.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	output_format	-	string	0 - RAW (default) 1 - JSON	How to process response. Used only by HTTP agent items.
	post_type	-	string	0 - RAW (default) 2 - JSON 3 - XML	Type of post data body. Used only by HTTP agent items.
	posts	-	string		HTTP(S) request body data.
query_fields		-			Used only by HTTP agent items. Root element for query parameters.
	name	x	string		Used only by HTTP agent items. Parameter name.
	value	-	string		Parameter value.
	request_method	-	string	0 - GET (default) 1 - POST 2 - PUT 3 - HEAD	Request method. Used only by HTTP agent items.
	retrieve_mode	-	string	0 - BODY (default) 1 - HEADERS 2 - BOTH	What part of response should be stored.
	ssl_cert_file	-	string		Used only by HTTP agent items. Public SSL Key file path.
	ssl_key_file	-	string		Used only by HTTP agent items. Private SSL Key file path.
	ssl_key_password	-	string		Used only by HTTP agent items. Password for SSL Key file.
					Used only by HTTP agent items.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	status_codes	-	string		<p>Ranges of required HTTP status codes separated by commas. Supports user macros. Example: 200,200-{\$M},{ \$M},200-400</p> <p>Used only by HTTP agent items.</p>
	timeout	-	string		<p>Item data polling request timeout. Supports user macros.</p>
	verify_host	-	string	0 - NO (default) 1 - YES	<p>Used by HTTP agent and Script items. Validate if host name in URL is in Common Name field or a Subject Alternate Name field of host certificate.</p>
	verify_peer	-	string	0 - NO (default) 1 - YES	<p>Used only by HTTP agent items. Validate if host certificate is authentic.</p>
	parameters	-			<p>Used only by HTTP agent items. Root element for user-defined parameters.</p>
	name	x	string		<p>Used only by Script items. Parameter name.</p>
	value	-	string		<p>Used only by Script items. Parameter value.</p>
	value map	-			<p>Used only by Script items. Value map.</p>

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	name	x	string		Name of the value map to use for the item.
preprocessing		-			Root element for item value preprocessing.
step		-			Individual item value preprocessing step.
	type	x	string	1 - MULTIPLIER 2 - RTRIM 3 - LTRIM 4 - TRIM 5 - REGEX 6 - BOOL_TO_DECIMAL 7 - OCTAL_TO_DECIMAL 8 - HEX_TO_DECIMAL 9 - SIMPLE_CHANGE (calculated as (received value-previous value)) 10 - CHANGE_PER_SECOND (calculated as (received value-previous value)/(time now-time of last check)) 11 - XMLPATH 12 - JSONPATH 13 - IN_RANGE 14 - MATCHES_REGEX 15 - NOT_MATCHES_REGEX 16 - CHECK_JSON_ERROR 17 - CHECK_XML_ERROR 18 - CHECK_REGEX_ERROR 19 - DISCARD_UNCHANGED 20 - DISCARD_UNCHANGED_HEARTBEAT 21 - JAVASCRIPT 22 - PROMETHEUS_PATTERN 23 - PROMETHEUS_TO_JSON 24 - CSV_TO_JSON 25 - STR_REPLACE 26 - CHECK_NOT_SUPPORTED	Type of the item value preprocessing step.
	parameters	-			Root element for parameters of the item value preprocessing step.
	parameter	x	string		Individual parameter of the item value preprocessing step.
	error_handler	-	string	0 - ORIGINAL_ERROR (default) 1 - DISCARD_VALUE 2 - CUSTOM_VALUE 3 - CUSTOM_ERROR	Action type used in case of preprocessing step failure.
	error_handler_params	-	string		Error handler parameters used with 'error_handler'.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
master_item		-			Individual item master item.
	key	x	string		Required by dependent items. Dependent item master item key value.
triggers		-			Recursion up to 3 dependent items and maximum count of dependent items equal to 29999 are allowed. Root element for simple triggers.
	For trigger element tag values, see template trigger tags .				
tags		-			Root element for item tags.
	tag	x	string		Tag name.
	value	-	string		Tag value.

Template low-level discovery rule タグ

Element	Element プロパティ	必須?	型	レンジ	説明
discovery_rules		-			Root element for low-level discovery rules.
	For most of the element tag values, see element tag values for a regular item. Only the tags that are specific to low-level discovery rules, are described below.				

Element	Element プロパティ	必須?	型	レンジ	説明
filter	type	-	string	0 - ZAB-BIX_PASSIVE (default) 2 - TRAP 3 - SIMPLE 5 - INTERNAL 7 - ZAB-BIX_ACTIVE 10 - EXTERNAL 11 - ODBC 12 - IPMI 13 - SSH 14 - TELNET 16 - JMX 18 - DEPENDENT 19 - HTTP_AGENT 20 - SNMP_AGENT	Item type.
	lifetime	-	string	Default: 30d	Time period after which items that are no longer discovered will be deleted. Seconds, time unit with suffix or user macro.
	evaltype	-	string	0 - AND_OR (default) 1 - AND 2 - OR 3 - FORMULA	Individual filter. Logic to use for checking low-level discovery rule filter conditions.
	formula	-	string		Custom calculation formula for filter conditions.
	conditions	-			Root element for filter conditions.
	macro	x	string		Low-level discovery macro name.
	value	-	string		Filter value: regular expression or global regular expression.
conditions	operator	-	string	8 - MATCHES_REGEX (default) 9 - NOT_MATCHES_REGEX	Condition operator.

Element	Element プロパティ	必須?	型	レンジ	説明
	formulaid	x	character		Arbitrary unique ID that is used to reference a condition from the custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward.
lld_macro_paths		-			Root element for LLD macro paths.
	lld_macro	x	string		Low-level discovery macro name.
	path	x	string		Selector for value which will be assigned to the corresponding macro.
preprocessing		-			LLD rule value preprocessing.
step		-			Individual LLD rule value preprocessing step.
	type	x	string	5 - REGEX 11 - XMLPATH 12 - JSONPATH 15 - NOT_MATCHES_REGEX 16 - CHECK_JSON_ERROR 17 - CHECK_XML_ERROR 20 - DIS-CARD_UNCHANGED_HEARTBEAT 21 - JAVASCRIPT 23 - PROMETHEUS_TO_JSON 24 - CSV_TO_JSON 25 - STR_REPLACE	For most of the element tag values, see element tag values for a template item value preprocessing. Only the tags that are specific to template low-level discovery value preprocessing, are described below. Type of the item value preprocessing step.
trigger_prototypes		-			Root element for trigger prototypes.
					For trigger prototype element tag values, see regular template trigger tags.
graph_prototypes		-			Root element for graph prototypes.

Element	Element プロパティ	必須?	型	レンジ	説明
	For graph prototype element tag values, see regular template graph tags.				
host_prototypes		-			Root element for host prototypes.
	For host prototype element tag values, see regular host tags.				
item_prototypes		-			Root element for item prototypes.
	For item prototype element tag values, see regular template item tags.				
master_item		-			Individual item prototype master item/item prototype data.
	key	x	string		Dependent item prototype master item/item prototype key value.
					Required for a dependent item.

Template trigger タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
triggers		-			Root element for triggers.
	uuid	x	string		Unique identifier for this trigger.
	expression	x	string		Trigger expression.
	recovery_mode	-	string	0 - EXPRESSION (default) 1 - RECOVERY_EXPRESSION 2 - NONE	Basis for generating OK events.
	recovery_expression	-	string		Trigger recovery expression.
	name	x	string		Trigger name.
	correlation_mode	-	string	0 - DISABLED (default) 1 - TAG_VALUE	Correlation mode (no event correlation or event correlation by tag).
	correlation_tag	-	string		The tag name to be used for event correlation.
	url	-	string		URL associated with the trigger.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Trigger status.
	priority	-	string	0 - NOT_CLASSIFIED (default) 1 - INFO 2 - WARNING 3 - AVERAGE 4 - HIGH 5 - DISASTER	Trigger severity.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	description	-	text		Trigger description.
	type	-	string	0 - SINGLE (default) 1 - MULTIPLE	Event generation type (single problem event or multiple problem events).
	manual_close	-	string	0 - NO (default) 1 - YES	Manual closing of problem events.
dependencies		-			Root element for dependencies.
	name	x	string		Dependency trigger name.
	expression	x	string		Dependency trigger expression.
	recovery_expression	-	string		Dependency trigger recovery expression.
tags		-			Root element for trigger tags.
	tag	x	string		Tag name.
	value	-	string		Tag value.

Template graph タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
graphs		-			Root element for graphs.
	uuid	x	string		Unique identifier for this graph.
	name	x	string		Graph name.
	width	-	integer	20-65535 (default: 900)	Graph width, in pixels. Used for preview and for pie/exploded graphs.
	height	-	integer	20-65535 (default: 200)	Graph height, in pixels. Used for preview and for pie/exploded graphs.
	yaxismin	-	double	Default: 0	Value of Y axis minimum.
	yaxismax	-	double	Default: 0	Value of Y axis maximum.
					Used if 'ymin_type_1' is FIXED.
					Used if 'ymax_type_1' is FIXED.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	show_work_period	-	string	0 - NO 1 - YES (default)	Highlight non-working hours.
	show_triggers	-	string	0 - NO 1 - YES (default)	Used by normal and stacked graphs. Display simple trigger values as a line.
	type	-	string	0 - NORMAL (default) 1 - STACKED 2 - PIE 3 - EXPLODED	Used by normal and stacked graphs. Graph type.
	show_legend	-	string	0 - NO 1 - YES (default)	Display graph legend.
	show_3d	-	string	0 - NO (default) 1 - YES	Enable 3D style.
	percent_left	-	double	Default:0	Used by pie and exploded pie graphs. Show the percentile line for left axis.
	percent_right	-	double	Default:0	Used only for normal graphs. Show the percentile line for right axis.
	ymin_type_1	-	string	0 - CALCULATED (default) 1 - FIXED 2 - ITEM	Used only for normal graphs. Minimum value of Y axis.
	ymax_type_1	-	string	0 - CALCULATED (default) 1 - FIXED 2 - ITEM	Used by normal and stacked graphs. Maximum value of Y axis.
	ymin_item_1	-			Used by normal and stacked graphs. Individual item details.
	host	x	string		Required if 'ymin_type_1' is ITEM. Item host.
	key	x	string		Item key.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
ymax_item_1		-			Individual item details. Required if 'ymax_type_1' is ITEM.
graph_items	host	x	string		Item host.
	key	x	string		Item key.
	sortorder	-	integer		Root element for graph items. Draw order. The smaller value is drawn first. Can be used to draw lines or regions behind (or in front of) another.
	drawtype	-	string	0 - SINGLE_LINE (default) 1 - FILLED_REGION 2 - BOLD_LINE 3 - DOTTED_LINE 4 - DASHED_LINE 5 - GRADIENT_LINE	Draw style of the graph item. Used only by normal graphs.
	color	-	string		Element color (6 symbols, hex).
	yaxisside	-	string	0 - LEFT (default) 1 - RIGHT	Side of the graph where the graph item's Y scale will be drawn.
	calc_fnc	-	string	1 - MIN 2 - AVG (default) 4 - MAX 7 - ALL (minimum, average and maximum; used only by simple graphs) 9 - LAST (used only by pie and exploded pie graphs)	Used by normal and stacked graphs. Data to draw if more than one value exists for an item.
	type	-	string	0 - SIMPLE (default) 2 - GRAPH_SUM (value of the item represents the whole pie; used only by pie and exploded pie graphs)	Graph item type.
item		x			Individual item.
	host	x	string		Item host.
	key	x	string		Item key.

Template web シナリオタグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
httpstests		-			Root element for web scenarios.
	uuid	x	string		Unique identifier for this web scenario.
	name	x	string		Web scenario name.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	delay	-	string	Default: 1m	Frequency of executing the web scenario. Seconds, time unit with suffix or user macro.
	attempts	-	integer	1-10 (default: 1)	The number of attempts for executing web scenario steps.
	agent	-	string	Default: Zabbix	Client agent. Zabbix will pretend to be the selected browser. This is useful when a website returns different content for different browsers.
	http_proxy	-	string		Specify an HTTP proxy to use, using the format: http://[username[:password@]hostname[:port]]
variables		-			Root element for scenario-level variables (macros) that may be used in scenario steps.
	name	x	text		Variable name.
	value	x	text		Variable value.
headers		-			Root element for HTTP headers that will be sent when performing a request. Headers should be listed using the same syntax as they would appear in the HTTP protocol.
	name	x	text		Header name.
	value	x	text		Header value.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Web scenario status.
	authentication	-	string	0 - NONE (default) 1 - BASIC 2 - NTLM	Authentication method.
	http_user	-	string		User name used for basic, HTTP or NTLM authentication.
	http_password	-	string		Password used for basic, HTTP or NTLM authentication.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	verify_peer	-	string	0 - NO (default) 1 - YES	Verify the SSL certificate of the web server.
	verify_host	-	string	0 - NO (default) 1 - YES	Verify that the Common Name field or the Subject Alternate Name field of the web server certificate matches.
	ssl_cert_file	-	string		Name of the SSL certificate file used for client authentication (must be in PEM format).
	ssl_key_file	-	string		Name of the SSL private key file used for client authentication (must be in PEM format).
	ssl_key_password	-	string		SSL private key file password.
steps		x			Root element for web scenario steps.
	name	x	string		Web scenario step name.
	url	x	string		URL for monitoring.
query_fields		-			Root element for query fields - an array of HTTP fields that will be added to the URL when performing a request.
	name	x	string		Query field name.
	value	-	string		Query field value.
posts		-			HTTP POST variables as a string (raw post data) or as an array of HTTP fields (form field data).
	name	x	string		Post field name.
	value	x	string		Post field value.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
variables		-			Root element of step-level variables (macros) that should be applied after this step. If the variable value has a 'regex:' prefix, then its value is extracted from the data returned by this step according to the regular expression pattern following the 'regex:' prefix
	name	x	string		Variable name.
	value	x	string		Variable value.
headers		-			Root element for HTTP headers that will be sent when performing a request. Headers should be listed using the same syntax as they would appear in the HTTP protocol.
	name	x	string		Header name.
	value	x	string		Header value.
	follow_redirects	-	string	0 - NO 1 - YES (default)	Follow HTTP redirects.
	retrieve_mode	-	string	0 - BODY (default) 1 - HEADERS 2 - BOTH	HTTP response retrieve mode.
	timeout	-	string	Default: 15s	Timeout of step execution. Seconds, time unit with suffix or user macro.
	required	-	string		Text that must be present in the response. Ignored if empty.
	status_codes	-	string		A comma delimited list of accepted HTTP status codes. Ignored if empty. For example: 200-201,210-299
tags		-			Root element for web scenario tags.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	tag	x	string		Tag name.
	value	-	string		Tag value.

Template dashboard タグ

Element	Element プロパティ	必須	型	レンジ ¹	説明
dashboards		-			Root element for template dashboards.
	uuid	x	string		Unique identifier for this dashboard.
	name	x	string		Template dashboard name.
	display period	-	integer		Display period of dashboard pages.
	auto_start	-	string	0 - no 1 - yes	Slideshow auto start.
pages		-			Root element for template dashboard pages.
	name	-	string		Page name.
	display period	-	integer		Page display period.
	sortorder	-	integer		Page sorting order.
widgets		-			Root element for template dashboard widgets.
	type	x	string		Widget type.
	name	-	string		Widget name.
	x	-	integer	0-23	Horizontal position from the left side of the template dashboard.
	y	-	integer	0-62	Vertical position from the top of the template dashboard.
	width	-	integer	1-24	Widget width.
	height	-	integer	2-32	Widget height.
	hide_header	-	string	0 - no 1 - yes	Hide widget header.
fields		-			Root element for the template dashboard widget fields.
	type	x	string	0 - INTEGER 1 - STRING 3 - HOST 4 - ITEM 5 - ITEM_PROTOTYPE 6 - GRAPH 7 - GRAPH_PROTOTYPE	Widget field type.

Element	Element プロパティ	必須	型	レンジ ¹	説明
	name	x	string		Widget field name.
	value	x	mixed		Widget field value, depending on the field type.

脚注

文字列値の場合は、番号付けは行われず、文字列のみがエクスポートされます (例: 「ZABBIX_ACTIVE」)。
 このテーブルの範囲値 (API 値に対応) の番号は、順序付けにのみ使用されます。

3 ホスト

概要

ホストは、多くの関連するオブジェクトと **export** されます。
 オブジェクトとオブジェクトの関係が含まれます。

ホストのエクスポートには以下の内容が含まれます。
 - リンクされたホストグループ - ホストデータ - テンプレート・リンケージ - ホストグループリンケージ - ホストインターフェース - 直接リンクされた項目 - 直接リンクされたトリガー - 直接リンクされたグラフ - すべてのプロトタイプに直接リンクされたディスカバリールール - 直接リンクされた Web シナリオ - ホストマクロ - ホストインベントリデータ - バリュemap

export

ホストの export を行うには、以下のようにします。

- Configuration → Hosts に移動します。
- export するホストのチェックボックスにマークを付けます。
- リストの下にある Export をクリックします。

選択した形式に応じて、ホストはローカルにデフォルト名のファイルとして export されます。

- zabbix_export_hosts.yaml - YAML export の場合 (export のデフォルトオプション)
- zabbix_export_hosts.xml - XML export の場合
- zabbix_export_hosts.json - JSON 形式で export する場合

Import

ホストを Import するには、次のようにします。

- Configuration→Hosts をクリックします。
- 右側の Import をクリックします。
- import ファイルを選択します。
- import ルールの必要なオプションにマークを付けます
- Import をクリックします。

Import ✕

* Import file Choose file zbx_export_hosts.yaml

Rules	Update existing	Create new	Delete missing
Groups	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Hosts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Value mappings	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Template linkage		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Items	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Discovery rules	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Triggers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Graphs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Web scenarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Import
Cancel

import の成功または失敗のメッセージは、フロントエンドに表示されます。

Import ルール:

Rule	Description
Update existing	既存の要素は、import ファイルから取得したデータで更新されます。そうでない場合は、更新されません。
Create new	import は、import ファイルのデータを使用して新しい要素を追加します。そうでない場合は、追加されません。
Delete missing	import は、import ファイルに存在しない既存の要素を削除します。そうでない場合は、削除されません。

テンプレート・リンクに Delete missing が指定されている場合、import ファイルに存在しない既存のテンプレート・リンクは、リンクされていない可能性のあるテンプレートから継承されたすべてのエンティティ（アイテム、トリガーなど）と共にホストから削除されます。|

Export フォーマット

YAML の Export フォーマット:

```
zabbix_export:
  version: '6.0'
  date: '2021-09-28T12:20:07Z'
  groups:
    -
      uuid: f2481361f99448eeea617b7b1d4765566
      name: 'Discovered hosts'
    -
      uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Zabbix servers'
  hosts:
    -
```

```

host: 'Zabbix server 1'
name: 'Main Zabbix server'
templates:
-
  name: 'Linux by Zabbix agent'
-
  name: 'Zabbix server health'
groups:
-
  name: 'Discovered hosts'
-
  name: 'Zabbix servers'
interfaces:
-
  ip: 192.168.1.1
  interface_ref: if1
items:
-
  name: 'Zabbix trap'
  type: TRAP
  key: trap
  delay: '0'
  history: 1w
  preprocessing:
  -
    type: MULTIPLIER
    parameters:
    - '8'
  tags:
  -
    tag: Application
    value: 'Zabbix server'
  triggers:
  -
    expression: 'last(/Zabbix server 1/trap)=0'
    name: 'Last value is zero'
    priority: WARNING
    tags:
    -
      tag: Process
      value: 'Internal test'
tags:
-
  tag: Process
  value: Zabbix
macros:
-
  macro: '{$HOST.MACRO}'
  value: '123'
-
  macro: '{$PASSWORD1}'
  type: SECRET_TEXT
inventory:
  type: 'Zabbix server'
  name: yyyyyy-HP-Pro-3010-Small-Form-Factor-PC
  os: 'Linux yyyyyy-HP-Pro-3010-Small-Form-Factor-PC 4.4.0-165-generic #193-Ubuntu SMP Tue Sep 17 17
inventory_mode: AUTOMATIC
graphs:
-
  name: 'CPU utilization server'
  show_work_period: 'NO'
  show_triggers: 'NO'

```

```

graph_items:
-
  drawtype: FILLED_REGION
  color: FF5555
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,steal]'
-
  sortorder: '1'
  drawtype: FILLED_REGION
  color: 55FF55
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,softirq]'
-
  sortorder: '2'
  drawtype: FILLED_REGION
  color: '009999'
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,interrupt]'
-
  sortorder: '3'
  drawtype: FILLED_REGION
  color: '990099'
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,nice]'
-
  sortorder: '4'
  drawtype: FILLED_REGION
  color: '999900'
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,iowait]'
-
  sortorder: '5'
  drawtype: FILLED_REGION
  color: '990000'
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,system]'
-
  sortorder: '6'
  drawtype: FILLED_REGION
  color: '000099'
  calc_fnc: MIN
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,user]'
-
  sortorder: '7'
  drawtype: FILLED_REGION
  color: '009900'
  item:
    host: 'Zabbix server 1'
    key: 'system.cpu.util[,idle]'

```

Element タグ

Element タグの値については、下表のとおりです。

Host タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
groups		x			Root element for host groups.
	name	x	string		Host group name.
hosts		-			Root element for hosts.
	host	x	string		Unique host name.
	name	-	string		Visible host name.
	description	-	text		Host description.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Host status.
	ipmi_authtype	-	string	-1 - DEFAULT (default) 0 - NONE 1 - MD2 2 - MD5 4 - STRAIGHT 5 - OEM	IPMI session authentication type.
	ipmi_privilege	-	string	1 - CALLBACK 2 - USER (default) 3 - OPERATOR 4 - ADMIN 5 - OEM	IPMI session privilege level.
	ipmi_username	-	string		Username for IPMI checks.
	ipmi_password	-	string		Password for IPMI checks.
proxy		-			Proxy.
	name	x	string		Name of the proxy (if any) that monitors the host.
templates		-			Root element for linked templates.
	name	x	string		Template name.
interfaces		-			Root element for host interfaces.
	default	-	string	0 - NO 1 - YES (default)	Whether this is the primary host interface. There can be only one primary interface of one type on a host.
	type	-	string	1 - ZABBIX (default) 2 - SNMP 3 - IPMI 4 - JMX	Interface type.
	useip	-	string	0 - NO 1 - YES (default)	Whether to use IP as the interface for connecting to the host (if not, DNS will be used).

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	ip	-	string		IP address, can be either IPv4 or IPv6.
	dns	-	string		Required if the connection is made via IP. DNS name.
	port	-	string		Required if the connection is made via DNS. Port number. Supports user macros.
	interface_ref	x	string	Format: if<N>	Interface reference name to be used in items.
details		-			Root element for interface details.
	version	-	string	1 - SNMPV1 2 - SNMP_V2C (default) 3 - SNMP_V3	Use this SNMP version.
	community	-	string		SNMP community.
	contextname	-	string		Required by SNMPv1 and SNMPv2 items. SNMPv3 context name.
	securityname	-	string		Used only by SNMPv3 items. SNMPv3 security name.
	securitylevel	-	string	0 - NOAUTHNOPRIV (default) 1 - AUTHNOPRIV 2 - AUTHPRIV	Used only by SNMPv3 items. SNMPv3 security level.
	authprotocol	-	string	0 - MD5 (default) 1 - SHA1 2 - SHA224 3 - SHA256 4 - SHA384 5 - SHA512	Used only by SNMPv3 items. SNMPv3 authentication protocol.
	authpassphrase	-	string		Used only by SNMPv3 items. SNMPv3 authentication passphrase.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	privprotocol	-	string	0 - DES (default) 1 - AES128 2 - AES192 3 - AES256 4 - AES192C 5 - AES256C	SNMPv3 privacy protocol. Used only by SNMPv3 items.
	privpassphrase	-	string		SNMPv3 privacy passphrase. Used only by SNMPv3 items.
	bulk	-	string	0 - NO 1 - YES (default)	Use bulk requests for SNMP.
items		-			Root element for items.
	For item element tag values, see host item tags.				
tags		-			Root element for host tags.
	tag	x	string		Tag name.
	value	-	string		Tag value.
macros		-			Root element for macros.
	macro	x			User macro name.
	type	-	string	0 - TEXT (default) 1 - SECRET_TEXT 2 - VAULT	Type of the macro.
	value	-	string		User macro value.
	description	-	string		User macro description.
inventory		-			Root element for host inventory.
	<inventory_property>	-			Individual inventory property. All available inventory properties are listed under the respective tags, e.g. <type>, <name>, <os> (see example above).
inventory_mode		-	string	-1 - DISABLED 0 - MANUAL (default) 1 - AUTOMATIC	Inventory mode.
valuemaps		-			Root element for host value maps.
	name	x	string		Value map name.
	mapping	-			Root element for mappings.
	value	x	string		Value of a mapping.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	newvalue	x	string		New value of a mapping.

Host item タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
items		-			Root element for items.
	name	x	string		Item name.
	type	-	string	0 - ZABBIX_PASSIVE (default) 2 - TRAP 3 - SIMPLE 5 - INTERNAL 7 - ZABBIX_ACTIVE 10 - EXTERNAL 11 - ODBC 12 - IPMI 13 - SSH 14 - TELNET 15 - CALCULATED 16 - JMX 17 - SNMP_TRAP 18 - DEPENDENT 19 - HTTP_AGENT 20 - SNMP_AGENT 21 - ITEM_TYPE_SCRIPT	Item type.
	snmp_oid	-	string		SNMP object ID.
	key	x	string		Required by SNMP items. Item key.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	delay	-	string	Default: 1m	<p>Update interval of the item.</p> <p>Note that delay will be always '0' for trapper items.</p> <p>Accepts seconds or a time unit with suffix (30s, 1m, 2h, 1d). Optionally one or more custom intervals can be specified either as flexible intervals or scheduling. Multiple intervals are separated by a semicolon. User macros may be used. A single macro has to fill the whole field. Multiple macros in a field or macros mixed with text are not supported. Flexible intervals may be written as two macros separated by a forward slash (e.g. <code>{\$FLEX_INTERVAL}/{\$FLEX_INTERVAL}</code>).</p>
	history	-	string	Default: 90d	<p>A time unit of how long the history data should be stored. Time unit with suffix, user macro or LLD macro.</p>
	trends	-	string	Default: 365d	<p>A time unit of how long the trends data should be stored. Time unit with suffix, user macro or LLD macro.</p>
	status	-	string	0 - ENABLED (default) 1 - DISABLED	<p>Item status.</p>

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	value_type	-	string	0 - FLOAT 1 - CHAR 2 - LOG 3 - UNSIGNED (default) 4 - TEXT	Received value type.
	allowed_hosts	-	string		List of IP addresses (comma delimited) of hosts allowed sending data for the item.
	units	-	string		Used by trapper and HTTP agent items. Units of returned values (bps, B, etc).
	params	-	text		Additional parameters depending on the type of the item: - executed script for Script, SSH and Telnet items; - SQL query for database monitor items; - formula for calculated items.
	ipmi_sensor	-	string		IPMI sensor.
	authtype	-	string	Authentication type for SSH agent items: 0 - PASSWORD (default) 1 - PUBLIC_KEY Authentication type for HTTP agent items: 0 - NONE (default) 1 - BASIC 2 - NTLM	Used only by IPMI items. Authentication type. Used only by SSH and HTTP agent items.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	username	-	string		<p>Username for authentication. Used by simple check, SSH, Telnet, database monitor, JMX and HTTP agent items.</p> <p>Required by SSH and Telnet items. When used by JMX agent, password should also be specified together with the username or both properties should be left blank.</p>
	password	-	string		<p>Password for authentication. Used by simple check, SSH, Telnet, database monitor, JMX and HTTP agent items.</p> <p>When used by JMX agent, username should also be specified together with the password or both properties should be left blank.</p>
	publickey	-	string		<p>Name of the public key file.</p>
	privatekey	-	string		<p>Required for SSH agent items. Name of the private key file.</p>
	description	-	text		<p>Required for SSH agent items. Item description.</p>
	inventory_link	-	string	0 - NONE	<p>Host inventory field that is populated by the item.</p> <p>Capitalized host inventory field name. For example: 4 - ALIAS 6 - OS_FULL 14 - HARDWARE etc.</p> <p>Refer to the host inventory page for a list of supported host inventory fields and their IDs.</p>

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	logtimefmt	-	string		Format of the time in log entries. Used only by log items.
	interface_ref	-	string	Format: if<N>	Reference to the host interface.
	jmx_endpoint	-	string		JMX endpoint.
	url	-	string		Used only by JMX agent items. URL string.
	allow_traps	-	string	0 - NO (default) 1 - YES	Required only for HTTP agent items. Allow to populate value as in a trapper item.
	follow_redirects	-	string	0 - NO 1 - YES (default)	Used only by HTTP agent items. Follow HTTP response redirects while pooling data.
headers		-			Used only by HTTP agent items. Root element for HTTP(S) request headers, where header name is used as key and header value as value.
	name	x	string		Used only by HTTP agent items. Header name.
	value	x	string		Header value.
	http_proxy	-	string		HTTP(S) proxy connection string.
	output_format	-	string	0 - RAW (default) 1 - JSON	Used only by HTTP agent items. How to process response.
					Used only by HTTP agent items.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	post_type	-	string	0 - RAW (default) 2 - JSON 3 - XML	Type of post data body. Used only by HTTP agent items.
	posts	-	string		HTTP(S) request body data. Used only by HTTP agent items.
query_fields		-			Root element for query parameters. Used only by HTTP agent items.
	name	x	string		Parameter name.
	value	-	string		Parameter value.
	request_method	-	string	0 - GET (default) 1 - POST 2 - PUT 3 - HEAD	Request method. Used only by HTTP agent items.
	retrieve_mode	-	string	0 - BODY (default) 1 - HEADERS 2 - BOTH	What part of response should be stored.
	ssl_cert_file	-	string		Used only by HTTP agent items. Public SSL Key file path.
	ssl_key_file	-	string		Used only by HTTP agent items. Private SSL Key file path.
	ssl_key_password	-	string		Used only by HTTP agent items. Password for SSL Key file.
					Used only by HTTP agent items.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	status_codes	-	string		<p>Ranges of required HTTP status codes separated by commas. Supports user macros. Example: 200,200-{\$M},{ \$M},200-400</p> <p>Used only by HTTP agent items.</p>
	timeout	-	string		<p>Item data polling request timeout. Supports user macros.</p>
	verify_host	-	string	0 - NO (default) 1 - YES	<p>Used by HTTP agent and Script items. Validate if host name in URL is in Common Name field or a Subject Alternate Name field of host certificate.</p>
	verify_peer	-	string	0 - NO (default) 1 - YES	<p>Used only by HTTP agent items. Validate if host certificate is authentic.</p>
	parameters	-			<p>Used only by HTTP agent items. Root element for user-defined parameters.</p>
	name	x	string		<p>Used only by Script items. Parameter name.</p>
	value	-	string		<p>Used only by Script items. Parameter value.</p>
	value map	-			<p>Used only by Script items. Value map.</p>

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	name	x	string		Name of the value map to use for the item.
preprocessing		-			Root element for item value preprocessing.
step		-			Individual item value preprocessing step.
	type	x	string	1 - MULTIPLIER 2 - RTRIM 3 - LTRIM 4 - TRIM 5 - REGEX 6 - BOOL_TO_DECIMAL 7 - OCTAL_TO_DECIMAL 8 - HEX_TO_DECIMAL 9 - SIMPLE_CHANGE (calculated as (received value-previous value)) 10 - CHANGE_PER_SECOND (calculated as (received value-previous value)/(time now-time of last check)) 11 - XMLPATH 12 - JSONPATH 13 - IN_RANGE 14 - MATCHES_REGEX 15 - NOT_MATCHES_REGEX 16 - CHECK_JSON_ERROR 17 - CHECK_XML_ERROR 18 - CHECK_REGEX_ERROR 19 - DISCARD_UNCHANGED 20 - DISCARD_UNCHANGED_HEARTBEAT 21 - JAVASCRIPT 22 - PROMETHEUS_PATTERN 23 - PROMETHEUS_TO_JSON 24 - CSV_TO_JSON 25 - STR_REPLACE 26 - CHECK_NOT_SUPPORTED 27 - XML_TO_JSON	Type of the item value preprocessing step.
	parameters	-			Root element for parameters of the item value preprocessing step.
	parameter	x	string		Individual parameter of the item value preprocessing step.
	error_handler	-	string	0 - ORIGINAL_ERROR (default) 1 - DISCARD_VALUE 2 - CUSTOM_VALUE 3 - CUSTOM_ERROR	Action type used in case of preprocessing step failure.
	error_handler_params	-	string		Error handler parameters used with 'error_handler'.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
master_item		-			Individual item master item.
	key	x	string		Required by dependent items. Dependent item master item key value.
triggers		-			Recursion up to 3 dependent items and maximum count of dependent items equal to 29999 are allowed. Root element for simple triggers.
	For trigger element tag values, see host trigger tags .				
tags		-			Root element for item tags.
	tag	x	string		Tag name.
	value	-	string		Tag value.

Host low-level discovery rule タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
discovery_rules		-			Root element for low-level discovery rules.
	For most of the element tag values, see element tag values for a regular item. Only the tags that are specific to low-level discovery rules, are described below.				
	type	-	string	0 - ZABBIX_PASSIVE (default) 2 - TRAP 3 - SIMPLE 5 - INTERNAL 7 - ZABBIX_ACTIVE 10 - EXTERNAL 11 - ODBC 12 - IPMI 13 - SSH 14 - TELNET 16 - JMX 18 - DEPENDENT 19 - HTTP_AGENT 20 - SNMP_AGENT	Item type.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	lifetime	-	string	Default: 30d	Time period after which items that are no longer discovered will be deleted. Seconds, time unit with suffix or user macro.
filter	evaltype	-	string	0 - AND_OR (default) 1 - AND 2 - OR 3 - FORMULA	Individual filter. Logic to use for checking low-level discovery rule filter conditions.
	formula	-	string		Custom calculation formula for filter conditions.
conditions		-			Root element for filter conditions.
	macro	x	string		Low-level discovery macro name.
	value	-	string		Filter value: regular expression or global regular expression.
	operator	-	string	8 - MATCHES_REGEX (default) 9 - NOT_MATCHES_REGEX	Condition operator.
	formulaid	x	character		Arbitrary unique ID that is used to reference a condition from the custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward.
lld_macro_paths		-			Root element for LLD macro paths.
	lld_macro	x	string		Low-level discovery macro name.
	path	x	string		Selector for value which will be assigned to the corresponding macro.
preprocessing		-			LLD rule value preprocessing.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
step		-			Individual LLD rule value preprocessing step.
	For most of the element tag values, see element tag values for a host item value preprocessing. Only the tags that are specific to low-level discovery value preprocessing, are described below.				
	type	x	string	5 - REGEX 11 - XMLPATH 12 - JSONPATH 15 - NOT_MATCHES_REGEX 16 - CHECK_JSON_ERROR 17 - CHECK_XML_ERROR 20 - DISCARD_UNCHANGED_HEARTBEAT 21 - JAVASCRIPT 23 - PROMETHEUS_TO_JSON 24 - CSV_TO_JSON 25 - STR_REPLACE 27 - XML_TO_JSON	Type of the item value preprocessing step.
trigger_prototypes		-			Root element for trigger prototypes.
	For trigger prototype element tag values, see regular host trigger tags.				
graph_prototypes		-			Root element for graph prototypes.
	For graph prototype element tag values, see regular host graph tags.				
host_prototypes		-			Root element for host prototypes.
	For host prototype element tag values, see regular host tags.				
item_prototypes		-			Root element for item prototypes.
	For item prototype element tag values, see regular host item tags.				
master_item		-			Individual item prototype master item/item prototype data.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	key	x	string		Dependent item prototype master item/item prototype key value. Required for a dependent item.

Host trigger タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
triggers		-			Root element for triggers.
	expression	x	string		Trigger expression.
	recovery_mode	-	string	0 - EXPRESSION (default) 1 - RECOVERY_EXPRESSION 2 - NONE	Basis for generating OK events.
	recovery_expression	-	string		Trigger recovery expression.
	name	x	string		Trigger name.
	correlation_mode	-	string	0 - DISABLED (default) 1 - TAG_VALUE	Correlation mode (no event correlation or event correlation by tag).
	correlation_tag	-	string		The tag name to be used for event correlation.
	url	-	string		URL associated with the trigger.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Trigger status.
	priority	-	string	0 - NOT_CLASSIFIED (default) 1 - INFO 2 - WARNING 3 - AVERAGE 4 - HIGH 5 - DISASTER	Trigger severity.
	description	-	text		Trigger description.
	type	-	string	0 - SINGLE (default) 1 - MULTIPLE	Event generation type (single problem event or multiple problem events).
	manual_close	-	string	0 - NO (default) 1 - YES	Manual closing of problem events.
dependencies		-			Root element for dependencies.
	name	x	string		Dependency trigger name.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	expression	x	string		Dependency trigger expression.
	recovery_expression	-	string		Dependency trigger recovery expression.
tags		-			Root element for event tags.
	tag	x	string		Tag name.
	value	-	string		Tag value.

Host graph タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
graphs		-			Root element for graphs.
	name	x	string		Graph name.
	width	-	integer	20-65535 (default: 900)	Graph width, in pixels. Used for preview and for pie/exploded graphs.
	height	-	integer	20-65535 (default: 200)	Graph height, in pixels. Used for preview and for pie/exploded graphs.
	yaxismin	-	double	Default: 0	Value of Y axis minimum.
	yaxismax	-	double	Default: 0	Value of Y axis maximum.
	show_work_period	-	string	0 - NO 1 - YES (default)	Used if 'ymin_type_1' is FIXED. Highlight non-working hours.
	show_triggers	-	string	0 - NO 1 - YES (default)	Used by normal and stacked graphs. Display simple trigger values as a line.
	type	-	string	0 - NORMAL (default) 1 - STACKED 2 - PIE 3 - EXPLODED	Used by normal and stacked graphs. Graph type.
	show_legend	-	string	0 - NO 1 - YES (default)	Display graph legend.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	show_3d	-	string	0 - NO (default) 1 - YES	Enable 3D style.
	percent_left	-	double	Default:0	Used by pie and exploded pie graphs. Show the percentile line for left axis.
	percent_right	-	double	Default:0	Used only for normal graphs. Show the percentile line for right axis.
	ymin_type_1	-	string	0 - CALCULATED (default) 1 - FIXED 2 - ITEM	Used only for normal graphs. Minimum value of Y axis.
	ymax_type_1	-	string	0 - CALCULATED (default) 1 - FIXED 2 - ITEM	Used by normal and stacked graphs. Maximum value of Y axis.
	ymin_item_1	-			Used by normal and stacked graphs. Individual item details.
	host	x	string		Required if 'ymin_type_1' is ITEM. Item host.
	key	x	string		Item key.
	ymax_item_1	-			Individual item details.
	host	x	string		Required if 'ymax_type_1' is ITEM. Item host.
	key	x	string		Item key.
	graph_items	x			Root element for graph items.
	sortorder	-	integer		Draw order. The smaller value is drawn first. Can be used to draw lines or regions behind (or in front of) another.
	drawtype	-	string	0 - SINGLE_LINE (default) 1 - FILLED_REGION 2 - BOLD_LINE 3 - DOTTED_LINE 4 - DASHED_LINE 5 - GRADIENT_LINE	Draw style of the graph item.
	color	-	string		Used only by normal graphs. Element color (6 symbols, hex).

Element	Element プロパティ	必須?	型	レンジ ¹	説明
	yaxisside	-	string	0 - LEFT (default) 1 - RIGHT	Side of the graph where the graph item's Y scale will be drawn.
	calc_fnc	-	string	1 - MIN 2 - AVG (default) 4 - MAX 7 - ALL (minimum, average and maximum; used only by simple graphs) 9 - LAST (used only by pie and exploded pie graphs)	Used by normal and stacked graphs. Data to draw if more than one value exists for an item.
	type	-	string	0 - SIMPLE (default) 2 - GRAPH_SUM (value of the item represents the whole pie; used only by pie and exploded pie graphs)	Graph item type.
item		x			Individual item.
	host	x	string		Item host.
	key	x	string		Item key.

Host web scenario タグ

Element	Element プロパティ	必須?	型	レンジ ¹	説明
httptests		-			Root element for web scenarios.
	name	x	string		Web scenario name.
	delay	-	string	Default: 1m	Frequency of executing the web scenario. Seconds, time unit with suffix or user macro.
	attempts	-	integer	1-10 (default: 1)	The number of attempts for executing web scenario steps.
	agent	-	string	Default: Zabbix	Client agent. Zabbix will pretend to be the selected browser. This is useful when a website returns different content for different browsers.
	http_proxy	-	string		Specify an HTTP proxy to use, using the format: http://[username[:password

Element	Element プロパティ	必須?	型	レンジ ¹	説明
variables		-			Root element for scenario-level variables (macros) that may be used in scenario steps.
	name	x	text		Variable name.
	value	x	text		Variable value.
headers		-			Root element for HTTP headers that will be sent when performing a request. Headers should be listed using the same syntax as they would appear in the HTTP protocol.
	name	x	text		Header name.
	value	x	text		Header value.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Web scenario status.
	authentication	-	string	0 - NONE (default) 1 - BASIC 2 - NTLM	Authentication method.
	http_user	-	string		User name used for basic, HTTP or NTLM authentication.
	http_password	-	string		Password used for basic, HTTP or NTLM authentication.
	verify_peer	-	string	0 - NO (default) 1 - YES	Verify the SSL certificate of the web server.
	verify_host	-	string	0 - NO (default) 1 - YES	Verify that the Common Name field or the Subject Alternate Name field of the web server certificate matches.
	ssl_cert_file	-	string		Name of the SSL certificate file used for client authentication (must be in PEM format).
	ssl_key_file	-	string		Name of the SSL private key file used for client authentication (must be in PEM format).
	ssl_key_password	-	string		SSL private key file password.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
steps		x			Root element for web scenario steps.
	name	x	string		Web scenario step name.
	url	x	string		URL for monitoring.
query_fields		-			Root element for query fields - an array of HTTP fields that will be added to the URL when performing a request.
	name	x	string		Query field name.
	value	-	string		Query field value.
posts		-			HTTP POST variables as a string (raw post data) or as an array of HTTP fields (form field data).
	name	x	string		Post field name.
	value	x	string		Post field value.
variables		-			Root element of step-level variables (macros) that should be applied after this step.
					If the variable value has a 'regex:' prefix, then its value is extracted from the data returned by this step according to the regular expression pattern following the 'regex:' prefix
	name	x	string		Variable name.
	value	x	string		Variable value.

Element	Element プロパティ	必須?	型	レンジ ¹	説明
headers		-			Root element for HTTP headers that will be sent when performing a request. Headers should be listed using the same syntax as they would appear in the HTTP protocol.
	name	x	string		Header name.
	value	x	string		Header value.
	follow_redirects	-	string	0 - NO 1 - YES (default)	Follow HTTP redirects.
	retrieve_mode	-	string	0 - BODY (default) 1 - HEADERS 2 - BOTH	HTTP response retrieve mode.
	timeout	-	string	Default: 15s	Timeout of step execution. Seconds, time unit with suffix or user macro.
	required	-	string		Text that must be present in the response. Ignored if empty.
	status_codes	-	string		A comma delimited list of accepted HTTP status codes. Ignored if empty. For example: 200-201,210-299
tags		-			Root element for web scenario tags.
	tag	x	string		Tag name.
	value	-	string		Tag value.

脚注

¹ 文字列値の場合は、このテーブルで使用されている番号付けは行われず、
 文字列のみがエクスポートされます (例: 「ZABBIX_ACTIVE」)。
 このテーブルの範囲値 (API 値に対応) の番号は、順序付けにのみ使用されます。

4 ネットワークマップ

概要

ネットワークマップの **export** には、以下の内容が含まれます。

- すべての関連画像
- マップ構造 - すべてのマップ設定、すべての含まれる要素とその設定、マップリンク、およびマップリンクとマップリンクステータスインジケータ

Warning:

ホストグループ、ホスト、トリガー、他のマップ、または他の要素が含まれます。その他の要素は、export されません。
したがって、マップが参照する要素のうち少なくとも 1 つが欠落している場合 import に失敗します。

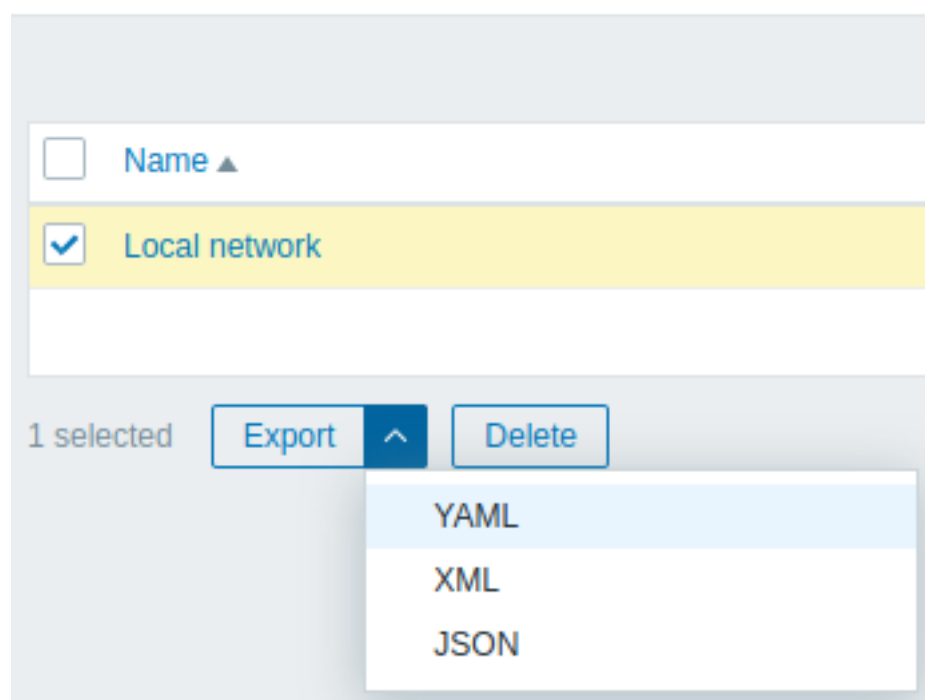
ネットワークマップの export /import は Zabbix1.8.2 以降でサポートされています。

export

ネットワークマップを export するには、次の操作を行います。

- Monitoring → Maps に移動します。
- export するネットワークマップのチェックボックスにマークを付けます。
- リストの下にある Export をクリックします。

≡ Maps



選択した形式に応じて、マップは次のようなローカルファイルにデフォルト名でエクスポートされます。

- zabbix_export_maps.yaml - YAML export の場合 (export の default option)
- zabbix_export_maps.xml - XML export の場合
- zabbix_export_maps.json - JSON export の場合

import

ネットワークマップを import するには、次のようにします。

- Monitoring → Maps に移動します。
- 右側の Import をクリックします。
- import ファイルを選択します。
- import ルールの必要なオプションにマークを付けます。
- import をクリックします。

すべての必須入力項目には赤いアスタリスクが表示されます。

import の成功または失敗のメッセージは、フロントエンドに表示されます。

Import ルール:

Rule	Description
Update existing	Existing maps will be updated with data taken from the import file. Otherwise they will not be updated.
Create new	The import will add new maps using data from the import file. Otherwise it will not add them.

マップのオプションを両方ともオフにし、イメージのオプションをそれぞれオンにすると、イメージのみが import されます。
画像の import は、Super Admin ユーザのみ可能です。

Warning:

既存の画像を置き換える場合、この画像を使用しているすべてのマップに影響します。

Export フォーマット

YAML への export :

```
zabbix_export:
  version: '6.0'
  date: '2021-08-31T12:55:10Z'
  images:
    -
      name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    -
      name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
```

```

label_type_host: '2'
label_type_hostgroup: '2'
label_type_trigger: '2'
label_type_map: '2'
label_type_image: '2'
label_string_host: ''
label_string_hostgroup: ''
label_string_trigger: ''
label_string_map: ''
label_string_image: ''
expand_macros: '1'
background: { }
iconmap: { }
urls: { }
selements:
  -
    elementtype: '0'
    elements:
      -
        host: 'Zabbix server'
        label: |
          {HOST.NAME}
          {HOST.CONN}
        label_location: '0'
        x: '111'
        'y': '61'
        elementsubtype: '0'
        areatype: '0'
        width: '200'
        height: '200'
        viewtype: '0'
        use_iconmap: '0'
        selementid: '1'
        icon_off:
          name: Zabbix_server_3D_(128)
        icon_on: { }
        icon_disabled: { }
        icon_maintenance: { }
        urls: { }
        evaltype: '0'
shapes:
  -
    type: '0'
    x: '0'
    'y': '0'
    width: '680'
    height: '15'
    text: '{MAP.NAME}'
    font: '9'
    font_size: '11'
    font_color: '000000'
    text_halign: '0'
    text_valign: '0'
    border_type: '0'
    border_width: '0'
    border_color: '000000'
    background_color: ''
    zindex: '0'
lines: { }
links: { }

```

Element タグの値は下表のとおりです。

Element	Element プロパティ	型	レンジ	説明
images	name	string		Root element for images. Unique image name.
	imagetype	integer	1 - image 2 - background	Image type.
	encodedImage			Base64 encoded image.
maps	name	string		Root element for maps. Unique map name.
	width	integer		Map width, in pixels.
	height	integer		Map height, in pixels.
	label_type	integer	0 - label 1 - host IP address 2 - element name 3 - status only 4 - nothing	Map element label type.
	label_location	integer	0 - bottom 1 - left 2 - right 3 - top	Map element label location by default.
	highlight	integer	0 - no 1 - yes	Enable icon highlighting for active triggers and host statuses.
	expandproblem	integer	0 - no 1 - yes	Display problem trigger for elements with a single problem.
	markelements	integer	0 - no 1 - yes	Highlight map elements that have recently changed their status.
	show_unack	integer	0 - count of all problems 1 - count of unacknowl- edged problems 2 - count of acknowl- edged and unacknowl- edged problems separately	Problem display.
	severity_min	integer	0 - not classified 1 - information 2 - warning 3 - average 4 - high 5 - disaster	Minimum trigger severity to show on the map by default.
	show_suppressed	integer	0 - no 1 - yes	Display problems which would otherwise be suppressed (not shown) because of host maintenance.
grid_size	integer	20, 40, 50, 75 or 100	Cell size of a map grid in pixels, if "grid_show=1"	

Element	Element プロパティ	型	レンジ	説明
grid_show		integer	0 - yes 1 - no	Display a grid in map configuration.
grid_align		integer	0 - yes 1 - no	Automatically align icons in map configuration.
label_format		integer	0 - no 1 - yes	Use advanced label configuration.
label_type_host		integer	0 - label 1 - host IP address 2 - element name 3 - status only 4 - nothing 5 - custom label	Display as host label, if "label_format=1"
label_type_hostgroup		integer	0 - label 2 - element name 3 - status only 4 - nothing 5 - custom label	Display as host group label, if "label_format=1"
label_type_trigger		integer	0 - label 2 - element name 3 - status only 4 - nothing 5 - custom label	Display as trigger label, if "label_format=1"
label_type_map		integer	0 - label 2 - element name 3 - status only 4 - nothing 5 - custom label	Display as map label, if "label_format=1"
label_type_image		integer	0 - label 2 - element name 4 - nothing 5 - custom label	Display as image label, if "label_format=1"
label_string_host		string		Custom label for host elements, if "label_type_host=5"
label_string_hostgroup		string		Custom label for host group elements, if "label_type_hostgroup=5"
label_string_trigger		string		Custom label for trigger elements, if "label_type_trigger=5"
label_string_map		string		Custom label for map elements, if "label_type_map=5"
label_string_image		string		Custom label for image elements, if "label_type_image=5"

Element	Element プロパティ	型	レンジ	説明
	expand_macros	integer	0 - no 1 - yes	Expand macros in labels in map configuration.
	background	id		ID of the background image (if any), if "imagetype=2"
	iconmap	id		ID of the icon mapping (if any).
	urls			Used by maps or each map element.
selements	name	string		Link name.
	url	string		Link URL.
	elementtype	integer	0 - host 1 - map 2 - trigger 3 - host group 4 - image	Map item type the link belongs to.
	elementtype	integer	0 - host 1 - map 2 - trigger 3 - host group 4 - image	Map element type.
	label	string		Icon label.
	label_location	integer	-1 - use map default 0 - bottom 1 - left 2 - right 3 - top	
	x	integer		Location on the X axis.
	y	integer		Location on the Y axis.
	elementsubtype	integer	0 - single host group 1 - all host groups	Element subtype, if "elementtype=3"
	areatype	integer	0 - same as whole map 1 - custom size	Area size, if "elementsubtype=1"
	width	integer		Width of area, if "areatype=1"
	height	integer		Height of area, if "areatype=1"
	viewtype	integer	0 - place evenly in the area	Area placement algorithm, if "elementsubtype=1"
tags	use_iconmap	integer	0 - no 1 - yes	Use icon mapping for this element. Relevant only if iconmapping is activated on map level.
	selementid	id		Unique element record ID.
	evaltype	integer		Evaluation type for tags. Problem tags (for host and host group elements). If tags are given, only problems with these tags will be displayed on the map.
	tag			Tag name.
	value			Tag value.

Element	Element プロパティ	型	レンジ	説明
elements	operator			Operator. Zabbix entities that are represented on the map (host, host group, map etc).
	host			
icon_off				Image to use when element is in 'OK' status.
icon_on				Image to use when element is in 'Problem' status.
icon_disabled				Image to use when element is disabled.
icon_maintenance				Image to use when element is in maintenance.
shapes	name	string		Unique image name.
	type	integer	0 - rectangle 1 - ellipse	Shape type.
	x	integer		X coordinates of the shape in pixels.
	y	integer		Y coordinates of the shape in pixels.
	width	integer		Shape width.
	height	integer		Shape height.
	border_type	integer	0 - none 1 - bold line 2 - dotted line 3 - dashed line	Type of the border for the shape.
	border_width	integer		Width of the border in pixels.
	border_color	string		Border color represented in hexadecimal code.
	text	string		Text inside of shape.

Element	Element プロパティ	型	レンジ	説明
	font	integer	0 - Georgia, serif 1 - "Palatino Linotype", "Book Antiqua", Palatino, serif 2 - "Times New Roman", Times, serif 3 - Arial, Helvetica, sans-serif 4 - "Arial Black", Gadget, sans-serif 5 - "Comic Sans MS", cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - "Lucida Sans Unicode", "Lucida Grande", sans-serif 8 - Tahoma, Geneva, sans-serif 9 - "Trebuchet MS", Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - "Courier New", Courier, monospace 12 - "Lucida Console", Monaco, monospace	Text font style.
	font_size	integer		Font size in pixels.
	font_color	string		Font color represented in hexadecimal code.
	text_halign	integer	0 - center 1 - left 2 - right	Horizontal alignment of text.
	text_valign	integer	0 - middle 1 - top 2 - bottom	Vertical alignment of text.

Element	Element プロパティ	型	レンジ	説明
lines	background_color	string		Background (fill) color represented in hexadecimal code.
	zindex	integer		Value used to order all shapes and lines (z-index).
	x1	integer		X coordinates of the line point 1 in pixels.
	y1	integer		Y coordinates of the line point 1 in pixels.
	x2	integer		X coordinates of the line point 2 in pixels.
	y2	integer		Y coordinates of the line point 2 in pixels.
	line_type	integer	0 - none 1 - bold line 2 - dotted line 3 - dashed line	Line type.
	line_width	integer		Line width in pixels.
	line_color	string		Line color represented in hexadecimal code.
	zindex	integer		Value used to order all shapes and lines (z-index).
links	drawtype	integer	0 - line 2 - bold line 3 - dotted line 4 - dashed line	Links between map elements. Link style.
	color	string		Link color (6 symbols, hex).
	label	string		Link label.
	selementid1	id		ID of one element to connect.
linktriggers	selementid2	id		ID of the other element to connect.
	drawtype	integer	0 - line 2 - bold line 3 - dotted line 4 - dashed line	Link status indicators. Link style when trigger is in the 'Problem' state.
trigger	color	string		Link color (6 symbols, hex) when trigger is in the 'Problem' state.
	description	string		Trigger used for indicating link status. Trigger name.
	expression	string		Trigger expression.
	recovery_expression	string		Trigger recovery expression.

5 メディアタイプ

概要

メディアタイプは、関連するすべてのオブジェクトやオブジェクトの関係性と一緒に**エクスポート**されます。

エクスポート

メディアタイプをエクスポートするには、次のようにします。

- 管理 → メディアタイプに移動します。
- エクスポートするメディアタイプのチェックボックスにマークを付けます。
- リストの下にあるエクスポートをクリックします。

Media types

<input type="checkbox"/>	Name ▲	Type
<input checked="" type="checkbox"/>	Helpdesk	Webhook

1 selected

Enable Disable Export ^ Delete

- YAML
- XML
- JSON

選択された形式に応じて、メディアタイプはローカルファイルにデフォルトの名前でエクスポートされます。

- zabbix_export_mediatypes.yaml - YAML 形式の場合 (export の default option)
- zabbix_export_mediatypes.xml - XML 形式の場合
- zabbix_export_mediatypes.json - JSON 形式の場合

インポート

メディアタイプをインポートするには、次のようにします。

- 管理 → メディアタイプに移動します。
- 右側のインポートをクリックします。
- インポートするファイルを選択します。
- インポートルールの必要なオプションにマークを付けます。
- インポートをクリックします。

Import

* Import file zbx_export_mediatypes.yaml

Rules

Update existing Create new Delete missing

Media types

Import Cancel

インポートの成功または失敗のメッセージが Web インターフェースに表示されます。

インポートルール:

ルール	説明
既存の設定を上書 Update existing 新規作成	既存の要素は、import ファイルから取得したデータで更新されます。そうでない場合は、更新されません。 import ファイルのデータを使用して新しい要素を追加します。そうでない場合は、追加されません。
存在しない場合に削除	インポートファイルに存在しない既存の要素を削除します。そうでない場合は、削除されません。

エクスポートフォーマット

YAML 形式の export:

```
zabbix_export:
  version: '6.0'
  date: '2021-08-31T13:34:17Z'
  media_types:
    -
      name: Pushover
      type: WEBHOOK
      parameters:
        8:
          name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        6:
          name: eventid
          value: '{EVENT.ID}'
        11:
          name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        12:
          name: event_source
          value: '{EVENT.SOURCE}'
        13:
          name: event_value
          value: '{EVENT.VALUE}'
        10:
          name: expire
          value: '1200'
        2:
          name: message
          value: '{ALERT.MESSAGE}'
        14:
          name: priority_average
          value: '0'
        15:
          name: priority_default
          value: '0'
        16:
          name: priority_disaster
          value: '0'
        17:
          name: priority_high
          value: '0'
        18:
          name: priority_information
          value: '0'
        19:
          name: priority_not_classified
          value: '0'
        20:
          name: priority_warning
```

```

    value: '0'
  9:
    name: retry
    value: '60'
  3:
    name: title
    value: '{ALERT.SUBJECT}'
  0:
    name: token
    value: '<PUSHOVER TOKEN HERE>'
  7:
    name: triggerid
    value: '{TRIGGER.ID}'
  4:
    name: url
    value: '{$ZABBIX.URL}'
  5:
    name: url_title
    value: Zabbix
  1:
    name: user
    value: '{ALERT.SENDTO}'
max_sessions: '0'
script: |
  try {
    var params = JSON.parse(value),
        request = new HttpRequest(),
        data,
        response,
        severities = [
          {name: 'not_classified', color: '#97AAB3'},
          {name: 'information', color: '#7499FF'},
          {name: 'warning', color: '#FFC859'},
          {name: 'average', color: '#FFA059'},
          {name: 'high', color: '#E97659'},
          {name: 'disaster', color: '#E45959'},
          {name: 'resolved', color: '#009900'},
          {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
      request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
      throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0 or 1';
    }

    if (params.event_value !== '0' && params.event_value !== '1'
      && (params.event_source === '0' || params.event_source === '3')) {
      throw 'Incorrect "event_value" parameter given: "' + params.event_value + '".\nMust be 0 or 1';
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
      params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
      params.event_nseverity = '6';
    }
  }

```

```

priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

if (isNaN(priority) || priority < -2 || priority > 2) {
    throw '"priority" should be -2..2';
}

if (params.event_source === '0' && isNaN(params.triggerid)) {
    throw 'field "triggerid" is not a number';
}

if (isNaN(params.eventid)) {
    throw 'field "eventid" is not a number';
}

if (typeof params.message !== 'string' || params.message.trim() === '') {
    throw 'field "message" cannot be empty';
}

data = {
    token: params.token,
    user: params.user,
    title: params.title,
    message: params.message,
    url: (params.event_source === '0')
        ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
        : params.url,
    url_title: params.url_title,
    priority: priority
};

if (priority == 2) {
    if (isNaN(params.retry) || params.retry < 30) {
        throw 'field "retry" should be a number with value of at least 30 if "priority" is set
    }

    if (isNaN(params.expire) || params.expire > 10800) {
        throw 'field "expire" should be a number with value of at most 10800 if "priority" is
    }

    data.retry = params.retry;
    data.expire = params.expire;
}

data = JSON.stringify(data);
Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

request.addHeader('Content-Type: application/json');
response = request.post(params.endpoint, data);

Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus());

if (response !== null) {
    try {
        response = JSON.parse(response);
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
        response = null;
    }
}

if (request.getStatus() != 200 || response === null || typeof response !== 'object' || response

```

```

        if (response !== null && typeof response === 'object' && typeof response.errors === 'object'
            && typeof response.errors[0] === 'string') {
            throw response.errors[0];
        }
        else {
            throw 'Unknown error. Check debug log for more information.';
        }
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
    throw 'Pushover notification failed: ' + error;
}
description: |
    Please refer to setup guide here: https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/template

    Set token parameter with to your Pushover application key.
    When assigning Pushover media to the Zabbix user - add user key into send to field.
message_templates:
-
    event_source: TRIGGERS
    operation_mode: PROBLEM
    subject: 'Problem: {EVENT.NAME}'
    message: |
        Problem started at {EVENT.TIME} on {EVENT.DATE}
        Problem name: {EVENT.NAME}
        Host: {HOST.NAME}
        Severity: {EVENT.SEVERITY}
        Operational data: {EVENT.OPDATA}
        Original problem ID: {EVENT.ID}
        {TRIGGER.URL}
-
    event_source: TRIGGERS
    operation_mode: RECOVERY
    subject: 'Resolved in {EVENT.DURATION}: {EVENT.NAME}'
    message: |
        Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE}
        Problem name: {EVENT.NAME}
        Problem duration: {EVENT.DURATION}
        Host: {HOST.NAME}
        Severity: {EVENT.SEVERITY}
        Original problem ID: {EVENT.ID}
        {TRIGGER.URL}
-
    event_source: TRIGGERS
    operation_mode: UPDATE
    subject: 'Updated problem in {EVENT.AGE}: {EVENT.NAME}'
    message: |
        {USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
        {EVENT.UPDATE.MESSAGE}

        Current problem status is {EVENT.STATUS}, age is {EVENT.AGE}, acknowledged: {EVENT.ACK.STATUS}
-
    event_source: DISCOVERY
    operation_mode: PROBLEM
    subject: 'Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
    message: |
        Discovery rule: {DISCOVERY.RULE.NAME}

        Device IP: {DISCOVERY.DEVICE.IPADDRESS}

```

```

Device DNS: {DISCOVERY.DEVICE.DNS}
Device status: {DISCOVERY.DEVICE.STATUS}
Device uptime: {DISCOVERY.DEVICE.UPTIME}

Device service name: {DISCOVERY.SERVICE.NAME}
Device service port: {DISCOVERY.SERVICE.PORT}
Device service status: {DISCOVERY.SERVICE.STATUS}
Device service uptime: {DISCOVERY.SERVICE.UPTIME}
-
event_source: AUTOREGISTRATION
operation_mode: PROBLEM
subject: 'Autoregistration: {HOST.HOST}'
message: |
  Host name: {HOST.HOST}
  Host IP: {HOST.IP}
  Agent port: {HOST.PORT}

```

要素のタグ

要素のタグの値は下表のとおりです。

Element	Element プロパティ	必須	型	レンジ ¹	説明
media_types		-			Root element for media_types.
	name	x	string		Media type name.
	type	x	string	0 - EMAIL 1 - SMS 2 - SCRIPT 4 - WEBHOOK	Transport used by the media type.
	status	-	string	0 - ENABLED (default) 1 - DISABLED	Whether the media type is enabled.
	max_sessions	-	integer	Possible values for SMS: 1 - (default) Possible values for other media types: 0-100, 0 - unlimited	The maximum number of alerts that can be processed in parallel.
	attempts	-	integer	1-10 (default: 3)	The maximum number of attempts to send an alert.
	attempt_interval	-	string	0-60s (default: 10s)	The interval between retry attempts.
	description	-	string		Accepts seconds and time unit with suffix. Media type description.
message_templates		-			Root element for media type message templates.
	event_source	x	string	0 - TRIGGERS 1 - DISCOVERY 2 - AUTOREGISTRATION 3 - INTERNAL	Event source.
	operation_mode	x	string	0 - PROBLEM 1 - RECOVERY 2 - UPDATE	Operation mode.

Element	Element プロパティ	必須	型	レンジ ¹	説明	
Used only by e-mail media type	subject	-	string		Message subject.	
	message	-	string		Message body.	
	smtp_server	x	string		SMTP server.	
	smtp_port	-	integer	Default: 25	SMTP server port to connect to.	
	smtp_helo	x	string		SMTP helo.	
	smtp_email	x	string		Email address from which notifications will be sent.	
	smtp_security	-	string	0 - NONE (default) 1 - STARTTLS 2 - SSL_OR_TLS	SMTP connection security level to use.	
	smtp_verify_host	-	string	0 - NO (default) 1 - YES	SSL verify host for SMTP. Optional if smtp_security is STARTTLS or SSL_OR_TLS.	
	smtp_verify_peer	-	string	0 - NO (default) 1 - YES	SSL verify peer for SMTP. Optional if smtp_security is STARTTLS or SSL_OR_TLS.	
	smtp_authentication	-	string	0 - NONE (default) 1 - PASSWORD	SMTP authentication method to use.	
Used only by SMS media type	username	-	string		Username.	
	password	-	string		Authentication password.	
	content_type	-	string	0 - TEXT 1 - HTML (default)	Message format.	
	gsm_modem	x	string		Serial device name of the GSM modem.	
	Used only by script media type parameters	script name	x	string		Script name.
			-			Root element for script parameters.
	Used only by webhook media type	script	x	string		Script.

Element	Element プロパティ	必須	型	レンジ ¹	説明
	timeout	-	string	1-60s (default: 30s)	Javascript script HTTP request timeout interval.
	process_tags	-	string	0 - NO (default) 1 - YES	Whether to process returned tags.
	show_event_menu	-	string	0 - NO (default) 1 - YES	If {EVENT.TAGS.*} were successfully resolved in event_menu_url and event_menu_name fields, this field indicates presence of entry in the event menu.
	event_menu_url	-	string		URL of the event menu entry. Supports {EVENT.TAGS.*} macro.
	event_menu_name	-	string		Name of the event menu entry. Supports {EVENT.TAGS.*} macro.
	parameters	-			Root element for webhook media type parameters.
	name	x	string		Webhook parameter name.
	value	-	string		Webhook parameter value.

脚注

¹ 文字列値については、文字列のみがエクスポートされ (例:"EMAIL " など)、この表で使用される番号付けは行われません。
 このテーブルの範囲値 (API 値に対応) の番号は、順序付けのためにのみ使用されます。

15. ディスカバリ

サイドバーからディスカバリセクションのコンテンツにアクセスしてください。

1 ネットワークディスカバリ

概要

Zabbix は、効果的でとても柔軟な自動ネットワークディスカバリ機能を提供しています。

ネットワークディスカバリを適切に設定すると、次のような効果があります：

- Zabbix の展開スピードアップ

- 管理のシンプル化
- 急速に変化する環境で、過剰な管理なしで Zabbix を使用可能

Zabbix のネットワークディスカバリは、次の情報に基づいています：

- IP アドレスの範囲
- 外部サービスの可用性 (FTP, SSH, WEB, POP3, IMAP, TCP, etc)
- Zabbix エージェントから受信する情報 (非暗号化モードのみサポート)
- SNMP エージェントから受信する情報

以下の機能は提供「しません」：

- ネットワークトポロジの検出

ネットワークディスカバリは、基本的に、ディスカバリとアクションの2段階で構成されます。

ディスカバリ

Zabbix はネットワークディスカバリルールで定義された IP 範囲を定期的にスキャンします。チェックの頻度はルールごと個別に設定できます。

1つのディスカバリルールは、常に1つのディスカバリプロセスによって処理されることに注意してください。IP 範囲は、複数のディスカバリプロセス間で分割されません。

各ルールには、IP 範囲に対して実行するように定義された一連のサービスチェックがあります。

Note:

ディスカバリチェックは、他のチェックとは別に処理されます。チェックでサービスが見つからない(または失敗した)場合でも、他のチェックは引き続き処理されます。

ネットワークディスカバリモジュールによって実行されるサービスとホスト (IP) のチェックごとに、ディスカバリイベントが生成されます。

イベント	サービス結果の確認
Service Discovered	サービスは'ダウン'した後または初めて検出されたときに'アップ'しています。
Service Up	サービスは'アップ'した後、引き続き'アップ'のままです。
Service Lost	サービスは'アップ'した後に'ダウン'しています。
Service Down	サービスは'ダウン'した後、引き続き'ダウン'のままです。
Host Discovered	ホストのすべてのサービスが'ダウン'した後、または登録されていないホストに属するサービスが検出された後、少なくとも1つのサービスが'アップ'しています。
Host Up	少なくとも1つのサービスが既に'アップ'していた後、ホストの少なくとも1つのサービスが引き続き'アップ'のままです。
Host Lost	ホストのすべてのサービスは、少なくとも1つが'アップ'した後、'ダウン'しています。
Host Down	ホストのすべてのサービスは、'ダウン'した後、引き続き'ダウン'のままです。

アクション

ディスカバリイベントは、次のような関連するアクションの基盤になります：

- 通知の送信
- ホストの追加/削除
- ホストの有効化/無効化
- グループへのホストの追加
- グループからのホストの削除
- テンプレートへのホストのリンク/テンプレートからのホストのリンクの削除
- リモートスクリプトの実行

これらのアクションはデバイスタイプ、IP、ステータス、アップタイム/ダウンタイムなどに関して設定可能です。イベントに基づくネットワークディスカバリに対するアクションの設定に関する詳細は、アクションのオペレーションとコンディションのページを参照してください。

ネットワーク検出アクションはイベントベースであるため、検出されたホストがオンラインの場合とオフラインの場合の両方でトリガーされます。Service Lost/ServiceDown イベントで Addhost がトリガーされるなどのアクションを回避するために、アクション条件 Discovery status:up を追加することを強くお勧めします。それ以外の場合、検出されたホストが手動で削除された場合でも、Service Lost/Service Down イベントが生成され、次の検出サイクル中に再作成されます。

Note:

検出されたホストのテンプレートへのリンクは、リンク可能なテンプレートのいずれかに、ホストまたは別のリンク可能なテンプレートにすでに存在する一意のエントティ (アイテムキーなど) と同じ一意のエントティ (アイテムキーなど) がある場合、まとめて失敗します。

ホストの作成

ホストの追加操作が選択されている場合にホストが追加されますが、ホストの追加操作を選択していない場合でも、ホストでアクションが発生する操作を選択すると、ホストも追加されます。そのような操作は次のとおりです。

- ホストを有効にする
- ホストを無効にする
- ホストをホストグループに追加する
- テンプレートをホストにリンクする

作成されたホストは Discovered hosts グループに追加されます。(デフォルトでは、管理 → 一般 → **その他**で設定可能) ホストを別のグループに追加する場合は * Remove from hostgroups 操作を追加します。("Discovered hosts" を指定) また、ホストはホストグループに属している必要があるため、Add to host groups* 操作 (追加先は別のホストグループを指定) を追加します。

ホストの命名

ホストを追加する場合、ホスト名は逆引き DNS ルックアップの結果または逆引きルックアップが失敗した場合の IP アドレスです。ルックアップは、検出を行っている Zabbix サーバーまたは Zabbix プロキシから実行されます。プロキシでルックアップが失敗した場合でもサーバーで再試行されません。同じ名前のホストがすでに存在する場合、次のホストには名前に **_2** を追加し、また次のホストには **_3** というように追加されます。

DNS/IP ルックアップをオーバーライドして、ホスト名にアイテム値を使用することもできます。次に例を示します。

- 検出用の Zabbix エージェントアイテムを使用して実行されている Zabbix エージェントで、複数のサーバーを検出し、このアイテムによって返される文字列値に基づいて、適切な名前をサーバーに自動的に割り当てる
- SNMP エージェントアイテムを使用して、複数の SNMP ネットワークデバイスを検出し、このアイテムによって返される文字列値に基づいて、適切な名前を自動的に割り当てる

ホスト名がアイテム値を使用して設定されている場合、次の検出チェックでもホスト名は更新されません。項目値でホスト名を設定できない場合は、デフォルト値 (DNS 名) を使用します。

検出された IP アドレスを持つホストがすでに存在する場合、新しいホストは作成されません。ただし、検出アクションに操作 (リンクテンプレート、ホストグループへの追加など) が含まれている場合、それらは既存のホストで実行されます。

ホストの削除

ネットワーク検出ルールによって検出されたホストは、検出されたエントティがルールの IP 範囲内がない場合、監視 → 検出からすぐ自動的に削除されます。

ホスト追加時のインターフェース作成

ホストがネットワークディスカバリの結果として追加されると、次のルールに従って作成されたインターフェースを取得します。

- 検出されたサービス - たとえば、SNMP チェックが成功した場合、SNMP インターフェースが作成されます
- ホストが Zabbix エージェントと SNMP リクエストの両方に応答した場合、両方のタイプのインターフェースが作成されます
- 一意性の基準が Zabbix エージェントまたは SNMP から返されたデータである場合、ホストに対して最初に見つかったインターフェースがデフォルトとして作成されます。その他の IP アドレスは、追加のインターフェースとして追加されます。アクションの条件 (ホスト IP など) は、インターフェースの追加には影響しません。注意これは、すべてのインターフェースが同じディスカバリルールによって検出された場合に機能することに注意してください。別のディスカバリルールが同じホストの別のインターフェースを検出した場合、追加のホストが追加されます。
- ホストがエージェントチェックのみに応答した場合、ホストはエージェントインターフェースのみで作成されます。後で SNMP への応答を開始した場合は、追加の SNMP インターフェースが追加されます。
- 'IP' による一意性基準によって検出された 3 つの個別のホストが最初に作成され、ホスト A、B、および C が同一の一意性基準結果を持つようにディスカバリルールが変更された場合、B および C は A の追加インターフェースとして作成されます。最初のホストである個々のホスト B と C は残ります。監視データ → ディスカバリでは、追加されたインターフェースは " 検出されたデバイス " 列に黒いフォントでインデントされて表示されますが " 監視対象ホスト " 列には最初に作成されたホストである A のみが表示されます。追加のインターフェースと見なされる IP については " アップタイム/ダウンタイム " は測定されません。

プロキシ設定の変更

異なるプロキシによって検出されたホストは、常に異なるホストとして扱われます。これにより、異なるサブネットで使われる同一 IP 範囲を検出できますが、プロキシの変更は検出されたすべてのホストにも適用する必要があるため、すでに監視されているサブネットのプロキシの変更は複雑です。

たとえば、検出ルールでプロキシを置き換える手順は次のとおりです。

1. 検出ルールを無効にする
2. プロキシ設定を同期する
3. 検出ルールのプロキシを置き換える
4. このルールによって検出されたすべてのホストのプロキシを置き換える
5. 検出ルールを有効にする

1 ネットワークディスカバリルールの設定

概要

Zabbix で使用されるネットワークディスカバリルールでホストやサービスを検出するように設定する方法は次の通りです。

- [設定](#)→[ディスカバリ](#) を選択
- [\[ディスカバリルールの作成\]](#) をクリック (またはルール名をクリックして既存のルールを編集)
- ディスカバリルールの属性を編集

ルールの属性

* Name

Discovery by proxy

* IP range

* Update interval

* Checks

Type	Discovery check
HTTP	Check type <input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="SNMPv2 agent"/>
HTTPS	* Port range <input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="161"/>
SNMPv2 agent "iso.3.6.1.2.1.1.1.0"	* SNMP community <input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="public"/>
Zabbix agent "system.uptime"	* SNMP OID <input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="iso.3.6.1.2.1.1.1.0"/>
Add	

Device uniqueness criteria

IP address

SNMPv2 agent "iso.3.6.1.2.1.1.1.0"

Zabbix agent "system.uptime"

Host name

DNS name

IP address

SNMPv2 agent "iso.3.6.1.2.1.1.1.0"

Zabbix agent "system.uptime"

Visible name

Host name

DNS name

IP address

SNMPv2 agent "iso.3.6.1.2.1.1.1.0"

Zabbix agent "system.uptime"

Enabled

必須入力フィールドは、赤いアスタリスクでマークされています。

パラメーター	説明
名前 プロキシによるディスクバリ	一意なルール名。例：“ローカルネットワーク”。 ディスクバリを実行するもの： プロキシなし - Zabbix サーバーがディスクバリを実行します < プロキシ名 > - このプロキシがディスクバリを実行します 検出用の IP アドレス範囲。次の形式である可能性があります。 単一 IP：192.168.1.33 IP アドレス範囲：192.168.1-10.1-255。範囲は対象となるアドレスの総数 (64K 未満) によって制限されます。 IP マスク：192.168.4.0/24 サポートされる IP マスク： /16-/ 30 (IPv4 アドレスの場合) /112-/128 (IPv6 アドレスの場合) リスト：192.168.1.1-255、192.168.2.1-100、192.168.2.200、192.168.4.0/24
IP アドレスの範囲	Zabbix 3.0.0 以降、このフィールドはスペース、集計、および複数行をサポートします。
監視間隔	このパラメータは、Zabbix がルールを実行する間隔を定義します。間隔は前の検出インスタンスの実行が終了した後に測定されるため重複はありません。 Zabbix 3.4.0 移行、 タイムサフィックス がサポートされています。 例：30s,1m,2h,1d ユーザーマクロ は、Zabbix 3.4.0 以降でサポートされています。 注意ユーザーマクロが使用され、その値が変更された場合 (たとえば、1w→1h) 次のチェックは前の値に従って実行されます (将来的にはサンプル値を使用)。
チェック	Zabbix はこのチェックリストを検出に使用します。 Add をクリックして、ポップアップウィンドウで新しいチェックを設定します。 サポートされているチェック：SSH、LDAP、SMTP、FTP、HTTP、HTTPS、POP、NNTP、IMAP、TCP、Telnet、Zabbix エージェント、SNMPv1 エージェント、SNMPv2 エージェント、SNMPv3 エージェント、ICMPPing プロトコルベースの検出では、 net.tcp.service[] 機能を使用して、SNMP OID を照会する SNMP を除いて、各ホストをテストします。 Zabbix エージェントは、暗号化されていないモードでアイテムをクエリすることによってテストされます。詳細については エージェントアイテム を参照してください。 'Ports' パラメータは次のいずれかになります。 単一のポート:22 ポート範囲:22-45 リスト:22-45,55,60-70
デバイスの固有性を特定する基準	一意性の基準は次のとおりです。 IP アドレス - 複数の単一 IP デバイスの処理はありません。同じ IP を持つデバイスがすでに存在する場合、そのデバイスはすでに検出されていると見なされ、新しいホストは追加されません。 < 検出チェック > - Zabbix エージェントまたは SNMP エージェントチェック
ホスト名	以下を使用して作成したホストの技術ホスト名を設定します。 ** DNS 名 ** - DNS 名 (デフォルト) IP アドレス - IP アドレス < 検出チェック > - 検出チェックの受信文字列値 (例：Zabbix エージェント、SNMP エージェントチェック) 関連項目： ホストの名前付け
表示名	このオプションは 4.2.0 以降でサポートされています。 以下を使用して作成したホストの表示ホスト名を設定します。 ホスト名 - 技術的なホスト名 (デフォルト) DNS 名 - DNS 名 IP アドレス - IP アドレス < 検出チェック > - 検出チェックの受信文字列値 (例：Zabbix エージェント、SNMP エージェントチェック) 関連項目： ホストの名前付け このオプションは 4.2.0 以降でサポートされています。

パラメーター	説明
有効	チェックボックスがマークされている場合、ルールはアクティブであり、Zabbix サーバーによって実行されます。 マークされていない場合、ルールは無効です。実行されません。

実際に使用するシナリオ

この例では、ローカルネットワークの 192.168.1.1 ~ 192.168.1.254 の範囲の IP アドレスをチェックするネットワークディスカバリを設定します。

このシナリオでは、以下のことをおこなう設定をします：

- Zabbix エージェントが動作しているホストを検出
- 検出は 10 分間隔で実行
- アップタイムが 1 時間以上のホストは監視対象に追加
- ダウンタイムが 24 時間以上のホストは監視対象から削除
- Linux ホストはグループ「Linux servers」に追加
- Windows ホストはグループ「Windows servers」に追加
- Linux ホストではテンプレート Linux を使用
- Windows ホストではテンプレート Windows を使用

ステップ 1

IP アドレス範囲のネットワークディスカバリルールを定義します。

* Name	Local network							
Discovery by proxy	No proxy ▼							
* IP range	192.168.1.1-254							
* Update interval	10m							
* Checks	<table border="1"> <thead> <tr> <th>Type</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Zabbix agent "system.uname"</td> <td>Edit Remove</td> </tr> <tr> <td>Add</td> <td></td> </tr> </tbody> </table>	Type	Actions	Zabbix agent "system.uname"	Edit Remove	Add		
Type	Actions							
Zabbix agent "system.uname"	Edit Remove							
Add								
Device uniqueness criteria	<input checked="" type="radio"/> IP address <input type="radio"/> Zabbix agent "system.uname"							
Host name	<input type="radio"/> DNS name <input type="radio"/> IP address <input checked="" type="radio"/> Zabbix agent "system.uname"							
Visible name	<input checked="" type="radio"/> Host name <input type="radio"/> DNS name <input type="radio"/> IP address <input type="radio"/> Zabbix agent "system.uname"							
Enabled	<input checked="" type="checkbox"/>							

Zabbix は、Zabbix エージェントに接続し、**system.uname** キーから値を取得することにより、IP アドレス範囲 192.168.1.1～192.168.1.254 のホストを検出しようとしています。エージェントから受け取った値を使用して、ホストに名前を付けたり、オペレーティングシステムごとに異なるアクションを適用したりできます。たとえば、Windows サーバーをテンプレート Windows にリンクし、Linux サーバーをテンプレート Linux にリンクするなどです。

ルールは 10 分ごとに実行されます。

このルールが追加されると、Zabbix は自動的に検出を開始し、検出ベースのイベントを生成してさらに処理します。

ステップ 2

新たに検出された Linux サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。

Action **Operations**

* Name

Type of calculation A and B and C and D

Conditions

Label	Name
A	Received value contains <i>Linux</i>
B	Discovery status equals <i>Up</i>
C	Service type equals <i>Zabbix agent</i>
D	Uptime/Downtime is greater than or equals <i>3600</i>

[Add](#)

このアクションは以下のコンディションが成立する場合有効になります。 - サービス“Zabbix agent”が“動作”している。 - system.uname（ルールを定義する際に使用した Zabbix エージェントのキー）の値に“Linux”という文字列が含まれる。 - アップタイムが 1 時間（3600 秒）を超えている。

Action **Operations**

Default subject

Default message
 Device IP: {DISCOVERY.DEVICE.IPADDRESS}
 Device DNS: {DISCOVERY.DEVICE.DNS}
 Device status: {DISCOVERY.DEVICE.STATUS}
 Device uptime: {DISCOVERY.DEVICE.UPTIME}
 Device service name: {DISCOVERY.SERVICE.NAME}"/>

Operations

[Details](#)

Add to host groups: Linux servers

Link to templates: Linux

[Add](#)

アクションによって以下のオペレーションが実行されます。

- 新たに検出したホストをグループ“Linux servers”に追加する。（まだ追加されていないホストの場合はホストの追加も実行する）
- ホストをテンプレート“Linux”にリンクする。“Linux”テンプレートに含まれるアイテムとトリガーを使用してホストの監視が自動的に開始される。

ステップ 3

新たに検出された Windows サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。

Action Operations

* Name

Type of calculation A and B and C and D

Conditions

Label	Name
A	Received value contains <i>Windows</i>
B	Discovery status equals <i>Up</i>
C	Service type equals <i>Zabbix agent</i>
D	Uptime/Downtime is greater than or equals <i>3600</i>

[Add](#)

Action Operations

Default subject

Default message

Operations

Details

Add to host groups: Windows servers

Link to templates: Windows

[Add](#)

ステップ 4

動作していないサーバーを削除するための検出アクションを定義します。

Action	Operations								
* Name	Remove lost servers								
Type of calculation	And ▼ A and B and C								
Conditions	<table border="1"> <thead> <tr> <th>Label</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Uptime/Downtime is greater than or equals 86400</td> </tr> <tr> <td>B</td> <td>Discovery status equals Down</td> </tr> <tr> <td>C</td> <td>Service type equals Zabbix agent</td> </tr> </tbody> </table> <p>Add</p>	Label	Name	A	Uptime/Downtime is greater than or equals 86400	B	Discovery status equals Down	C	Service type equals Zabbix agent
Label	Name								
A	Uptime/Downtime is greater than or equals 86400								
B	Discovery status equals Down								
C	Service type equals Zabbix agent								

Action	Operations				
Default subject	Discovery: {DISCOVERY.DEVICE.STATUS}, {DISCOVERY.DEVICE.IPAD}				
Default message	Discovery rule: {DISCOVERY.RULE.NAME} Device IP: {DISCOVERY.DEVICE.IPADDRESS} Device DNS: {DISCOVERY.DEVICE.DNS} Device status: {DISCOVERY.DEVICE.STATUS} Device uptime: {DISCOVERY.DEVICE.UPTIME} Device service name: {DISCOVERY.SERVICE.NAME}				
Operations	<table border="1"> <thead> <tr> <th>Details</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Remove host</td> <td>Edit Remove</td> </tr> </tbody> </table> <p>Add</p>	Details	Action	Remove host	Edit Remove
Details	Action				
Remove host	Edit Remove				

"Zabbix agent" サービスが 24 時間 (86400 秒) 以上 down している場合、サーバーは削除されます。A server will be removed if "Zabbix agent" service is 'down' for more than 24 hours (86400 seconds).

2 アクティブエージェントの自動登録

概要

サーバが監視を開始可能になった後は、アクティブな Zabbix エージェントの自動登録が可能です。自動登録によって、サーバ上で手動で設定をしなくても新しいホストを監視対象として追加できます。

前に認知されていないアクティブエージェントがチェックを求めたときに、自動登録がおこなわれます。

この機能は、新しいクラウドのノードの自動監視にとっても便利です。クラウド環境で新しいノードをもつと同時に、Zabbix が自動的にそのホストのパフォーマンスと可用性のデータの収集を開始します。

アクティブエージェントの自動登録は、パッシブチェックで追加されたホストの監視もサポートしています。アクティブエージェントがチェックを求めたとき、自動登録をおこなうと、設定ファイルで定義された「ListenIP」または「ListenPort」設定のパラメータをもつことになり、これらはサーバに送信されます。(複数の IP アドレスが指定されている場合は、最初のものがサーバに送信されます。)

新しい自動登録ホストを追加したとき、サーバは受信した IP アドレスとポート番号をエージェントの設定に使用します。IP アドレスの値が 1 つも受信されない場合は、接続に使用されたアドレスが使用されます。ポート番号が 1 つも受信されない場合は、10050 が

使用されます。

デフォルトのエージェントインターフェイスとして**DNS 名**を使用してホストを自動登録するように指定できます。

次の場合は自動登録が再実行されます。

- ホスト**メタデータ**情報が変更された場合：
 - ホストメタデータが変更され、エージェントが再起動したため
 - ホストメタデータアイテムによって返される値が変更されたため
- メタデータが欠落している、手動で作成されたホストの場合
- ホストを手動で変更して別の Zabbix プロキシで監視する場合
- 同じホストの自動登録が新しい Zabbix プロキシからのものである場合

設定

サーバーを指定する

エージェント**設定ファイル** - zabbix_agentd.conf で Zabbix サーバーが識別されていることを確認してください

ServerActive=10.0.0.1

zabbix_agentd.conf でホスト名を明確に定義しない限り、エージェントの場所のシステムホスト名がサーバーによってホストの命名に使用されます。Linux のシステムホスト名は hostname コマンドを実行することで取得できます。

ホスト名が Zabbix エージェント設定でホストのコンマ区切りリストとして定義されている場合、リストされているすべてのホスト名に対してホストが作成されます。

設定ファイル変更後、エージェントを再起動します。

アクティブエージェントの自動登録用アクション

サーバーは、エージェントから自動登録要求を受信すると、**アクション**を呼び出します。イベントソース“自動登録”のアクションは、エージェントの自動登録用に設定する必要があります。

Note:

ネットワーク検出を設定するために、アクティブなエージェントを自動登録する必要はありません。

Zabbix フロントエンドで、設定 → アクションに移動し、イベントソースとして自動登録を選択して、アクションの作成をクリックします。

- アクションタブで、アクションに名前を付けます
- 必要に応じて**条件**を指定します。ホスト名/ホストメタデータの条件で、部分文字列の一致または正規表現の一致を行うことができます。“ホストメタデータ”条件を使用する場合は、次のセクションを参照してください。
- 操作タブで‘ホストの追加’や‘ホストグループへの追加’（たとえば検出されたホスト）、‘テンプレートへのリンク’などの関連する操作を追加します。

Note:

自動登録されるホスト（Zabbix サーバーからファイアウォールで保護されているホストなど）がアクティブモニタリングのみでサポートされる可能性が高い場合は、リンク用に Template_Linux-active 等、特定のテンプレートを作成することをお勧めします。

作成されたホストは Discovered hosts グループに追加されます。（デフォルトでは管理 → 一般 → **その他**で設定可能です）ホストを別のグループに追加する場合は、ホストグループから削除操作（Discovered hosts を指定）を追加し、ホストグループに追加操作（Discovered hosts 以外の別ホストグループを指定）も追加します。これは、ホストはホストグループに属している必要があるためです。

安全な自動登録

暗号化された接続を使用して PSK ベース認証を設定することにより、安全な自動登録が可能になります。

暗号化レベルは、右側のドロップダウンからアクセスできる [自動登録] セクションの [管理] → **一般*** でグローバルに設定でき、暗号化なし、PSK 認証を使用した TLS 暗号化、またはその両方を選択できます。（一部のホストは暗号化なしで登録、他のホストは暗号化を介して登録などもできます）

PSK による認証は、ホストを追加する前に Zabbix サーバーによって検証されます。検証が成功すると、ホストが追加され、**ホストから/へ接続**がグローバル自動登録設定と同じ ID/事前共有キーでのみ PSK に設定されます。

Attention:

プロキシ経由での自動登録のセキュリティを確保するには、Zabbix サーバーとプロキシ間の暗号化を有効にする必要があります。

DNS をデフォルトのインターフェースとして使用する

HostInterface および HostInterfaceItem **パラメーター** を使用すると、自動登録にホストインターフェイスのカスタム値を指定できます。

具体的には、ホストを IP アドレスではなくデフォルトのエージェントインターフェイスとして DNS 名で自動登録する必要がある場合に役立ちます。その場合、DNS 名は HostInterface または HostInterfaceItem どちらかのパラメーターの値として指定または返される必要があります。2 つのパラメータのいずれかの値が変更されると、自動登録されたホストインターフェイスが更新されることに注意してください。上記からデフォルトのインターフェースを別の DNS 名に変更したり、IP アドレスに変更したりすることもできます。ただし、変更を有効にするには、エージェントを再起動する必要があります。

ホストメタデータの使用

エージェントがサーバーに自動登録要求を送信すると、エージェントはホスト名も送信しますが、場合によっては（たとえば Amazon クラウドノード）ホスト名は Zabbix サーバーが検出されたホストを区別するのに十分では無いこともあります。オプションでホストメタデータを使用してエージェントからサーバーに他の情報を送信することもできます。

ホストメタデータはエージェント **設定ファイル** - zabbix_agentd.conf で設定されます。ホストメタデータの指定箇所は方法は 2 つあります。

HostMetadata

HostMetadataItem

上記リンクのオプションの説明を参照してください。

Attention:

自動登録の試行は、アクティブエージェントがアクティブチェックの更新要求をサーバーに送信するたびに発生します。リクエスト間の遅延は、エージェントの **RefreshActiveChecks** パラメーターで指定されます。最初の要求は、エージェントが再起動された直後に送信されます。

例 1

ホストメタデータを使用して Linux ホストと Windows ホストを区別します。

ホストを Zabbix サーバーによって自動登録したいとします。ネットワーク上にアクティブな Zabbix エージェント（上記の“設定”セクションを参照）があります。ネットワーク上に Windows ホストと Linux ホストがあり、Zabbix フロントエンドで“Linux by Zabbix agent”と“Windows by Zabbix agent”のテンプレートを使用したいとします。その場合、ホストの登録時に適切な Linux/Windows テンプレートを適用する必要があります。デフォルトでは、自動登録時にホスト名のみがサーバーに送信されますが、これでは不十分な場合があります。適切なテンプレートがホストに適用されていることを確認するには、ホストメタデータを使用する必要があります。

フロントエンドの設定

最初に行うことはフロントエンドを構成することです。今回は 2 つのアクションを作成します。

まず

- 名前: Linux ホストの自動登録
- 条件: ホストメタデータに Linux が含まれる。
- 操作: テンプレートへのリンク: Linux

Note:

この場合は「ホストの追加」操作をスキップできます。本来テンプレートにリンクするには、サーバーが自動的にそれを行うように、最初にホストを追加する必要があります。

次に

- 名前: Windows ホストの自動登録
- 条件: ホストメタデータに Windows が含まれる
- 操作: テンプレートへのリンク: Windows

エージェントの設定

次にエージェントを設定する必要があります。次の行をエージェント設定ファイルに追加します。

```
HostMetadataItem=system.uname
```

このようにしてエージェントが実行されているホストに応じて、ホストメタデータに“Linux”または“Windows”が含まれるようにします。この場合のホストメタデータの例は以下です。

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
```

```
Windows: Windows WIN-OPXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

設定ファイル変更後、エージェントの再起動を忘れないでください。

例 2

ステップ 1

ホストメタデータを使用して、不要ホストの登録回避を可能にします。

フロントエンドの設定

推測しにくい秘密のコードを使用してフロントエンドでアクションを作成し、不要なホストを禁止します。

名前：自動登録アクションLinux

条件：

```
* Type of calculation: AND
* Condition (A): Host metadata contains //Linux//
* Condition (B): Host metadata contains //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
* Operations:
  * Send message to users: Admin via all media
  * Add to host groups: Linux servers
  * Link to templates: Linux
```

データはプレーンテキストで送信されるため、この方法だけでは強力な保護は提供されないことに注意してください。変更をすぐに有効にするには、設定キャッシュのリロードが必要です。

エージェントの設定

次の行をエージェント構成ファイルに追加します。

```
HostMetadata=Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

ここでの“Linux”はプラットフォームであり、文字列の残りの部分は推測しにくい秘密のテキストです。

設定ファイル変更後は、エージェントを再起動することを忘れないでください。

ステップ 2

すでに登録されているホストの監視を追加することができます。

フロントエンドの設定

フロントエンドのアクションを更新します。

- 名前: 自動登録アクション Linux
- 条件:

```
* Type of calculation: AND
* Condition (A): Host metadata contains Linux
* Condition (B): Host metadata contains 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
* Operations:
  * Send message to users: Admin via all media
  * Add to host groups: Linux servers
  * Link to templates: Linux
  * Link to templates: MySQL by Zabbix Agent
```

エージェントの設定

エージェント構成ファイルの次の行を更新します。

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

設定ファイル変更後は、エージェントの再起動を忘れないでください。

3 ローレベルディスクバリ

概要 ローレベルディスクバリは、コンピューター上のさまざまなエンティティのアイテム、トリガー、およびグラフを自動的に作成する方法を提供します。たとえば、Zabbix は各ファイルシステムまたはネットワークインターフェイスのアイテムを手動で作成しなくても、マシン上のファイルシステムまたはネットワークインターフェイスの監視を自動的に開始できます。さらに、定期的に行われるディスクバリの結果に基づいて、不要なエンティティを自動的に削除するように Zabbix を設定することができます。

ユーザーは特定の JSON プロトコルに基づいて独自のディスクバリを定義できます。

検出プロセスの一般的なアーキテクチャは次のとおりです。

まずユーザーは“設定”→“テンプレート”→“ディスカバリ”列でディスカバリルールを作成します。ディスカバリルールは、(1) 必要なエンティティ（ファイルシステムやネットワークインターフェイスなど）を検出するアイテムと、(2) アイテムの値に基づいて作成する必要のあるアイテム、トリガー、グラフのプロトタイプで構成されます。

必要なエンティティを検出するアイテムは、他の場所で見られる通常のアイテムのようなものです。サーバーは Zabbix エージェント（またはアイテムのタイプが設定されているもの）にそのアイテムの値を要求し、エージェントはテキスト値で応答します。違いは、エージェントが応答する値には、検出されたエンティティのリストが JSON 形式で含まれている必要があることです。この形式の詳細は、カスタムディスカバリチェックの実装者にとってのみ重要ですが、戻り値にはマクロ → 値のペアのリストが含まれていることを知っておく必要があります。たとえば、アイテム“net.if.discovery”は“{#IFNAME}” → “lo” と “{#IFNAME}” → “eth0” の 2 つのペアを返す場合があります。

これらのマクロは、名前、キー、およびその他のプロトタイプフィールドで使用され、検出された各エンティティの実際のアイテム、トリガー、グラフ、さらにはホストを作成するために、受け取った値に置き換えられます。ローレベルディスカバリマクロの使用については **オプション** のリストを参照してください。

サーバーは検出したアイテムの値を受け取ると、マクロ → 値のペアを確認し、ペアごとにプロトタイプに基づいて実際のアイテム、トリガー、およびグラフを生成します。上記の“net.if.discovery”の例では、サーバーはループバックインターフェイス“lo”用のアイテム、トリガー、およびグラフの 1 つのセットと、インターフェイス“eth0”用の別のセットを生成します。

Zabbix 4.2 以降、ローレベルディスカバリルールによって返される JSON の形式が変更されていることに注意してください。JSON に“データ”オブジェクトが含まれることはもはや期待されていません。ローレベルディスカバリは、JSON ドキュメント内のローレベルディスカバリマクロ値へのアイテム値の前処理やカスタムパスなどの新機能をサポートするために、配列を含む通常の JSON を受け入れるようになりました。

組み込みのディスカバリキーが更新され、JSON ドキュメントのルートにあるローレベルディスカバリ行の配列が返されるようになりました。配列フィールドで {#MACRO} 構文がキーとして使用されている場合、Zabbix はマクロと値を自動的に抽出します。新しいネイティブディスカバリチェックでは“データ”要素なしで新しい構文が使用されます。ローレベルディスカバリの値を処理する場合、最初に root が特定されます（\$、または \$.data の配列）。

“data”要素はディスカバリに関連するすべてのネイティブアイテムから削除されていますが、下位互換性のために Zabbix は“data”要素を含む JSON 表記を引き続き受け入れます。ですがその使用は推奨されていません。JSON に“data”配列要素が 1 つしかないオブジェクトが含まれている場合、JSONPath\$.data を使用して要素のコンテンツを自動的に抽出します。ローレベルディスカバリは JSONPath 構文で指定されたカスタムパスを持つオプションのユーザー定義ローレベルディスカバリマクロを受け入れるようになりました。

Warning:

上記の変更の結果、新しいエージェントは古い Zabbix サーバーと連携できなくなりました。

参照：[検出されたエンティティ](#)

ローレベルディスカバリの設定 ファイルシステム検出の例に基づいて、ローレベルディスカバリについて説明します。

次の手順でディスカバリを構成します。

- 設定 → テンプレートまたはホストを開く
- 適当なテンプレートまたはホストのディスカバリ行をクリックする

≡ Templates

<input type="checkbox"/>	Name ▲	Hosts	Applications	Items	Triggers	Graphs	Dashboards	Discovery
<input type="checkbox"/>	Linux OS agent	Hosts 1	Applications 11	Items 42	Triggers 14	Graphs 8	Dashboards 1	Discovery 3

- 画面右上にあるディスカバリルールの作成をクリックする
- ディスカバリルールフォームに必要な詳細を記入します

ディスカバリルール

ディスカバリルールフォームには、ディスカバリ中のデータフローを左から右に表す 5 つのタブが含まれています。

- ディスカバリルール - 最も重要なこととして、検出されたデータを取得するための組み込みアイテムまたはカスタムスクリプトを指定します

- 保存前処理 - 検出されたデータに前処理を適用します
- LLD マクロ - 検出されたアイテム、トリガーなどで使用するいくつかのマクロ値を設定できます
- フィルター - 検出された値をフィルタリングできます
- オーバーライド - 特定の検出されたオブジェクトに適用するときに、アイテム、トリガー、グラフ、またはホストプロトタイプを変更できます。

ディスカバリルールタブには、検出に使用するアイテムキー（およびいくつかの一般的なディスカバリルール属性）が含まれています。

Discovery rule
Preprocessing
LLD macros
Filters 4
Overrides

* Name

Type

* Key

* Update interval

Custom intervals	Type	Interval	Period
	Flexible	Scheduling	50s
			1-7,00:00-24:00
	Add		

* Keep lost resources period

Description

Enabled

すべての必須入力フィールドは、赤いアスタリスクでマークされています。

パラメータ	説明
名前	ディスカバリルール名
タイプ	ディスカバリを実行するためのチェックタイプ。 この例では Zabbix エージェントアイテムキーを使用しています。 アイテムによっては、ディスカバリルールを 依存アイテム にすることもできます。別のディスカバリルールに依存することはできません。依存アイテムの場合は、それぞれのタイプ（依存アイテム）を選択し、'マスターアイテム' フィールドでマスターアイテムを指定します。マスターアイテムは実在する必要があります。 検出項目キーを入力します（最大 2048 文字） たとえば組み込みの"vfs.fs.discovery" アイテムキーを使用して、コンピューターに存在するファイルシステムとそのタイプのリストを含む JSON を返すことができます。 ファイルシステム検出の別のオプションは、Zabbix 4.4.5 以降でサポートされている"vfs.fs.get" エージェントキーによる検出結果を使用することです（例を参照）
キー	

パラメータ	説明
監視間隔	このフィールドは Zabbix がディスカバリを実行する頻度を指定します。最初にファイルシステムの検出を設定するだけの場合は、短い間隔に設定することをお勧めしますが、ファイルシステムは通常あまり頻繁に変更されないため、機能することがわかったら 30 分以上に再設定します。 Zabbix3.4.0 以降、 タイムサフィックス がサポートされています。 例：30s,1m,2h,1d Zabbix3.4.0 以降、 ユーザーマクロ がサポートされています。 注：更新間隔はゼロ以外の値のカスタム間隔が存在する場合のみ'0'に設定できます。更新間隔が'0'に設定され、ゼロ以外の値のカスタム間隔（フレキシブルまたはスケジュール済み）が存在する場合、アイテムはカスタム間隔の期間中にポーリングされます。 注既存のディスカバリルールの場合、監視データ取得ボタンを押すと検出がすぐに実行されます。
監視間隔のカスタマイズ	アイテムをチェックするためのカスタムルールを作成できます。 例外設定 - 監視間隔の例外を作成します（異なる頻度の間隔） 定期設定 - カスタムポーリングスケジュールを作成します。 詳細については、 カスタム間隔 を参照してください。Zabbix 3.0.0 以降、スケジュールリングがサポートされています。
存在しなくなったリソースの保持期間	このフィールドでは、検出ステータスが“取得不可”（1 時間から 25 年、または“0”）になった後、検出されたエンティティが保持される（削除されない）期間を指定できます。 Zabbix3.4.0 以降、 タイムサフィックス がサポートされています。例：2h,1d ユーザーマクロ は Zabbix3.4.0 以降でサポートされています。 注：“0”に設定すると、エンティティはすぐに削除されます。フィルタを誤って編集すると、すべての履歴データとともにエンティティが削除される可能性があるため“0”の使用はお勧めしません。
説明 有効	説明を入力します チェックするとルールが処理されます。

Note:

ディスカバリルールの履歴は保持されません。

保存前処理

保存前処理タブでは、検出結果に適用する変換ルールを定義できます。このステップでは、1 つまたは複数の変換が可能です。変換は定義された順序で実行されます。すべての保存前処理は Zabbix サーバーによって行われます。

参照：

- [保存前処理の詳細](#)
- [保存前処理のテスト](#)

Discovery rule **Preprocessing 2** LLD macros Filters Overrides

Preprocessing steps	Name	Parameters
1:	Regular expression	pattern
2:	JSONPath	\$.pool
Add		

タイプ	トランスフォーメーション	説明
テキスト	正規表現	<p>受信した値を <pattern> 正規表現に一致させ、値を抽出した <output> に置き換えます。正規表現は、\N シーケンスで最大 10 個のキャプチャされたグループの抽出をサポートします。</p> <p>パラメータ：</p> <p>pattern - 正規表現</p> <p>output - 出力フォーマットテンプレート。 \N (N = 1...9) のエスケープシーケンスは、N 番目に一致するグループに置き換えられます。 \0 エスケープシーケンスは一致したテキストに置き換えられます。</p> <p>失敗時のカスタマイズチェックボックスをオンにすると、値を破棄、値を設定、エラーメッセージを設定いずれかのカスタムエラー処理オプションを指定できます。</p>
	置換	<p>検索文字列を見つけて、別の文字列に置き換え（または削除）します。検索文字列のすべての出現箇所が置き換えられます。パラメーター：</p> <p>search string - 検索して置換する文字列、大文字と小文字を区別（必須）</p> <p>replacement - 検索文字列を置き換える文字列。置換文字列が空の場合は見つかった検索文字列を削除します。</p> <p>エスケープシーケンスを使用して、改行、キャリッジリターン、タブ、スペース "\n \r \t \s" を検索または置換することができます。バックslashは「\」としてエスケープでき、エスケープシーケンスは "\\n" としてエスケープできます。改行、キャリッジリターン、タブのエスケープは、ローレベルディスクバリエーション中に自動的に実行されます。</p> <p>5.0.0 以降でサポートされています。</p>
構造化データ	JSONPath	<p>JSONPath 機能を使用して JSON データから値またはフラグメントを抽出します。</p> <p>失敗時のカスタマイズチェックボックスをオンにすると、前処理ステップが失敗した場合に実行されるカスタムエラー処理オプションを指定できます。値を破棄するか、設定する値を指定するか、設定するエラーメッセージを指定してください。</p>
	XML XPath	<p>XPath 機能を使用して XML データから値またはフラグメントを抽出します。</p> <p>このオプションを機能させるには、Zabbix サーバーを libxml サポートでコンパイルする必要があります。</p> <p>例：</p> <pre>number(/ document/item/value) から 10 を抽出 <document><item><value>10</ value></ item></ document></pre> <pre>number(/document/item/@attribute) から 10 を抽出 <document><item attribute="10"></ item></ document></pre> <pre>/document/item から <item><value>10</value></item> を抽出 <document><item><value>10</ value></ item></document></pre> <p>名前付けはサポートされていないことに注意してください。</p> <p>4.4.0 以降でサポートされています。</p> <p>失敗時のカスタマイズチェックボックスをオンにすると、前処理ステップが失敗した場合に実行されるカスタムエラー処理オプションを指定できます。値を破棄するか、設定する値を指定するか、設定するエラーメッセージを指定してください。</p>
	CSV から JSON	<p>CSV ファイルデータを JSON 形式に変換します。</p> <p>詳細については CSV から JSON への前処理 を参照してください。</p> <p>4.4.0 以降でサポートされています。</p>
	XML から JSON	<p>XML 形式のデータを JSON に変換します。</p> <p>詳細については、シリアル化ルール を参照してください。</p> <p>失敗時のカスタマイズチェックボックスをオンにすると、カスタムエラー処理オプションを指定できます。値を破棄するか、設定する値を指定するか、設定するエラーメッセージを指定してください。</p>

タイプ	
カスタムスクリプト	<p>JavaScript</p> <p>パラメータフィールドをクリックしたとき、またはクリックして表示または変更をクリックしたときに表示されるブロックに JavaScript コードを入力します。 使用可能な JavaScript の長さは使用するデータベースによって異なります。
 詳細については、JavaScript の前処理を参照してください。</p>
バリデーション	<p>正規表現と一致しない</p> <p>値が一致してはならない正規表現を指定します。 例: <code>Error:(.*?)\.</code> 失敗時のカスタマイズチェックボックスをオンにすると、カスタムエラー処理オプションを指定できます: 値を破棄するか、指定された値を設定するか、指定されたエラーメッセージを設定します。</p> <p>JSON 内のエラーチェック</p> <p>JSONpath にあるアプリケーションレベルのエラーメッセージを確認します。成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この保存前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、保存前処理ステップ情報を追加せずに、そのままユーザーに報告されることに注意してください。 例: <code>\$.errors</code> が <code>{"errors":"e1"}</code> のような JSON を受信した場合、次の保存前処理ステップは実行されません。 * 失敗時のカスタマイズ * チェックボックスをオンにすると、カスタムエラー処理オプションを指定できます。値を破棄するか、設定する値を指定するか、設定するエラーメッセージを指定してください ^ *XML 内のエラーチェック * xpath にあるアプリケーションレベルのエラーメッセージを確認します。成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この保存前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、保存前処理ステップ情報を追加せずに、そのままユーザーに報告されることに注意してください。 無効な XML の解析に失敗しても、エラーは報告されません。 4.4.0 以降でサポートされています。 * 失敗時のカスタマイズ * チェックボックスをオンにすると、カスタムエラー処理オプションを指定できます。値を破棄するか、設定する値を指定するか、設定するエラーメッセージを指定してください 絞り込み < < < * 指定秒内に変化が無ければ破棄 * 定義された期間 (秒単位) 内に値が変更されなかった場合は、値を破棄します。 秒を指定するために正の整数値がサポートされています。(最小 - 1 秒) このフィールドではタイムサフィックスを使用できます。(30s,1m,2h,1d など) このフィールドではユーザーマクロとローレベルディスカバリマクロを使用できます。 ディスカバリアイテムに指定できるスロットルオプションは 1 つだけです。 例: 1m' 同一のテキストが 60 秒以内に 2 回このルールに渡された場合、そのテキストは破棄されます。 注: アイテムのプロトタイプを変更しても、スロットリングはリセットされません。スロットリングは保存前処理ステップが変更された場合にのみリセットされます。</p>
Prometheus	<p>Prometheus から JSON</p> <p>Prometheus メトリックを JSON に変換します。 詳細については Prometheus チェック を参照してください。</p>

テンプレートを介してディスカバリルールがホストに適用されている場合、このタブのコンテンツは読み取り専用であることに注意してください。

LLD マクロ

LLD マクロタブではカスタマイズしたローレベルディスカバリマクロを指定できます。

カスタムマクロは、返された JSON に必要なマクロがまだ定義されていない場合に役立ちます。例：

- ファイルシステム検出用のネイティブの `vfs.fs.discovery` キーは `{#FSNAME}`、`{#FSTYPE}` などの事前定義された LLD マクロを含む JSON を返します。これらのマクロは、アイテムで使用でき、プロトタイプをトリガーします（ページの後続のセクションを参照）。カスタムマクロを定義する必要はありません。
- `vfs.fs.get` エージェントアイテムは **ファイルシステムデータ** を含む JSON を返しますが、事前定義された LLD マクロはありません。この場合マクロを自分で定義し、JSONPath を使用して JSON の値にマップすることができます。

The screenshot shows the 'LLD macros' tab in the configuration interface. It features a table with two columns: 'LLD macro' and 'JSONPath'. Two entries are listed: one with the macro `{#FSNAME}` and JSONPath `$.fsname`, and another with `{#FSTYPE}` and `$.fstype`. Below the table is an 'Add' button with a plus sign icon.

抽出された値は、ディスカバリアイテム、トリガーなどで使用できます。値は、検出の結果とこれまでの前処理ステップから抽出されることに注意してください。

パラメータ	説明
LLD マクロ	次の構文を使用したローレベルディスカバリマクロの名前： <code>{#MACRO}</code>
JSONPath	JSONPath 構文を使用して LLD 行から LLD マクロ値を抽出するために使用されるパス。 たとえば、 <code>\$.foo</code> はこの JSON から "bar" と "baz" を抽出します： <code>[{"foo": "bar"}, {"foo": "baz"}]</code> 返された JSON から抽出された値は、アイテム、トリガーなどのプロトタイプフィールドの LLD マクロを置き換えるために使用されます。 JSONPath はドット表記またはブラケット表記を使用して指定できます。 <code>['unicode + special chars#1']['unicode + special chars#2']</code> のように、特殊文字や Unicode の場合は、角かっこ表記を使用する必要があります。

フィルター

フィルターを使用して、条件に一致するエンティティのアイテム、トリガー、およびグラフを生成できます。フィルタータブには、検出値をフィルター処理できる検出ルールフィルター定義が含まれています。

The screenshot shows the 'Filters' tab in the configuration interface. At the top, 'Type of calculation' is set to 'And' with a dropdown arrow and the text '(A and B) and (C and D)'. Below this is a table with the following structure:

Filters	Label	Macro		Regular expression
A	<code>{#FSNAME}</code>	matches	▼	<code>{\$VFS.FS.FSNAME.MATCH</code>
B	<code>{#FSNAME}</code>	does not match	▼	<code>{\$VFS.FS.FSNAME.NOT_M</code>
C	<code>{#FSTYPE}</code>	matches	▼	<code>{\$VFS.FS.FSTYPE.MATCH</code>
D	<code>{#FSTYPE}</code>	does not match	▼	<code>{\$VFS.FS.FSTYPE.NOT_M</code>

Below the table is an 'Add' button with a plus sign icon.

パラメータ	説明
計算のタイプ	<p>フィルタを計算するための次のオプションを使用できます。</p> <p>And - すべてのフィルタに合格する必要があります;</p> <p>Or - 1つのフィルタに合格すれば十分です;</p> <p>** And/Or ** - 異なるマクロ名で And を使用し、同じマクロ名で Or を使用します;</p> <p>カスタム条件式 - フィルタのカスタム計算を定義します。数式にはリスト内のすべてのフィルタが含まれている必要があります。255 シンボルに制限されています。</p>
フィルター	<p>次のフィルター条件演算子を使用できます。一致する、一致しない、存在する、存在しない。</p> <p>一致するおよび一致しない演算子は、Perl 互換正規表現(PCRE)を受け付けます。たとえば C:,D:, および E: ファイルシステムのみに関心がある場合、{#FSNAME} を "マクロ" に、"^C ^D ^E" を "正規表現" に入れることで実現できます。</p>
<p>フィルタリングは、 {#FSTYPE} マクロを使用するファイルシステムタイプ (例:"^ext ^reiserfs") および (Windows エージェントのみでサポートされる){#FSDRIVETYPE} マクロを使用するドライブタイプ (例:"fixed")</p> <p>正規表現フィールドに正規表現を入力するか、グローバル正規表現を参照できます。</p> <p>正規表現のテストには "grep -E" を使用できます。たとえば、次のようになります。</p> <pre>for f in ext2 nfs reiserfs smbfs; do echo \$f grep -E '^ext ^reiserfs' echo "SKIP: \$f"; done</pre>	<p>Windows の {#FSDRIVETYPE} マクロは Zabbix3.0.0 以降でサポートされています。</p> <p>存在するおよび存在しない演算子を使用すると、応答内の指定された LLD マクロの有無に基づいてエンティティをフィルタリングできます (バージョン 5.4.0 以降でサポートされています)</p> <p>Zabbix ** 2.4.0 ** 以降、複数のフィルターの定義がサポートされています。</p> <p>フィルターのマクロが応答にない場合、このマクロに存在しない条件が指定されていない限り、見つかったエンティティは無視されることに注意してください。</p>

Warning:

LLD ルールで使用される正規表現の誤りまたはタイプミス（たとえば誤った“ファイルシステムディスクカバリ”正規表現）により、多くのホストの数千の構成要素、履歴値、およびイベントが削除される可能性があります。

Attention:

大文字と小文字を区別するファイルシステム名を正しく検出するためには、MySQL の Zabbix データベースを大文字と小文字を区別するように作成する必要があります。

オーバーライド

オーバーライドタブでは、アイテム、トリガー、グラフ、ホストのプロトタイプのリスト、または特定の条件を満たす検出されたオブジェクトの属性をルールに基づいて変更できます。

オーバーライドは、（存在する場合）並べ替え可能なドラッグアンドドロップリストに表示され、定義された順序で実行されます。新しいオーバーライドの詳細を構成するには、オーバーライドブロックの **Add** をクリックします。既存のオーバーライドを編集するには、オーバーライド名をクリックすると、ポップアップウィンドウが開き、オーバーライドルールの詳細を編集できます。

必須パラメーターは赤いアスタリスクでマークされています。

パラメータ	説明
名前	一意の（LLD ルールごとの）オーバーライド名
フィルターが一致した場合	フィルタ条件が満たされたときに次のオーバーライドを処理するかどうかを設定します。上書きを続行 - 後続のオーバーライドが処理されます。処理の停止 - 先行する操作（存在する場合）およびこのオーバーライドは実行されると、一致した LLD 行の後続のオーバーライドは実行されません。

パラメータ	説明
フィルター	検出されたエンティティにオーバーライドを適用する必要があるかどうかを決定します。オーバーライドフィルターは、ディスカバリルール フィルター の後に処理されます。
実行内容	オーバーライド操作は、次の詳細とともに表示されます。 トリガー条件式 - オブジェクト (アイテムのプロトタイプ/トリガーのプロトタイプ/グラフのプロトタイプ/ホストプロトタイプ) およびトリガー条件式 (等しい/等しくない/含む/含まない/一致する/一致しない) アクション - 実行内容を変更または削除するためのリンクが表示されます。

実行内容の設定

新しい実行内容を設定するには、実行内容ブロックの **Add** をクリックします。既存の実行内容を編集するには各実行内容の **Edit** をクリックすると詳細を編集できるポップアップウィンドウが開きます。

New operation

Object

Condition

Create enabled Original

Discover

Severity Original

Tags Original

Add

パラメータ	説明
オブジェクト	次の 4 種類のオブジェクトを使用できます。 アイテムのプロトタイプ トリガーのプロトタイプ グラフのプロトタイプ ホストプロトタイプ
トリガー条件式	操作を適用する必要があるエンティティをフィルタリングします。
条件式	サポートされている条件： 等しい - パターンと等しいプロトタイプに適用 等しくない - パターンと等しくないプロトタイプに適用 含む - プロトタイプ名に文字列パターンが含まれる場合に適用 含まない - プロトタイプ名に文字列パターンが含まれていない場合は適用 一致する - プロトタイプ名がパターン内の正規表現と一致する場合は適用します 一致しない - プロトタイプ名がパターン内の正規表現と一致しない場合に適用

パターンオブジェクト: アイテムのプロトタイプ

自動生成時のステータス有効	<p>正規表現または検索する文字列</p> <p>チェックボックスをオンにするとボタンが表示され、元のアイテムのプロトタイプ設定を上書きできます。</p> <p>はい - アイテムは有効な状態で追加されます。</p>
ディスカバリ	<p>いいえ - アイテムは検出されたエンティティに追加されますが、無効な状態になります。</p> <p>チェックボックスをオンにするとボタンが表示され、元のアイテムのプロトタイプ設定を上書きできます。</p> <p>はい - アイテムが追加されます。</p> <p>いいえ - アイテムは追加されません。</p>
監視間隔	<p>チェックボックスをオンにすると 2 つのオプションが表示され、アイテムに異なる監視間隔を設定できます。</p> <p>間隔 - アイテムの更新間隔。ユーザーマクロおよびタイムサフィックス (例: 30s,1m,2h,1d) がサポートされています。監視間隔のカスタマイズを使用する場合は 0 に設定する必要があります。</p> <p>監視間隔のカスタマイズ - Add をクリックして、柔軟なスケジュール間隔を指定します。詳細については、監視間隔のカスタマイズ を参照してください。</p>
履歴の保存期間	<p>チェックボックスをオンにするとボタンが表示され、異なる履歴の保存期間を設定できます。</p> <p>履歴を保存しない - 履歴は保存されません。</p> <p>保存期間 - 選択すると保存期間を指定するための入力フィールドが右側に表示されます。ユーザーマクロと LLD マクロがサポートされています。</p>
トレンドの保存期間	<p>チェックボックスをオンにするとボタンが表示され、異なるトレンドの保存期間を設定できます。</p> <p>トレンドを保存しない - トレンドは保存されません。</p> <p>保存期間 - 選択すると保存期間を指定するための入力フィールドが右側に表示されます。ユーザーマクロと LLD マクロがサポートされています。</p>
タグ	<p>チェックボックスをオンにすると新しいブロックが表示され、タグと値のペアを指定できます。</p> <p>これらのタグはタグ名が一致していても、アイテムのプロトタイプで指定されたタグに追加されます。</p>

オブジェクト:
トリガーの
プロトタイプ

自動生成時のステータス有効	チェックボックスをオンにするとボタンが表示され、元のトリガーのプロトタイプ設定を上書きできます。 はい - トリガーは有効な状態で追加されます。
ディスカバリ	いいえ - トリガーは検出されたエンティティに追加されますが、無効な状態になります。 チェックボックスをオンにすると、ボタンが表示され、元のトリガーのプロトタイプ設定を上書きできます。 はい - トリガーが追加されます。
深刻度	いいえ - トリガーは追加されません。 チェックボックスをオンにするとトリガーの深刻度ボタンが表示され、深刻度を変更できます。
タグ	チェックボックスをオンにすると新しいブロックが表示され、タグと値のペアを指定できます。 これらのタグはタグ名が一致していても、トリガープロトタイプで指定されたタグに追加されます。

オブジェクト:
グラフの
プロトタイプ

ディスカバリ	チェックボックスをオンにするとボタンが表示され、元のグラフのプロトタイプ設定を上書きできます。 はい - グラフが追加されます。 いいえ - グラフは追加されません。
--------	---

オブ
ジ
ェ
ク
ト：
ホ
ス
ト
プ
ロ
ト
タ
イ
プ

自動生成時のステータス有効	チェックボックスをオンにするとボタンが表示され、元のホストプロトタイプ設定を上書きできます。 はい - ホストは有効な状態で追加されます。 いいえ - ホストは無効な状態で追加されます。
ディスクバリ	チェックボックスをオンにするとボタンが表示され、元のホストプロトタイプ設定を上書きできます。 はい - ホストが検出されます。 いいえ - ホストは検出されません。
テンプレートをリンク	チェックボックスをオンにするとテンプレートを指定するための入力フィールドが表示されます。テンプレート名を入力するか、フィールドの横にある選択をクリックして、ポップアップウィンドウのリストからテンプレートを選択します。 ホストプロトタイプにリンクされているすべてのテンプレートは、このオーバーライドのテンプレートに置き換えられます。
タグ	チェックボックスをオンにすると、新しいブロックが表示され、タグと値のペアを指定できます。 これらのタグは、タグ名が一致していても、ホストプロトタイプで指定されたタグに追加されます。
ホストインベントリ	チェックボックスをオンにするとボタンが表示され、ホストプロトタイプの別のインベントリモードを選択できます。 無効 - ホストインベントリにデータを入力しない マニュアル - 詳細を手動で入力します 自動 - 収集されたメトリックに基づいてホストインベントリデータを自動入力します。

フォームボタン

フォームの下部にあるボタンを使用すると、いくつかの操作を実行できます。

Add

ディスクバリルールを追加します。このボタンは新しいディスクバリルールでのみ使用できます。

Update

ディスクバリルールのプロパティを更新します。このボタンは既存のディスクバリルールでのみ使用できます。

Clone

現在のディスクバリルールのプロパティに基づいて、別のディスクバリルールを作成します。

Check now

ディスクバリルールに基づいて検出をすぐ実行します。ディスクバリルールはすでに存在している必要があります。[詳細を参照してください](#)。
注すぐに検出を実行すると、設定キャッシュは更新されないため、結果にはディスクバリルール設定の最近の変更が反映されません。

ディスカバリルールを削除します。

ディスカバリルールのプロパティ編集をキャンセルします。

検出されたエンティティ 以下のスクリーンショットは、検出されたアイテム、トリガー、およびグラフがホストの構成でどのように見えるかを示しています。検出されたエンティティには、元のディスカバリルールへのオレンジ色のリンクがプレフィックスとして付けられます。

Items				
All hosts / Remote proxy: New host Enabled ZBX SNMP JMX IPMI Applications 11 Items 41				
<input type="checkbox"/>	Wizard	Name	Triggers	Key
<input type="checkbox"/>	...	Mounted filesystem discovery: Free disk space on / (percentage)	Triggers 1	vfs.fs.size[/,pfr
<input type="checkbox"/>	...	Mounted filesystem discovery: Used disk space on /		vfs.fs.size[/,use
<input type="checkbox"/>	...	Mounted filesystem discovery: Free disk space on /		vfs.fs.size[/,fre
<input type="checkbox"/>	...	Mounted filesystem discovery: Free inodes on / (percentage)	Triggers 1	vfs.fs.inode[/,p

同じキーのアイテムや同じ名前のグラフなど、同じ一意性基準を持つ既存のエンティティがすでに存在する場合、検出されたエンティティは作成されないことに注意してください。この場合、フロントエンドにローレベルディスカバリルールが特定のエンティティを作成できなかったことを示すエラーメッセージが表示されます。ただ、一部のエンティティを作成できずにスキップする必要があったためであり、ディスカバリルール自体がサポートされなくなることはありません。ディスカバリルールは、他のエンティティの作成/更新を続行します。

ローレベルディスカバリルールによって作成されたアイテム（同様に、トリガーとグラフ）は、検出されたエンティティ（ファイルシステム、インターフェイスなど）が検出されなくなった場合（またはフィルターを通過しなくなった場合）、アイテム、トリガー、およびグラフは存在しなくなったりソースの保持期間フィールドで定義された日数が経過すると自動的に削除されます。

検出されたエンティティが「もう検出されていません」になると、アイテムリストに有効期間インジケータが表示されます。マウスポインタをその上に移動すると、アイテムが削除されるまでの残り日数を示すメッセージが表示されます。

1m	7d	1y	Zabbix agent	Enabled	
The item is not discovered anymore and will be deleted in 29d 23h 44m (on 2015-08-31 at 23:27).					

エンティティに削除のマークが付けられているが、予定された時間に削除されなかった場合（無効なディスカバリルールまたはアイテムホスト）次にディスカバリルールが処理されるときにエンティティは削除されます。

削除のマークが付けられている他のエンティティを含むエンティティは、ディスカバリルールレベルで変更されても更新されません。たとえば、LLD ベースのトリガーには、削除のマークが付けられたアイテムが含まれている場合、更新されません。

Triggers			Group
			all
All hosts / Remote proxy: New host Enabled ZBX SNMP JMX IPMI Applications 11 Items 41			
<input type="checkbox"/>	Severity	Name ▲	
<input type="checkbox"/>	Warning	Mounted filesystem discovery: Free disk space is less than 20% on volume /	
<input type="checkbox"/>	Warning	Mounted filesystem discovery: Free inodes is less than 20% on volume /	

Graphs

Group

All hosts / Remote proxy: New host Enabled **ZBX** SNMP JMX IPMI Applications 11 Items 41 T

- Name ▲
- [Template OS Linux: CPU jumps](#)
- [Template OS Linux: CPU load](#)
- [Template OS Linux: CPU utilization](#)
- [Mounted filesystem discovery: Disk space usage /](#)

他のタイプの検出 次のセクションでは、他のタイプのすぐに使用可能な検出の詳細とハウツーを利用できます。

- ネットワークインターフェースの検出;
- CPU と CPU コアの検出;
- SNMP OIDsの検出;
- JMX オブジェクトの検出;
- ODBC SQL クエリを使った検出;
- Windows サービスの検出;
- Zabbix のホストインターフェース検出;

検出項目の JSON 形式の詳細と、独自のファイルシステム検出器を Perl スクリプトとして実装する方法の例については、[カスタム LLD ルールの作成](#) を参照してください。

1 アイテムのプロトタイプ

ルールが作成されたら、そのルールのアイテムに移動し、“アイテムプロトタイプの作成” を押してアイテムプロトタイプを作成します。

ファイルシステム名が必要な場所で、{#FSNAME} マクロがどのように使用されているかに注意してください。ディスカバリーが正しく処理されるようにするには、項目キーでローレベルディスカバリマクロを使用することは必須です。ディスカバリルールが処理されると、このマクロは検出されたファイルシステムに置き換えられます。

Item prototype Tags Preprocessing

* Name

Type

* Key

Type of information

Units

* Update interval

Custom intervals

Type	Interval	Period
Flexible Scheduling	50s	1-7,00:00-24:00

[Add](#)

* History storage period

* Trend storage period

Value mapping

Description

Create enabled

Discover

ローレベルディスクカバリマクロとユーザーマクロは、アイテムプロトタイプの設定とアイテム値の前処理パラメーターでサポートされています。更新間隔で使用する場合、1つのマクロでフィールド全体を埋める必要があることに注意してください。1つのフィールド内に複数のマクロ、またはテキストと混合されたマクロはサポートされていません。

Note:

ローレベルディスクカバリマクロのコンテキスト固有エスケープは、正規表現および XPath 前処理パラメーターで安全に使用するために実行されます。

アイテムプロトタイプ固有属性:

パラメータ	説明
自動生成時のステータス有効	チェックされている場合、アイテムは検出されたエンティティに有効な状態で追加されます。チェックされていない場合、アイテムは検出されたエンティティに追加されますが、無効になります。

パラメータ	説明
ディスカバリ	チェックされている場合 (デフォルト)、アイテムは検出されたエンティティに追加されます。チェックを外すと、検出ルールでこの設定が 上書き されていない限り、アイテムは検出されたエンティティに追加されません。

関心のあるファイルシステムメトリックごとに、いくつかのアイテムプロトタイプを作成できます:

☰ Item prototypes

All templates / Template Module Windows filesystem... Discovery list / Mounted filesystem discovery

Item prototypes 3 Trigger prototypes 2 Graph prototypes 1 Host prototypes

<input type="checkbox"/>	Name ▲	Key	Interval
<input type="checkbox"/>	... {#FSNAME}: Space utilization	vfs.fs.size[{#FSNAME},pused]	1m
<input type="checkbox"/>	... {#FSNAME}: Total space	vfs.fs.size[{#FSNAME},total]	1m
<input type="checkbox"/>	... {#FSNAME}: Used space	vfs.fs.size[{#FSNAME},used]	1m

0 selected

三点アイコンをクリックすると、特定のアイテムのプロトタイプのメニューが開き、次のオプションが表示されます: - Create trigger prototype - このアイテムプロトタイプに基づいてトリガープロトタイプを作成する - Trigger prototypes - クリックして、このアイテムプロトタイプの構成済みトリガープロトタイプへのリンクを含むリストを表示します - 依存アイテムの作成 - このアイテムプロトタイプの依存アイテムを作成する

一括更新 オプションは、複数のアイテムプロトタイプのプロパティを一度に更新する場合に使用できます。

2 トリガーのプロトタイプ

アイテムのプロトタイプと同様の方法でトリガーのプロトタイプを作成します。

Trigger prototype **Dependencies**

* Name

Severity

* Expression

[Expression constructor](#)

OK event generation

PROBLEM event generation mode

OK event closes

Tags

<input type="text" value="tag"/>	<input type="text" value="value"/>
----------------------------------	------------------------------------

[Add](#)

Allow manual close

URL

Description

Create enabled

Discover

トリガーのプロトタイプ固有の属性：

パラメータ	説明
自動生成時のステータス有効	オンにすると、トリガーは有効な状態で追加されます。オフにすると、トリガーは検出されたエンティティに追加されませんが、無効な状態になります。

パラメータ	説明
ディスカバリ	オンにすると (デフォルト)、トリガーは検出されたエンティティに追加されます。 チェックをオフにすると、この設定が検出ルールで上書きされない限りトリガーは検出されたエンティティに追加されません。

プロトタイプから実際のトリガーを作成する場合、式の比較にどの定数 (この例では「20」) を使用するかについて柔軟に対応する必要があります。コンテキスト付きのユーザーマクロがこのような柔軟性を実現するためにどのように役立つかをご確認ください。

トリガープロトタイプ間で依存関係を定義することもできます。(Zabbix 3.0以降でサポート) これを行うには依存関係タブに移動します。トリガープロトタイプは、同じローレベルディスカバリ (LLD) ルールから別のトリガープロトタイプまたは通常のトリガーに依存する場合があります。トリガープロトタイプは、別の LLD ルールのトリガープロトタイプまたはトリガープロトタイプから作成されたトリガーに依存しない場合があります。ホストトリガーのプロトタイプは、テンプレートからのトリガーに依存することはできません。

The screenshot shows the 'Trigger prototypes' page in Zabbix. The breadcrumb trail is: All templates / Template OS Linux / Discovery list / Mounted filesystem discovery / Item prototypes 5. The table below lists two triggers:

<input type="checkbox"/>	SEVERITY	NAME ▲	EXPRESSION
<input type="checkbox"/>	Warning	Free disk space is less than 20% on volume {#FSNAME}	{Template OS
<input type="checkbox"/>	Warning	Free inodes is less than 20% on volume {#FSNAME}	{Template OS

3 グラフのプロトタイプ

グラフのプロトタイプを作成することもできます。

Graph prototype
Preview

* Name

* Width

* Height

Graph type

Show legend

3D view

* Items

Name	Type
1: Template Module Linux filesystems by Zabbix agent: {#FSNAME}: Total space	Graph
2: Template Module Linux filesystems by Zabbix agent: {#FSNAME}: Used space	Simple

[Add](#) [Add prototype](#)

Discover

グラフのプロトタイプ固有の属性：

パラメータ	説明
ディスカバリ	オンにすると（デフォルト）グラフは検出されたエンティティに追加されます。 チェックをオフにすると、この設定が検出ルールで上書きされない限りグラフは検出されたエンティティに追加されません。

Graph prototypes

All templates / Template OS Linux Discovery list / Mounted filesystem discovery Item prototypes 5

<input type="checkbox"/> NAME ▲	WIDTH
<input type="checkbox"/> Disk space usage {#FSNAME}	600

これまでの例で最終的に、次のようなディスカバリルールが作成されます。5つのアイテムのプロトタイプ、2つのトリガーのプロトタイプ、および1つのグラフのプロトタイプがあります。

Discovery rules

All templates / Template Module Linux filesystems... Items Triggers Graphs Dashboards **Disco**

<input type="checkbox"/>	Template	Name ▲	Items
<input type="checkbox"/>	Template Module Linux filesystems by Zabbix agent	Mounted filesystem discovery	Item prototypes 4

注：ホストプロトタイプ構成については、仮想マシンの監視における[ホストプロトタイプ](#)の設定に関するセクションを参照してください。

4 ローレベルディスカバリにおける注意

ユーザーマクロコンテキストでの LLD マクロの使用

LLD マクロは、[トリガープロトタイプ](#)など、ユーザーマクロコンテキスト内で使用できます。

同じアイテムに対する複数の LLD ルール

Zabbix エージェントバージョン 3.2 以降、同じディスカバリ項目で複数のローレベルディスカバリルールを定義することが可能です。

これにはエイリアスエージェント[パラメータ](#)を定義する必要があります。これによりさまざまな検出ルールで変更された検出アイテムキーを使用できるようになります。たとえば `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` などです。

戻り値のデータ制限

ローレベルディスカバリルールの JSON データが Zabbix サーバーによって直接受信される場合、戻り値はデータベースに保存されずに処理されるため、制限はありません。カスタムのローレベルディスカバリルールにも制限はありませんが、ユーザーパラメータを使用してカスタム LLD データを取得する場合は、ユーザーパラメータの戻り値の制限 (512 KB) が適用されます。

データが Zabbix プロキシを経由する必要がある場合は、このデータをデータベースに保存する必要があるため[データベース制限](#)が適用されます。

5 ディスカバリルール

サイドバーからさまざまなディスカバリルールの設定例を確認してください。

1 ファイルシステムマウントの検出

概要

マウントされたファイルシステムとそのプロパティ (マウントポイント名、マウントポイントタイプ、ファイルシステムサイズ、iノード統計) を検出できます。

これを行うには、次の組み合わせを使用します。

- マスターアイテムとして `vfs.fs.get` エージェントアイテム
- ローレベルディスカバリルールとアイテムのプロトタイプ

設定

マスターアイテム

次のキーを使用して Zabbix エージェントアイテムを作成します。

`vfs.fs.get`

Item	Tags	Preprocessing
* Name	vfs.fs.get item	
Type	Zabbix agent	
* Key	vfs.fs.get	
* Host interface	127.0.0.1 : 10050	
Type of information	Text	

JSON データが大きくなる可能性がある場合は、データ型を"テキスト"に設定します。

このアイテムによって返されるデータ、マウントされたファイルシステムには、次のようなものが含まれます。

```
{
  "fsname": "/",
  "fstype": "rootfs",
  "bytes": {
    "total": 1000,
    "free": 500,
    "used": 500,
    "pfree": 50.00,
    "pused": 50.00
  },
  "inodes": {
    "total": 1000,
    "free": 500,
    "used": 500,
    "pfree": 50.00,
    "pused": 50.00
  }
}
```

依存 LLD ルール

"依存アイテム"タイプとしてローレベルディスクバリルールを作成します。

Discovery rule	Preprocessing	LLD macros	Filters	Overrides
* Name	Discovery rule for vfs.fs.get			
Type	Dependent item			
* Key	fs.mountpoint.discovery			
* Master item	Zabbix server: vfs.fs.get item			
* Keep lost resources period	30d			

マスターアイテムとして、作成した `vfs.fs.get` アイテムを選択します。

“LLD マクロ” タブで、対応する JSONPath を使用してカスタムマクロを定義します。

LLD macro	JSONPath
{#FSNAME}	\$.fsname
{#FSTYPE}	\$.fstype

[Add](#)

依存アイテムのプロトタイプ

この LLD ルールで“依存アイテム”タイプのアイテムプロトタイプを作成します。このプロトタイプのマスターアイテムとして、作成した `vfs.fs.get` アイテムを選択します。

* Name: Free disk space on {#FSNAME}, type: {#FSTYPE}

Type: Dependent item

* Key: free[{#FSNAME}]

* Master item: Zabbix server: vfs.fs.get item

Type of information: Numeric (unsigned)

アイテムのプロトタイプ名とキーでカスタムマクロが使用されていることに注意してください。

- Name: Free disk space on {#FSNAME}, type: {#FSTYPE}
- Key: Free[{#FSNAME}]

情報の種類として、次を使用します。

- 数値 (整数) 次のメトリックの場合 'free', 'total', 'used'
- 数値 (浮動小数) 次のメトリックの場合 'pfree', 'pused' (percentage)

アイテムのプロトタイプの“保存前処理”タブで、JSONPath を選択し、次の JSONPath 式をパラメーターとして使用します。

```
$. [?(@.fsname=='{#FSNAME}')].bytes.free.first()
```

Preprocessing steps	Name	Parameters
1:	JSONPath	\$. [?(@.fsname=='{#FSNAME}')].bytes.free.first()

[Add](#)

検出されるとマウントポイントごとに 1 つのアイテムが作成されます。この項目は、指定されたマウントポイントの空きバイト数を返します。

2 ネットワークインターフェースの検出

ファイルシステムが検出されるのと同様の方法で、ネットワークインターフェイスも検出できます。

アイテムキー

検出ルールで使用するアイテムキーは次のとおりです。

`net.if.discovery`

このアイテムは Zabbix エージェント 2.0 以降でサポートされています。

サポートされているマクロ

ディスカバリルールフィルターで `{#IFNAME}` マクロを使用したり、アイテム、トリガー、グラフのプロトタイプを使用したりできます。

“net.if.discovery” に基づいて作成する可能性のあるアイテムのプロトタイプの例：

- “net.if.in[`{#IFNAME}`],bytes”
- “net.if.out[`{#IFNAME}`],bytes”

Windows では `{#IFGUID}` も返されることに注意してください。

3 CPU と CPU コアの検出

ファイルシステムが検出されるのと同様の方法で、CPU と CPU コアも検出できます。

アイテムキー

検出ルールで使用するアイテムキーは次のとおりです。

`system.cpu.discovery`

このアイテムは、Zabbix エージェント 2.4 以降でサポートされています。

サポートされているマクロ

このディスカバリキーは、CPU モデル番号とステータスをそれぞれ識別する `{#CPU.NUMBER}` と `{#CPU.STATUS}` の 2 つのマクロを返します。実際の物理プロセッサ、コア、ハイパースレッドを明確に判別できないことに注意してください。Linux、UNIX、および BSD システムの `{#CPU.STATUS}` はプロセッサのステータスを返します。これは“online”または“offline”のいずれかになります。Windows システムではこの同じマクロが 3 番目の値 (“unknown”) を表す場合があります。これは、プロセッサが検出されたもののそのプロセッサに関する情報がまだ収集されていないことを示します。

CPU ディスカバリは、エージェントのコレクタープロセスに依存して、コレクターによって提供されるデータとの整合性を維持し、データの取得にかかるリソースを節約します。これには、このアイテムキーがエージェントバイナリのテスト (-t) コマンドラインフラグで機能しないという効果があります。これにより、NOT_SUPPORTED ステータスと、コレクタプロセスが開始されていないことを示す付随メッセージが返されます。

CPU の検出に基づいて作成できるアイテムのプロトタイプには、たとえば次のものがあります。

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

アイテムキーの詳細な説明については、[Zabbix エージェントアイテムキー](#)を参照してください。

4 SNMP OIDs の検出

概要

このセクションでは、スイッチで SNMP の検出を実行します。

アイテムキー

ファイルシステムやネットワークインターフェイスの検出とは異なり、アイテムは必ずしも“snmp.discovery”キーを持っている必要はありません。SNMP エージェントのアイテムタイプで十分です。

SNMP OID の検出は、Zabbix サーバー/プロキシ 2.0 以降でサポートされています。

ディスカバリルールを設定するには、次のようにします。

- 設定 → テンプレートを開きます
- 適当なテンプレート行のディスカバリーをクリックします

≡ Templates

Name ▲ Hosts Items Triggers Graphs Dashboards Discovery

Interfaces SNMP Hosts Items Triggers Graphs Dashboards 1 Discovery 1

- 画面の右上隅にある [ディスカバリルールの作成] をクリックします
- 以下のスクリーンショットのように、必要な詳細をディスカバリルールフォームに入力します

Discovery rule Preprocessing LLD macros Filters 12 Overrides

* Name

Type

* Key

* SNMP OID

* Update interval

Custom intervals

Type	Interval	Period
Flexible Scheduling	50s	1-7,00:00-24:00

[Add](#)

* Keep lost resources period

Description

必須入力フィールドは、赤いアスタリスクでマークされています。

検出する OID は SNMP OID フィールドで次の形式で定義されます : `discovery[{{#MACRO1}}, oid1, {{#MACRO2}}, oid2, ...]`

ここでの `{{#MACRO1}}`, `{{#MACRO2}}`... は有効な lld マクロ名であり、`oid1`, `oid2`... はこれらのマクロに意味のある値を生成できる OID です。検出された OID のインデックスを含む組み込みマクロ `{{#SNMPINDEX}}` が、検出されたエンティティに適用されます。検出されたエンティティは `{{#SNMPINDEX}}` マクロ値によってグループ化されます。

何を意味するのかを理解するために、スイッチでいくつかの `snmpwalk` を実行してみましょう。

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2
```

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

そして SNMP OID を次のように設定します : `discovery[{{#IFDESCR}}, ifDescr, {{#IFPHYSADDRESS}}, ifPhysAddress]`

このルールは次のエンティティを検出します。 `{{# IFDESCR}}` マクロ : **WAN, LAN1, LAN2** `{{# IFPHYSADDRESS}}` マクロ : **8:0:27:90:7a:75, 8:0:27:90:7a:76, 8:0:27:2b:af:9e** `{{# SNMPINDEX}}` マクロ : **1, 2, 3**

```
[
  {
    "{{#SNMPINDEX}}": "1",
    "{{#IFDESCR}}": "WAN",
```

```

    "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
  },
  {
    "{#SNMPINDEX}": "2",
    "{#IFDESCR}": "LAN1",
    "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
  },
  {
    "{#SNMPINDEX}": "3",
    "{#IFDESCR}": "LAN2",
    "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
  }
]

```

エンティティに指定された OID がない場合は、対応するマクロはこのエンティティで省略されます。たとえば、次のデータがある場合：

```

ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

```

```

ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"

```

この場合、SNMP ディスカバリ discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias] は次の構造を返します。

```

[
  {
    "{#SNMPINDEX}": 1,
    "{#IFDESCR}": "Interface #1",
    "{#IFALIAS}": "eth0"
  },
  {
    "{#SNMPINDEX}": 2,
    "{#IFDESCR}": "Interface #2",
    "{#IFALIAS}": "eth1"
  },
  {
    "{#SNMPINDEX}": 3,
    "{#IFALIAS}": "eth2"
  },
  {
    "{#SNMPINDEX}": 4,
    "{#IFDESCR}": "Interface #4"
  },
  {
    "{#SNMPINDEX}": 5,
    "{#IFALIAS}": "eth4"
  }
]

```

アイテムプロトタイプ

次のスクリーンショットは、アイテムのプロトタイプでこれらのマクロを使用する方法を示しています。

Item prototype Tags Preprocessing 2

* Name

Type

* Key

Type of information

* SNMP OID

Units

* Update interval

繰り返して必要な数のアイテムのプロトタイプを作成します。

☰ Item prototypes

All templates / Linux SNMP Discovery list / Network interfaces discovery **Item prototypes 9** Trigger prototypes 4 Graph prototypes 1 Host prototypes

<input type="checkbox"/>	Name ▲	Key	Interval	History	Trends	Type	Create enabled
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]]	3m	7d	365d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]]	3m	7d	365d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]]	3m	7d	365d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]]	3m	7d	365d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Interface type	net.if.type[ifType.{#SNMPINDEX}]]	1h	7d	0d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]]	1m	7d	0	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]]	3m	7d	365d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]]	3m	7d	365d	SNMP agent	Yes
<input type="checkbox"/>	... Interface {#IFNAME}{#IFALIAS}: Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]]	5m	7d	0d	SNMP agent	Yes

トリガーのプロトタイプ

次のスクリーンショットはトリガーのプロトタイプでこれらのマクロを使用する方法を示しています。

* Name Interface {#IFNAME}{#IFALIAS}: Link down

Event name Interface {#IFNAME}{#IFALIAS}: Link down

Operational data Current state: {ITEM.LASTVALUE1}

Severity Not classified Information Warning Average High Disaster

* Problem expression `{$IFCONTROL:"{#IFNAME}"}=1 and last(/SNMP host/net.if.status[ifOperStatus.{#SNMPINDEX}])=2 and (last(/SNMP host/net.if.status[ifOperStatus.{#SNMPINDEX}], #1)<>last(/SNMP host/net.if.status[ifOperStatus.{#SNMPINDEX}], #2))`

[Expression constructor](#)

OK event generation Expression Recovery expression None

* Recovery expression `last(/SNMP host/net.if.status[ifOperStatus.{#SNMPINDEX}])<>2 or {$IFCONTROL:"{#IFNAME}"}=0`

Trigger prototypes

Severity	Name	Operational data	Expression	Create enabled
Information	Interface {#IFNAME} ({#IFALIAS}): Ethernet has changed to lower speed than it was before Depends on: Linux SNMP: Interface {#IFNAME}{#IFALIAS}: Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: <code>change(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}])<0 and last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}])>0 and (last(/Linux SNMP/net.if.type[ifType.{#SNMPINDEX}])=6 or last(/Linux SNMP/net.if.type[ifType.{#SNMPINDEX}])=7 or last(/Linux SNMP/net.if.type[ifType.{#SNMPINDEX}])=11 or last(/Linux SNMP/net.if.type[ifType.{#SNMPINDEX}])=62 or last(/Linux SNMP/net.if.type[ifType.{#SNMPINDEX}])=69 or last(/Linux SNMP/net.if.type[ifType.{#SNMPINDEX}])=117) and (last(/Linux SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])<>2)</code> Recovery: <code>(change(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}])>0 and last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}], #2)>0) or (last(/Linux SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])=2)</code>	Yes
Warning	Interface {#IFNAME} ({#IFALIAS}): High bandwidth usage Depends on: Linux SNMP: Interface {#IFNAME}{#IFALIAS}: Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: <code>(avg(/Linux SNMP/net.if.in[ifHCInOctets.{#SNMPINDEX}], 15m)>{\${F.UTIL.MAX:"{#IFNAME}"})/100)*last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}]) or avg(/Linux SNMP/net.if.out[ifHCOutOctets.{#SNMPINDEX}], 15m)>{\${F.UTIL.MAX:"{#IFNAME}"})/100)*last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}]) and last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}])>0</code> Recovery: <code>avg(/Linux SNMP/net.if.in[ifHCInOctets.{#SNMPINDEX}], 15m)<{\${F.UTIL.MAX:"{#IFNAME}"})/100)*last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}]) and avg(/Linux SNMP/net.if.out[ifHCOutOctets.{#SNMPINDEX}], 15m)<{\${F.UTIL.MAX:"{#IFNAME}"})/100)*last(/Linux SNMP/net.if.speed[ifHighSpeed.{#SNMPINDEX}])</code>	Yes
Warning	Interface {#IFNAME} ({#IFALIAS}): High error rate Depends on: Linux SNMP: Interface {#IFNAME}{#IFALIAS}: Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: <code>min(/Linux SNMP/net.if.in.errors[ifInErrors.{#SNMPINDEX}], 5m)>{\${F.ERRORS.WARN:"{#IFNAME}"}) or min(/Linux SNMP/net.if.out.errors[ifOutErrors.{#SNMPINDEX}], 5m)>{\${F.ERRORS.WARN:"{#IFNAME}"})</code> Recovery: <code>max(/Linux SNMP/net.if.in.errors[ifInErrors.{#SNMPINDEX}], 5m)<{\${F.ERRORS.WARN:"{#IFNAME}"})*0.8 and max(/Linux SNMP/net.if.out.errors[ifOutErrors.{#SNMPINDEX}], 5m)<{\${F.ERRORS.WARN:"{#IFNAME}"})*0.8</code>	Yes
Average	Interface {#IFNAME} ({#IFALIAS}): Link down	Current state: {ITEM.LASTVALUE1}	Problem: <code>{\$IFCONTROL:"{#IFNAME}"}=1 and last(/Linux SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])=2 and (last(/Linux SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}], #1)<>last(/Linux SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}], #2))</code>	Yes

グラフのプロトタイプ

次のスクリーンショットは、グラフのプロトタイプでこれらのマクロを使用する方法を示しています。

Graph prototype [Preview](#)

* Name

* Width

* Height

Graph type

Show legend

Show working time

Show triggers

Percentile line (left)

Percentile line (right)

Y axis MIN value

Y axis MAX value

* Items

Name	Function	Draw style	Y axis side	Color
1: SNMP host: Interface {#IFNAME}({#IFALIAS}): Bits received	avg	Gradient line	Left	
2: SNMP host: Interface {#IFNAME}({#IFALIAS}): Bits sent	avg	Bold line	Left	
3: SNMP host: Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	avg	Line	Right	
4: SNMP host: Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	avg	Line	Right	
5: SNMP host: Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	avg	Line	Right	
6: SNMP host: Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	avg	Line	Right	

[Add](#) [Add prototype](#)

≡ Graph prototypes

All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 **Graph prototypes 1** Host prototypes

<input type="checkbox"/> Name ▲	Width	Height
<input type="checkbox"/> Interface {#IFNAME}({#IFALIAS}): Network traffic	900	200

ディスカバリルールの概要：

All templates / Linux SNMP Items 26 Triggers 10 Graphs 5 Dashboards 2 **Discovery rules 5** Web scenarios

<input type="checkbox"/> Template	Name ▲	Items	Triggers	Graphs
<input type="checkbox"/> Linux SNMP	Network interfaces discovery	Item prototypes 9	Trigger prototypes 4	Graph prototypes 1

発見されたエンティティ

サーバーが実行されると、SNMP ディスカバリルールが返す値に基づいて、実際のアイテム、トリガー、およびグラフが作成されます。ホスト設定では、プレフィックスとして、元のディスカバリルールへのオレンジ色のリンクが付けられます。

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios									
<input type="checkbox"/>	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled	
<input type="checkbox"/>	... Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled	

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios					
<input type="checkbox"/>	Severity	Value	Name ▲	Operational data	Expression
<input type="checkbox"/>	Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before Depends on: SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: change (/SNMP host/net.if.speed[ifHighSpeed.2])<0 and last (/SNMP host/net.if.speed[ifHighSpeed.2])>0 and (last (/SNMP host/net.if.type[ifType.2])=6 or last (/SNMP host/net.if.type[ifType.2])=7 or last (/SNMP host/net.if.type[ifType.2])=11 or last (/SNMP host/net.if.type[ifType.2])=62 or last (/SNMP host/net.if.type[ifType.2])=69 or last (/SNMP host/net.if.type[ifType.2])=117) and (last (/SNMP host/net.if.status[ifOperStatus.2])<>2) Recovery: (change (/SNMP host/net.if.speed[ifHighSpeed.2])>0 and last (/SNMP host/net.if.speed[ifHighSpeed.2],#2)>0) or (last (/SNMP host/net.if.status[ifOperStatus.2])=2)
<input type="checkbox"/>	Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage Depends on: SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (avg (/SNMP host/net.if.in[ifHCInOctets.2],15m)>({\$IF.UTIL.MAX:"enp4s0"}/100)* last (/SNMP host/net.if.speed[ifHighSpeed.2])) or (avg (/SNMP host/net.if.out[ifHCOutOctets.2],15m)>({\$IF.UTIL.MAX:"enp4s0"}/100)* last (/SNMP host/net.if.speed[ifHighSpeed.2])) and (last (/SNMP host/net.if.speed[ifHighSpeed.2])>0) Recovery: avg (/SNMP host/net.if.in[ifHCInOctets.2],15m)<(({\$IF.UTIL.MAX:"enp4s0"}-3)/100)* last (/SNMP host/net.if.speed[ifHighSpeed.2]) and avg (/SNMP host/net.if.out[ifHCOutOctets.2],15m)<(({\$IF.UTIL.MAX:"enp4s0"}-3)/100)* last (/SNMP host/net.if.speed[ifHighSpeed.2])
<input type="checkbox"/>	Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate Depends on: SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: min (/SNMP host/net.if.in.errors[ifInErrors.2],5m)>{\$IF.ERRORS.WARN:"enp4s0"} or min (/SNMP host/net.if.out.errors[ifOutErrors.2],5m)>{\$IF.ERRORS.WARN:"enp4s0"} Recovery: max (/SNMP host/net.if.in.errors[ifInErrors.2],5m)<{\$IF.ERRORS.WARN:"enp4s0"}*0.8 and max (/SNMP host/net.if.out.errors[ifOutErrors.2],5m)<{\$IF.ERRORS.WARN:"enp4s0"}*0.8
<input type="checkbox"/>	Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {\$IFCONTROL:"enp4s0"}=1 and last (/SNMP host/net.if.status[ifOperStatus.2])=2 and (last (/SNMP host/net.if.status[ifOperStatus.2],#1)<< last (/SNMP host/net.if.status[ifOperStatus.2],#2)) Recovery: last (/SNMP host/net.if.status[ifOperStatus.2])<>2 or {\$IFCONTROL:"enp4s0"}=0

≡ Graphs

All hosts / SNMP host Enabled **SNMP** Items 81 Triggers 23 **Graphs 14** Discovery rules 6 Web scenarios

- Name ▲
- Mounted filesystem discovery: /: Disk space usage
- Linux SNMP: CPU jumps
- CPU discovery: CPU usage
- CPU discovery: CPU utilization
- Network interfaces discovery: Interface enp4s0(): Network traffic

5 JMX オブジェクトの検出

概要

すべての JMX MBean または MBean 属性を検出したり、これらのオブジェクトの検出パターンを指定したりできます。

ディスカバリルールを設定するには、MBean と MBean 属性の違いを理解しておく必要があります。MBean は、管理が必要なデバイス、アプリケーション、または任意のリソースを表すことができるオブジェクトです。

たとえば、Web サーバーを表す MBean があります。その属性は、接続数、スレッド数、リクエストタイムアウト、http ファイルキャッシュ、メモリ使用量などです。この考えを人間の包括的な言語で表現すると、コーヒーマシンを豆と水の補充時間などを監視対象の属性として持つ MBean として定義できます。

アイテムキー

ディスカバリルール設定のタイプフィールドで **JMX** エージェントを選択します。

JMX オブジェクトの検出では、jmx.discovery[] と jmx.get[] の 2 つのアイテムキーがサポートされています。

アイテムキー

	戻り値	パラメータ	コメント
jmx.discovery [<discovery mode>,<object name>,<unique short description>]			

この項目は、MBean オブジェクト またはその属性を記述する LLD マクロを含む JSON 配列を返します。

discovery例:

mode - →
 次のいずれか: - すべて
 at- の JMX
 tributes MBean
 (JMX 属性を
 MBean 取得す
 属性の
 取得、 →
 デフォルト) or
 beans の JMX
 (JMX MBean
 MBean を取得
 を取得) する
object →
name - jmx.discovery[attribu
 取得す - すべて
 る のガベ
 MBean ージコ
 名を識 レクタ
 別する ー属性
 オブジ を取得
 ェクト する
 名パタ →
 ーン jmx.discovery[beans,
 (ドキュ - すべて
 メント のガベ
 を参照) ージコ
 (デフォ レクタ
 ルトで ーを取
 は空で、 得する
 登録さ マクロ
 れてい 名の生
 るすべ 成でサ
 ての ポート
 Bean されて
 を取得 いる制
 します) 限され
unique た文字
short に基づ
de- いて、
scrip- このア
tion - イテム
 ホスト が返す
 上で同 ことが
 じ検出 できる
 モード MBean
 とオブ プロパ
 ジェク ティに
 ト名を はいく
 持つ複 つか
 数の の制
 JMX ア **限**
 イテム **限**があ
 を許可 ります
 する一 (サポー
 意の説 トされ
 明(オ ている
 プショ 文字は、
 ン) 次の正
 規表現
 で記述
 できま

アイテムキー

jmx.get[<discovery mode>,<object name>,<unique short description>]

この項目は、MBean オブジェクト またはその属性を含む JSON 配列を返します。

`jmx.discovery` [] の違いは LLD マクロを定義しないことです。

discovery mode - 次のいずれか: `at-tributes` (JMX MBean 属性を取得、デフォルト) または `beans` [] (JMX MBean を取得) **object name** - 取得する MBean 名を識別するオブジェクト名パターン (ドキュメントを参照) (デフォルトでは空で、登録されているすべての Bean を取得します) **unique short description** - ホスト上で同じ検出モードとオブジェクト名を持つ複数の JMX アイテムを許可する一意の説明 (オプション)

このアイテムを使用する場合、JSON-Path を使用して返された JSON から抽出された値を指す、カスタムの低レベル検出マクロを定義する必要があります。 .

Zabbix Java Gateway 4.4 以降でサポートされています。

Attention:

パラメータが渡されない場合、JMX からのすべての MBean 属性が要求されます。JMX ディスカバリー用のパラメーターを指定しない、または `*:type=*,name=*` のように広範囲のすべての属性を取得しようとすると、パフォーマンスの問題が発生する可能性があります。

jmx.discovery の使用

この項目は、MBean オブジェクトまたは属性を記述するローレベルディスカバリマクロを含む JSON オブジェクトを返します。たとえば、MBean 属性のディスカバリーでは次の通りです。(明確にするために再フォーマットされています)

```
[
  {
    "{#JMXVALUE}": "0",
    "{#JMXTYPE}": "java.lang.Long",
    "{#JMXOBJ}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXDESC}": "java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
    "{#JMXATTR}": "CollectionCount"
  },
  {
    "{#JMXVALUE}": "0",
    "{#JMXTYPE}": "java.lang.Long",
    "{#JMXOBJ}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXDESC}": "java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
    "{#JMXATTR}": "CollectionTime"
  },
  {
    "{#JMXVALUE}": "true",
    "{#JMXTYPE}": "java.lang.Boolean",
    "{#JMXOBJ}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXDESC}": "java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
    "{#JMXATTR}": "Valid"
  },
  {
    "{#JMXVALUE}": "PS Scavenge",
    "{#JMXTYPE}": "java.lang.String",
    "{#JMXOBJ}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXDESC}": "java.lang:type=GarbageCollector,name=PS Scavenge,Name",
    "{#JMXATTR}": "Name"
  },
  {
    "{#JMXVALUE}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXTYPE}": "javax.management.ObjectName",
    "{#JMXOBJ}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXDESC}": "java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
    "{#JMXATTR}": "ObjectName"
  }
]
```

MBean のディスカバリーでは次の通りです。(明確にするために再フォーマットされています):

```
[
  {
    "{#JMXDOMAIN}": "java.lang",
    "{#JMXTYPE}": "GarbageCollector",
    "{#JMXOBJ}": "java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXNAME}": "PS Scavenge"
  }
]
```

サポートされているマクロ

次のマクロはディスカバリルールフィルター およびアイテム、トリガー、グラフのプロトタイプでの使用がサポートされています。

マクロ	説明
MBean 属性の検出	
{#JMXVALUE}	属性値
{#JMXTYPE}	属性タイプ
{#JMXOBJ}	オブジェクト名
{#JMXDESC}	属性名を含むオブジェクト名
{#JMXATTR}	属性名
MBean の検出	
{#JMXDOMAIN}	MBean ドメイン. (Zabbix 予約名)
{#JMXOBJ}	オブジェクト名. (Zabbix 予約名)
{#JMX<key property>}	MBean プロパティ (like {#JMXTYPE}, {#JMXNAME}) (参照 Limitations below).

制限事項

MBean プロパティ名から LLD マクロ名を作成するアルゴリズムには、いくつかの制限があります。

- 属性名を大文字に変更
- 属性名が LLD マクロ名でサポートされていない文字で構成されている場合、属性名は無視されます (LLD マクロは生成されません)。サポートされている文字は、次の正規表現で記述できます: `A-Z0-9_\.`
- 属性が "obj" または "domain" と呼ばれる場合、予約された Zabbix プロパティ {#JMXOBJ} および {#JMXDOMAIN} (Zabbix 3.4.3 以降でサポート) の値と重複するため、それらは無視されます。

次の `jmx.discovery` ("beans" モードを使用) の例を確認してください。MBean には次のプロパティが定義されています。

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

JMX 検出の結果、次の LLD マクロが生成されます。

- {#JMXDOMAIN} - Zabbix 内部、MBean のドメインを記述
- {#JMXOBJ} - Zabbix 内部、MBean オブジェクトについて説明
- {#JMXNAME} - "name" プロパティから作成

次のプロパティは以下の理由により無視されます。

- `тип` : 名前にサポートされていない文字 (非 ASCII) が含まれています
- `attributes[]` : 名前にサポートされていない文字が含まれています (角括弧はサポートされていません)
- `Name` : 定義済み (`name=test`)
- `domain` : Zabbix の予約名

例

Mbean を使用して LLD ルールを作成するさらに 2 つの実用的な例を見てみましょう。Mbean を収集する LLD ルールと Mbean 属性を収集する LLD ルールの違いをよりよく理解するには、次の表をご覧ください。

MBean1	MBean2	MBean3
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

例 1: Mbeans の検出

このルールは 3 つのオブジェクトを返します: 列の一番上の行: MBean1, MBean2, MBean3.

オブジェクトの詳細については、[サポートされているマクロ](#) 表の MBean の検出セクションを参照してください。

Mbean (属性なし) を収集するディスカバリー・ルール構成は、次のようになります。

Discovery rule Preprocessing LLD macros Filters Overrides

* Name

Type

* Key

* Host interface

ここで使用されるキー:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

属性のないすべてのガベージコレクターが検出されます。ガベージコレクターには同じ属性セットがあるため、アイテムプロトタイプで必要な属性を次のように使用できます。

Item prototypes

All hosts / JMX Enabled **JMX** Discovery list / JMX garbage collectors Item prototypes Trigger p

<input type="checkbox"/> Name ▲	Key
<input type="checkbox"/> GC {#JMXNAME} CollectionCount	jmx[{#JMXOBJ},CollectionCount]
<input type="checkbox"/> GC {#JMXNAME} CollectionTime	jmx[{#JMXOBJ},CollectionTime]
<input type="checkbox"/> GC {#JMXNAME} Valid	jmx[{#JMXOBJ},Valid]

ここで使用されるキー:

```
jmx[{#JMXOBJ},CollectionCount]
```

```
jmx[{#JMXOBJ},CollectionTime]
```

```
jmx[{#JMXOBJ},Valid]
```

LLD ディスカバリルールは、これに近い結果になります (アイテムは 2 つのガベージコレクターで検出されます)。

<input type="checkbox"/> Name ▲	Triggers	Key
<input type="checkbox"/> ... JMX garbage collectors: GC PS MarkSweep CollectionCount		jmx["java.lang:type=GarbageCollector,name=PS MarkSweep",CollectionCount]
<input type="checkbox"/> ... JMX garbage collectors: GC PS MarkSweep CollectionTime		jmx["java.lang:type=GarbageCollector,name=PS MarkSweep",CollectionTime]
<input type="checkbox"/> ... JMX garbage collectors: GC PS MarkSweep Valid		jmx["java.lang:type=GarbageCollector,name=PS MarkSweep",Valid]
<input type="checkbox"/> ... JMX garbage collectors: GC PS Scavenge CollectionCount		jmx["java.lang:type=GarbageCollector,name=PS Scavenge",CollectionCount]
<input type="checkbox"/> ... JMX garbage collectors: GC PS Scavenge CollectionTime		jmx["java.lang:type=GarbageCollector,name=PS Scavenge",CollectionTime]
<input type="checkbox"/> ... JMX garbage collectors: GC PS Scavenge Valid		jmx["java.lang:type=GarbageCollector,name=PS Scavenge",Valid]

例 2: Mbean 属性の検出

このルールは、次のフィールドを持つ 9 つのオブジェクトを返します: MBean1Attribute1, MBean2Attribute1, Mbean3Attribute1, MBean1Attribute2, Mbean3Attribute2, MBean1Attribute3, MBean2Attribute3, Mbean3Attribute3.

オブジェクトの詳細については、サポートされているマクロ表の MBean 属性の検出セクションを参照してください。

Mbean 属性を収集する検出ルールの構成は、次のようになります。

Discovery rule Preprocessing LLD macros Filters Overrides

* Name

Type

* Key

* Host interface

ここで使用されるキー:

```
jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]
```

単一の項目属性を持つすべてのガベージコレクターが検出されます。

Item prototypes

All hosts / JMX Enabled **JMX** Discovery list / JMX garbage collectors Item prototypes Trigger p

<input type="checkbox"/> Name ▲	Key
<input type="checkbox"/> {#JMXOBJ} {#JMXATTR}	jmx[{#JMXOBJ},{#JMXATTR}]

この特定のケースでは、すべての MBean 属性のプロトタイプからアイテムが作成されます。この構成の主な欠点は、すべての属性に対してアイテムプロトタイプが 1 つしかないため、トリガープロトタイプからトリガーを作成できないことです。したがって、この設定はデータ収集には使用できますが、自動監視にはお勧めできません。

jmx.get の使用

jmx.get[] は jmx.discovery[] アイテムに似ていますが、Java オブジェクトのプロパティをローレベルディスカバリマクロ名に変換しないため、制限なしで値を返すことができます。ハイフンや非 ASCII 文字などの LLD マクロ名の生成に関連付けられています。

検出に「jmx.get[]」を使用する場合、ローレベルディスカバリのカスタム LLD マクロ タブで、必要な値の JSONPath を使用してローレベルディスカバリマクロを個別に定義できます。

MBeans の検出

検出アイテム: jmx.get[beans,"com.example:type=*,*"]

レスポンス:

```
[
  {
    "object": "com.example:type=Hello,data-src=data-base, = ",
    "domain": "com.example",
    "properties": {
      "data-src": "data-base",
      " ": " ",
      "type": "Hello"
    }
  },
  {
    "object": "com.example:type=Atomic",
    "domain": "com.example",
    "properties": {
      "type": "Atomic"
    }
  }
]
```

```
}  
]
```

MBean 属性の検出

検出アイテム: `jmx.get[attributes,"com.example:type=*,*"]`

レスポンス:

```
[  
  {  
    "object": "com.example:type=*",  
    "domain": "com.example",  
    "properties": {  
      "type": "Simple"  
    }  
  },  
  {  
    "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand, = ,obj=true",  
    "domain": "com.zabbix",  
    "properties": {  
      "type": "Hello",  
      "domain": "com.example",  
      "data-source": "/dev/rand",  
      " ": " ",  
      "obj": true  
    }  
  }  
]
```

6 IPMI センサーの検出

概要

IPMI センサーを自動的に検出することが可能です。

これを行うには次の組み合わせを使用します。

- マスターアイテムとしての `ipmi.getIPMI` アイテム (Zabbix **5.0.0** 以降でサポート)
- 依存するローレベルディスカバリルールとアイテムのプロトタイプ

設定

マスターアイテム

次のキーを使用して IPMI アイテムを作成します。

`ipmi.get`

Item Tags Preprocessing

* Name IPMI get item

Type IPMI agent

* Key ipmi.get

* Host interface 127.0.0.1 : 623

IPMI sensor

Type of information Text

JSON データが大きくなる可能性がある場合は、データ型を"テキスト"に設定します。

依存 LLD ルール

"依存アイテム"タイプとしてローレベルディスクバリールールを作成します。

Discovery rule Preprocessing LLD macros Filters Overrides

* Name Discovery rule for ipmi.get

Type Dependent item

* Key ipmi.sensor.discovery

* Master item Zabbix server: IPMI get item

マスターアイテムとして作成した ipmi.get アイテムを選択します。

"LLD マクロ" タブで対応する JSONPath を使用してカスタムマクロを定義します。

Discovery rule Preprocessing LLD macros 1 Filters Overrides

LLD macros	LLD macro	JSONPath
	{#SENSOR_ID}	\$.id

Add

依存アイテムのプロトタイプ

この LLD ルールで"依存アイテム"タイプのアイテムのプロトタイプを作成します。このプロトタイプのマスターアイテムとして、作成した ipmi.get アイテムを選択します。

Item prototype Tags Preprocessing

* Name

Type

* Key

* Master item

Type of information

アイテムのプロトタイプ名とキーで {#SENSOR_ID} マクロが使用されていることに注意してください。

- 名前: IPMI value for sensor {#SENSOR_ID}
- キー: ipmi_sensor[{#SENSOR_ID}]

データ型として数値 (整数)

アイテムのプロトタイプの "保存前処理" タブで JSONPath を選択し、次の JSONPath 式をパラメーターとして使用します。

`$.[?(@.id=='{#SENSOR_ID}')] .value.first()`

Item prototype Tags Preprocessing 1

Preprocessing steps	Name	Parameters
1:	<input type="text" value="JSONPath"/>	<input type="text" value="\$.[?(@.id=='{#SENSOR_ID}')] .value.first()"/>

[Add](#)

ディスカバリが開始されると、IPMI センサーごとに 1 つのアイテムが作成されます。このアイテムは、指定されたセンサーの整数値を返します。

7 systemd サービスの検出

概要

Zabbix を使用して systemd ユニット (デフォルトではサービス) を検出することができます。

アイテムキー

ディスカバリルールで使用するアイテムは

`systemd.unit.discovery`

Attention:

このアイテムキーは、Zabbix エージェント 2 でのみサポートされています。

このアイテムは systemd ユニットに関する情報を含む JSON を返します。次に例を示します。

```
[{
  "{#UNIT.NAME}": "mysqld.service",
  "{#UNIT.DESCRPTION}": "MySQL Server",
  "{#UNIT.LOADSTATE}": "loaded",
  "{#UNIT.ACTIVESTATE}": "active",
  "{#UNIT.SUBSTATE}": "running",
  "{#UNIT.FOLLOWED}": "",
```

```

    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
  }, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
  }
]}

```

無効になっている systemd ユニットの検出

Zabbix 6.0.1 以降、無効になっている systemd ユニットの検出することもできます。この場合、結果の JSON で 3 つのマクロが返されます。

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:

無効になっている systemd ユニットのプロトタイプからアイテムとトリガーを作成するには、{#UNIT.ACTIVESTATE} と {#UNIT.UNITFILESTATE} の LLD フィルターの禁止を調整（または削除）してください。

サポートされているマクロ

次のマクロはディスカバリルール **フィルター** およびアイテム、トリガー、グラフのプロトタイプでの使用がサポートされています。

マクロ	説明
{#UNIT.NAME}	プライマリユニット名
{#UNIT.DESCRPTION}	人間が読みやすい形式の説明。
{#UNIT.LOADSTATE}	ロードステータス（ユニットファイルが正常にロードされたかどうか）
{#UNIT.ACTIVESTATE}	アクティブステータス（ユニットが現在起動しているかどうか）
{#UNIT.SUBSTATE}	サブステータス（アクティブ状態ではない。ユニットタイプに固有のアクティブ状態のより詳細な情報）
{#UNIT.FOLLOWED}	このユニットが存在する場合、このユニットがフォローしているユニット。それ以外の場合は空の文字列
{#UNIT.PATH}	ユニットのオブジェクトパス
{#UNIT.JOBID}	ジョブ単位のキューに入れられたジョブがある場合はジョブ ID の数値。それ以外の場合は 0
{#UNIT.JOBTYPE}	ジョブタイプ
{#UNIT.JOBPATH}	ジョブのオブジェクトパス
{#UNIT.UNITFILESTATE}	ユニットファイルのインストール状態

アイテムのプロトタイプ

systemd サービスディスカバリに基づいて作成できるアイテムのプロトタイプには、たとえば次のものがあります。

- アイテム名: {#UNIT.DESCRPTION}; アイテムキー: `systemd.unit.info["{#UNIT.NAME}"]`
- アイテム名: {#UNIT.DESCRPTION}; アイテムキー: `systemd.unit.info["{#UNIT.NAME}",LoadState]`

`systemd.unit.info` エージェントアイテムは、Zabbix 4.4 以降でサポートされています。

8 Windows サービスの検出

概要

ファイルシステムが検出されるのと同様の方法で、Windows サービスも検出できます。

アイテムキー

ディスカバリルールで使用するアイテムは

`service.discovery`

このアイテムは、ZabbixWindows エージェント 3.0 以降でサポートされています。

サポートされているマクロ

次のマクロはディスカバリルールフィルターおよびアイテム、トリガー、グラフのプロトタイプで使用がサポートされています。

マクロ	説明
{#SERVICE.NAME}	サービス名
{#SERVICE.DISPLAYNAME}	表示サービス名
{#SERVICE.DESCRPTION}	サービスの説明
{#SERVICE.STATE}	サービス状態の数値： 0 - 実行中 1 - 一時停止 2 - 保留中の開始 3 - 保留中の一時停止 4 - 保留中の続行 5 - 保留中の停止 6 - 停止 7 - 不明
{#SERVICE.STATENAME}	サービス状態の名前（実行中、一時停止、保留中の開始、保留中の一時停止、保留中の続行、保留中の停止、停止、または不明）
{#SERVICE.STATENAME}	サービス状態名（実行中、一時停止、保留中の開始、保留中の一時停止、保留中の続行、保留中の停止、停止、または不明）
{#SERVICE.PATH}	サービスのパス
{#SERVICE.USER}	サービスユーザー
{#SERVICE.STARTUP}	サービススタートアップタイプの数値： 0 - 自動 1 - 自動遅延 2 - 手動 3 - 無効 4 - 不明
{#SERVICE.STARTUPNAME}	サービススタートアップタイプの名前（自動、自動遅延、手動、無効、不明）
{#SERVICE.STARTUPTRIGGER}	サービスの起動タイプに次の値があるかどうかを示す数値： 0 - 起動トリガーなし 1 - 起動トリガーあり このマクロは Zabbix3.4.4 以降でサポートされています。自動（トリガー開始）、自動遅延（トリガー開始）、手動（トリガー開始）などのサービス起動タイプを見つけると便利です。

Windows サービスディスカバリに基づいて、次のようなアイテムのプロトタイプを作成できます。

```
service.info[{#SERVICE.NAME},<param>]
```

ここで `param` は次の値を受け入れます：`state`、`displayname`、`path`、`user`、`startup`、または `description`

たとえば、サービスの表示名を取得するには `"service.info[{#SERVICE.NAME},displayname]"` アイテムを使用します。`param` 値が指定されていない場合 (`"service.info[{#SERVICE.NAME}]"`) デフォルトの `state` パラメーターが使用されます。

9 Windows パフォーマンスカウンターインスタンスの検出

概要

Windows パフォーマンスカウンターのオブジェクトインスタンスを検出することができます。これはマルチインスタンスのパフォーマンスカウンターに役立ちます。

アイテムキー

ディスカバリルールで使用するアイテムは

```
perf_instance.discovery[object]
```

また、OS のローカリゼーションとは関係なく、英語でのみオブジェクト名を提供できるようにするには、次のようにします。

```
perf_instance_en.discovery[object]
```

例:

```
perf_instance.discovery[Processador]  
perf_instance_en.discovery[Processor]
```

これらのアイテムは ZabbixWindows エージェント 5.0.1 以降でサポートされています。

サポートされているマクロ

ディスクカバリは指定されたオブジェクトのすべてのインスタンスを `{#INSTANCE}` マクロで返します。これは `perf_count` および `perf_count_en` アイテムのプロトタイプで使用できます。

```
[  
  {"{#INSTANCE}": "0"},  
  {"{#INSTANCE}": "1"},  
  {"{#INSTANCE}": "_Total"}  
]
```

たとえばディスクカバリルールで使用されるアイテムキーが次の場合、

```
perf_instance.discovery[Processor]
```

アイテムのプロトタイプを作成できます。

```
perf_counter["\\Processor({#INSTANCE})\\% Processor Time"]
```

注:

- 指定されたオブジェクトが見つからないか、変数インスタンスをサポートしていない場合、検出アイテムは `NOTSUPPORTED` になります。
- 指定されたオブジェクトが変数インスタンスをサポートしているが、現在インスタンスがない場合は、空の JSON 配列が返されます。
- 重複するインスタンスの場合、スキップされます。

10 WMI クエリを使用した検出

概要

WMIは Windows の強力なインターフェイスであり、インストールされている Windows コンポーネント、サービス、状態、およびソフトウェアに関するさまざまな情報を取得するために使用できます。

物理ディスクの検出とそのパフォーマンスデータの収集、ネットワークインターフェイスの検出、Hyper-V ゲストの検出、Windows サービスの監視など WindowsOS の他の多くのことに使用できます。

このタイプのローレベルディスクカバリは、結果がローレベルディスクカバリに適した JSON オブジェクトに自動的に変換される WQL クエリを使用して実行されます。

アイテムキー

ディスクカバリルールで使用するアイテムは

```
wmi.getall[<namespace>,<query>]
```

このアイテムはクエリ結果を JSON 配列に変換します。例えば:

```
select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'
```

次のようなものが返されます。

```
[  
  {  
    "DeviceID" : "\\.\PHYSICALDRIVE0",  
    "BytesPerSector" : 512,  
    "Capabilities" : [  
      3,  
      4  
    ],  
    "CapabilityDescriptions" : [  
      "Random Access",  
      "Supports Writing"  
    ],  
  },  
]
```

```

    "Caption" : "VBOX HARDDISK ATA Device",
    "ConfigManagerErrorCode" : "0",
    "ConfigManagerUserConfig" : "false",
    "CreationClassName" : "Win32_DiskDrive",
    "Description" : "Disk drive",
    "FirmwareRevision" : "1.0",
    "Index" : 0,
    "InterfaceType" : "IDE"
  },
  {
    "DeviceID" : "\\.\PHYSICALDRIVE1",
    "BytesPerSector" : 512,
    "Capabilities" : [
      3,
      4
    ],
    "CapabilityDescriptions" : [
      "Random Access",
      "Supports Writing"
    ],
    "Caption" : "VBOX HARDDISK ATA Device",
    "ConfigManagerErrorCode" : "0",
    "ConfigManagerUserConfig" : "false",
    "CreationClassName" : "Win32_DiskDrive",
    "Description" : "Disk drive",
    "FirmwareRevision" : "1.0",
    "Index" : 1,
    "InterfaceType" : "IDE"
  }
]

```

このアイテムは ZabbixWindows エージェント 4.4 以降でサポートされています。

ローレベルディスクカバリマクロ

返された JSON にローレベルディスクカバリマクロは作成されませんが、これらのマクロは、返された JSON で検出された値を指す JSONPath で **カスタム LLD マクロ機能** を使用して、追加の手順としてユーザーが定義できます。

その後、マクロを使用して、アイテム、トリガーなどのプロトタイプを作成できます。

11 ODBCSQL クエリを使用した検出

概要

このタイプのローレベル**ディスクカバリ**は SQL クエリで実行され、その結果はローレベルディスクカバリに適した JSON オブジェクトに自動的に変換されます。

アイテムキー

SQL クエリは“データベースモニター”アイテムタイプを使用して実行されます。したがって**ODBC モニタリング**ページのほとんどの手順は、機能する“データベースモニター”ディスクカバリルールを取得するために適用されます。

“データベースモニター”ディスクカバリルールでは、次の 2 つのアイテムキーを使用できます。

- **db.odbc.discovery**[<unique short description>,<dsn>,<connection string>] - このアイテムは、SQL クエリ結果を JSON 配列に変換し、クエリ結果の列名を検出されたフィールド値とペアになったローレベルディスクカバリマクロ名に変換します。これらのマクロは、アイテム、トリガーなどのプロトタイプの作成に使用できます。参照：[db.odbc.discovery の使用](#)。
- **db.odbc.get**[<unique short description>,<dsn>,<connection string>] - このアイテムは、SQL クエリ結果を JSON 配列に変換し、クエリ結果の元の列名を、検出された値とペアになった JSON のフィールド名として保持します。db.odbc.discovery [] と比較すると、このアイテムは返された JSON にローレベルディスクカバリマクロを作成しないため、列名が有効なマクロ名であるかどうかを確認する必要はありません。ローレベルディスクカバリマクロは、必要に応じて追加のステップとして定義できます。**カスタム LLD マクロ機能**を使用して、返される JSON で検出された値を指す JSONPath を使用します。参照：[using db.odbc.get](#)。

db.odbc.discovery を使用する

SQL クエリが JSON に変換される方法を説明する実際的な例として、Zabbix データベースで ODBC クエリを実行することによる Zabbix プロキシのローレベルディスカバリについて考えてみましょう。これは、"zabbix[proxy,<name>,lastaccess]"内部アイテムを自動作成して、どのプロキシが有効であるかを監視するのに役立ちます。

ディスカバリールールの設定から始めましょう。

Discovery rule Preprocessing LLD macros Filters Overrides

* Name Proxy discovery

Type Database monitor

* Key db.odbc.discovery[proxies,{SDSN}]

User name

Password

* SQL query SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxy_hostid WHERE h1.status IN (5, 6) GROUP BY h1.host;

* Update interval 30s

必須入力フィールドは、赤いアスタリスクでマークされています。

ここでは Zabbix データベースに対する直接クエリを使用して、監視しているホストの数とともに、すべての Zabbix プロキシを選択します。たとえばホストの数を使用して、空のプロキシを除外できます。

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxy_hostid
+-----+-----+
| host    | count |
+-----+-----+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+-----+-----+
3 rows in set (0.01 sec)
```

"db.odbc.discovery[,{SDSN}]" アイテムの内部動作により、このクエリの結果は自動的に次の JSON に変換されます。

```
[
  {
    "#HOST": "Japan 1",
    "#COUNT": "5"
  },
  {
    "#HOST": "Japan 2",
    "#COUNT": "12"
  },
  {
    "#HOST": "Latvia",
    "#COUNT": "3"
  }
]
```

列名がマクロ名になり、選択した行がこれらのマクロの値になることがわかります。

Note:

列名がマクロ名にどのように変換されるかが明確でない場合は、上記の例でいう“COUNT(h2.host) AS count”のような列エイリアスを使用することをお勧めします。

列名を有効なマクロ名に変換できない場合、ディスカバリルールはサポートされなくなり、エラーメッセージに問題のある列番号の詳細が表示されます。追加のヘルプが必要な場合は、取得した列名が Zabbix サーバーログファイルの DebugLevel=4 で提供されます。

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
...
23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 I
23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
```

SQL クエリが JSON オブジェクトに変換される方法を理解したので、アイテムのプロトタイプで {#HOST} マクロを使用できます。

Item prototype	Tags	Preprocessing
* Name	Last access time of proxy {#HOST}	
Type	Zabbix internal	
* Key	zabbix[proxy,{#HOST},lastaccess]	
Type of information	Numeric (unsigned)	
Units	unixtime	
* Update interval	60s	

ディスカバリが実行されると、プロキシごとにアイテムが作成されます。

<input type="checkbox"/>	Name	Triggers	Key ▲
<input type="checkbox"/>	... Proxy discovery: Last access time of proxy Japan1		zabbix[proxy,Japan1,lastacce
<input type="checkbox"/>	... Proxy discovery: Last access time of proxy Japan2		zabbix[proxy,Japan2,lastacce
<input type="checkbox"/>	... Proxy discovery: Last access time of proxy Latvia		zabbix[proxy,Latvia,lastaccess

db.odbc.get を使用する

db.odbc.get[,{\$DSN}] と次の SQL の例を使用します。

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxy_ho
+-----+-----+
| host    | count |
+-----+-----+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+-----+-----+
3 rows in set (0.01 sec)
```

この JSON が返されます：

```
[
  {
    "host": "Japan 1",
    "count": "5"
  },
  {
    "host": "Japan 2",
    "count": "12"
  },
  {
    "host": "Latvia",
    "count": "3"
  }
]
```

ご覧のとおりそこにはローレベルディスカバリマクロはありません。ただしカスタムのローレベルディスカバリマクロは、JSONPath を使用してディスカバリルールの **LLD マクロ** タブで作成できます。次に例を示します。

{#HOST} → \$.host

これで、この {#HOST} マクロをアイテムのプロトタイプで使用できるようになりました。

Item prototype	Tags	Preprocessing
* Name	Last access time of proxy {#HOST}	
Type	Zabbix internal	
* Key	zabbix[proxy,{#HOST},lastaccess]	
Type of information	Numeric (unsigned)	
Units	unixtime	
* Update interval	60s	

12 Prometheus データを使用した検出

概要

Prometheus ラインフォーマットで提供されるデータは、ローレベルディスカバリに使用できます。

Zabbix での Prometheus データクエリの実装方法の詳細については [Prometheus チェック](#) を参照してください。

設定

ローレベルディスカバリルールは、Prometheus データを収集する HTTP マスターアイテムへの **依存アイテム** として作成する必要があります。

プロメテウスから JSON

ディスカバリルールで **保存前処理** タブに移動し、Prometheus から JSON オプションを選択します。ディスカバリには JSON 形式のデータが必要であり、Prometheus から JSON オプションは、次の属性を使用してそれを返します。

- metric name
- metric value
- help (if present)
- type (if present)
- labels (if present)
- raw line

例: wmi_logical_disk_free_bytes をクエリします。

Preprocessing steps

Name

Parameters

1:	Prometheus to JSON	wmi_logical_disk_free_bytes{volume=~".*"} Add
----	--------------------	--

これらのプロメテウスラインから：

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

以下のように返します

```
[
  {
    "name": "wmi_logical_disk_free_bytes",
    "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
    "type": "gauge",
    "labels": {
      "volume": "C:"
    },
    "value": "3.5180249088e+11",
    "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
  },
  {
    "name": "wmi_logical_disk_free_bytes",
    "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
    "type": "gauge",
    "labels": {
      "volume": "D:"
    },
    "value": "2.627731456e+09",
    "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
  },
  {
    "name": "wmi_logical_disk_free_bytes",
    "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
    "type": "gauge",
    "labels": {
      "volume": "HarddiskVolume4"
    },
    "value": "4.59276288e+08",
    "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
  }
]
```

LLD マクロマッピング

次に LLD マクロタブに移動して次のマッピングを行います。

```
{#VOLUME}=${.labels['volume']}
{#METRIC}=${['name']}
{#HELP}=${['help']}
```

アイテムのプロトタイプ

次のようなアイテムのプロトタイプを作成することをお勧めします。

Item prototype Tags Preprocessing

* Name

Type

* Key

* Master item

Type of information

Units

* History storage period

* Trend storage period

Value mapping

Description

Create enabled

Discover

前処理オプション :

Item prototype Tags Preprocessing 1

Preprocessing steps	Name	Parameters
1:	<input type="text" value="Prometheus pattern"/>	<input type="text" value="{#METRIC}{volume=" {#volume}"}"=""/>

[Add](#)

13 ブロックデバイスの検出

ファイルシステムが検出されるのと同様の方法で、ブロックデバイスとそのタイプも検出できます。

アイテムキー

ディスカバリルールで使用するアイテムキーは次のとおりです

`vfs.dev.discovery`

Zabbix エージェント 4.4 以降、このアイテムは Linux プラットフォームでのみサポートされます。

このディスカバリアイテムを使用してディスカバリルールを作成し、次のようなことを行うことができます。

- filter: **{#DEVNAME} matches sd[\D]\$** - 名前の付いたデバイスを検出する"sd0", "sd1", "sd2", ...
- filter: **{#DEVTYPE} matches disk AND {#DEVNAME} does not match ^loop.*** - 名前が loop で始まらないディスクタイプのデバイスを検出する

サポートされているマクロ

このディスクバリキーは、ブロックデバイスの名前とタイプをそれぞれ識別する {#DEVNAME} と {#DEVTYPE} の 2 つのマクロを返します。次に例を示します。

```
[
  {
    "#DEVNAME": "loop1",
    "#DEVTYPE": "disk"
  },
  {
    "#DEVNAME": "dm-0",
    "#DEVTYPE": "disk"
  },
  {
    "#DEVNAME": "sda",
    "#DEVTYPE": "disk"
  },
  {
    "#DEVNAME": "sda1",
    "#DEVTYPE": "partition"
  }
]
```

ブロックデバイスの検出では、`vfs.dev.read[]` および `vfs.dev.write[]` アイテムと {#DEVNAME} マクロを使用してアイテムのプロトタイプを作成できます。次に例を示します。

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} は、デバイスのフィルタリングを目的としています。

14 Zabbix でのホストインターフェース検出

概要

ホストの Zabbix フロントエンドで設定されているすべてのインターフェースを**検出**することができます。

アイテムキー

ディスクバリルールで使用するアイテムは以下です。

`zabbix[host,discovery,interfaces]`

内部アイテムです。このアイテムは Zabbix サーバー 3.4 以降でサポートされています。

このアイテムは、次のようなインターフェースの説明を含む JSON を返します。

- IP アドレス/DNS ホスト名 ("接続先" ホスト設定によって異なります)
- ポート番号
- インターフェースタイプ (Zabbix エージェント, SNMP, JMX, IPMI)
- デフォルトのインターフェースかどうか
- バルクリクエスト機能が有効か - SNMP インターフェイスの場合のみ

例:

```
[{"#IF.CONN": "192.168.3.1", "#IF.IP": "192.168.3.1", "#IF.DNS": "", "#IF.PORT": "10050", "#IF.TYPE": "AG"}
```

複数のインターフェースを使用する場合、JSON のレコードは次の順序で並べられます。

- インターフェースタイプ
- デフォルト - デフォルトのインターフェイスは、それ以外のインターフェイスの前に配置されます。
- インターフェース ID (昇順)

サポートされているマクロ

次のマクロはディスクバリルール**フィルター**およびアイテム、トリガー、グラフのプロトタイプでの使用がサポートされています。

マクロ	説明
{#IF.CONNN}	インターフェースの IP アドレスまたは DNS ホスト名
{#IF.IP}	インターフェースの IP アドレス
{#IF.DNS}	インターフェースの DNS ホスト名
{#IF.PORT}	インターフェースのポート番号
{#IF.TYPE}	インターフェースタイプ ("AGENT", "SNMP", "JMX", or "IPMI").
{#IF.DEFAULT}	インターフェースのデフォルトステータス : 0 - デフォルトではない 1 - デフォルトインターフェイス
{#IF.SNMP.BULK}	インターフェースの SNMP 一括処理ステータス : 0 - 無効 1 - 有効 このマクロはインターフェースタイプが"SNMP" の場合にのみ返されます。

7 Custom LLD rules

Overview

It is also possible to create a completely custom LLD rule, discovering any type of entities - for example, databases on a database server.

To do so, a custom item should be created that returns JSON, specifying found objects and optionally - some properties of them. The amount of macros per entity is not limited - while the built-in discovery rules return either one or two macros (for example, two for filesystem discovery), it is possible to return more.

Example

The required JSON format is best illustrated with an example. Suppose we are running an old Zabbix 1.8 agent (one that does not support "vfs.fs.discovery"), but we still need to discover file systems. Here is a simple Perl script for Linux that discovers mounted file systems and outputs JSON, which includes both file system name and type. One way to use it would be as a UserParameter with key "vfs.fs.discovery_perl":

```
#####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\t\"#{FSNAME}\" : \"$fsname\" ,\n";
    print "\t\t\t\"#{FSTYPE}\" : \"$fstype\" \n";
    print "\t}\n";
}

print "]\n";
```

Attention:

Allowed symbols for LLD macro names are **0-9** , **A-Z** , **_** , **.** . Lowercase letters are not supported in the names.

An example of its output (reformatted for clarity) is shown below. JSON for custom discovery checks has to follow the same format.

```
[
  { "#{FSNAME}": "/",           "#{FSTYPE}": "rootfs"  },
  { "#{FSNAME}": "/sys",       "#{FSTYPE}": "sysfs"   },
  { "#{FSNAME}": "/proc",      "#{FSTYPE}": "proc"    },
  { "#{FSNAME}": "/dev",       "#{FSTYPE}": "devtmpfs"},
  { "#{FSNAME}": "/dev/pts",   "#{FSTYPE}": "devpts"  },
```

```

{ "#{FSNAME}":"/lib/init/rw",           "#{FSTYPE}":"tmpfs"   },
{ "#{FSNAME}":"/dev/shm",             "#{FSTYPE}":"tmpfs"   },
{ "#{FSNAME}":"/home",                "#{FSTYPE}":"ext3"    },
{ "#{FSNAME}":"/tmp",                 "#{FSTYPE}":"ext3"    },
{ "#{FSNAME}":"/usr",                 "#{FSTYPE}":"ext3"    },
{ "#{FSNAME}":"/var",                 "#{FSTYPE}":"ext3"    },
{ "#{FSNAME}":"/sys/fs/fuse/connections", "#{FSTYPE}":"fusectl" }
]

```

In the previous example it is required that the keys match the LLD macro names used in prototypes, the alternative is to extract LLD macro values using JSONPath `{#FSNAME} → $.fsname` and `{#FSTYPE} → $.fstype`, thus making such script possible:

```

####!/usr/bin/perl

$first = 1;

print "\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"fsname\": \"$fsname\", \n";
    print "\t\t\"fstype\": \"$fstype\" \n";
    print "\t}\n";
}

print "]\n";

```

An example of its output (reformatted for clarity) is shown below. JSON for custom discovery checks has to follow the same format.

```

[
  { "fsname": "/",           "fstype": "rootfs"   },
  { "fsname": "/sys",       "fstype": "sysfs"    },
  { "fsname": "/proc",     "fstype": "proc"     },
  { "fsname": "/dev",      "fstype": "devtmpfs" },
  { "fsname": "/dev/pts",  "fstype": "devpts"   },
  { "fsname": "/lib/init/rw", "fstype": "tmpfs"    },
  { "fsname": "/dev/shm",  "fstype": "tmpfs"    },
  { "fsname": "/home",     "fstype": "ext3"     },
  { "fsname": "/tmp",      "fstype": "ext3"     },
  { "fsname": "/usr",      "fstype": "ext3"     },
  { "fsname": "/var",      "fstype": "ext3"     },
  { "fsname": "/sys/fs/fuse/connections", "fstype": "fusectl" }
]

```

Then, in the discovery rule's "Filter" field, we could specify `"{#FSTYPE}"` as a macro and `"rootfs|ext3"` as a regular expression.

Note:

You don't have to use macro names FSNAME/FSTYPE with custom LLD rules, you are free to use whatever names you like. In case JSONPath is used then LLD row will be an array element that can be an object, but it can be also another array or a value.

Note that, if using a user parameter, the return value is limited to 512 KB. For more details, see [data limits for LLD return values](#).

16 分散監視

概要 Zabbix は、Zabbixプロキシを使用して分散 IT インフラストラクチャを監視する効果的で信頼性の高い方法を提供します。プロキシを使用して、一元化された Zabbix サーバーに代わってローカルでデータを収集し、サーバーにデータを報告できます。プロキシ機能

プロキシの使用不使用選択時には、いくつか考慮事項があります。

	Proxy
軽量性	Yes
GUI	No
独立動作	Yes
メンテナンス性	Yes
自動 DB 作成 ¹	Yes
ローカル admin	No
組み込みハードウェア	Yes
一方向 TCP 接続	Yes
構成の一元化	Yes
通知の生成	No

Note:

[1] 自動 DB 作成機能は SQLite でのみ機能します。他のデータベースでは**手動でセットアップ**する必要があります。

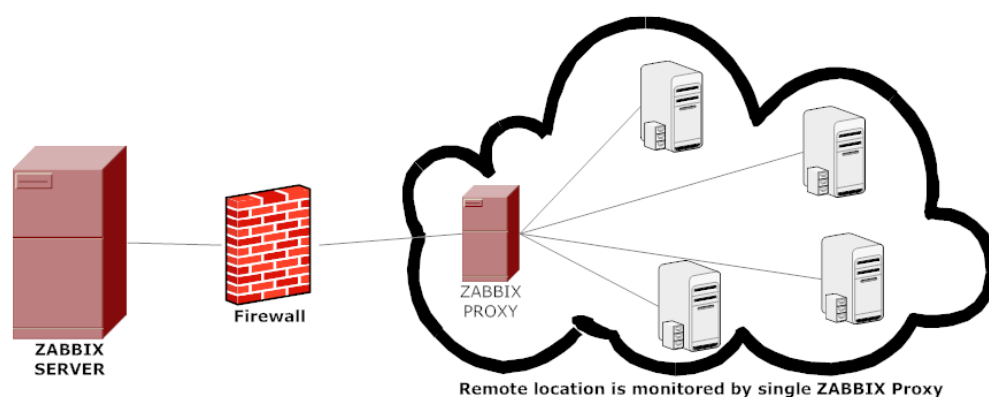
1 プロキシ

概要 Zabbix プロキシは、Zabbix サーバーに代わってパフォーマンスと可用性のデータを収集できます。プロキシはデータ収集の負荷の一部を引き受け、Zabbix サーバーの負荷を軽減できます。

また、すべてのエージェントとプロキシが 1 つの Zabbix サーバーにレポートし、すべてのデータが一元的に収集される場合、プロキシを使用するのが集中型および分散型のモニタリングを実装する最も簡単な方法です。

Zabbix プロキシは次の目的で使用できます。

- 遠隔地を監視する
- 通信の信頼性が低い場所を監視する
- 何千ものデバイスを監視する時に Zabbix サーバーの負荷を下げる
- 分散監視の保守を簡素化する



プロキシは、Zabbix サーバーへの TCP 接続を 1 つだけ必要とします。この方法ではファイアウォールルールを 1 つ設定するだけなので、ファイアウォールを回避するのが簡単です。

Attention:

Zabbix プロキシはサーバーとは別のデータベースを使用する必要があります。Zabbix サーバーデータベースを指定すると、設定が破損します。

プロキシによって収集されたすべてのデータは、サーバーに送信される前にローカルに保存されます。これによりサーバーとの一時的な通信の問題が発生してもデータが失われることはありません。**プロキシ設定ファイル**の ProxyLocalBuffer パラメーターと ProxyOfflineBuffer パラメーターでデータがローカルに保持される期間を制御します。

Attention:

Zabbix サーバーデータベースから直接最新の設定変更を受信するプロキシは、CacheUpdateFrequency の値が原因で更新が遅延した Zabbix サーバーよりも最新の設定になっている場合があります。その結果、プロキシは収集したデータを無視する Zabbix サーバーにデータを送信する場合があります。

Zabbix プロキシはデータコレクターです。トリガーの計算、イベントの処理、アラートの送信は行いません。プロキシ機能の概要については、次の表を確認してください。

機能	プロキシによるサポート
items	
	Zabbix agent checks Yes
	Zabbix agent checks (active) Yes ¹
	Simple checks Yes
	Trapper items Yes
	SNMP checks Yes
	SNMP traps Yes
	IPMI checks Yes
	JMX checks Yes
	Log file monitoring Yes
	Internal checks Yes
	SSH checks Yes
	Telnet checks Yes
	External checks Yes
	Dependent items Yes
	Script items Yes
Built-in web monitoring	Yes
Item value preprocessing	Yes
Network discovery	Yes
Active agent autoregistration	Yes
Low-level discovery	Yes
Remote commands	Yes
Calculating triggers	No
Processing events	No
Event correlation	No
Sending alerts	No

Note:

[1] エージェントが（サーバーではなく）プロキシにアクティブチェックを要求することを確認するには、プロキシがエージェント構成ファイルの **ServerActive** パラメーターにリストされている必要があります。

過負荷からの保護

Zabbix サーバーがしばらくダウンして、その間にプロキシが大量のデータを収集した後に Zabbix サーバーが起動すると、サーバーが過負荷になる可能性があります。（履歴キャッシュの使用率はしばらくの間 95~100%のままです）この過負荷により、パフォーマンスが低下し、チェック処理が本来よりも遅くなる可能性があります。このシナリオからの保護は、履歴キャッシュの過負荷によって発生する問題を回避するために実装されました。

Zabbix サーバーの履歴キャッシュがいっぱいになると、履歴キャッシュの書き込みアクセスが制限され、サーバーのデータ収集プロセスが停止します。最も一般的な履歴キャッシュの過負荷のケースは、プロキシが収集したデータをアップロードしているサーバーのダウンタイム後です。このプロキシスロットリングを回避するために履歴キャッシュの書き込みアクセス制限が追加されました。（現在、無効にすることはできません）

履歴キャッシュの使用率が 80%に達すると、Zabbix サーバーはプロキシからのデータの受け入れを停止します。代わりに、これらのプロキシはスロットルリストに追加されます。これは、キャッシュ使用量が 60%に低下するまで続きます。60%に低下後、サーバーはスロットリングリストで定義されたプロキシからのデータの受け入れを 1 つずつ再開します。これは、スロットリング期間中にデータをアップロードしようとした最初のプロキシが最初に提供され、それが完了するまで、サーバーは他のプロキシからのデータを受け入れないことを意味します。

このスロットリングモードは、キャッシュ使用量が再び 80%に達するか、20%に低下するか、スロットリングリストが空になるまで続きます。80%のケースでは、サーバーはプロキシデータの受け入れを再び停止します。他の 2 つのケースでは、サーバーは正常に動作を開始し、すべてのプロキシからのデータを受け入れます。

上記情報を次の表に示します:

書き込み履歴	Zabbix サーバーモード	Zabbix サーバーアクション
キャッシュの使用状況		
80% 到達	Wait	プロキシデータの受け入れを停止しますが、スロットリングリスト (後で連絡するプロキシの優先順位付けされたリスト) を維持します。
60% に低下	Throttled	スロットリングリストの処理を開始しますが、まだプロキシデータを受け入れません。
20% に低下	Normal	スロットリングリストを削除し、通常どおりプロキシデータの受け入れを開始します。

zabbix[wcache,history,pused] 内部アイテムを使用して、Zabbix サーバーのこの動作をメトリックと関連付けることができます。

設定 Zabbix プロキシサーバーをインストールして設定したのち、Zabbix フロントエンドでプロキシの設定をします。

プロキシの追加

Zabbix フロントエンドでプロキシを設定するには：

- 管理 → プロキシに移動します
- プロキシの作成をクリックします

パラメータ	説明
プロキシ名	プロキシ名を入力します。プロキシ構成ファイルの Hostname パラメーターと同じ名前である必要があります。
プロキシモード	プロキシモードを選択します。 アクティブ - プロキシが Zabbix サーバーに接続し、設定データを要求します パッシブ - Zabbix サーバーがプロキシに接続します 注意暗号化されていない (機密性の高い) プロキシ設定データは、アクティブなプロキシを使用しているときに Zabbix サーバーのトラッパーポートにアクセスできる関係者が利用できるようになる可能性があります。これが可能なのは、認証が行われない場合、またはプロキシアドレスがプロキシアドレスフィールドで制限されていない場合、誰でもアクティブなプロキシのふりをして構成データを要求する可能性があるためです。

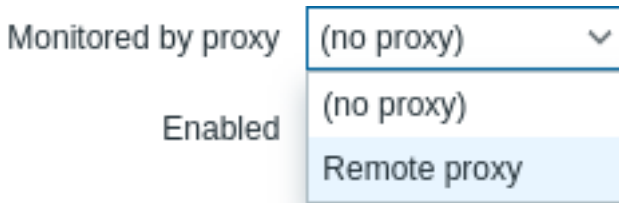
パラメータ	説明
プロキシのアドレス	指定した場合、アクティブなプロキシリクエストは、このコマンド切りの IP アドレスのリスト (オプションで CIDR 表記またはアクティブな Zabbix プロキシの DNS 名) からのみ受け入れられます。 このフィールドはプロキシモードフィールドでアクティブが選択されている場合にのみ使用できます。マクロはサポートされていません。
インターフェース	このオプションは Zabbix4.0.0 以降でサポートされています。パッシブプロキシのインターフェース詳細を入力します。 このフィールドはプロキシモードフィールドでパッシブが選択されている場合にのみ使用できます。
IP アドレス	パッシブプロキシの IP アドレス (オプション)
DNS 名	パッシブプロキシの DNS 名 (オプション)
接続方法	それぞれのボタンをクリックすると、Zabbix サーバーにプロキシからデータを取得するために使用するものが通知されます。 IP - プロキシに IP アドレスで接続します (推奨) DNS - プロキシに DNS 名で接続します
ポート	パッシブプロキシの TCP/UDP ポート番号 (デフォルトでは 10051)
説明	プロキシの説明を入力します

暗号化タブではプロキシとの暗号化された接続を設定できます。

パラメータ	説明
プロキシへの接続	サーバーがパッシブプロキシに接続する方法: 暗号化なし (デフォルト)、PSK (事前共有キー)、証明書
プロキシからの接続	アクティブなプロキシから許可される接続のタイプを選択します。複数の接続タイプを同時に選択できます (テストおよび他の接続タイプへの切り替えに役立ちます)。デフォルトは "暗号なし" です。
発行者	証明書発行者。証明書は最初に CA (認証局) で検証されます。CA によって署名された有効な証明書は、発行者フィールドを使用して許可される CA をさらに制限できます。このフィールドはオプションであり Zabbix インストールで複数の CA からの証明書を使用する場合に使用することを目的としています。
サブジェクト	許可された証明書名。証明書は最初に CA で検証されます。CA によって署名された有効な証明書はサブジェクトフィールドを使用して、サブジェクト文字列の値を 1 つだけ許可できます。このフィールドが空の場合、構成された CA によって署名された有効な証明書が受け入れられます。
PSK アイデンティティ	事前共有キー ID 文字列 機密情報を PSKID に入れないでください。機密情報は暗号化されずにネットワーク経由で送信され、使用する PSK を受信者に通知します。
PSK	事前共有キー (16 進文字列) 最大長: Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 hex-digits (256 バイト PSK)、Zabbix が mbed TLS (PolarSSL) ライブラリを使用する場合は 64 hex-digits (32-byte PSK) 例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

ホストの設定

Monitored by proxy フィールドを使用して、**host configuration** フォームでプロキシによって個々のホストを監視するように指定できます。



ホストの**一括更新**は、ホストをプロキシで監視指定するもう 1 つの方法です。

17. 暗号化

概要 Zabbix は、トランスポートレイヤーセキュリティ (TLS) プロトコル v.1.2、1.3 (crypto ライブラリによる) を使用した Zabbix コンポーネント間の
 暗号化通信をサポートしています。証明書ベースと事前共有キーベースの暗号化がサポートされています。

暗号化は以下の接続に設定することができます:

- Zabbix server, Zabbix proxy, Zabbix agent, zabbix_sender と zabbix_get ユーティリティ間
- Zabbix データベース [from Zabbix frontend and server/proxy](#)

暗号化はオプションで、個々のコンポーネントに設定可能:

- proxy や agent の中には、server との間で証明書ベースの暗号化を使用するように設定できるものもあれば、事前共有鍵ベースの暗号化を使用できるものもあり、また暗号化されていない通信を続けるものもあります。(以前と同様です)
- server (proxy) は、ホストごとに異なる暗号化設定を使用することができます。

Zabbix のデーモンプログラムでは、暗号化された接続と暗号化されていない接続に 1 つのリスニングポートを使用します。
 暗号化を追加する場合、ファイアウォールに新しいポートを開く必要はありません。

制限事項

- 秘密鍵は起動時に Zabbix コンポーネントが読み込み可能なファイルにプレーンテキストで保存されます。
- 事前共有鍵は Zabbix フロントエンドで入力され、Zabbix データベースにプレーンテキストで保存されます。
- 内蔵の暗号化では、通信は保護されません。
 - Zabbix フロントエンドが動作するウェブサーバとユーザのウェブブラウザ間
 - Zabbix フロントエンドと Zabbix サーバ間
- 現在、暗号化された接続は完全な TLS ハンドシェイクで開かれ、セッションキャッシュとチケットは実装されていません。
- 暗号化を追加すると、ネットワーク遅延に応じてアイテムのチェックとアクションにかかる時間が増加します。
 - 例えば、パケット遅延が 100ms の場合、TCP 接続を開き、暗号化されていないリクエストを送信するのに約 200ms かかります。暗号化では、TLS 接続の確立に約 1000ms が追加されます。
 - タイムアウトを増やす必要があるかもしれません。そうしないと、agent 上でリモートスクリプトを実行するいくつかのアイテムやアクションが、暗号化されていない接続では動作するのに、暗号化されているとタイムアウトで失敗する可能性があります。
- 暗号化は、[network discovery](#)ではサポートされていません。Zabbix agent が暗号化されていない接続を拒否するように設定されている場合、ネットワークディスカバリーが実行する Zabbix agent のチェックは成功しません。

暗号化をサポートした **Zabbix** のコンパイル 暗号化をサポートするために、Zabbix は以下のいずれかのライブラリとコンパイル、リンクする必要があります:

- GnuTLS - from version 3.1.18
- OpenSSL - versions 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. Note that 3.0.x is supported since Zabbix 6.0.4.
- LibreSSL - tested with versions 2.7.4, 2.8.2:
 - LibreSSL 2.6.x はサポートされていません。
 - LibreSSL は OpenSSL の互換性のある代替品としてサポートされています。新しい `tls_*`() LibreSSL 固有の API 関数は使用されません。LibreSSL を使用してコンパイルされた Zabbix コンポーネントは PSK を使用できず、証明書のみ使用可能です。

ライブラリの選択は、“configure” スクリプトにそれぞれのオプションを指定することで行います:

- `--with-gnutls [=DIR]`
- `--with-openssl [=DIR]` (also used for LibreSSL)

例えば、server と agent のソースを OpenSSL で設定する場合、次のようになります:

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --wit
```


異なる Zabbix コンポーネントは異なる暗号化ライブラリを使用してコンパイルされている場合があります。
 (例：サーバは OpenSSL、エージェントは GnuTLS など)

Attention:

事前共有鍵 (PSK) の使用を計画している場合、PSK を使用している Zabbix コンポーネントで GnuTLS または OpenSSL 1.1.0
 (またはそれ以降) ライブラリの使用を検討してください。GnuTLS と OpenSSL 1.1.0 ライブラリは、[Perfect Forward Secrecy](#) の
 PSK 暗号をサポートしています。
 古いバージョンの OpenSSL ライブラリ (1.0.1、1.0.2c) も PSK をサポートしていますが、利用可能な PSK 暗号スイートは
 Perfect Forward Secrecy を提供しません。

接続の暗号化管理 Zabbix の接続は以下を使用することができます:

- no encryption (default)
- **RSA certificate-based encryption**
- **PSK-based encryption**

Zabbix コンポーネント間の暗号化を指定するために、2 つの重要なパラメータがあります:

- TLSConnect - 発信接続に使用する暗号化方式を指定します (非暗号化、PSK、または証明書)。
- TLSAccept - 着信接続で許可される接続の種類を指定します (非暗号化、PSK、または証明書)。1 つまたは複数の値を指定できます。

TLSConnect は、Zabbix proxy (アクティブモードで、server への接続のみを指定) と Zabbix agent (アクティブチェック用) の
 設定ファイルで使用されています。Zabbix フロントエンドでは、TLSConnect に相当するのは
 Configuration → Hosts → <some host> → Encryption タブの Connections to host フィールドと
 Administration → Proxies → <some proxy> → Encryption タブの Connections to proxy フィールドになります。
 接続に設定された暗号化方式が失敗した場合、他の暗号化方式は試されません。

TLSAccept は Zabbix proxy (パッシブモード、server からの接続のみ指定) と Zabbix agent (パッシブチェック用) の
 設定ファイルで使用されています。Zabbix フロントエンドでは、TLSAccept に相当するのは
 Configuration → Hosts → <some host> → Encryption* タブの Connections from host フィールドと
 Administration → Proxies → <some proxy> → Encryption タブの Connections from proxy フィールドになります。

通常、受信する暗号化には 1 種類の暗号化しか設定しません。しかし、ダウンタイムやロールバックを最小限に抑えながら、
 暗号化方式を非暗号化から証明書方式に変更したい場合があります。これを実現するために:

- agent 設定ファイルに TLSAccept=unencrypted,cert を設定し、Zabbix agent を再起動する。
- zabbix_get を使用して、証明書を使用するエージェントへの接続をテストします。うまくいった場合、Zabbix フロントエンドでその agent の暗号化を再設定することができます。フロントエンドの Configuration → Hosts → <some host> → Encryption タブで Connections to host を "Certificate" に設定します。
- server 設定キャッシュが更新されたとき (およびホストが監視されている場合は proxy 設定が更新されたとき) proxy で監視しているホストの場合は proxy 設定も更新されると、その agent への接続が暗号化されます。
- 期待通りに動作していれば、agent 設定ファイルに TLSAccept=cert を設定し、Zabbix agent を再起動することができます。これで、agent は暗号化された証明書ベースの接続のみを受け入れるようになります。暗号化されていない、PSK ベースの接続は拒否されます。

同様に、server と proxy でも動作します。Zabbix フロントエンドのホスト設定で Connections from host を "Certificate" に
 設定すると、agent (アクティブチェック) と zabbix_sender(トラッパーアイテム) からは証明書ベースの暗号化接続のみが
 許可されます。

ほとんどの場合、受信接続と送信接続で同じ暗号化タイプを使用するか、まったく暗号化しないように設定します。しかし、
 技術的には非対称に設定することが可能です。

各ホストの暗号化設定は、Zabbix フロントエンドの Configuration → Hosts の Agent encryption 列に表示されます。例:

例	Connections to host	Allowed connections from host	Rejected connections from host
NONE	Unencrypted	Unencrypted	Encrypted, certificate and PSK-based encrypted
CERT NONE PSK CERT	Encrypted, certificate-based	Encrypted, certificate-based	Unencrypted and PSK-based encrypted
PSK NONE PSK CERT	Encrypted, PSK-based	Encrypted, PSK-based	Unencrypted and certificate-based encrypted
PSK NONE PSK CERT	Encrypted, PSK-based	Unencrypted and PSK-based encrypted	Certificate-based encrypted
CERT NONE PSK CERT	Encrypted, certificate-based	Unencrypted, PSK or certificate-based encrypted	-

Attention:

デフォルトでは、接続は暗号化されていません。暗号化は、各ホストとプロキシに個別に設定する必要があります。

zabbix_get と **zabbix_sender** の暗号化機能 **zabbix_get** と **zabbix_sender** manpages のマニュアルを参照してください。

暗号スイート 暗号スイートはデフォルトでは Zabbix 起動時に内部で設定され、Zabbix 4.0.19 以前、4.4.7 以前ではユーザが設定することは
 できません。

Zabbix 4.0.19、4.4.7 から GnuTLS と OpenSSL のユーザ設定可能な暗号スイートがサポートされるようになりました。
 ユーザはセキュリティポリシーに従って暗号スイートを **configure** することができます。
 この機能の使用は任意です。(組み込みのデフォルトの暗号スイートはそのまま機能します)

デフォルト設定でコンパイルされた暗号ライブラリの場合、Zabbix のビルトインルールでは通常以下の暗号スイートを
 使用します。(優先度の高いものから低いものの順):

ライブラリ	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256	TLS_ECDHE_PSK_AES_128_CBC_SHA256
	TLS_ECDHE_RSA_AES_128_CBC_SHA256	TLS_ECDHE_PSK_AES_128_CBC_SHA1
	TLS_ECDHE_RSA_AES_128_CBC_SHA1	TLS_PSK_AES_128_GCM_SHA256
	TLS_RSA_AES_128_GCM_SHA256	TLS_PSK_AES_128_CBC_SHA256
	TLS_RSA_AES_128_CBC_SHA256	TLS_PSK_AES_128_CBC_SHA1
	TLS_RSA_AES_128_CBC_SHA1	
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256	PSK-AES128-CBC-SHA
	ECDHE-RSA-AES128-SHA256	
	ECDHE-RSA-AES128-SHA	
	AES128-GCM-SHA256	
	AES128-SHA256	
OpenSSL 1.1.0	AES128-SHA	
	ECDHE-RSA-AES128-GCM-SHA256	ECDHE-PSK-AES128-CBC-SHA256
	ECDHE-RSA-AES128-SHA256	ECDHE-PSK-AES128-CBC-SHA
	ECDHE-RSA-AES128-SHA	PSK-AES128-GCM-SHA256
	AES128-GCM-SHA256	PSK-AES128-CCM8
	AES128-CCM8	PSK-AES128-CCM
	AES128-CCM	PSK-AES128-CBC-SHA256
	AES128-SHA256	PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	AES128-SHA	
	TLS_AES_256_GCM_SHA384	TLS_CHACHA20_POLY1305_SHA256
	TLS_CHACHA20_POLY1305_SHA256	TLS_AES_128_GCM_SHA256
	TLS_AES_128_GCM_SHA256	ECDHE-PSK-AES128-CBC-SHA256
	ECDHE-RSA-AES128-GCM-SHA256	ECDHE-PSK-AES128-CBC-SHA
	ECDHE-RSA-AES128-SHA256	PSK-AES128-GCM-SHA256
	ECDHE-RSA-AES128-SHA	PSK-AES128-CCM8
	AES128-GCM-SHA256	PSK-AES128-CCM
	AES128-CCM8	PSK-AES128-CBC-SHA256
	AES128-CCM	PSK-AES128-CBC-SHA
	AES128-SHA256	

ユーザー設定暗号スイート 内蔵の暗号スイート選択基準は、ユーザーが設定した暗号スイートで上書きすることができます。

Attention:

ユーザー設定暗号スイートは、TLS 暗号スイートとその安全性を理解し、TLS のトラブルシューティングに精通している
 上級ユーザー向けの機能です。

組み込みの暗号スイート選択基準は、以下のパラメータを使用して上書きすることができます:

上書きスコープ	パラメータ	値	説明
Ciphersuite selection for certificates	TLSCipherCert13	Valid OpenSSL 1.1.1 cipher strings for TLS 1.3 protocol (their values are passed to the OpenSSL function <code>SSL_CTX_set_ciphersuites()</code>).	Certificate-based ciphersuite selection criteria for TLS 1.3 Only OpenSSL 1.1.1 or newer.
	TLSCipherCert	Valid OpenSSL cipher strings for TLS 1.2 or valid GnuTLS priority strings . Their values are passed to the <code>SSL_CTX_set_cipher_list()</code> or <code>gnutls_priority_init()</code> functions, respectively.	Certificate-based ciphersuite selection criteria for TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Ciphersuite selection for PSK	TLSCipherPSK13	Valid OpenSSL 1.1.1 cipher strings for TLS 1.3 protocol (their values are passed to the OpenSSL function <code>SSL_CTX_set_ciphersuites()</code>).	PSK-based ciphersuite selection criteria for TLS 1.3 Only OpenSSL 1.1.1 or newer.
	TLSCipherPSK	Valid OpenSSL cipher strings for TLS 1.2 or valid GnuTLS priority strings . Their values are passed to the <code>SSL_CTX_set_cipher_list()</code> or <code>gnutls_priority_init()</code> functions, respectively.	PSK-based ciphersuite selection criteria for TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Combined ciphersuite list for certificate and PSK	TLSCipherAll13	Valid OpenSSL 1.1.1 cipher strings for TLS 1.3 protocol (their values are passed to the OpenSSL function <code>SSL_CTX_set_ciphersuites()</code>).	Ciphersuite selection criteria for TLS 1.3 Only OpenSSL 1.1.1 or newer.
	TLSCipherAll	Valid OpenSSL cipher strings for TLS 1.2 or valid GnuTLS priority strings . Their values are passed to the <code>SSL_CTX_set_cipher_list()</code> or <code>gnutls_priority_init()</code> functions, respectively.	Ciphersuite selection criteria for TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

`zabbix_get` と `zabbix_sender` ユーティリティで暗号化スイート選択を
 上書きするには、次のコマンドラインパラメータを使用します。:

- `--tls-cipher13`
- `--tls-cipher`

新しいパラメータはオプションである。パラメータが指定されていない場合、内部のデフォルト値が使用されます。
 パラメータが定義されている場合、空にすることはできません。

crypto ライブラリの TLSCipher* 値の設定に失敗すると、server、proxy、または agent は起動せず、エラーが記録されます。

各パラメータがどのような場合に適用されるかを理解することが重要です。

発信コネクション

最も単純なケースは、発信コネクションです:

- 証明書を使用した発信コネクションの場合 - TLSCipherCert13 または TLSCipherCert を使用します。
- PSK を使用する発信コネクションには、TLSCipherPSK13 と TLSCipherPSK を使用します。
- `zabbix_get` と `zabbix_sender` ユーティリティの場合、コマンドラインパラメータ `--tls-cipher13` と `--tls-cipher` を使用できます。(暗号化は一義的に `--tls-connect` パラメータで指定されます)

着信コネクション

着信コネクションの場合、ルールがコンポーネントや構成によって異なるため、少し複雑になります。

Zabbix agent の場合:

Agent connection setup	Cipher configuration
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13

Agent connection setup	Cipher configuration
TLSCipherCert=cert,psk	TLSCipherAll, TLSCipherAll13

Zabbix **server** と **proxy** の場合:

Connection setup	Cipher configuration
Outgoing connections using PSK	TLSCipherPSK, TLSCipherPSK13
Incoming connections using certificates	TLSCipherAll, TLSCipherAll13
Incoming connections using PSK if server has no certificate	TLSCipherPSK, TLSCipherPSK13
Incoming connections using PSK if server has certificate	TLSCipherAll, TLSCipherAll13

上の 2 つの表でいくつかのパターンを見ることができます:

- TLSCipherAll および TLSCipherAll13 は、証明書と PSK ベースの暗号化方式を組み合わせたリストを使用する場合にのみ指定できます。証明書が設定されている server(proxy)(PSK 暗号スイートは、crypto ライブラリが PSK をサポートしている場合、server、proxy で常に設定されます)、証明書と PSK ベースの両方の受信コネクションを受け入れるように設定された agent の場合
- その他の場合は、TLSCipherCert と TLSCipherPSK の両方またはどちらかで十分です。

以下の表は、TLSCipher* 組み込みのデフォルト値を示しています。これらは、独自のカスタム値のための良い出発点になる
 可能性があります。

Parameter	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Parameter	OpenSSL 1.1.1d ¹
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

¹ デフォルト値は、古いバージョンの OpenSSL (1.0.1、1.0.2、1.1.0)、LibreSSL、および OpenSSL が PSK を
 サポートせずにコンパイルされた場合、それぞれ異なっています。

**** ユーザー設定暗号スイートの例 ****

ユーザーが設定した暗号スイートは、以下の例を参照してください:

- [Testing cipher strings and allowing only PFS ciphersuites](#)
- [Switching from AES128 to AES256](#)

暗号文字列をテストし、PFS 暗号スイートのみを許可します。

どの暗号スイートが選択されたかを確認するには、設定ファイルで DebugLevel=4 を設定するか、zabbix_sender の
 -vv オプションを使用する必要があります。

希望する暗号スイートが得られるまで、TLSCipher* パラメータの実験が必要な場合があります。TLSCipher* パラメータを調整するために、Zabbix server、proxy、agent を何度も再起動するのは不便です。より便利なオプションは zabbix_sender または openssl コマンドを使用することです。ここではその両方を紹介します。

1. zabbix_sender を使う

テスト用の設定ファイルを作成してみましょう。zabbix_agentd.conf ファイルの構文で、/home/zabbix/test.conf を作成します:

```
Hostname=nonexisting
ServerActive=nonexisting
```

```
TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

この例では、有効な CA とエージェントの証明書と PSK が必要です。証明書と PSK のファイルパスとファイル名は、
 環境に合わせて調整してください。

証明書を使用せず、PSK のみを使用する場合は、より単純なテストファイルを作成することができます：

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

選択された暗号スイートは、zabbix_sender (OpenSSL 1.1.d でコンパイルされた例) を実行することで確認できます：

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

ここでは、デフォルトで選択されている暗号スイートが表示されています。これらのデフォルト値は、OpenSSL の
 バージョンが 1.0.1 より古いシステムで動作する Zabbix agent との相互運用性を確保するために選ばれました。

新しいシステムでは、PFS(Perfect Forward Secrecy) を使用する暗号化方式のみなど、一部の暗号化方式のみを許可することで
 セキュリティを強化することが可能です。ここでは、TLSCipher* パラメータを使って、PFS を持つ暗号スイートだけを
 許可するようにしてみましょう。

Attention:

PSK を使用した場合、OpenSSL 1.0.1 および 1.0.2 を使用したシステムと相互運用できません。
 証明書ベースの暗号化は動作するはずですが、

設定ファイル test.conf に 2 行を追加します：

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

再度テストします：

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

"certificate ciphersuites" と "PSK ciphersuites" のリストが変更され、以前より短くなり、予想通り TLS 1.3 暗号スイートと
 TLS 1.2 ECDHE- * 暗号スイートのみが含まれています。

2. TLSCipherAll と TLSCipherAll13 は zabbix_sender ではテストできません。TLSCipherAll と TLSCipherAll13 を
 調整するには、agent、proxy、または server で実験する必要があります。

したがって、PFS 暗号スイートのみを許可するには、最大で 3 つのパラメータを追加する必要があります。

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

を zabbix_agentd.conf, zabbix_proxy.conf, zabbix_server_conf に追加する必要があります。

Zabbix 環境が PSK ベースの暗号化のみを使用し、証明書を使用しない場合は、1 つだけにしてください：

```
TLSCipherPSK=kECDHEPSK+AES128
```

この仕組みを理解した上で、Zabbix の外でも openssl コマンドを使用して暗号化方式をテストすることができます。
 3 つの TLSCipher* パラメータ値すべてをテストしてみましょう。:

```
$ openssl ciphers ECDH+aRSA+AES128 | sed 's:// /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's:// /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers ECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's:// /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

より詳細な出力を得たい場合は、openssl ciphers に -V オプションを指定することをお勧めします:

```
$ openssl ciphers -V ECDH+aRSA+AES128:kECDHEPSK+AES128
0x13,0x02 - TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any Enc=AESGCM(256) Mac=AEAD
0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any Au=any Enc=CHACHA20/POLY1305(256) Mac=AEAD
0x13,0x01 - TLS_AES_128_GCM_SHA256 TLSv1.3 Kx=any Au=any Enc=AESGCM(128) Mac=AEAD
0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(128) Mac=AEAD
0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA256
0xC0,0x13 - ECDHE-RSA-AES128-SHA TLSv1 Kx=ECDH Au=RSA Enc=AES(128) Mac=SHA1
0xC0,0x37 - ECDHE-PSK-AES128-CBC-SHA256 TLSv1 Kx=ECDHEPSK Au=PSK Enc=AES(128) Mac=SHA256
0xC0,0x35 - ECDHE-PSK-AES128-CBC-SHA TLSv1 Kx=ECDHEPSK Au=PSK Enc=AES(128) Mac=SHA1
```

同様に、GnuTLS の優先度文字列をテストすることができます:

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-ALL
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-ALL
TLS_ECDHE_RSA_AES_128_GCM_SHA256 0xc0, 0x2f TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256 0xc0, 0x27 TLS1.2
```

```
Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GROUP-FFDHE3072
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-ECDSA-SHA384, SIGN-ECDSA-SHA512
```

AES128 から AES256 への切り替え

Zabbix はデータ用のビルトインデフォルトとして AES128 を使用しています。証明書を使用していて、OpenSSL 1.1.1 の
 AES256 に切り替えたい場合を考えてみましょう。

これは、zabbix_server.conf に各パラメータを追加することで実現可能です:

```
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:

証明書関連の暗号スイートのみが使用されますが、より広い相互運用性のために、安全性の低い暗号を含むデフォルト値を避けるために、TLSCipherPSK* パラメータも定義されています。PSK 暗号スイートは、server / proxy で完全に無効にすることはできません。

zabbix_agentd.conf でも:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

1 証明書の利用

概要

Zabbix では、公開または社内 CA によって署名された PEM 形式の RSA 証明書を使用することができます。証明書を検証は、事前に設定された CA 証明書と照合します。オプションとして、証明書リボケーションリスト (CRL) を使用することができます。各 Zabbix コンポーネントに設定できる証明書は 1 つだけです。

内部 CA の設定と操作方法、証明書要求の生成と署名方法、証明書の失効方法の詳細については、[OpenSSL PKI Tutorial v1.1] (<http://pki-tutorial.readthedocs.org/en/latest/>) などのオンライン HowTo を参照してください。

証明書の拡張を慎重に検討し、テストしてください。 [Limitations on using X.509 v3 certificate extensions](#) を参照してください。

証明書の設定パラメータ

パラメータ	必須	説明
TLSCAFile	*	ピア証明書検証用の最上位 CA(s) 証明書を含むファイルのフルパス名。 複数のメンバーを持つ証明書チェーンの場合は、それらを順序付ける必要があります。最初に下位レベルの CA 証明書、次に上位レベルの CA(s) の証明書が続きます。 複数の CA(s) からの証明書を 1 つのファイルに纏めることができます。
TLSCRLFile		証明書失効リストを含むファイルのフルパス名。 証明書失効リスト (CRL) の注釈を参照してください。
TLSCertFile	*	証明書 (証明書チェーン) を含むファイルのフルパス名。 複数のメンバーを持つ証明書チェーンの場合、最初にサーバー、プロキシ、またはエージェントの証明書、次に下位レベルの CA 証明書、次に上位レベルの CA(s) 証明書の順に並べる必要があります。
TLSKeyFile	*	秘密鍵を含むファイルのフルパス名。このファイルは Zabbix ユーザーだけが読み取れるようにアクセス権を設定する必要があります。
TLSServerCertIssuer		許可されたサーバー証明書の発行者。
TLSServerCertSubject		許可されたサーバー証明書のサブジェクト。

Zabbix サーバでの証明書設定

1. 相手の証明書を検証するために、Zabbix サーバは相手のトップレベルの自己署名付きルート CA 証明書のファイルにアクセスする必要があります。例えば、2 つの独立したルート CA からの証明書が必要な場合、これらの証明書を以下のようにファイル `/home/zabbix/zabbix_ca_file` に格納します:

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA

...

Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

...

X509v3 extensions:

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

X509v3 Basic Constraints: critical

CA:TRUE

```
...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLQGQ
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVm0
-----END CERTIFICATE-----
```

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
  ...
  Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
  ...
  X509v3 extensions:
    X509v3 Key Usage: critical
      Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
      CA:TRUE
```

```
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLQGQ
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Zabbix サーバ証明書チェーンをファイル (例: /home/zabbix/zabbix_server.crt) に格納します:

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
  ...
  Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
  ...
  X509v3 extensions:
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Basic Constraints:
      CA:FALSE
```

```
-----BEGIN CERTIFICATE-----
MIIEDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
```

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 2 (0x2)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
  ...
  Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
  Subject Public Key Info:
```



```
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
...
X509v3 extensions:
X509v3 Key Usage: critical
Certificate Sign, CRL Sign
X509v3 Basic Constraints: critical
CA:TRUE, pathlen:0
...
```

```
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLQGQ
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVivWXdMIFJMOHw==
-----END CERTIFICATE-----
```

ここでは、最初に Zabbix server 証明書、その後中間 CA 証明書を掲載しています。

3. Zabbix server の秘密鍵をファイル (例: /home/zabbix/zabbix_server.key) に格納します:

```
-----BEGIN PRIVATE KEY-----
MIIEWAIBADANBgkqhkiG9w0BAQEFAASCBAKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Zabbix サーバ設定ファイルの TLS パラメータを以下のように編集してください:

```
TLSCAFile=/home/zabbix/zabbix_ca_file
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix proxy における証明書ベースの暗号化設定

1. Zabbix サーバの証明書設定 (/manual/encryption/using_certificates#configuring_certificate_on_zabbix_server) に従って、
トップレベル CA 証明書、proxy 証明書 (チェーン)、秘密鍵のファイルを準備します。
proxy 設定のパラメータ TLSCAFile、TLSCertFile、TLSKeyFile を適宜編集してください。

2. アクティブ proxy は、TLSConnect パラメータを編集してください:

```
TLSConnect=cert
```

パッシブ proxy では、TLSAccept パラメータを編集します:

```
TLSAccept=cert
```

3. これで最小限の証明書ベースのプロキシ設定ができました。TLSServerCertIssuer と TLSServerCertSubject パラメータを
設定して、プロキシのセキュリティを向上させることもできます (参照 [Restricting allowed certificate issuer and subject](#))。

4. 最終的なプロキシ設定ファイルでは、TLS パラメータは次のようになります:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

5. Zabbix フロントエンドでこの proxy の暗号化を設定します:

- Administration → Proxies に移動します。
- proxy を選択し、**Encryption** タブをクリックします

以下の例では、Issuer と Subject のフィールドに入力されています。
[Restricting allowed certificate issuer and subject] (/manual/encryption/using_certificates#restricting_allowed_certificate_issuer_and_subject) を参照してください。
これらのフィールドを使用する理由と方法について説明します。

アクティブ proxy

Proxy Encryption

Connections to proxy No encryption PSK Certificate

Connections from proxy No encryption PSK Certificate

Issuer

Subject

パッシブ proxy

Proxy Encryption

Connections to proxy No encryption PSK Certificate

Connections from proxy No encryption PSK Certificate

Issuer

Subject

Zabbix agent の証明書ベースの暗号化設定

1. トップレベル CA 証明書、agent 証明書 (チェーン)、秘密鍵のファイルを準備します。Configuring certificate on Zabbix server の説明に従って、トップレベルの CA 証明書 (チェーン) と秘密鍵のファイルを準備します。
 それに応じて、agent 設定のパラメータ TLSCAFile、TLSCertFile、TLSKeyFile を編集してください。

2. アクティブチェックのために、TLSConnect パラメータを編集してください:

```
TLSCConnect=cert
```

パッシブチェックのためには、TLSAccept パラメータを編集してください:

```
TLSAaccept=cert
```

3. これで、最小限の証明書ベースのエージェント設定ができました。TLSServerCertIssuer と TLSServerCertSubject
 パラメータを設定することで、エージェントのセキュリティを向上させることができます。
 (Restricting allowed certificate Issuer and Subject を参照してください)。

4. 最終的なエージェント設定ファイルでは、TLS パラメータは以下のようになります:

```
TLSCConnect=cert
TLSAaccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

(この例では、proxy 経由でホストを監視しているため、proxy 証明書の Subject を想定しています)

5. Zabbix フロントエンドでこの agent の暗号化を設定します:

- Configuration → Hosts に移動します。
- ホストを選択し、**Encryption** タブをクリックします。

以下の例では、Issuer と Subject フィールドが入力されています。
これらのフィールドを使用する理由と方法は、[Restricting allowed certificate Issuer and Subject (/manual/encryption/using_certificates#restricting_allowed_certificate_issuer_and_subject)] を参照してください。

The screenshot shows the Zabbix web interface for configuring encryption. The 'Encryption' tab is selected. The configuration is as follows:

Connections to host	<input type="radio"/> No encryption	<input type="radio"/> PSK	<input checked="" type="radio"/> Certificate
Connections from host	<input type="checkbox"/> No encryption	<input type="checkbox"/> PSK	<input checked="" type="checkbox"/> Certificate
Issuer	CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com		
Subject	CN=www01,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com		

Buttons: Update, Clone, Full clone, Delete, Cancel

許可された証明書の発行者とサブジェクトを制限する

2 つの Zabbix コンポーネント (サーバーとエージェントなど) が TLS 接続を確立すると、両者は互いの証明書をチェックします。ピア証明書が信頼できる CA によって署名されている場合 (TLSCAFile に最上位の証明書が事前構成されている場合)、有効で、有効期限が切れておらず、他のチェックに合格すると、通信を続行できます。この最も単純なケースでは、証明書の発行者とサブジェクトはチェックされません。

これにはリスクがあり、有効な証明書を持っている人なら誰でも、他の人になりすますことができます (たとえば、ホスト証明書を使用してサーバーになりすますことが可能)。これは証明書が専用の社内 CA によって署名され、なりすましのリスクが低い小規模な環境では許容される場合があります。

トップレベルの CA が、Zabbix によって受け入れられない他の証明書の発行に使用されている場合、またはなりすましのリスクを軽減したい場合は、発行者とサブジェクトの文字列を指定して、許可される証明書を制限できます。

たとえば Zabbix プロキシ構成ファイルに次のように記述します。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

これらの設定では、アクティブプロキシは証明書の異なる発行者またはサブジェクト文字列を持つ Zabbix サーバーと通信しません。パッシブプロキシはそのようなサーバーからの要求を受け入れません。

発行者または件名の文字列の一致に関するいくつかの注意事項:

1. 発行者と件名の文字列は個別にチェックされます。どちらもオプションです。
2. UTF-8 文字を使用できます。
3. 文字列が指定されていない場合、任意の文字列が受け入れられます。
4. 文字列は "そのまま" 比較されます。一致するにはまったく同じでなければなりません。
5. 一致では、ワイルドカードと正規表現はサポートされていません。
6. [RFC 4514 Lightweight Directory Access Protocol \(LDAP\): String Representation of Distinguished Names](#)の一部要件のみが実装されています。
 1. 任意の場所のエスケープ文字 ' (U+0022)、' + ' (U+002B)、' ; ' (U+002C)、' ; ' (U+003B)、' < ' (U+003C)、' > ' (U+003E)、' \ ' (U+005C)
 2. 文字列の先頭にあるエスケープ文字スペース (' ' (U+0020) または番号記号 (# ' (U+0023)
 3. 文字列の末尾のエスケープ文字スペース (' ' (U+0020)
7. null 文字 (U+0000) が検出された場合、一致は失敗します (RFC4514 で許可されています)
8. [RFC 4517 Lightweight Directory Access Protocol \(LDAP\): Syntaxes and Matching Rules](#) および [RFC 4518 Lightweight Directory Access Protocol \(LDAP\): Internationalized String Preparation](#) は、多くの作業が必要なためサポートされていません。

Issuer および Subject 文字列のフィールドの順序とフォーマットは重要です。Zabbix は RFC4514 の推奨に従い、フィールドの“逆”順を使用します。

逆順例:

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

低レベル (CN) で始まり、中レベル (OU、O) に進み、最上位 (DC) フィールドで終わることに注意してください。

OpenSSL では、使用する追加オプションに応じて、デフォルトで証明書の発行者フィールドとサブジェクトフィールドが“通常の”順序で表示されます。

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy
```

```
$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
```

```
...
    Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
...
    Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

ここで、発行者とサブジェクトの文字列はトップレベル (DC) で始まり、ローレベル (CN) フィールドで終わります。スペースとフィールド区切り文字は、使用するオプションによって異なります。これらの値はいずれも、Zabbix 発行者フィールドとサブジェクトフィールドで一致しません!

Attention:

Zabbix で使用できる適切な発行者と件名の文字列を取得するには、特別なオプションを指定して OpenSSL を呼び出します
-nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname:

```
$ openssl x509 -noout -issuer -subject \
    -nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname \
    -in /home/zabbix/zabbix_proxy.crt
issuer= CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject= CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

文字列フィールドが逆順になり、フィールドがカンマで区切られ、Zabbix 構成ファイルとフロントエンドで使用できるようになりました。

X.509 v3 証明書拡張の使用に関する制限

- 件名の別名 (**subjectAltName**) 拡張
subjectAltName 拡張からの代替件名 (IP アドレス、電子メールアドレス等) は、Zabbix ではサポートされていません。Zabbix で確認できるのは“Subject”フィールドの値のみです (許可された証明書の発行者とサブジェクトの制限を参照してください)。証明書が subjectAltName 拡張子を使用している場合、結果は、Zabbix コンポーネントがコンパイルされている暗号化ツールキットの特定の組み合わせによって異なります (動作する場合と動作しない場合があります、Zabbix はピアからのそのような証明書の受け入れを拒否する場合があります)。
- 拡張キーの使用拡張。
使用する場合、一般に clientAuth (TLS WWW クライアント認証) と serverAuth (TLS WWW サーバー認証) の両方が必要です。たとえば、パッシブチェックでは、Zabbix エージェントは TLS サーバーの役割で動作するため、serverAuth をエージェント証明書に設定する必要があります。アクティブチェックのエージェント証明書には、clientAuth を設定する必要があります。GnuTLS は、鍵の使用法に違反した場合に警告を発行しますが、通信は続行できます。
- 名前制約拡張
すべての暗号化ツールキットがサポートしているわけではありませんが、この拡張機能により、このセクションが重要とマークされている CA 証明書を Zabbix がロードできない場合があります (暗号化ツールキットによって異なります)。

証明書失効リスト (CRL)

証明書が漏洩した場合、CA は CRL に記載することで証明書を失効させることができます。CRL は server、proxy、agent のコンフィギュレーションファイルで TLSCRLFile というパラメータを使って設定することができます。例えば、以下のようになります:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file
```

zabbix_crl_file には複数の CA の CRL が含まれ、以下のように表示されることがあります:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
```

```
treZeUPjb7LSmZ3K2hpbZN7So0ZcAoHQ3Gwd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLQGGRYDY29t
...
CAEebS2CND3ShBedZ8YSi15906JvaDP611R51Ns=
-----END X509 CRL-----
```

CRL ファイルは Zabbix 起動時のみ読み込まれます。CRL の更新には再起動が必要です

Attention:

Zabbix コンポーネントが OpenSSL でコンパイルされ、CRL を使用する場合、証明書チェーンのトップレベル及び
 中間レベルの各 CA は、TLSCRLFile に対応する CRL(空でも可) を持つ必要があります。

2 事前共有鍵の使用

概要

Zabbix の各事前共有キー (PSK) は、次のペアです。

- 非秘匿 PSK ID 文字列、
- 秘匿 PSK 文字列値。

PSK ID 文字列は、空ではない UTF-8 の文字列です。たとえば“PSK ID 001 Zabbix agentd” などです。この特定の PSK が Zabbix コンポーネントによって参照される一意の名前です。暗号化されずにネットワーク経由で送信されますので、機密情報を PSK ID 文字列に含めないでください。

PSK 文字列値は推測しにくい 16 進数の文字列です。たとえば、“e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9” などです。

サイズ制限

Zabbix には PSK ID と値のサイズ制限があります。場合によっては、暗号化ライブラリの制限がより低くなる場合があります。

コンポーネント	PSK ID 最大サイズ	PSK 値最小サイズ	PSK 値最大サイズ
Zabbix	128 UTF-8 文字列	128-bit (16-bytes PSK, 16 進数 32 桁)	2048-bit (256-bytes PSK, 16 進数 512 桁)
GnuTLS	128 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)
OpenSSL 1.0.x, 1.1.0	127 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)
OpenSSL 1.1.1	127 bytes (UTF-8 文字列を含む)	-	512-bit (64-byte PSK, 16 進数 128 桁)
OpenSSL 1.1.1a 以降	127 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)

Attention:

Zabbix フロントエンドでは使用する暗号化ライブラリに関係なく、最大 128 文字の PSK ID 文字列と 2048 ビットの PSK を設定できます。一部の Zabbix コンポーネントが下限をサポートしている場合、これらのコンポーネントに許可された長さで PSK ID 文字列と PSK 値を設定するのはユーザーの責任です。長さの制限を超えると、Zabbix コンポーネント間で通信障害が発生します。

Zabbix サーバーが PSK を使用してエージェントに接続する前に、サーバーはそのエージェント用に構成された PSK ID と PSK 値をデータベース (実際には構成キャッシュ) で検索します。接続を受信すると、エージェントは設定ファイルから PSK ID と PSK 値を使用します。双方が同じ PSK ID 文字列と PSK 値を持っている場合、接続が成功します。

Attention:

各 PSK ID は、1 つの値のみとペアにする必要があります。ID 文字列が同じで値が異なる 2 つの PSK が存在しないことを確認するのは、ユーザーの責任です。そうしないと、予測できないエラーが発生したり、この PSK ID 文字列を持つ PSK を使用する Zabbix コンポーネント間の通信が中断したりする可能性があります。

PSK の生成

たとえば、256 ビット (32 バイト) の PSK は、次のコマンドを使用して生成できます。

- with OpenSSL:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- with GnuTLS:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk
```

```
$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

上記の「psktool」は、PSK ID とそれに関連付けられた PSK を含むデータベースファイルを生成することに注意してください。Zabbix は PSK ファイルに PSK のみを想定しているため、識別文字列とコロン (':') をファイルから削除する必要があります。

サーバーエージェント通信用の PSK の構成 (例)

エージェントホストで PSK 値をファイルに書き込みます (例: /home/zabbix/zabbix_agentd.psk)。ファイルには、最初のテキスト文字列に PSK が含まれている必要があります。次に例を示します。

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

PSK ファイルへのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

エージェント構成ファイル zabbix_agentd.conf で TLS パラメータを編集します。たとえば次のように設定します。

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

エージェントはサーバーに接続し、PSK を使用したサーバー (アクティブチェック) と zabbix_get からの接続のみを受けつけます。PSK ID は "PSK 001" です

エージェント再起動後、zabbix_get を使用して接続をテストできます。例えば:

```
$ zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk \
--tls-psk-identity="PSK 001" --tls-psk-file=/home/zabbix/zabbix_agentd.psk
```

(ダウンタイムを最小限に抑えるには、[接続暗号化の管理](#)の接続タイプを変更する方法を参照してください。)

Zabbix フロントエンドでこのエージェントの PSK 暗号化を設定します。

- 設定 → ホストに移動
- ホストを選択して暗号化タブをクリックします

例:

The screenshot shows the Zabbix web interface for configuring a host's encryption settings. The 'Encryption' tab is selected. Under 'Connections to host', the 'PSK' option is selected. Under 'Connections from host', the 'PSK' option is also selected. The 'PSK identity' field contains 'PSK 001'. The 'PSK' field contains the long hexadecimal string: '1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952'. At the bottom, there are buttons for 'Update', 'Clone', 'Full clone', 'Delete', and 'Cancel'.

すべての必須入力フィールドには赤いアスタリスクが付いています。

構成キャッシュがデータベースと同期されると、新しい接続は PSK を使用します。サーバーとエージェントのログファイルでエラーメッセージを確認してください。

サーバーの PSK の構成 - アクティブなプロキシ通信 (例)

プロキシで PSK 値をファイルに書き込みます (例: /home/zabbix/zabbix_proxy.psk)。ファイルの最初のテキスト文字列に PSK が含まれている必要があります。次に例を示します。

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

PSK ファイルへのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

プロキシ構成ファイル zabbix_proxy.conf で TLS パラメータを編集します。たとえば、次のように設定します。

```
TLSCConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

プロキシは PSK を使用してサーバーに接続します。PSK ID は "PSK 002" になります。

(ダウンタイムを最小限に抑えるには、[接続暗号化の管理](#) で接続タイプを変更する方法を参照してください)。

Zabbix フロントエンドでこのプロキシの PSK を構成します。[管理]→[プロキシ](#) に移動し、プロキシを選択して、[暗号化](#) タブに移動します。"プロキシからの接続" で PSK にチェックします。"PSK アイデンティティ" フィールドに "PSK 002" を貼り付け、"e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9" を "PSK" フィールドに入力して、"更新" をクリックします。

プロキシを再起動して、サーバーへの PSK ベースの暗号化された接続の使用を開始します。サーバーとプロキシのログファイルでエラーメッセージを確認してください。

パッシブプロキシの場合、手順は非常に似ています。唯一の違いは、プロキシ構成ファイルで `TLSCAccept=psk` を設定し、Zabbix フロントエンドで "プロキシへの接続" を PSK に設定することです。

3 トラブルシューティング

一般的な推奨事項

- 問題が発生した場合、どのコンポーネントが TLS クライアントとして機能し、どのコンポーネントが TLS サーバーとして機能するかを理解することから始めます。Zabbix サーバー、プロキシ、およびエージェントは、それらの間の相互作用に応じて、すべて TLS サーバーおよびクライアントとして機能します。たとえば、Zabbix パッシブチェックのためにエージェントに接続するサーバーは、TLS クライアントとして機能します。エージェントは TLS サーバーの役割を果たします。Zabbix エージェントは、プロキシからアクティブなチェックのリストを要求し、TLS クライアントとして機能します。プロキシは TLS サーバーの役割を果たします。'zabbix_get' および 'zabbix_sender' ユーティリティは常に TLS クライアントとして機能します。
- Zabbix は相互認証を使用します。それぞれの側がピアを検証し、接続を拒否する場合があります。たとえば、エージェントに接続している Zabbix サーバーは、エージェントの証明書が無効な場合、すぐに接続を閉じることができます。逆もまた同様です - サーバーがエージェントによって信頼されていない場合、サーバーからの接続を受け入れる Zabbix エージェントは接続を閉じることができます。
- TLS クライアントと TLS サーバーの両方でログファイルを調べます。接続を拒否する側は、拒否された正確な理由をログに記録する場合があります。反対側は、かなり一般的なエラーを報告することがよくあります (たとえば、「ピアによって接続が閉じられました」、「接続が正しく終了されませんでした」など)。
- 暗号化の設定を誤ると、実際の原因を示すものではなく、混乱を招くエラーメッセージが表示されることがあります。以下のサブセクションでは、トラブルシューティングに役立つ可能性のあるメッセージと考えられる原因の (完全ではない) コレクションを提供しようとしています。さまざまな暗号化ツールキット (OpenSSL、GnuTLS) は、同じ問題の状況で異なるエラーメッセージを生成することがよくあります。場合によっては、エラーメッセージが両側の暗号化ツールキットの特定の組み合わせに依存することもあります。

1 接続タイプまたは権限の問題

サーバーは PSK を使用してエージェントに接続するように設定されているが、エージェントは暗号化されていない接続のみを受け入れる場合

サーバーまたはプロキシログ (GnuTLS 3.3.16 を使用)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
-110 The TLS connection was non-properly terminated.
```

サーバーまたはプロキシログ (OpenSSL 1.0.2c を使用)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
Connection closed by peer. Check allowed connection types and access rights
```

片側は証明書で接続するが、反対側は PSK のみを受け入れる、またはその逆

任意のログ (GnuTLS を使用):

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
-21 Could not negotiate a supported cipher suite.
```

任意のログ (OpenSSL 1.0.2c を使用):

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
TLS write fatal alert "handshake failure"
```

TLS サポートでコンパイルされた Zabbix センダーを使用して、TLS なしでコンパイルされた Zabbix サーバー/プロキシにデータを
送信した場合

接続元ログ:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL library (version OpenSSL 1.1.1 11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by peer'
```

Windows:

```
...OpenSSL library (version OpenSSL 1.1.1a 20 Nov 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed successfully'
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O error: [0] Success
```

受付側ログ:

```
...failed to accept an incoming connection: from 127.0.0.1: support for TLS was not compiled in
```

一方は PSK で接続するが、もう一方は LibreSSL を使用しているか、暗号化サポートなしでコンパイルされている場合

LibreSSL は PSK をサポートしていません。

接続元ログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

受付側ログ:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Zabbix フロントエンド:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

一方は PSK で接続しますが、もう一方は PSK サポートを無効にして OpenSSL を使用している場合

接続元ログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERROR_SSL
```

受付側ログ:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 証明書の問題

CRL で使用される OpenSSL と証明書チェーンの一部の CA では、CRL が TLSCRLFile に含まれていない

OpenSSL ピアの場合の TLS サーバーログ:


```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
TLS write fatal alert "unknown CA"
```

GnuTLS ピアの場合の TLS サーバーログ:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padding
```

CRL が期限切れになった、またはサーバー操作中に期限切れになった

OpenSSL サーバーログ:

- 期限切れ前:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
SSL routines:ssl3_get_server_certificate:certificate verify failed:\
TLS write fatal alert "certificate revoked"
```

- 期限切れ後:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
SSL routines:ssl3_get_server_certificate:certificate verify failed:\
TLS write fatal alert "certificate expired"
```

ここでのポイントは、有効な CRL を使用すると、取り消された証明書が「証明書が取り消された」として報告されることです。CRL の有効期限が切れると、エラーメッセージが「証明書の有効期限が切れました」に変わります。これは誤解を招く可能性があります。

GnuTLS サーバーログ:

- 有効期限の前後で同じ:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

自己署名証明書や不明な CA

OpenSSL ログ:

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
TLS write fatal alert "unknown CA"
```

これは、サーバー証明書が CA によって署名されているにもかかわらず、間違って発行者とサブジェクトの文字列が同じである場合に発生します。最上位の CA 証明書では発行者とサブジェクトは同じですが、サーバー証明書では同じにすることはできません。(プロキシ証明書とエージェント証明書も同様です)

3 PSK の問題

PSK に奇数個の 16 進数が含まれている

プロキシまたはエージェントが起動しない、プロキシまたはエージェントログのメッセージ:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

128 バイトを超える PSK ID 文字列が GnuTLS に渡された

TLS クライアント側のログ:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

TLS サーバー側のログ:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

OpenSSL 1.1.1 で使用される PSK 値が長すぎる

```
接続側ログ: ...OpenSSL library (version OpenSSL 1.1.1 11 Sep 2018) initialized ... ..In zbx_tls_connect(): psk_identity:"PSK
1" ...zbx_psk_client_cb() requested PSK identity "PSK 1" ...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to
SSL_ERROR_SSL: file ssl\statem\extensions_clnt.c line 801: error:14212044:SSL routines:tls_construct_ctos_early_data:internal er-
ror: TLS write fatal alert "internal error"
```

受付側ログ:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

この問題は通常、OpenSSL を 1.0.x または 1.1.0 から 1.1.1 にアップグレードするときに、PSK 値が 512 ビット (64 バイト PSK、128 の 16 進数として入力) よりも長い場合に発生します。

参照: [値のサイズ制限](#)

18 Web インターフェース

概要 どこからでも、またどのプラットフォームからでも、Zabbix へ容易にアクセスするために、Web インターフェースが提供されています。

Note:

複数の Web インターフェースのインスタンスを使用する場合は、言語や地域の設定とライブラリ (LDAP、SAML など) がすべての Web インターフェースで同じようにインストールおよび構成されていることを確認してください。

1 メニュー

概要

サイドバーのメニューから、さまざまな Zabbix フロントエンドセクションにアクセスできます。



デフォルトのテーマでは、メニューは紺色です。

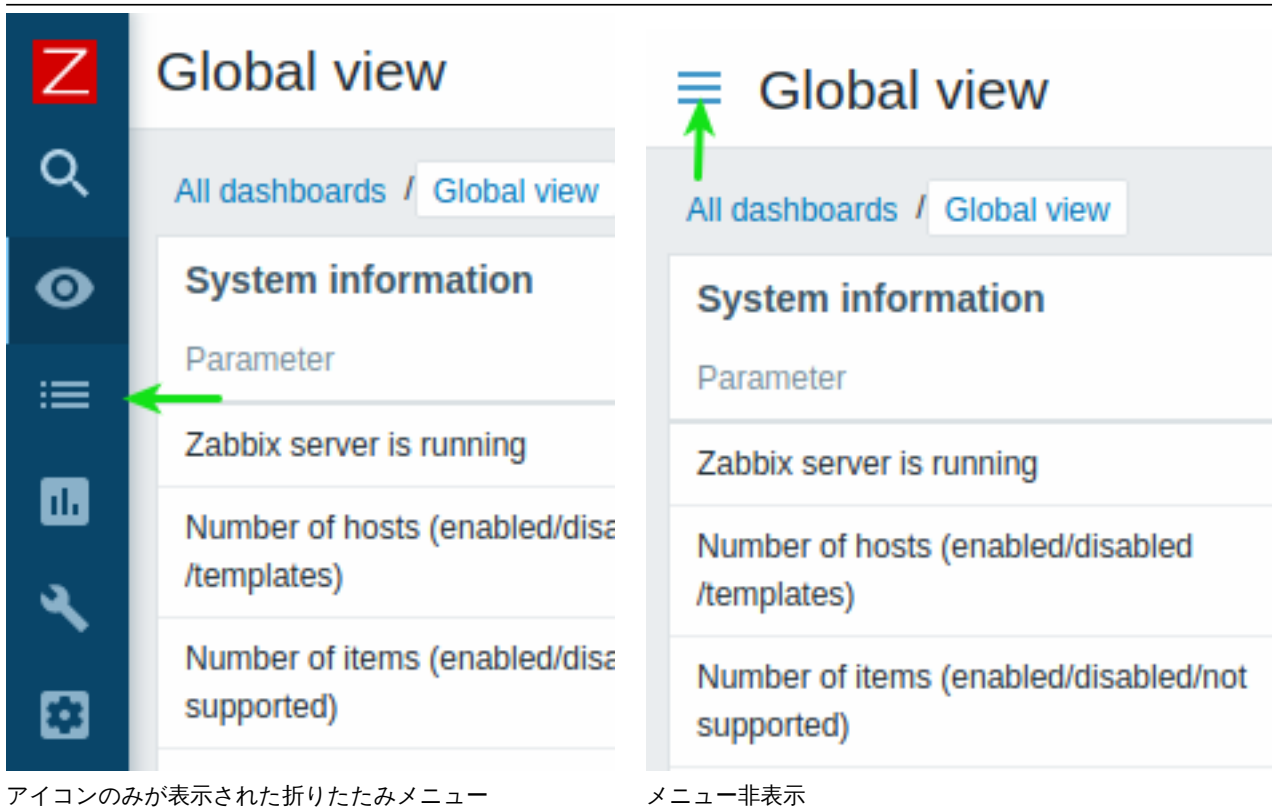
The screenshot shows the Zabbix web interface. On the left is a dark blue sidebar menu with the Zabbix logo at the top. The main area is titled 'Global view' and contains a status bar with colored boxes representing different severity levels: 0 Disaster (red), 2 High (orange), 0 Average (yellow), 0 Warning (light orange), 1 Information (blue), and 5 Not classified (grey). Below this are two tables: 'System information' and 'Problems'. The 'System information' table lists parameters like 'Zabbix server is running' (Yes), 'Number of hosts (enabled/disabled)' (5 / 4 / 1), and 'Number of templates' (140). The 'Problems' table lists recent events with timestamps, host names, and problem types like 'Nodata trigger'.

メニューの操作

Zabbix ログの下にグローバル検索ボックスがあります

メニューは折りたたんだり、完全に非表示にしたりできます。

- 折りたたむには Zabbix ログの横にある  をクリックします
- 非表示にするには、Zabbix ログの横にある  をクリックします



折りたたみメニュー

メニューを折りたたんでアイコンにした後、マウスカーソルをアイコンの上に置くとすぐに完全なメニューを再表示できます。メニューはページコンテンツの上に再表示されることに注意してください。ページコンテンツを右に移動するには、展開ボタンをクリックする必要があります。マウスカーソルを再びフルメニューの外に置くと、2秒後にメニューが再び折りたたまれます。

Tab キーを押して、折りたたまれたメニューを完全に再表示することもできます。Tab キーを繰り返し押すと、次のメニュー要素にフォーカスできます。

メニュー非表示

メニューが完全に非表示になっている場合でも、ハンバーガーアイコンを一回クリックするだけで完全なメニューを表示できます。ページコンテンツの上に再表示されることに注意してください。ページコンテンツを右に移動するには、サイドバーの表示ボタンをクリックしてメニューを再表示する必要があります。

2 フロントエンドセクション

サイドバーからフロントエンドセクションのコンテンツにアクセスしてください。

1 監視データ

概要

監視データメニューは、すべてデータの表示に関するものです。Zabbix が収集、視覚化、操作するために設定されている情報が何であれ、それは監視データメニューのさまざまなセクションに表示されます。

ビューモードボタン

右上隅にある次のボタンは、すべてのセクション共通です。



キオスクモードでページを表示します。このモードでは、ページコンテンツのみが表示されます。

キオスクモードを終了するにはコンテンツ上にマウスカーソルを移動すると表示される  終了ボタンをクリックします。

1 ダッシュボード

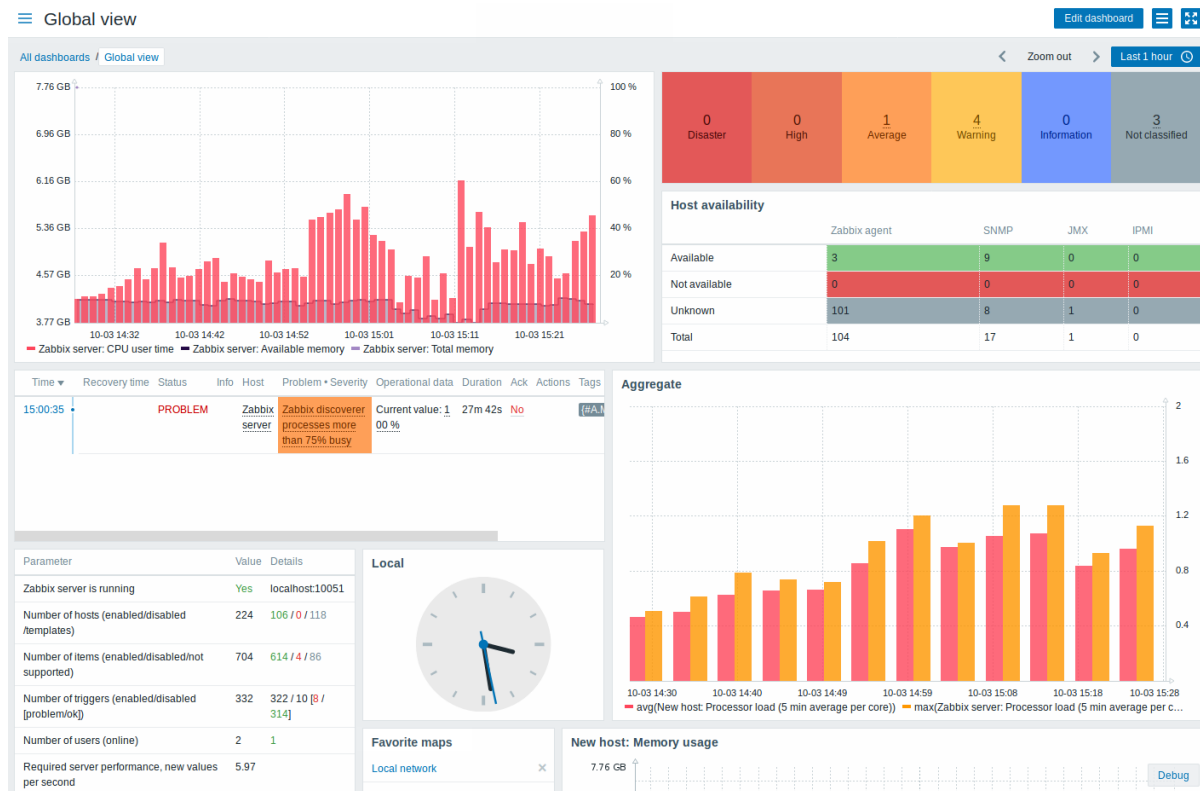
概要

監視データ → ダッシュボードセクションは、すべての重要な情報の概要をダッシュボードに表示するように設計されています。

一度に表示できるダッシュボードは1つだけですが、複数のダッシュボードを構成することもできます。各ダッシュボードには、スライドショーできる1つまたは複数のページが含まれる場合があります。

ダッシュボードページはウィジェットで構成され、各ウィジェットは、概要、地図、グラフ、時計など、特定の種類とソースの情報を表示するように設計されています。

ウィジェット内のホストへのアクセスは、ホストの権限によって異なります。



ページとウィジェットはダッシュボードに追加され、ダッシュボード編集モードで編集されます。ページは、ダッシュボード表示モードで表示およびスライドショー表示できます。

グラフウィジェットに表示される期間は、ウィジェットの上にある期間セレクターによって制御されます。右側にある期間セレクタラベルには、現在選択されている期間が表示されます。タブラベルをクリックすると、期間セレクターを展開および折りたたむことができます。

ダッシュボードがキオスクモードで表示され、ウィジェットのみが表示されている場合は、グラフをダブルクリックしてグラフ期間を縮小できます。

ダッシュボードサイズ

ダッシュボードの最小幅は 1200 ピクセルです。ダッシュボードはこの幅より小さくなりません。ブラウザウィンドウがそれよりも小さい場合は、水平スクロールバーが表示されます。

ダッシュボードの最大幅はブラウザウィンドウの幅です。ダッシュボードウィジェットはウィンドウに合わせて水平方向に伸縮します。ダッシュボードウィジェットをウィンドウの制限を超えて水平方向に配置することはできません。

技術的には、ダッシュボードは常に同じ幅の 12 個の水平列で構成され、動的に伸縮します (ただし、合計 1200 ピクセル未満にはなりません)。

ダッシュボードには、垂直方向に最大 64 行を含めることができます。各行の高さは 70 ピクセルに固定されています。ウィジェットの高さは最大 32 行です。

ダッシュボードの表示

構成されているすべてのダッシュボードを表示するには、セクションタイトルのすぐ下にあるすべてのダッシュボードをクリックします。



ダッシュボードは共有タグ付きで表示されます。

- 私が作成 - プライベートダッシュボードを表示します
- すべて - 任意のユーザーまたはユーザーグループと共有されるパブリックダッシュボードまたはプライベートダッシュボードを表示します

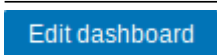
リストの右上にあるフィルターを使用すると、ダッシュボードを名前と現在のユーザーが作成したものでフィルターできます。


1 つまたは複数のダッシュボードを削除するには、それぞれのダッシュボードのチェックボックスをオンにして、リストの下にある削除をクリックします。

ダッシュボードの表示と編集

単一のダッシュボードを表示するには、ダッシュボードのリストでその名前をクリックします。

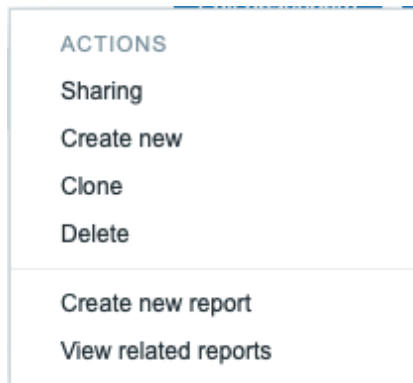
ダッシュボードを表示する場合、次のオプションを使用できます。



ダッシュボードの編集モードに切り替えます。
新しいダッシュボードが作成されているとき、およびウィジェットの  編集ボタンをクリックしたときも編集モードが開きます。



アクションメニューを開きます (以下のアクションの説明を参照)。



共有 - ダッシュボードの共有設定を編集します。ダッシュボードはパブリックまたはプライベートにすることができます。パブリックダッシュボードはすべてのユーザーに表示されます。プライベートダッシュボードは、所有者にのみ表示されます。

プライベートダッシュボードは、所有者が他のユーザーやユーザーグループと共有できます。共有の構成の詳細についてはマップの**設定**セクションを参照してください。

新規作成 - 新しいダッシュボードを**作成**します。

複製 - 既存のダッシュボードのプロパティをコピーして、新しいダッシュボードを作成します。まず、ダッシュボードのパラメータを入力するように求められます。次に、新しいダッシュボードが編集モードで開き、元のダッシュボードのすべてのウィジェットが表示されます。

削除 - ダッシュボードを削除します。

Create new report-レポート**設定フォーム**を含むポップアップウィンドウを開きます。ユーザーがスケジュールされたレポートを管理する権限を持っていない場合は無効になります。

View related reports - 現在のダッシュボードに基づいた既存のレポートのリストを含むポップアップウィンドウを開きます。関連するレポートがない場合、またはユーザーにスケジュールされたレポートを表示する権限がない場合は無効になります。

ページコンテンツのみを表示します。(キオスクモード)

キオスクモードには次の URL パラメータを使用してアクセスすることもできます：

```
/zabbix.php?action=dashboard.view&kiosk=1
```

通常モードに戻るには：

```
/zabbix.php?action=dashboard.view&kiosk=
```



ダッシュボードを編集する場合、次のオプションを使用できます。



ダッシュボードの**一般設定値**編集

+ Add ▼

Add widget
Add page
Paste widget
Paste page

新しいウィジェットを追加します。
矢印ボタンをクリックすると、アクションメニューが開きます（以下のアクションの説明を参照）

ウィジェットの追加 - 新しいウィジェットの追加
ページの追加 - 新しいページの追加
ウィジェットの貼り付け - コピーしたウィジェットの貼り付け。ウィジェットがコピーされていない場合、このオプションはグレーで表示されます。一度にコピーできるエンティティ（ウィジェットまたはページ）は1つだけです。
ページの貼り付け - コピーしたページを貼り付けます。ページがコピーされていない場合、このオプションはグレーで表示されます。

Save changes

Cancel

ダッシュボードの変更を保存

ダッシュボードの変更をキャンセル

ダッシュボードの作成

新しいダッシュボードは、次の2つの方法で作成できます。

- すべてのダッシュボードを表示してダッシュボードの作成をクリックします
- 単一のダッシュボードを表示しているときに、アクションメニューから新規作成を選択します

最初に、一般的なダッシュボードパラメータを入力するように求められます。

Dashboard properties [X]

* Owner: Admin (Zabbix Administrator) [X] [Select]

* Name: New dashboard

Default page display period: 30 seconds [v]

Start slideshow automatically:

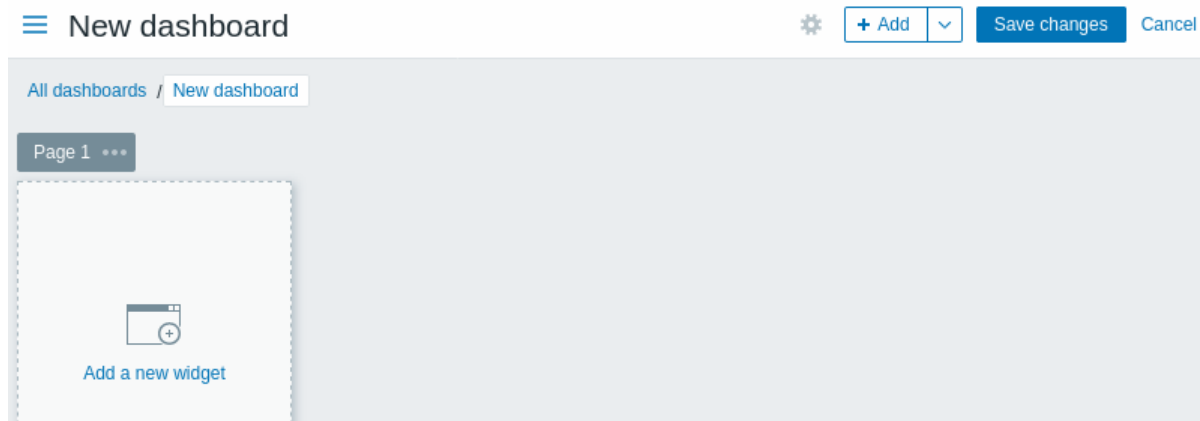
[Apply] [Cancel]

パラメーター

詳細

所有者	ダッシュボードの所有者となるシステムユーザーを選択します。
名前	ダッシュボード名を入力します。
デフォルトのページ表示間隔	スライドショーでダッシュボードページが表示される期間を選択します

適用をクリックすると、空のダッシュボードが開きます。




ダッシュボードでデータを表示するために、ウィジェットとページを追加できます。

[変更を保存] ボタンをクリックしてダッシュボードを保存します。[キャンセル] をクリックするとダッシュボードは作成されません。

ウィジェットの追加

ウィジェットをダッシュボードに追加するには

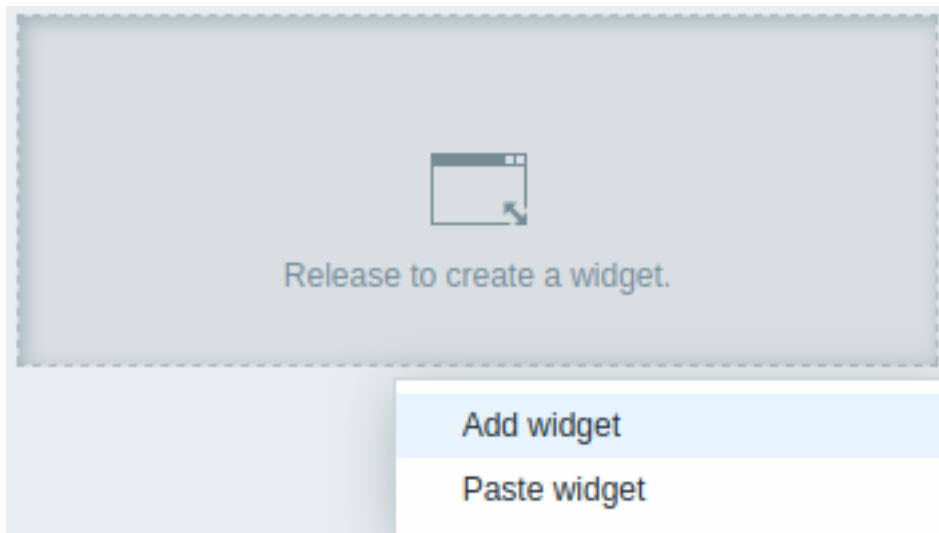


- 矢印をクリックして開くことができるアクションメニューの  ボタンまたはウィジェットの追加オプションをクリックして表示されるウィジェット設定フォームに入力します。ウィジェットはデフォルトのサイズで作成され、既存のウィジェット（存在する場合）の後に配置されます。

または

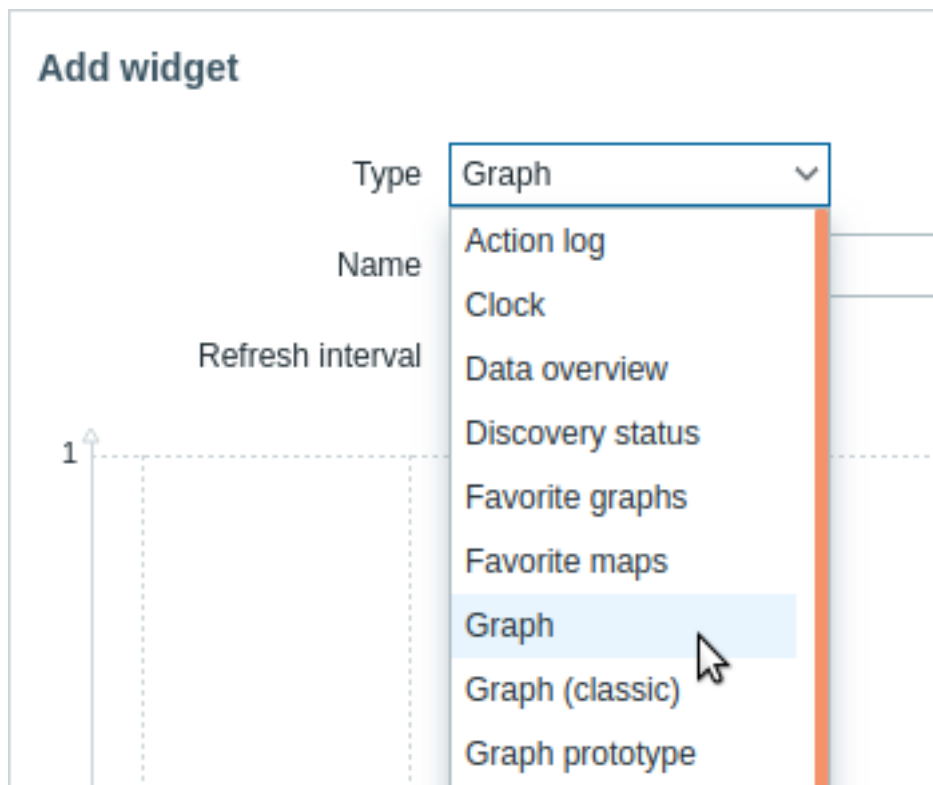
- 新しいウィジェット配置したい空白の箇所にマウスを移動します。マウスオーバーするとダッシュボードの空白箇所にプレースホルダーが表示されます。次にプレースホルダーをクリックしてウィジェット設定フォームを開きます。フォームに入力した後、ウィジェットはデフォルトサイズで作成されます。デフォルトサイズが使用可能なサイズよりも大きい場合は、使用可能なスペースを占有します。またはプレースホルダーをクリックして目的のウィジェットサイズにドラッグし、離してからウィジェット構成フォームに入力することもできます。（クリップボードにウィジェットがコピーされている場合、最初にウィジェットの追加オプションとウィジェットの貼り付けオプションのどちらかを選択してウィジェットを作成するように求められます。）





ウィジェット構成フォーム：

- ウィジェットのタイプを選択します
- ウィジェットのパラメータを入力します
- 追加をクリックします





ウィジェット

次のウィジェットをダッシュボードに追加できます。

- アクションログ
- 時刻
- データの概要
- ディスカバリのステータス
- お気に入りグラフ
- お気に入りのマップ
- 地理マップ
- グラフ
- グラフ (クラシック)
- グラフのプロトタイプ
- ホスト稼働状況
- アイテムの値
- マップ

- マップナビゲーション
- プレーンテキスト
- 障害中のホスト
- 障害
- SLA レポート
- システム情報
- 深刻度ごとの障害数
- 上位ホスト
- トリガーの概要
- URL
- Web 監視

ダッシュボード編集モードでは、ウィジェットのタイトルバーをクリックして新しい場所にドラッグすることで、ウィジェットのサイズを変更したり、ダッシュボード内を移動したりできます。また、ウィジェットの右上隅にある次のボタンをクリックして、次の操作を行うこともできます。

-  - ウィジェットの編集
-  - **ウィジェットメニュー**にアクセス

ダッシュボードの変更を保存をクリックして、ウィジェットへの変更を確定します。

ウィジェットのコピー/貼り付け

ダッシュボードウィジェットはコピーして貼り付けることで、既存のウィジェットのプロパティを使用して新しいウィジェットを作成できます。同じダッシュボード内、または異なるタブで開いたダッシュボード間でコピーして貼り付けることができます。

ウィジェットメニューを使用してウィジェットをコピーできます。ウィジェットを貼り付けるには

- ダッシュボードを編集するとき、追加ボタンの横にある矢印をクリックしてウィジェットの貼り付けオプションを選択します
- ダッシュボードの一部の領域を選択して新しいウィジェットを追加する場合は、ウィジェットの貼り付けオプションを使用します (貼り付けオプションを使用できるようにするには、ウィジェットを最初にコピーする必要があります)

コピーしたウィジェットを**ウィジェットメニュー**の貼り付けオプションで既存のウィジェットに貼り付けることもできます。

スライドショーの作成

ダッシュボードに 2 つ以上のページが含まれている場合、(**ページの追加参照**) 次のいずれかに該当するとスライドショーが自動的に実行されます。





- スライドショーの自動開始オプションがダッシュボードのプロパティでチェックされている
- ダッシュボードの URL に `slideshow=1` パラメータが含まれている

各ページはダッシュボードと個々のページのプロパティで指定された間隔に従って切り替わります。以下をクリックで動作を切り替えます。

- スライドショーの停止 - スライドショーを停止します
- スライドショーの開始 - スライドショーを開始します



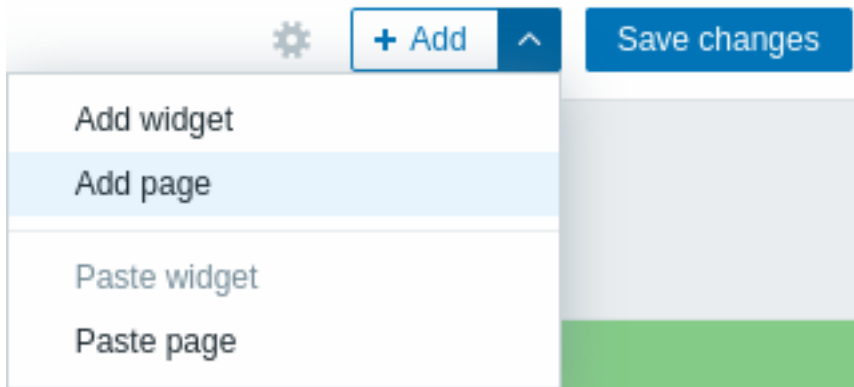
スライドショー関連のコントロールは **kiosk モード** (ページのコンテンツのみが表示される) でも使用できます。

-  - スライドショーの停止
-  - スライドショーの開始
-  - ひとつ前のページに
-  - 次のページに

ページの追加

ダッシュボードにページを追加するには

- ダッシュボードが**編集モード**になっていることを確認します
- 追加ボタンの横にある矢印をクリックしてページを追加オプションを選択します



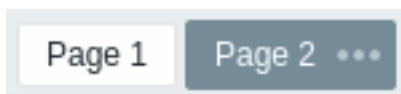
- 一般的なページパラメータを入力し、[適用] をクリックします。名前を空のままにすると、ページに「ページ N」という名前が追加されます。「N」はページの増分番号です。ページ表示期間でスライドショーでページを表示する期間をカスタマイズできます。

Dashboard page properties

Name

Page display period

新しいタブ（ページ 2）で表示される新しいページが追加されます。



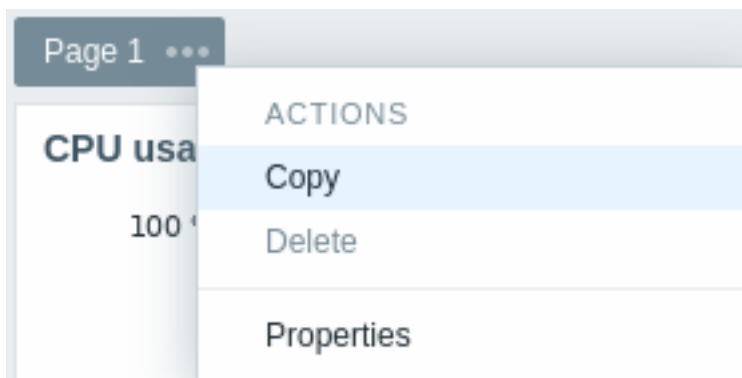
ページタブをドラッグアンドドロップすることで、ページを並べ替えることができます。並べ替えても元のページの名前が維持されます。タブをクリックすると、いつでも各ページに移動できます。

新しいページは空で追加されます。上記のようにウィジェットを追加できます。

ページのコピーと貼り付け

ダッシュボードページをコピーして貼り付けることで、既存のページのプロパティを使用して新しいページを作成できます。同じダッシュボードまたは別のダッシュボードから貼り付けることができます。


既存のページをダッシュボードに貼り付けるには、まず**ページメニュー**を使用してページをコピーします。

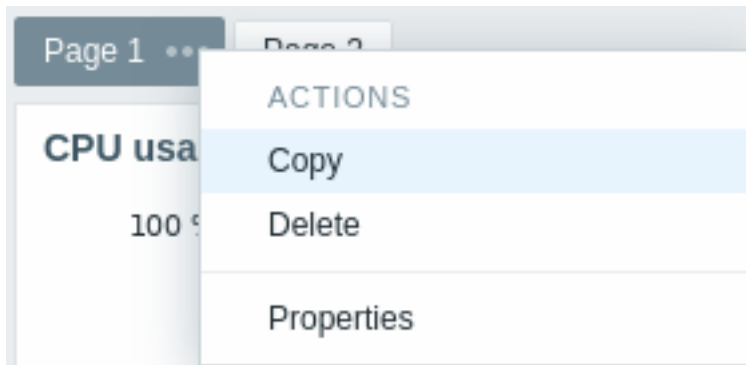


コピーしたページを貼り付けるには：

- ダッシュボードが**編集モード**になっていることを確認します
- 追加ボタンの横にある矢印をクリックしてページの貼り付けオプションを選択します

ページメニュー

ページメニューは、ページ名の横にある 3 つのドット  をクリックして開くことができます。



メニューには次のオプションが含まれています。

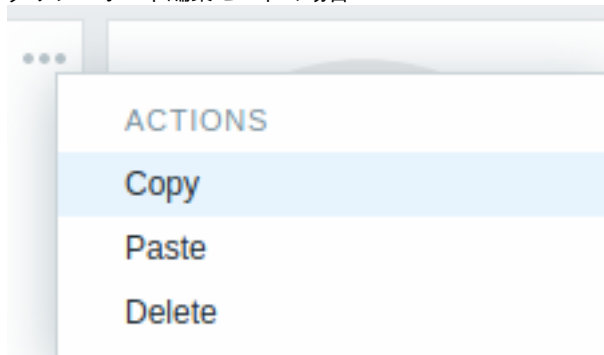
- コピー - ページをコピーします
- 削除 - ページを削除します (ページはダッシュボード編集モードでのみ削除できます)
- プロパティ - ページパラメータ (スライドショーの名前とページ表示期間) をカスタマイズします

ウィジェットメニュー

ウィジェットメニューには、ダッシュボードが編集モードか表示モードかに基づいてさまざまなオプションが含まれています。

ウィジェットメニュー

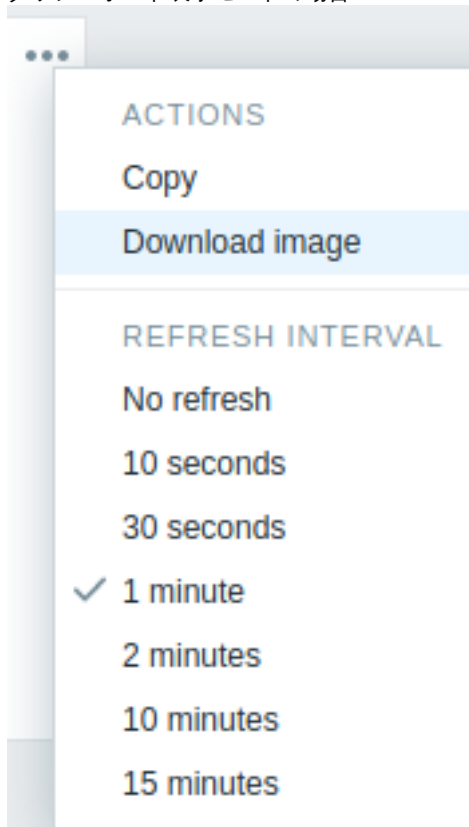
ダッシュボード編集モードの場合：



オプション

コピー - ウィジェットをコピーします
 貼り付け - コピーしたウィジェットをこのウィジェットに貼り付けます
 ウィジェットがコピーされていない場合、このオプションはグレー表示されます。
 削除 - ウィジェットを削除します

ダッシュボード表示モードの場合：



コピー - ウィジェットをコピーします
 イメージのダウンロード - ウィジェットを PNG 画像としてダウンロードします
 (グラフまたはグラフ (クラシック) でのみ使用可能)
 リフレッシュ間隔 - ウィジェットのコンテンツを更新する間隔を選択します

ダイナミックウィジェット

設定が以下のウィジェットの場合

- クラシックグラフ
- グラフプロトタイプ
- アイテムの値
- プレーンテキスト
- URL

ダイナミックアイテムと呼ばれる追加のオプションがあります。このボックスをオンにすると、ウィジェットを動的にすることができます。つまり、選択したホストに基づいてさまざまなコンテンツを表示できます。

これで、ダッシュボードを保存すると、ホストを選択するための新しいホスト選択フィールドがダッシュボードの上に表示されます (選択ボタンを使用すると、ポップアップでホストグループを選択できます)。



このようにして、選択されたホストからのデータに基づくコンテンツを表示できるウィジェットが作成できます。この利点は、たとえば、さまざまなホストからのデータを含む同じグラフを表示したいという理由だけで、追加のウィジェットを作成する必要がないことです。

ダッシュボードへのアクセス許可

通常のユーザーおよび「管理者」タイプのユーザーのダッシュボードへのアクセス許可は、次のように制限されます。

- ダッシュボードに対する読み取り権限が少なくともある場合は、ダッシュボードを表示して複製できます。
- ダッシュボードに対する読み取り/書き込み権限がある場合にのみ、ダッシュボードを編集および削除できます。
- ダッシュボードの所有者を変更することはできません。

ホストメニュー

障害ウィジェットでホストをクリックすると、ホストメニューが表示されます。ホストインベントリ、最新データ、障害、グラフ、ダッシュボード、Web シナリオ、および設定へのリンクが含まれています。ホストの設定は管理者およびスーパー管理者ユーザーのみが使用できます。

Time ▼	Info	Host	Problem * Severity	Duration	Ack	Actions
2020-02-12 10:06:35		Zabbix	Operating system description has	8m 13d 1h	No	1 3

Graph

150 Kbps

100 Kbps

グローバルスクリプトはホストメニューから実行することもできます。これらのスクリプトをホストメニューで使用するには、スコープを「手動ホストアクション」として定義する必要があります。

ホストメニューには、他のいくつかのフロントエンドセクションでホストをクリックしてアクセスできます。

- 監視データ → 障害
- 監視データ → 障害 → イベント詳細
- 監視データ → ホスト
- 監視データ → ホスト → Web 監視
- 監視データ → 最新データ
- 監視データ → マップ
- レポート → 障害発生数上位 100 項目

障害イベントのポップアップ

障害イベントのポップアップには、このトリガーの障害イベントのリストと、定義されている場合はトリガーの説明とクリック可能な URL が含まれます。

Time	Info	Host	Problem * Severity	Duration
05/07/2020 11:27:12 AM		Server3	/: Disk space is critically low (>90% used)	10m 22d 23

障害のイベントポップアップを表示するには:

- 障害ウィジェットの継続期間列で障害期間にマウスオーバーします。期間からマウスを離すとポップアップが消えます。
- 障害ウィジェットの継続期間列の期間をクリックします。期間をもう一度クリックするとポップアップが消えます。

1 ダッシュボードウィジェット

概要

このセクションではダッシュボードウィジェットに共通するパラメーターの詳細を説明します。

共通パラメータ

次のパラメータは、すべてのウィジェットで共通です。

名前	説明
リフレッシュ間隔	ウィジェット名を入力します デフォルトの更新間隔を構成します。ウィジェットのデフォルト更新間隔は、ウィジェットのタイプに応じてリフレッシュなしから 15 分の範囲です。例：URL ウィジェットの場合はリフレッシュなし、アクションログウィジェットの場合は * 1 分、時計ウィジェットの場合は 15 分 *
ヘッダーを表示	チェックボックスをオンにすると、ヘッダーが永続的に表示されます。 チェックを外すと、スペースを節約するためにヘッダーが非表示になり、表示モードと編集モードの両方で、マウスをウィジェット上に置くと上にスライドして表示されます。ウィジェットを新しい場所にドラッグするときにも半透明になります。

ウィジェットの更新間隔は、対応するすべてのユーザーのデフォルト値に設定できます。また、各ユーザーは独自の更新間隔値を設定できます。

- 対応するすべてのユーザーのデフォルト値を設定するには、編集モードに切り替え ([ダッシュボードの編集] ボタンをクリックして適切なウィジェットを見つけ、[編集] ボタンをクリックしてウィジェットの編集フォームを開きます) 更新間隔をドロップダウンリストから選択します。
- 表示モードでは、特定のウィジェットの ボタンをクリックすることにより、ユーザーごとに一意の更新間隔を個別に設定できます。

ユーザーが設定した一意の更新間隔は、ウィジェットの設定よりも優先され、一度設定すると、ウィジェットの設定が変更されても常に保持されます。

各ウィジェットの特定のパラメーターを確認するには、以下の個々のウィジェットページにアクセスしてください。

- [アクションログ](#)
- [時刻](#)
- [ディスクバリのステータス](#)

- お気に入りグラフ
- お気に入りのマップ
- 地理マップ
- グラフ
- グラフ (クラシック)
- グラフのプロトタイプ
- ホスト稼働状況
- アイテムの値
- マップ
- マップナビゲーション
- プレーンテキスト
- 障害中のホスト
- 障害
- SLA レポート
- システム情報
- 深刻度ごとの障害数
- 上位ホスト
- トリガーの概要
- URL
- Web 監視

廃止予定のウィジェット：

- データの概要

Attention:

廃止予定のウィジェットは、今後のメジャーリリースで削除されます。

1 アクションログ

概要

アクションログウィジェットでは、アクション操作の詳細（通知、リモートコマンド）を表示できます。管理 → 一般設定 → 監査ログからの情報を複製します。

設定

設定するには、タイプでアクションログを選択します。

Add widget ✕

Type Show header

Name

Refresh interval

Sort entries by

* Show lines

すべてのウィジェットの共通パラメーターに加えて、次の特定オプションを設定できます。

次による項目のソート

エントリを次の順序で並べ替えます：

時間（降順または昇順）

タイプ（降順または昇順）

ステータス（降順または昇順）

** 送信先 **（降順または昇順）

表示する行数

ウィジェットに表示されるアクションログの行数を設定します

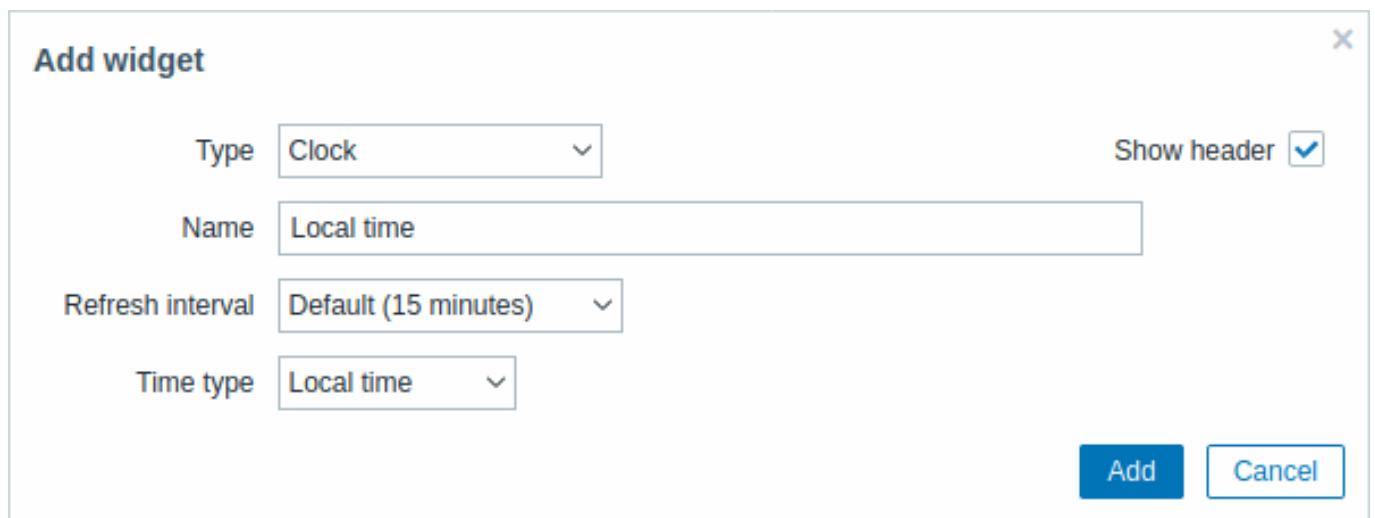
2 時刻

概要

時計ウィジェットでは、ローカル、サーバー、または指定されたホスト時間を表示できます。

設定

設定するには、タイプとして時刻を選択します。



Add widget ✕

Type Show header

Name

Refresh interval

Time type

すべてのウィジェットの共通パラメーターに加えて、次の特定オプションを設定できます。

時間の形式

ローカル、サーバー、または指定されたホストの時間を選択します。

Item

サーバー時間は、グローバルまたは Zabbix ユーザーに設定されたタイムゾーンと同じになります。

時間を表示する項目を選択します。ホスト時間を表示するには `system.localtime[local]` item を使用します。このアイテムはホストに存在する必要があります。

このフィールドはホスト時間が選択されている場合にのみ使用できます。

3 データの概要

Attention:

このウィジェットは非推奨であり、今後のメジャーリリースで削除される予定です。

概要

データの概要ウィジェットでは、ホストグループの最新データを表示できます。

障害項目の色は障害の深刻度の色に基づいており、障害の更新画面で調整できます。

By default, only values that fall within the last 24 hours are displayed. This limit has been introduced with the aim of improving initial loading times for large pages of latest data. This limit is configurable in Administration → General → GUI, using the Max

history display period option. デフォルトでは過去 24 時間以内の値のみが表示されます。この制限は最新データが大きなページとなった場合の初期読み込み時間を改善することを目的として導入されました。この制限は管理 → 一般設定 → 表示設定でヒストリの最大表示期間オプションを使用して変更できます。

データをクリックすると、いくつかの事前定義されたグラフまたは最新の値へのリンクが表示されます。

デフォルトでは 50 レコードが表示されます。(管理 → 一般設定 → 表示設定の概要テーブル内の列と行の最大数オプションで変更可能です) 最大数より多くのレコードが存在する場合は、テーブルの下部にメッセージが表示され、より具体的なフィルタリング基準を提供するように求められます。改ページはありません。

設定

設定するにはタイプとしてデータの概要を選択します。

The screenshot shows the 'Add widget' dialog box. The 'Type' dropdown is set to 'Data overview'. The 'Name' field contains 'default'. The 'Refresh interval' dropdown is set to 'Default (1 minute)'. There are two search fields for 'Host groups' and 'Hosts', both with 'type here to search' as a placeholder. The 'Tags' section has 'And/Or' selected. A tag configuration row shows 'tag' in the input, 'Contains' in the dropdown, and 'value' in the input. There is an 'Add' link below the tag configuration. The 'Show suppressed problems' checkbox is unchecked. The 'Hosts location' has 'Left' selected. At the bottom right are 'Add' and 'Cancel' buttons.

すべてのウィジェットの共通であるパラメーターに加えて、次のオプションを設定できます。

ホストグループ

ホストグループを選択します。このフィールドはオートコンプリートであるため、グループの名前を入力し始めると、一致するグループ名のドロップダウンが表示されます。スクロールして選択します。'x' をクリックすると選択したものを削除できます。

ホスト

ホストを選択します。このフィールドはオートコンプリートであるため、ホストの名前を入力し始めると、一致するホスト名のドロップダウンが表示されます。スクロールして選択します。'x' をクリックすると選択したものを削除できます。

タグ

タグを指定して、ウィジェットに表示されるアイテムデータの数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。各条件で使用できる演算子はいくつかあります。

Exists - 指定されたタグ名を含める

Equals - 指定されたタグ名と値を含める（大文字と小文字を区別）

Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める（文字列の部分一致、大文字と小文字を区別しない）

Does not exist - 指定されたタグ名を除外する

Does not equal - 指定されたタグ名と値を除外する（大文字と小文字を区別）

Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する（文字列の部分一致、大文字と小文字を区別しない）

条件には 2 つの計算タイプがあります：

And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる

フィルタリングするとタグ表示優先度（以下を参照）リストで上書きされない限り、ここで指定されたタグが障害とともに最初に表示されます。

チェックボックスをオンにすると、ホストのメンテナンス中に抑制される（表示されない）障害が表示されます。

ホストの場所を左または上で選択します。

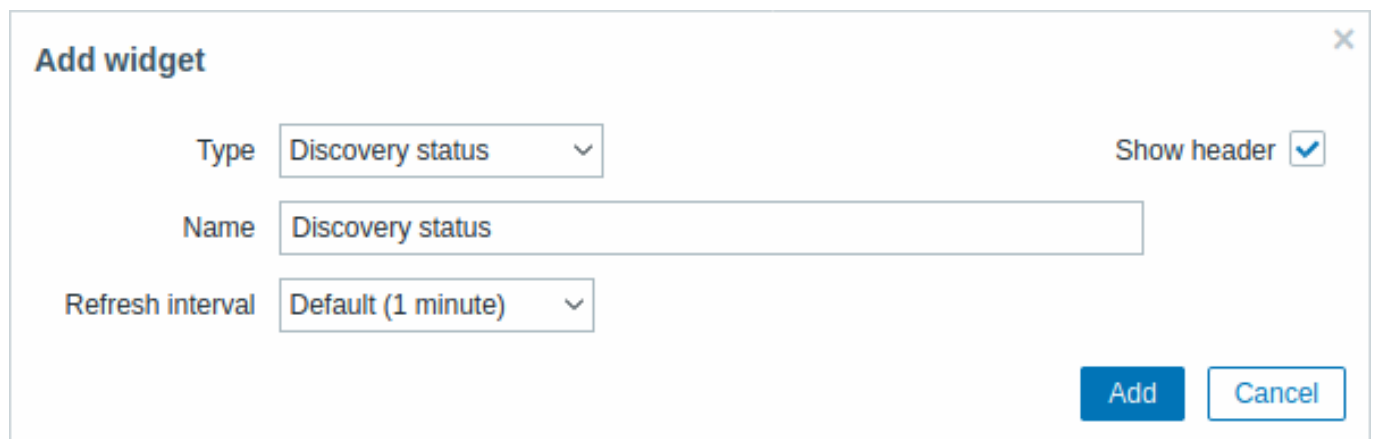
メンテナンス中の障害を表示

ホストの位置

4 ディスカバリのステータス

概要

このウィジェットは、アクティブなネットワークディスカバリルールのステータス概要を表示します。




The screenshot shows a dialog box titled "Add widget" with a close button (X) in the top right corner. It contains three configuration fields: "Type" with a dropdown menu showing "Discovery status", "Name" with a text input field containing "Discovery status", and "Refresh interval" with a dropdown menu showing "Default (1 minute)". To the right of the "Type" field is a "Show header" checkbox which is checked. At the bottom right of the dialog are two buttons: "Add" (highlighted in blue) and "Cancel".

すべての構成パラメーターは共通です。

5 お気に入りのグラフ


概要

このウィジェットにはアルファベット順にソートされた、お気に入りの追加されたグラフへのショートカットが含まれています。

グラフを表示してから  お気に入りの追加ボタンをクリックすると、ショートカットのリストに表示されます。


すべての構成パラメーターは共通です。

6 お気に入りのマップ

The list of shortcuts is populated when you **view** a map and then click on its  Add to favorites button.

概要

このウィジェットにはアルファベット順にソートされた、お気に入りの追加されたマップへのショートカットが含まれています。

マップを**表示**してから  お気に入りの追加ボタンをクリックすると、ショートカットのリストに表示されます。

すべての構成パラメーターは**共通**です。

7 地理マップ

概要

地理マップウィジェットは、オープンソースの JavaScript インタラクティブマップライブラリ Leaflet を使用して、ホストを地理マップ上のマーカーとして表示します。

Note:

Zabbix は複数の事前定義されたマップタイルサービスプロバイダーとカスタムタイルサービスプロバイダーまたはホストタイル自体を追加するオプションを提供します (管理 → 一般設定 → 地理マップ [メニューセクション](#) で設定可能)

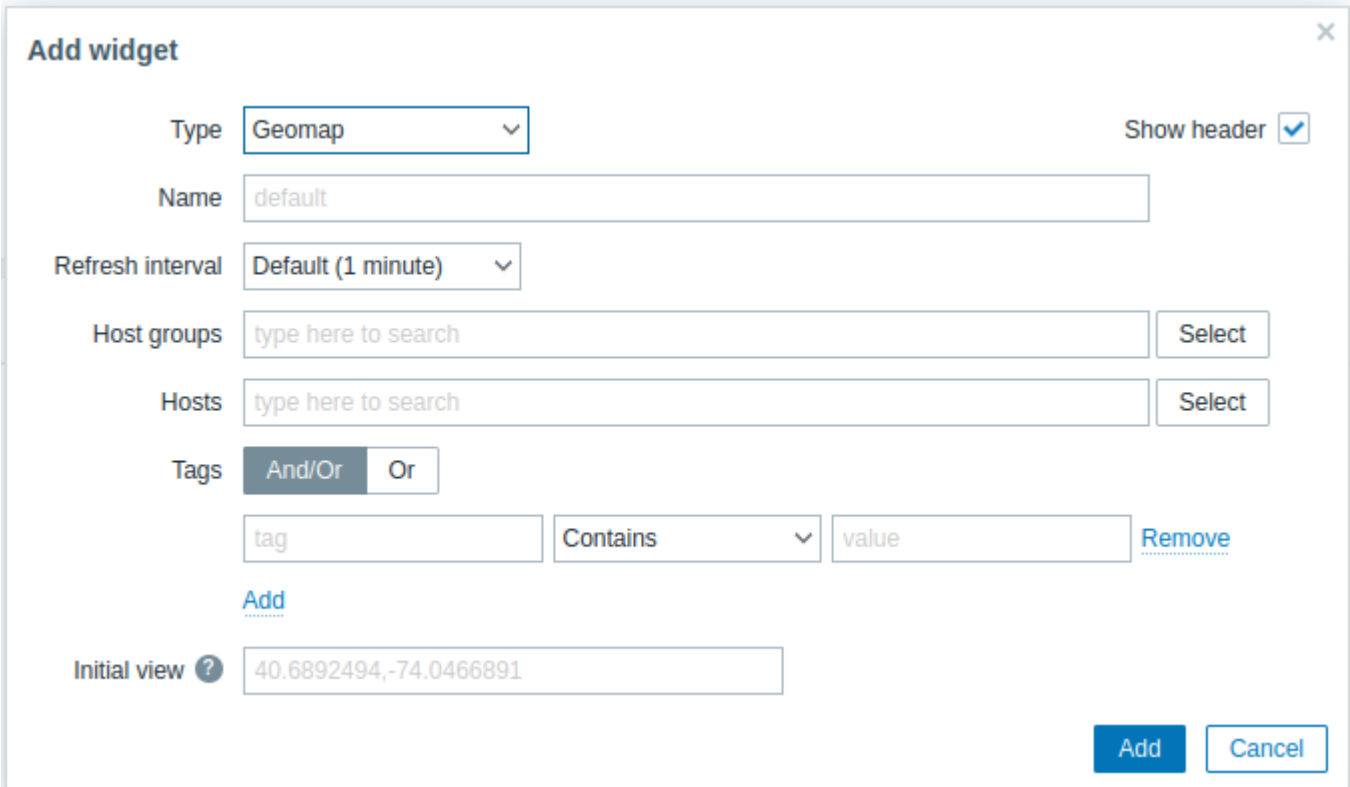
デフォルトでは、ウィジェットはホスト設定で定義された有効な地理座標を持つすべての有効なホストを表示します。ウィジェットパラメータでホストフィルタリングを設定することができます。

有効なホスト座標は次のとおりです。

- 緯度：-90~90 (整数または浮動小数点数)
- 経度：-180 から 180 まで (整数または浮動小数点数)

設定

ウィジェットを追加するにはタイプとして地理マップを選択します。



Add widget ✕

Type Show header

Name

Refresh interval

Host groups

Hosts

Tags

[Add](#)

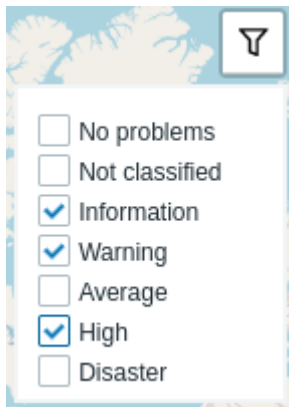
Initial view ?

すべてのウィジェットの**共通**パラメーターに加えて、次のオプションを設定できます。

ホストグループ	<p>マップに表示するホストグループを選択します。このフィールドはオートコンプリートであるため、グループの名前を入力し始めると、一致するグループ名がドロップダウンで表示されます。スクロールして選択します。'x' をクリックして選択したグループを削除できます。</p> <p>* ホストグループフィールドとホスト * フィールドの両方で何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。</p>
ホスト	<p>すべてのマップを表示するホストを選択します。このフィールドはオートコンプリートであるため、ホストの名前を入力し始めると、一致するホスト名がドロップダウンで表示されます。スクロールして選択します。'x' をクリックして選択したホストを削除できます。</p> <p>ホストグループフィールドとホストフィールドの両方で何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。</p>
タグ	<p>タグを指定して、ウィジェットに表示されるホストの数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。</p> <p>条件ごとに使用できる演算子がいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名と値を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める (文字列の部分一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (文字列の部分一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります :</p> <p>And/Or - すべての条件が満たされる必要があります、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる</p>
初期表示位置	<p>ウィジェットが最初に <latitude>、<longitude>、<zoom> の形式でロードされたときに表示する、カンマ区切りの中心座標とオプションのズームレベル</p> <p>初期ズームが指定されている場合、ジオマップウィジェットは与えられたズームレベルで表示します。それ以外の場合、初期ズームは特定のタイルプロバイダーの最大ズームの半分として計算されます。</p> <p>デフォルトビューが設定されている場合、初期ビューは無視されます (以下を参照)</p> <p>例 :</p> <p>=>40.6892494,-74.0466891,14</p> <p>=>40.6892494,-122.0466891</p>

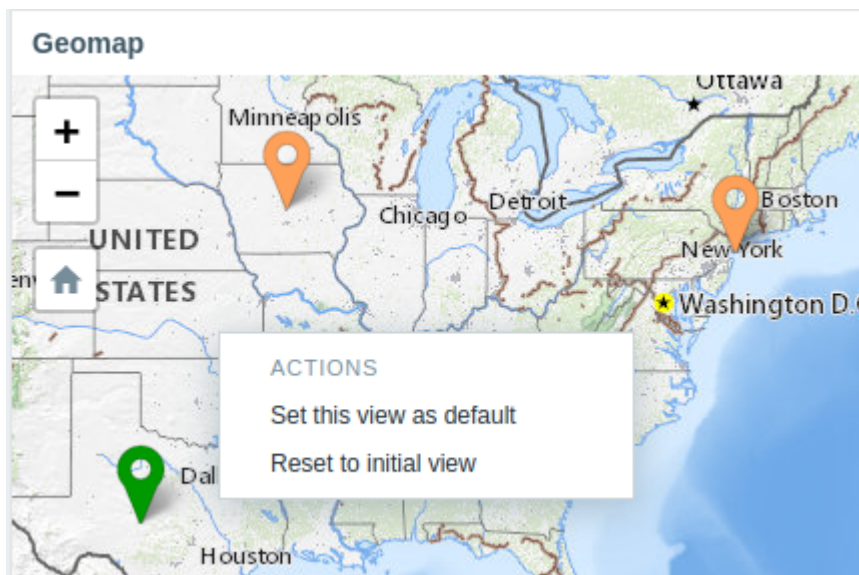
マップに表示されるホストマーカーはホストの最も深刻な障害の色です。ホストに障害がない場合は緑色になります。ホストマーカーをクリックすると、ホストの表示名と深刻度別にグループ化された未解決の障害の数を表示できます。表示されている名前をクリックすると、**ホストメニュー**が開きます。

マップに表示されるホストは障害の深刻度でフィルタリングできます。ウィジェットの右上隅にあるフィルターアイコンを押して、表示したい深刻度をチェックします。



ウィジェットの左上隅にあるプラスボタンとマイナスボタンを使用するか、マウスのスクロールホイールまたはタッチパッドを使用して、マップをズームインおよびズームアウトできます。現在のビューをデフォルトとして設定するには、マップ上の任意の場所を右クリックして、[この表示をデフォルトにセット]を選択します。この設定は、現在のユーザーの初期ビューウィジェットパラメータを上書きします。この操作を元に戻すには、マップ上の任意の場所をもう一度右クリックして、[初期表示にリセット]を選択します。

初期ビューまたはデフォルトビューが設定されている場合、左側のホームアイコンを押すと、いつでもこのビューに戻ることができます。



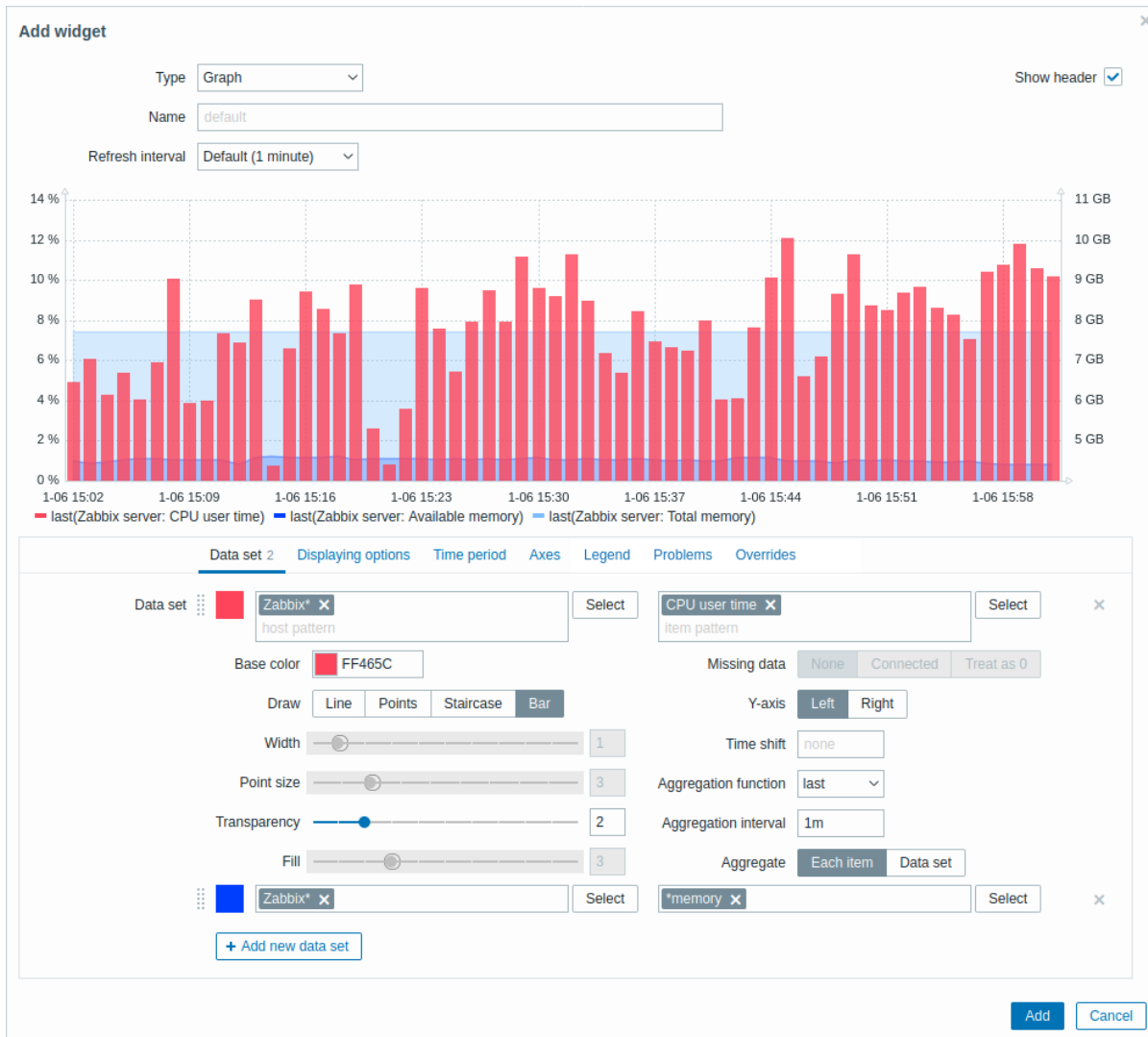
8 グラフ

概要

グラフウィジェットは、ベクター画像描画技術を使用して Zabbix によって収集されたデータを視覚化するための多目的な方法を提供します。このグラフウィジェットは Zabbix 4.0 以降でサポートされています。Zabbix 4.0 より前にサポートされていたグラフウィジェットは、引き続き **グラフ (クラシック)** として使用できます。

設定

設定するにはタイプとしてグラフを選択します。



データセットタブではデータセットを追加し、それらの視覚的表現を定義できます。

データセット

基本色

グラフに表示するホストとアイテムを選択またはホストとアイテムのパターンを入力することもできます。ワイルドカードパターンを使用できます。(たとえば * は 0 個以上の文字に一致する結果を返します) ワイルドカードパターンを指定するには、文字列を手動で入力して Enter を押します。入力中は、一致するすべてのホストがドロップダウンに表示されます。

グラフには最大 50 個のアイテムが表示される場合があります。

ホストパターンとアイテムパターンのフィールドは必須です。

ワイルドカード記号は常に解釈されるため、他に一致するアイテム (item2、item3 など) がある場合、たとえば「item*」という名前のアイテムを個別に追加することはできません。




カラーピッカーからまたは手動でベースカラーを調整します。基本色は、データセットの各項目の異なる色を計算するために使用されます。基本色の入力は必須です。

グラフの形式	<p>グラフ表示形式を選択します。選択可能な形式は、線（デフォルト）、* ポイント、階段、および棒 * です。</p> <p>線/階段グラフにデータポイントが1つしかない場合は、グラフ形式に関係なく、ポイントとして描画されます。ポイントサイズは線幅から計算されますが3ピクセル以上にすることはできません。</p>
幅	<p>線幅を設定します。このオプションは線または階段のグラフ形式が選択されている場合に使用できます。</p>
ポイントサイズ	<p>ポイントサイズを設定します。このオプションはポイントグラフ形式が選択されている場合に使用できます。</p>
透明度	<p>透明度を設定します。</p>
塗りつぶし	<p>塗りつぶしのレベルを設定します。このオプションは線または階段のグラフ形式が選択されている場合に使用できます。</p>
欠損データ	<p>欠損データを表示するためのオプションを選択します。</p> <p>なし - 欠損データは空白とします</p> <p>接続する - 欠損値の境界値が接続されます</p> <p>0 とみなす - 欠損データは 0 として表示されます</p> <p>ポイントおよびバーグラフ形式には適用されません。</p>
Y 軸	<p>Y 軸が表示されるグラフの側を選択します。</p>
タイムシフト	<p>必要に応じてタイムシフトを指定します。このフィールドでは、タイムサフィックスを使用できます。負の値が許可されます。</p>
アグリゲーション関数	<p>使用する集計関数を指定します。</p> <p>最小 - 最小値を表示します</p> <p>最大 - 最大値を表示します</p> <p>平均 - 平均値を表示します</p> <p>合計 - 値の合計を表示します</p> <p>個数 - 値の数を表示します</p> <p>第1 - 最初の値を表示します</p> <p>最新値 - 最後の値を表示します</p> <p>なし - すべての値を表示します (集計なし)</p> <p>集計を使用すると、すべての値ではなく、選択した間隔 (5 分、1 時間、1 日) の集計値を表示できます。参照: グラフの集計</p> <p>このオプションは、Zabbix4.4 以降でサポートされています。</p>
アグリゲーション間隔	<p>値を集計する間隔を指定します。このフィールドではタイムサフィックスを使用できます。サフィックスのない数値は秒と見なされます。</p> <p>このオプションは Zabbix4.4 以降でサポートされています。</p>

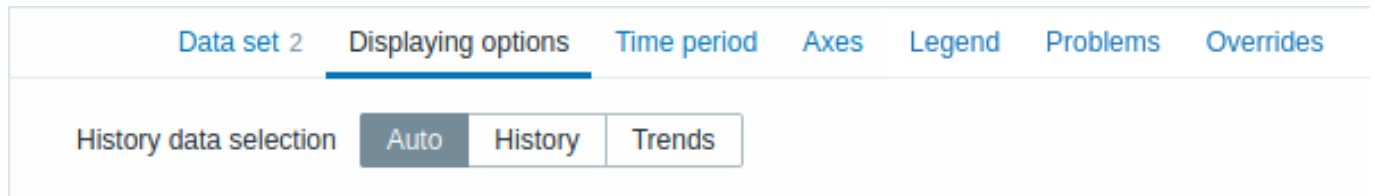
アグリゲート

集計対象を指定します。
アイテムごと - データセット内の各アイテムが集計され、個別に表示されます。
データセット - すべてのデータセットアイテムが集計され、1つの値として表示されます。
 このオプションは Zabbix4.4 以降でサポートされています。

既存のデータセットがリストに表示されます。次のこともできます。

-  - このボタンをクリックして、新しいデータセットを追加します
-  - カラーアイコンをクリックして、データセットの詳細を展開または折りたたみます
-  - 移動アイコンをクリックして、データセットをリスト内の新しい場所にドラッグします

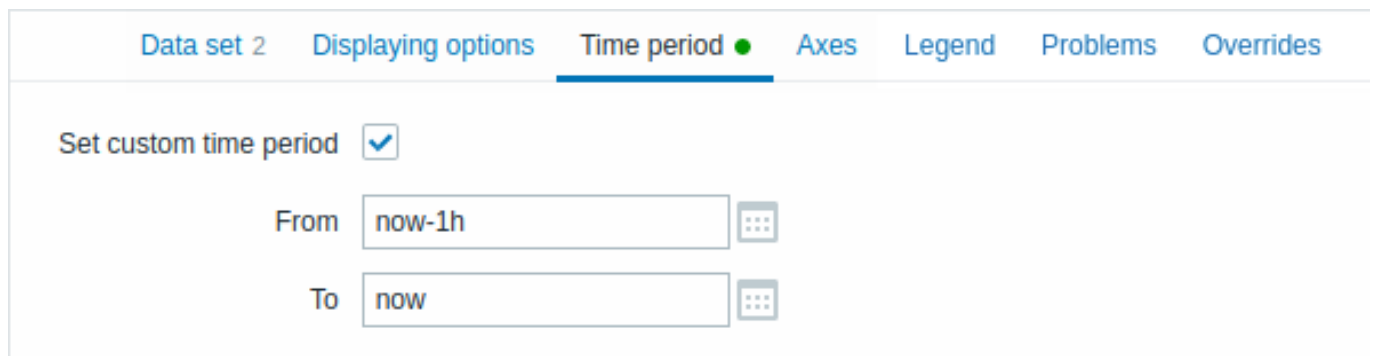
表示オプションタブでは、履歴データの選択を定義できます。



履歴データの選択

グラフデータのソースを設定します：
自動 - クラシックグラフアルゴリズム (デフォルト)
ヒストリ - 履歴からのデータ
トレンド - トレンドからのデータ

期間タブではカスタム期間を設定できます。



時間範囲のカスタマイズ このチェックボックスをオンにするとグラフのカスタム期間が設定されます (デフォルトはオフ)
開始 グラフのカスタム期間の開始時間を設定します
終了 グラフのカスタム期間の終了時間を設定します

軸タブでは軸の表示方法をカスタマイズできます。

Data set 2 Displaying options Time period **Axes** ● Legend Problems Overrides

Left Y Show Right Y Show X-Axis Show

Min Min

Max Max

Units ▼ Units ▼

左側 Y 軸	このチェックボックスをオンにすると、左の Y 軸が表示されます。データセットまたはオーバーライドタブのいずれかで選択されていない場合、チェックボックスが無効になります。
右側 Y 軸	このチェックボックスをオンにすると、右の Y 軸が表示されます。データセットまたはオーバーライドタブのいずれかで選択されていない場合、チェックボックスが無効になります。
X 軸	X 軸を非表示にするには、このチェックボックスをオフにします（デフォルトはオンです）
最小値	対応する Y 軸の可視範囲最小値を指定します。
最大値	対応する Y 軸の可視範囲最大値を指定します。
単位	ドロップダウンからグラフの軸の値の単位を選択します。自動オプションを選択すると、対応する軸の最初の項目の単位を使用して軸の値が表示されます。固定オプションを使用すると、対応する軸のカスタム名を割り当てることができます。固定オプションが選択され、値入力フィールドが空白のままの場合、対応する軸は数値のみで構成されます。

凡例タブでは、グラフの凡例をカスタマイズできます。

Data set 2 Displaying options Time period **Axes** **Legend** Problems Overrides

Show legend

Number of rows

凡例を表示	このチェックボックスをオフにすると、グラフの凡例が非表示になります（デフォルトはオンです）
行数	グラフに表示する行数を設定します。

障害タブでは障害の表示をカスタマイズできます。

Data set 2 Displaying options Time period Axes Legend **Problems** ● Overrides

Show problems

Selected items only

Problem hosts

Severity Not classified Warning High
 Information Average Disaster

Problem

Tags

[Add](#)

障害の表示

選択したアイテムのみ

障害中のホスト

深刻度
障害
タグ

このチェックボックスをオンにすると、グラフに障害を表示できるようになります（デフォルトは無効です）

選択したアイテムの障害のみグラフに表示するには、このチェックボックスをオンにします。

グラフに表示する障害のあるホストを選択します。ワイルドカードパターンを使用できます（たとえば、*は0個以上の文字に一致する結果を返します）。ワイルドカードパターンを指定するには、文字列を手動で入力してEnterを押します。入力している間、一致するすべてのホストがドロップダウンに表示されることに注意してください。

グラフに表示する障害の深刻度を選択します。

グラフに表示する障害名を指定します。

障害タグを指定して、ウィジェットに表示される障害の数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。

各条件で使用できる演算子はいくつかあります。

Exists - 指定されたタグ名を含める

Equals - 指定されたタグ名と値を含める（大文字と小文字を区別）

Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める（文字列の部分一致、大文字と小文字を区別しない）

Does not exist - 指定されたタグ名を除外する

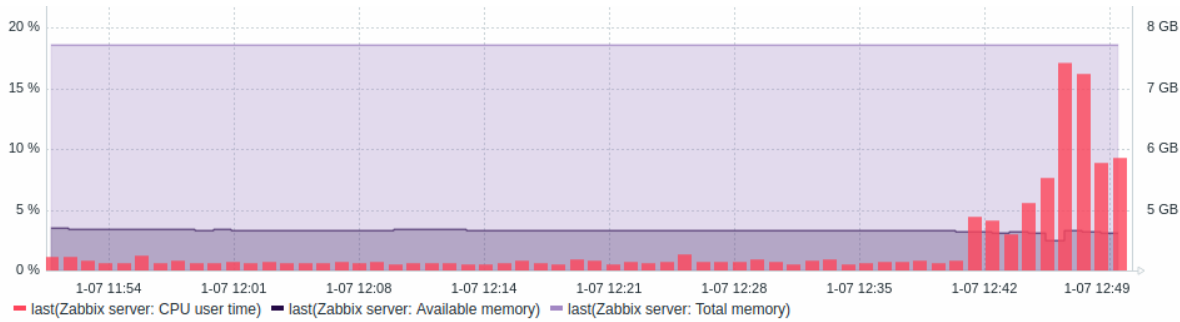
Does not equal - 指定されたタグ名と値を除外する（大文字と小文字を区別）

Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する（文字列の部分一致、大文字と小文字を区別しない）

条件には2つの計算タイプがあります：

And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件はOr条件によってグループ化されます
 Or - 1つの条件が満たされる

オーバーライドタブではデータセットにカスタムオーバーライドを追加できます。



Overrides 1

オーバーライドは、* ワイルドカードを使用してデータセットに対して複数のアイテムが選択されており、アイテムのデフォルトの表示方法（デフォルトのベースカラーやその他のプロパティなど）を変更する場合に役立ちます。

既存のオーバーライド（存在する場合）がリストに表示されます。新しいオーバーライドを追加するには：

- ボタンをクリックします
- オーバーライドするホストとアイテムを選択します。または、ホストとアイテムのパターンを入力することもできます。ワイルドカードパターンを使用できます。（たとえば * は 0 個以上の文字に一致する結果を返します）ワイルドカードパターンを指定するには、文字列を手動で入力して Enter を押します。入力している間、一致するすべてのホストがドロップダウンにどのように表示される事に注意してください。ワイルドカード記号は常に解釈されるため、他に一致するアイテム（item2、item3 など）がある場合、たとえば「item*」という名前のアイテムを個別に追加することはできません。ホストパターンとアイテムパターンのフィールドは必須です。
- をクリックして、オーバーライドパラメータを選択します。少なくとも 1 つのオーバーライドパラメータを選択する必要があります。パラメータの説明については、上記のデータセットタブを参照してください。

グラフウィジェットに表示される情報は **ウィジェットメニュー** を使用して .png 画像としてダウンロードできます。

ウィジェットのスクリーンショットがダウンロードフォルダーに保存されます。

9 グラフ（クラシック）

概要

従来のグラフウィジェットでは、単一のカスタムグラフまたは単純なグラフを表示できます。

設定

設定するにはタイプとしてグラフ（クラシック）を選択します。

Add widget [Close]

Type: Graph (classic) [v] Show header:

Name: System load

Refresh interval: Default (1 minute) [v]

Source: Graph Simple graph

* Graph: Zabbix server: System load [x] Select

Show legend:

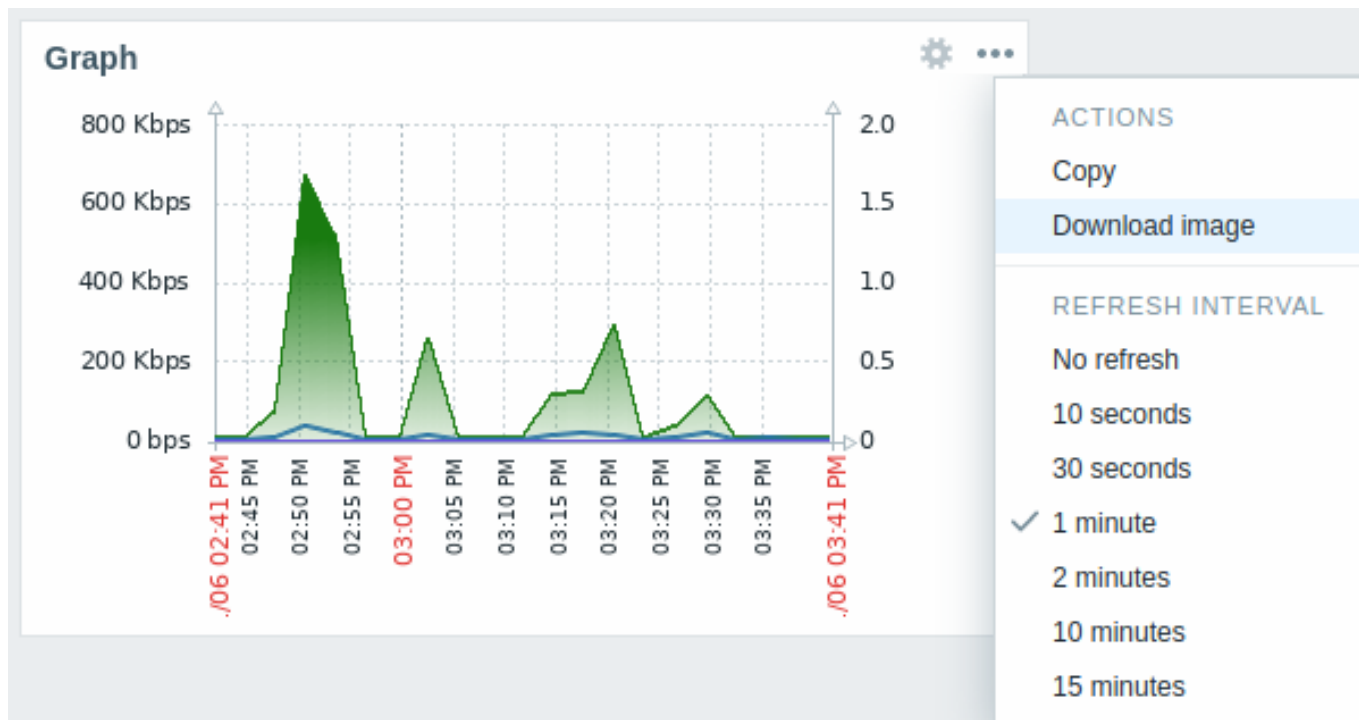
Dynamic item:

Add Cancel

すべてのウィジェットの共通パラメーターに加えて次のオプションを設定できます。

ソース	グラフの種類を選択 グラフ - カスタムグラフ シンプルグラフ - シンプルグラフ
グラフ	表示するカスタムグラフを選択します。 このオプションはソースがグラフとして選択されている場合に使用できます。
アイテム	シンプルグラフで表示する項目を選択します。 このオプションはソースがシンプルグラフとして選択されている場合に使用できます。
凡例を表示	グラフの凡例を非表示にするには、このチェックボックスをオフにします（デフォルトはオン）
ダイナミックアイテム	選択したホストに応じて異なるデータを表示するようにグラフを設定します。

クラシックグラフウィジェットによって表示される情報はウィジェットメニューを使用して.png 画像としてダウンロードできます。



ウィジェットのスクリーンショットがダウンロードフォルダーに保存されます。

10 グラフのプロトタイプ

概要

グラフのプロトタイプウィジェットでは、低レベルディスカバリによってプロトタイプグラフまたはプロトタイプアイテムのいずれかから作成されたグラフのグリッドを表示できます。

設定

設定するにはウィジェットタイプとしてグラフのプロトタイプを選択します。

Add widget ✕

Type Show header

Name

Refresh interval

Source

* Graph prototype

Show legend

Dynamic item

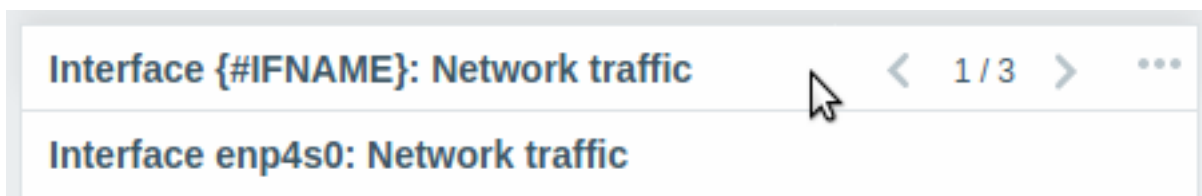
* Columns

* Rows

すべてのウィジェットの共通パラメーターに加えて次のオプションを設定できます。

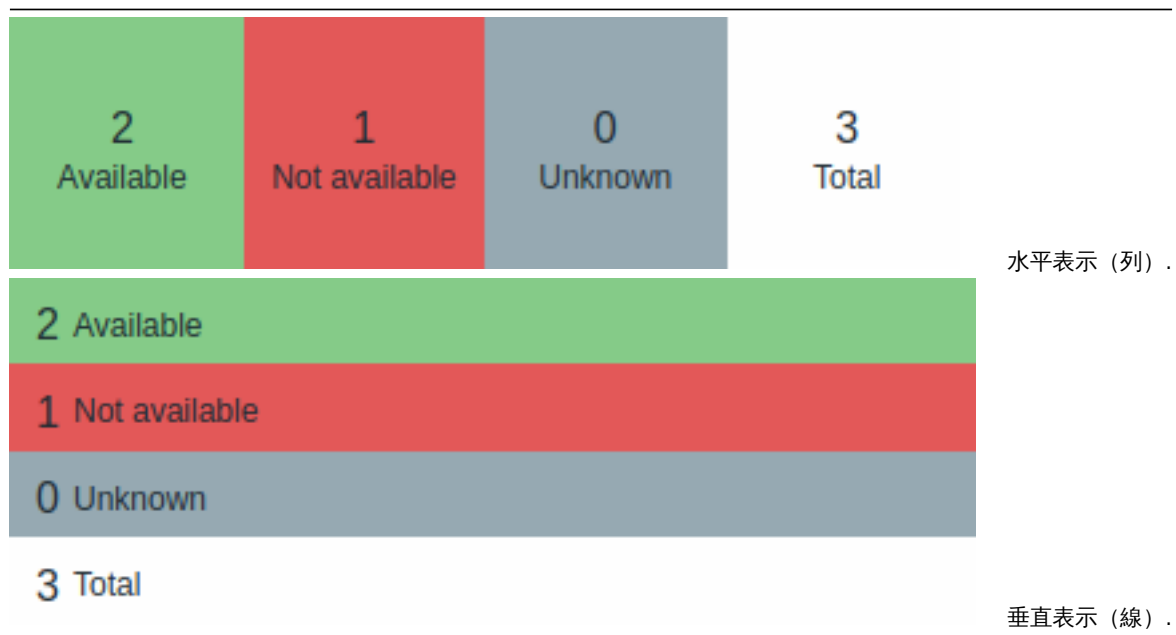
ソース	ソースをグラフのプロトタイプかシンプルグラフのプロトタイプから選択します。
グラフのプロトタイプ	グラフのプロトタイプを選択すると、グラフのプロトタイプでディスカバリされたグラフが表示されます。このオプションは、 ソース として グラフのプロトタイプ が選択されている場合に使用できます。
アイテムのプロトタイプ	アイテムのプロトタイプを選択すると、アイテムのプロトタイプでディスカバリされたアイテムに基づいてシンプルグラフが表示されます。このオプションは ソース として[シンプルグラフのプロトタイプ]が選択されている場合に使用できます。
凡例を表示	このチェックボックスをオンにすると、グラフに凡例が表示されます (デフォルトはオン)
ダイナミックアイテム	選択したホストに応じて異なるデータを表示するようにグラフを設定します。
列	グラフプロトタイプウィジェット内に表示するグラフの列数を入力します。
行	グラフプロトタイプウィジェット内に表示するグラフの行数を入力します。

列および行設定では、ウィジェットに複数のグラフを適合させることができますが、ウィジェットに列/行があるよりも多くのグラフが検出される場合があります。この場合はウィジェットでページングが使用可能になり、スライドアップヘッダーで左矢印と右矢印を使用してページを切り替えることができます。



概要

ホスト稼働状況ウィジェットでは、ホストの稼働状況に関する高レベルの統計が4つの色付きの列行に表示されます。



各列/行のホストの稼働状況は次のようにカウントされます。

- 利用可能 - すべてのインターフェースが利用可能なホスト
- 使用不可 - 少なくとも1つのインターフェースが使用できないホスト
- 不明 - 少なくとも1つのインターフェースが不明なホスト (使用不可)
- 合計 - すべてのホストの合計

設定

設定するには、タイプとしてホスト稼働状況を選択します。

Add widget ✕

Type Show header

Name

Refresh interval

Host groups Select

Interface type Zabbix agent
 SNMP
 JMX
 IPMI

Layout

Show hosts in maintenance

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

ホストグループ	ホストグループを選択します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。スクロールして選択します。'x'をクリックすると選択したものを削除できます。
インターフェースタイプ	稼働状況データを表示するホストインターフェースを選択します。何も選択されていない場合、デフォルトですべてのインターフェースの稼働状況が表示されます。
レイアウト	水平表示 (列) または垂直表示 (行) を選択します。
メンテナンス中のホストを表示	メンテナンス中のホストを統計に含めます。

12 アイテムの値

概要

アイテムの値ウィジェットでは、単一のアイテムの値を表示できます。

2022-01-05 22:43:15

2.38 ↑

Load average (5m avg)

値自体に加えて必要に応じて追加の要素を表示できます。

- メトリックの時間
- アイテム説明
- 変更した値のインジケータ
- アイテムユニット

ウィジェットは数値と文字列値を表示できます。文字列値は 1 行に表示され、必要に応じて切り捨てられます。アイテムの値がない場合は、データなしが表示されます。

値をクリックすると、数値項目のアドホックグラフまたは文字列項目の最新データが表示されます。

ウィジェットとその中のすべての要素は、高度な構成オプションを使用して視覚的に微調整でき、さまざまな視覚スタイルを作成できます。

2022-01-05 22:43:15

2.38 ↑

Load average (5m avg)

Agent status

UP ⇕

2022-01-05 22:40:42

Current
download
speed from
my favorite
website

20 ↑
KBps

2022-01-05 22:43:24

15.01
GB

Space left on drive C:

Nvidia GeForce R...

設定

設定するにはタイプとしてアイテムの値を選択します。

Edit widget ✕

Type Show header

Name

Refresh interval

* Item

* Show Description Value
 Time Change indicator

Advanced configuration

Dynamic item

すべてのウィジェットの**共通**パラメーターに加えて、次のオプションを設定できます。

アイテム 表示	アイテムを選択します チェックボックスをオンにすると、それぞれの要素（説明、値、時間、変化表示）が表示されます。非表示にするには、マークを外します。
高度な設定	少なくとも1つの要素を選択する必要があります。 チェックボックスをオンにすると、詳細設定オプションが表示されます。
ダイナミックアイテム	チェックボックスをオンにすると、選択したホストに応じて異なる値が表示されます。

高度な設定

高度な設定チェックボックスがオンになっている場合（スクリーンショットを参照）、**表示**フィールドで選択されている要素（上記を参照）に対してのみ高度な設定オプションが使用可能になります。

さらに高度な設定によりウィジェット全体の背景色を変更できます。

Advanced configuration

* Description ?

{ITEM.NAME}

Horizontal position Left Center Right Size %

Vertical position Top Middle Bottom Bold

Color

Value

Decimal places Size %

Horizontal position Left Center Right Size %

Vertical position Top Middle Bottom Bold

Color

Units

Position Size %

Bold

Color

Time

Horizontal position Left Center Right Size %

Vertical position Top Middle Bottom Bold

Color

Change indicator

Background color

説明

アイテムの説明を入力します。この説明は、デフォルトのアイテム名をオーバーライドする場合があります。複数行の説明がサポートされています。

テキストとサポートされているマクロを組み合わせることが可能です。

{HOST.}、{ITEM.}、{INVENTORY.*}、およびユーザーマクロがサポートされています。

水平の位置

アイテム説明の水平位置を左、右、または中央から選択します。

垂直の位置

アイテム説明の垂直位置を上、下、または中央から選択します。

サイズ

アイテム説明のフォントサイズを入力します (ウィジェットの高さ全体に対するパーセントで)

太字

チェックボックスをオンにすると、アイテム説明が太字で表示されます。

色	カラーピッカーからアイテム説明の色を選択します。 Dはデフォルトの色を表します(フロントエンドのテーマによって異なります)。デフォルト値に戻すには、カラーピッカーの[デフォルトを使用]ボタンをクリックします。
値	
小数点以下桁数	表示する小数点以下の桁数を入力します。この値は float アイテムにのみ影響します。
サイズ	小数点以下桁数のフォントサイズを入力します(ウィジェットの高さ全体に対するパーセントで)
水平の位置	アイテム値の水平位置を左、右、または中央から選択します。
垂直の位置	アイテム値の垂直位置を上、下、または中央から選択します。
サイズ	アイテム値のフォントサイズの高さを入力します(ウィジェットの高さ全体に対するパーセント)。 アイテム値のサイズが優先されることに注意してください。 他の要素は、アイテム値のためスペースを確保しなければなりません。アイテム値が大きすぎる場合、変化表示の変更インジケータを表示するために切り捨てられます。
太字	チェックボックスをオンにすると、アイテム値が太字で表示されます。
色	カラーピッカーからアイテム値の色を選択します。 Dはデフォルトの色を表します(フロントエンドのテーマによって異なります)。デフォルト値に戻すには、カラーピッカーの[デフォルトを使用]ボタンをクリックします。
単位	チェックボックスをオンにすると、アイテム値に単位が表示されます。単位名を入力すると、アイテム設定の単位が上書きされます。
位置	アイテム単位の位置(値の上、下、前、または後)を選択します。
サイズ	アイテム単位のフォントサイズを入力します(ウィジェットの大きさ全体に対するパーセントで)
太字	チェックボックスをオンにすると、アイテム単位が太字で表示されます。
色	カラーピッカーからアイテムユニットの色を選択します。 Dはデフォルトの色を表します(フロントエンドのテーマによって異なります)。デフォルト値に戻すには、カラーピッカーの[デフォルトを使用]ボタンをクリックします。
時間	時間は、アイテム履歴からのクロック値です。
水平の位置	時間の水平位置を左、右、または中央から選択します。
垂直の位置	時間の垂直位置を上、下、または中央から選択します。
サイズ	時間のフォントサイズを入力します(ウィジェットの大きさ全体に対するパーセントで)
太字	チェックボックスをマークして、時間を太字で表示します。
色	カラーピッカーから時間の色を選択します。 Dはデフォルトの色を表します(フロントエンドのテーマによって異なります)。デフォルト値に戻すには、カラーピッカーの[デフォルトを使用]ボタンをクリックします。
変化表示	カラーピッカーから変更インジケータの色を選択します。変化の指標は次のとおりです。 ↑ - アイテムの値が上昇しています(数値アイテムの場合) ↓ - アイテムの値が下降しています(数値アイテムの場合) ↕ - アイテムの値が変化しています(文字列アイテムとアイテムの場合) 値マッピングあり) Dはデフォルトの色を表します(フロントエンドのテーマによって異なります)。デフォルト値に戻すには、カラーピッカーの[デフォルトを使用]ボタンをクリックします。 変更インジケータの垂直方向のサイズは、値のサイズ(数値項目の値の整数部分)と同じです。 アップインジケータとダウンインジケータは、1つの値だけで表示されないことに注意してください。

背景色

カラーピッカーからウィジェット全体の背景色を選択します。Dはデフォルトの色を表します(フロントエンドのテーマによって異なります)。デフォルト値に戻すには、カラーピッカーの[デフォルトを使用] ボタンをクリックします。

複数の要素が同じスペースを占めることはできないことに注意してください。同じスペースに配置すると、エラーメッセージが表示されます。

13 マップ

概要

マップウィジェットでは、次のいずれかを表示できます。

- 単一の構成済みネットワークマップ
- マップナビゲーションツリーで設定されたネットワークマップの1つ (ツリーでマップ名をクリックした場合)

設定

設定するにはタイプとしてマップを選択します。

The screenshot shows the 'Add widget' dialog with the following settings:

- Type: Map
- Name: Local network
- Refresh interval: Default (15 minutes)
- Source type: Map
- * Map: Local network

すべてのウィジェットの**共通**パラメーターに加えて、次のオプションを設定できます。

ソースの種類

表示するソースを選択します：

マップ - ネットワークマップ
マップナビゲーション - 選択したマップナビゲーション内のマップの1つ

マップ

表示するマップを選択します。

このオプションはソースの種類としてマップが選択されている場合に使用できます。

フィルター

マップを表示するマップナビゲーションを選択します。

このオプションはソースの種類としてマップナビゲーションが選択されている場合に使用できます。

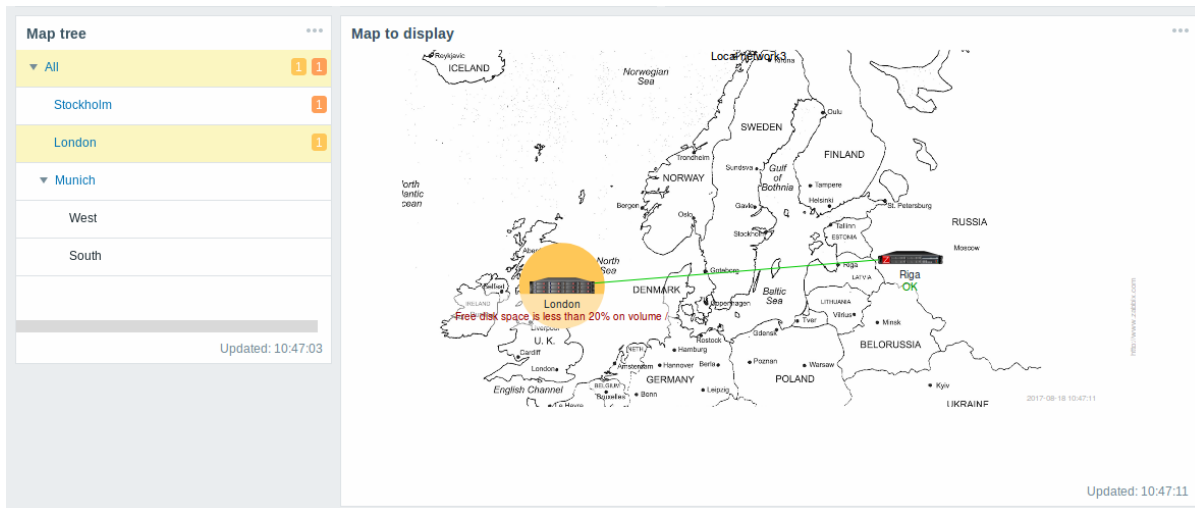
参照：[IE11の既知の問題](#)

14 マップナビゲーション

概要

このウィジェットを使用すると、既存のマップの階層を設定しながら各マップおよびマップグループの障害の統計を表示できます。

マップウィジェットをマップナビゲーションにリンクすると、さらに強力になります。マップナビゲーションでマップ名をクリックするとマップウィジェットにマップ全体が表示されます。



階層内の最上位マップを使用した統計では、すべてのサブマップの障害と障害の合計が表示されます。

設定

マップナビゲーションウィジェットを設定するには、タイプとしてマップナビゲーションを選択します。

Add widget

Type: Map navigation tree Show header

Name:

Refresh interval: Default (15 minutes)

Show unavailable maps:

Add Cancel

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

使用不可なマップの表示

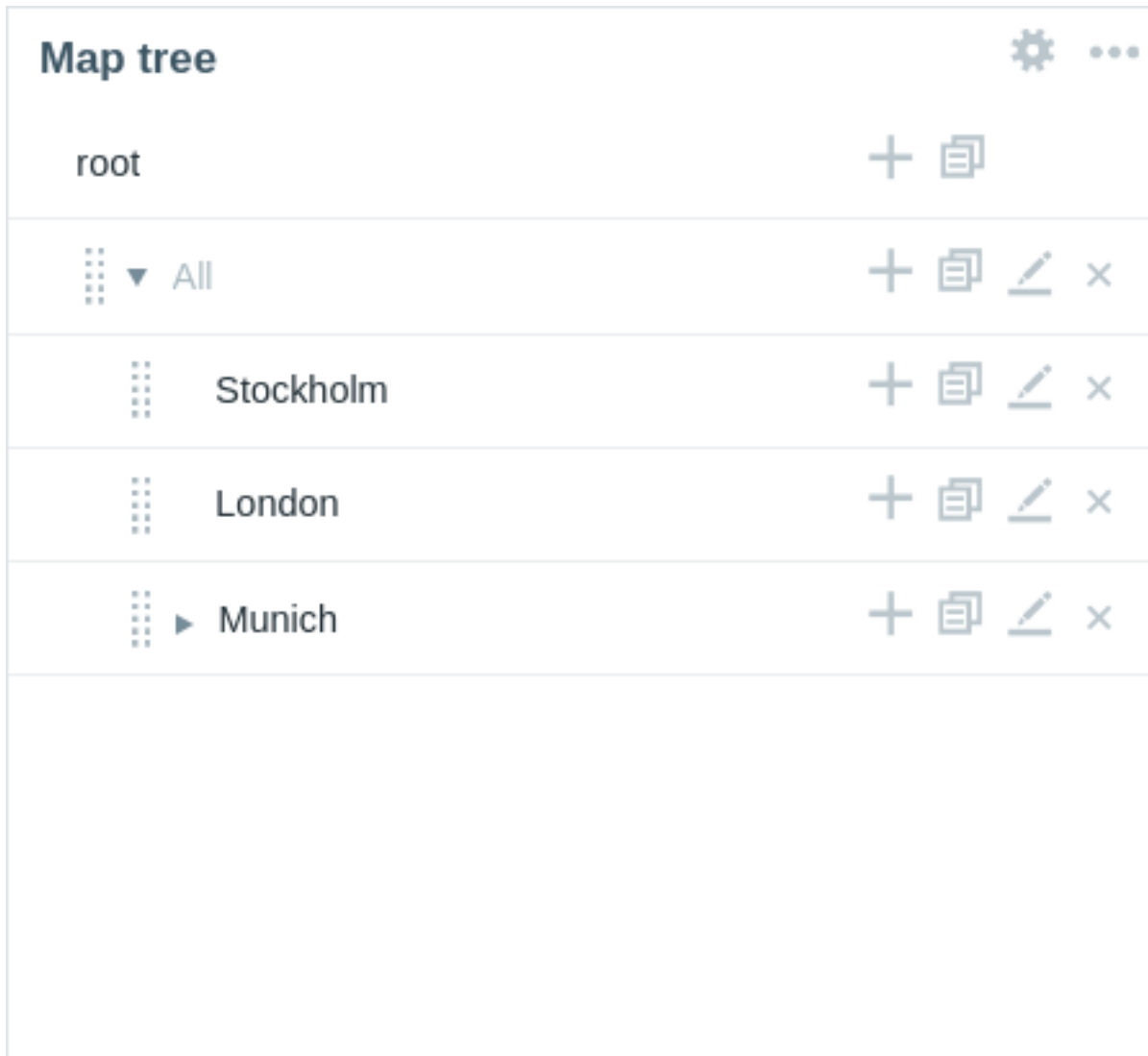
このチェックボックスをオンにすると、ユーザーが読み取り権限を持たないマップも表示されます。ナビゲーションツリーで使用できないマップは、灰色のアイコンで表示されます。

注：このチェックボックスをオンにすると、使用可能なサブマップが表示されます。親レベルマップが使用できない場合でも表示されます。オフの場合、使用できない親マップへの使用可能なサブマップはまったく表示されません。

障害数は、使用可能なマップと使用可能なマップ要素に基づいて計算されます。

Navigation tree elements are displayed in a list. You can:

- drag an element (including its child elements) to a new place in the list;
- expand or collapse an element to display or hide its child elements;
- add a child element (with or without a linked map) to an element;
- add multiple child elements (with linked maps) to an element;
- edit an element;
- remove an element (including its child elements).



Element configuration

To configure a navigation tree element, either add a new element or edit an existing element.

The 'Edit tree element' dialog box has a title bar with a close button (x). It contains the following fields and controls:

- Name:** A text input field containing 'London'.
- Linked map:** A dropdown menu showing 'London network' with a close button (x) on the right. A 'Select' button is positioned to the right of the dropdown.
- Add submaps:** A checkbox that is currently unchecked, followed by the text 'Add submaps'.
- Buttons:** Two buttons at the bottom right: a blue 'Add' button and a white 'Cancel' button with a blue border.

The following navigation tree element configuration parameters are available:

Name	Enter the navigation tree element name.
Linked map	Select the map to link to the navigation tree element. This field is auto-complete so starting to type the name of a map will offer a dropdown of matching maps.
Add submaps	Mark this checkbox to add the submaps of the linked map as child elements to the navigation tree element.

15 プレーンテキスト

概要

プレーンテキストウィジェットでは、最新のアイテムデータをプレーンテキストで表示できます。

設定

設定するにはタイプでプレーンテキストを選択します。

The screenshot shows the 'Add widget' dialog box with the following settings:

- Type: Plain text
- Show header:
- Name: Available memory
- Refresh interval: Default (1 minute)
- * Items: Zabbix server: Available memory (selected)
- Items location: Left (selected)
- * Show lines: 25
- Show text as HTML:
- Dynamic items:

Buttons: Add, Cancel

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

アイテム	アイテムを選択します
アイテム名の位置	ウィジェットに表示するアイテムの場所を選択します。
表示する行数	ウィジェットに表示される最新データ行の数を設定します。
HTMLとしてテキストを表示	テキストを HTML として表示するように設定します。
ダイナミックアイテム	選択したホストに応じて異なるデータを表示するように設定します。

16 障害中のホスト

概要

障害中のホストウィジェットでは、ホストの稼働状況に関する高レベルの情報を表示できます。

設定

設定するにはタイプとして障害中のホストを選択します。

Add widget



Type Show header

Name

Refresh interval

Host groups
type here to search

Exclude host groups

Hosts

Problem

Severity Not classified Warning High
 Information Average Disaster

Tags

[Add](#)

Show suppressed problems

Hide groups without problems

Problem display

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

パラメータ	詳細
ホストグループ	ウィジェットに表示するホストグループを入力します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択され、ホストデータがウィジェットに表示されます。ホストグループが入力されていない場合、すべてのホストグループが表示されます。
除外するホストグループ	ウィジェットから非表示にするホストグループを入力します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。これらのホストグループからのホストデータはウィジェットに表示されますが、たとえばホスト 001、002、003 はグループ A にあり、ホスト 002、003 もグループ B にあるとします。その上でグループ A を表示し、グループ B を除外した場合、ホスト 001 からのデータのみがダッシュボードに表示されます。
ホスト	ウィジェットに表示するホストを入力します。このフィールドはオートコンプリートであるため、ホストの名前を入力し始めると、一致するホスト名がドロップダウンで表示されます。ホスト名が入力されていない場合は、すべてのホストが表示されます。

パラメータ	詳細
障害	障害名で表示される障害中のホスト数を制限できます。ここに文字列を入力すると、名前に入力した文字列が含まれている障害中のホストのみが表示されます。マクロは展開されません。
深刻度 タグ	<p>ウィジェットに表示する障害の深刻度をマークします。</p> <p>障害タグを指定して、ウィジェットに表示される障害の数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。</p> <p>各条件で使用できる演算子はいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名と値を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める (文字列の部分一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (文字列の部分一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります：</p> <p>And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる</p>
メンテナンス中の障害を表示	チェックボックスをオンにすると、ホストのメンテナンスで抑制される (表示されない) 障害が表示されます
障害のないグループを隠す	ウィジェットで障害のないホストグループデータを非表示にするには、オプションをオンにします。
障害の表示	<p>障害数を表示方法を指定します：</p> <p>すべて - 完全な障害数が表示されます</p> <p>障害と未確認数を表示 - 未確認の障害数は合計障害数と分離して表示されます</p> <p>未確認のみ-未確認の障害数のみ表示されます。</p>

17 障害

概要

このウィジェットでは現在の障害を表示できます。このウィジェットの情報は監視データ → 障害に似ています。

設定

設定するにはタイプとして障害を選択します。

Add widget ✕

Type Show header

Name

Refresh interval

Show Recent problems Problems History

Host groups Select

Exclude host groups Select

Hosts Select

Problem

Severity Not classified Warning High
 Information Average Disaster

Tags And/Or Or

Contains

[Add](#)

Show tags None 1 2 3

Add Cancel

ウィジェットに表示される障害の数は、障害のステータス、障害の名前、深刻度、ホストグループ、ホスト、イベントタグ、確認応答のステータスなど、さまざまな方法で制限できます。

パラメータ	詳細
表示	障害のステータスでフィルタリングします： 最近の障害 - 未解決の障害と最近解決された障害が表示されます（デフォルト） 障害 - 未解決の障害が表示されます ヒストリ - すべてのヒストリイベントが表示されます
ホストグループ	ウィジェットに障害を表示するホストグループを指定します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると一致するグループ名がドロップダウンで表示されます。親ホストグループを指定するとネストされたすべてのホストグループが暗黙的に選択されます。ホストグループが入力されていない場合はすべてのホストグループの障害が表示されます。

パラメータ	詳細
除外するホストグループ	<p>ウィジェットから非表示にするホストグループを入力します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。</p> <p>これらのホストグループからのホストデータはウィジェットに表示されますが、たとえばホスト 001、002、003 はグループ A にあり、ホスト 002、003 もグループ B にあるとします。その上でグループ A を表示し、グループ B を除外した場合、ホスト 001 からのデータのみがダッシュボードに表示されます。</p>
ホスト	<p>ウィジェットに表示するホストを入力します。このフィールドはオートコンプリートであるため、ホストの名前を入力し始めると、一致するホスト名がドロップダウンで表示されます。ホスト名が入力されていない場合は、すべてのホストが表示されます。</p>
障害	<p>障害名で表示される障害中のホスト数を制限できます。ここに文字列を入力すると、名前に入力した文字列が含まれている障害中のホストのみが表示されます。マクロは展開されません。</p>
深刻度 タグ	<p>ウィジェットに表示する障害の深刻度をマークします。障害タグを指定してウィジェットに表示される障害の数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。</p> <p>各条件で使用できる演算子はいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名と値を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める (文字列の部分一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (文字列の部分一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります：</p> <p>And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる</p> <p>フィルタリングするとタグ表示優先度 (以下を参照) リストで上書きされない限り、ここで指定されたタグが障害とともに最初に表示されます。</p>
タグを表示	<p>表示するタグの数を選択します。</p> <p>なし - 監視データ → 障害にタグ列を表示しません。</p> <p>1 - タグ列に 1 つのタグが表示されます</p> <p>2 - タグ列には 2 つのタグが表示されます</p> <p>3 - タグ列には 3 つのタグが表示されます</p> <p>障害のすべてのタグを表示するには、3 ドットアイコンの上にマウスを置きます。</p>
タグ名	<p>タグ名の表示モードを選択します。</p> <p>すべて - タグの名前と値をすべて表示します</p> <p>短縮 - タグ名は 3 つの記号に短縮されます。タグ値は完全に表示されません</p> <p>なし - タグ値のみが表示されます。タグ名は表示されません</p>
タグの表示優先度	<p>障害のタグ表示優先度を、コンマ区切りリストとして入力します (例: Services, Applications, Application) タグ名のみを使用し、値は使用しないでください。このリストのタグは常に最初に表示され、アルファベット順による並べ替えを上書きします。</p>
運用データの表示	<p>運用データの表示方法を選択します：</p> <p>なし - 運用データは表示されません</p> <p>別々に - 運用データは別の列に表示します</p> <p>障害名と一緒に - 運用データを括弧でかこみ、問題名に運用データを追加します</p>

パラメータ	詳細
メンテナンス中の障害を表示	チェックボックスをオンにすると、ホストのメンテナンスのために抑制される（表示されない）障害が表示されます。
未確認のみ表示	未確認の障害のみを表示するには、チェックボックスをオンにします。
次の項目によるソート	エントリを次の順序で並べ替えます： 時間（降順または昇順） 深刻度（降順または昇順） 障害（降順または昇順） ホスト（降順または昇順）。
タイムラインを表示	チェックボックスをオンにすると視覚的なタイムラインが表示されます。
表示する行数	表示する障害の行数を指定します。

18 深刻度ごとの障害数

概要

このウィジェットでは深刻度別に障害を表示できます。ウィジェットに表示されるホストとトリガーを制限し、障害数の表示方法を定義できます。

設定

設定するにはタイプとして深刻度ごとの障害数を選択します。

Add widget ✕

Type Problems by severity Show header

Name

Refresh interval Default (1 minute)

Host groups Select

Exclude host groups Select

Hosts Select

Problem

Severity Not classified Warning High
 Information Average Disaster

Tags And/Or Or

Contains ▼

[Add](#)

Show Host groups Totals

Layout Horizontal Vertical

Show optional data None Count only With problem name

Add Cancel

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

パラメータ	詳細
除外するホストグループ	<p>ウィジェットから非表示にするホストグループを入力します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。</p> <p>これらのホストグループからのホストデータはウィジェットに表示されますが、たとえばホスト 001、002、003 はグループ A にあり、ホスト 002、003 もグループ B にあるとします。その上でグループ A を表示し、グループ B を除外した場合、ホスト 001 からのデータのみがダッシュボードに表示されます。</p>
ホスト	<p>ウィジェットに表示するホストを入力します。このフィールドはオートコンプリートであるため、ホストの名前を入力し始めると、一致するホスト名がドロップダウンで表示されます。</p> <p>ホスト名が入力されていない場合は、すべてのホストが表示されます。</p>
障害	<p>障害名で表示される障害中のホスト数を制限できます。ここに文字列を入力すると、名前に入力した文字列が含まれている障害中のホストのみが表示されます。マクロは展開されません。</p>
深刻度	<p>ウィジェットに表示する障害の深刻度をマークします。</p>

パラメータ	詳細
タグ	<p>障害タグを指定してウィジェットに表示される障害の数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。</p> <p>各条件で使用できる演算子はいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名と値を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める (文字列の部分一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (文字列の部分一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります :</p> <p>And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる</p>
表示	<p>表示オプションを選択します。</p> <p>ホストグループ - ホストグループごとに障害を表示します</p> <p>トータル - 選択したすべてのホストグループの障害の合計を、障害の深刻度に対応する色付きのブロックで表示します。</p>
レイアウト	<p>レイアウトオプションを選択します。</p> <p>横 - 色付きの合計ブロックが横方向に表示されます</p> <p>縦 - 色付きの合計ブロックが縦方向に表示されます</p> <p>このフィールドは表示オプションがトータルの場合編集できます。</p>
運用データの表示	<p>運用データの表示方法を選択します :</p> <p>なし - 運用データは表示されません</p> <p>別々に - 運用データは別の列に表示します</p> <p>障害名と一緒に - 運用データを括弧でかこみ、問題名に運用データを追加します</p>
メンテナンス中の障害を表示	<p>チェックボックスをオンにすると、ホストのメンテナンスのために抑制される (表示されない) 障害が表示されます。</p>
障害のないグループを隠す	<p>ウィジェットで障害のないホストグループのデータを非表示にするには障害のないグループを隠すチェックボックスをオンにします。</p>
障害の表示	<p>障害数を次のように表示します :</p> <p>すべて - 完全な障害数が表示されます</p> <p>障害数と未確認数を表示 - 未確認の障害数は、合計障害数から分離して表示されます</p> <p>未確認のみ - 未確認の障害数のみが表示されます。</p>
タイムラインを表示	<p>チェックボックスをオンにすると視覚的なタイムラインが表示されます。</p>

19 SLA レポート

概要

このウィジェットはSLA レポートを表示するのに役立ちます。機能的にはサービス -> SLA レポートセクションに似ています。

設定

設定するにはタイプとして SLA レポートを選択します。

Edit widget ✕

Type Show header

Name

Refresh interval

* SLA

Service

Show periods

From

To

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

SLA	レポートの SLA を選択します
サービス	レポートのサービスを選択します
表示期間	ウィジェットに表示される期間を設定します (デフォルトでは 20、最大 100)
開始	レポートの開始日を選択します。 相対的な日付がサポートされています : now, now/d, now/w-1w など。サポートされている日付修飾子 : d, w, M, y.
終了	レポートの終了日を選択します。 相対的な日付がサポートされています : now, now/d, now/w-1w など。サポートされている日付修飾子 : d, w, M, y.

20 システム情報

概要

このウィジェットはレポート → [システム情報](#)と同じ情報を表示しますが、単一のダッシュボードウィジェットは一度にシステム統計または HA ノードのいずれか (両方ではない) のみを表示できます。

設定

設定するにはタイプとしてシステム情報を選択します。

Add widget ×

Type Show header

Name

Refresh interval

Show

すべての構成パラメーターは共通パラメーターです。

21 トリガーの概要

概要

トリガー概要ウィジェットでは、ホストグループのトリガー状態を表示できます。

- トリガーの状態は色付きのブロックで表示されます。(障害のトリガー色は障害の申告度の色によって異なり**障害の更新**画面で調整できます) 最近のトリガーの変更 (過去 2 分以内) は点滅するブロックとして表示されることに注意してください。
- 青い上向き矢印と下向き矢印は依存関係があるトリガーを示します。マウスオーバーすると依存関係の詳細が表示されます。
- チェックボックスアイコンは確認済みの障害を示します。このアイコンを表示するにはトリガーのすべての障害または解決された障害を確認する必要があります。

トリガーブロックをクリックすると、トリガーの障害イベント、障害確認画面、トリガー構成、トリガー URL、またはシンプルグラフ/最新の値のリストへのリンクが表示されます。

デフォルトでは 50 レコードが表示されます。(管理 → 一般設定 → *表示設定の概要テーブル内の列と行の最大数 * オプションで設定可能です) 設定値よりも多くのレコードが存在する場合は、テーブルの下部にメッセージが表示され、より具体的なフィルタリング基準を提供するように求められます。改ページ機能はありません。

設定

設定するにはタイプとしてトリガーの概要を選択します。

Add widget ✕

Type Trigger overview Show header

Name

Refresh interval Default (1 minute)

Show Recent problems Problems Any

Host groups Select

Hosts Select

Tags And/Or Or

Contains value Remove

[Add](#)

Show suppressed problems

Hosts location Left Top

Add Cancel

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

表示	<p>障害のステータスでフィルタリング： 最近の障害 - 未解決の障害と最近解決された障害問題が表示されます（デフォルト） 障害 - 未解決の障害が表示されます</p>
ホストグループ	<p>すべて - すべての履歴イベントが表示されます ホストグループを選択します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループがドロップダウンで表示されます。</p>
ホスト	<p>ホストを選択します。このフィールドはオートコンプリートであるため、ホスト名を入力し始めると、一致するホスト名がドロップダウンで表示されます。スクロールして選択します。 'x' をクリックして選択したものを削除できます。</p>

タグ

タグを指定して、ウィジェットに表示されるアイテムとトリガーデータの数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。各条件で使用できる演算子はいくつかあります。

Exists - 指定されたタグ名を含める

Equals - 指定されたタグ名と値を含める（大文字と小文字を区別）

Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める（文字列の部分一致、大文字と小文字を区別しない）

Does not exist - 指定されたタグ名を除外する

Does not equal - 指定されたタグ名と値を除外する（大文字と小文字を区別）

Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する（文字列の部分一致、大文字と小文字を区別しない）

条件には 2 つの計算タイプがあります：

And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる

チェックボックスをオンにすると、ホストのメンテナンスで抑制される（表示されない）障害が表示されます。

ホストの場所を左または上で選択します。

メンテナンス中の障害を表示

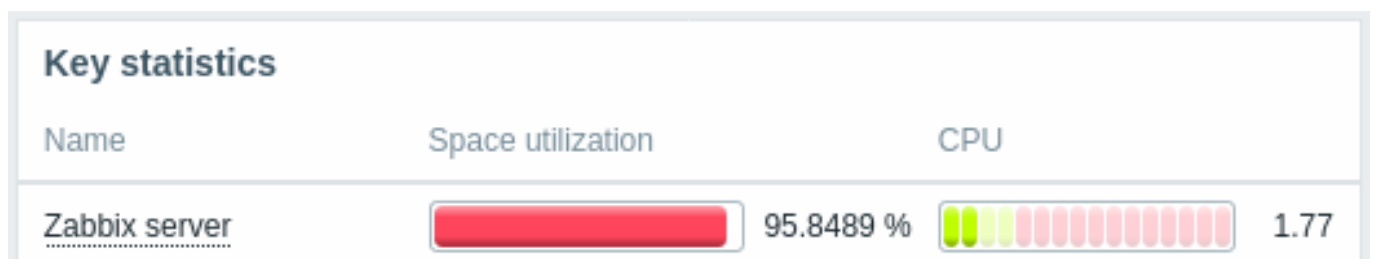
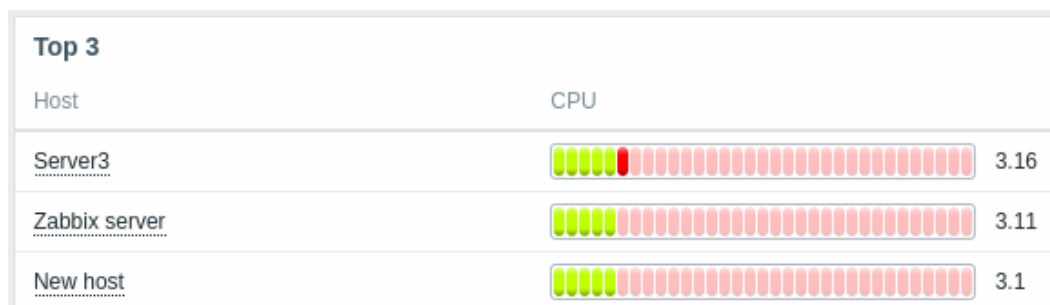
ホストの位置

21 上位ホスト

概要

このウィジェットは、データの状況を表示するためのカスタムテーブルを作成する方法を提供し、キャパシティプランニングに役立つ Top N のようなレポートとプログレスバーレポートを表示できるようにします。

表示できるホストの最大数は 100 です。



設定

設定するにはタイプとして上位ホストを選択します。

Add widget ✕

Type Show header

Name

Refresh interval

Host groups

Hosts

Host tags And/Or Or

[Add](#)

* Columns

Order Top N Bottom N

* Order column

* Host count

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

ホストグループ
 ホスト
 ホストのタグ

データを表示するホストグループ
 データを表示するホスト
 タグを指定して、ウィジェットに表示されるホストの数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。

各条件で使用できる演算子はいくつかあります。

Exists - 指定されたタグ名を含める

Equals - 指定されたタグ名と値を含める（大文字と小文字を区別）

Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める（文字列の部分一致、大文字と小文字を区別しない）

Does not exist - 指定されたタグ名を除外する

Does not equal - 指定されたタグ名と値を除外する（大文字と小文字を区別）

Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する（文字列の部分一致、大文字と小文字を区別しない）

条件には 2 つの計算タイプがあります：

And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる

表示するデータ列を追加します。

列の順序によって、左から右への表示が決まります。

列名の前のハンドルを上下にドラッグすると、列の順序を変更できます。

行の順序を指定します。

上位 **N** - 順序用列集計値の降順

下位 **N** - 順序用列集計値の昇順

列

表示順序

順序用列	定義された列リストから列を指定して、上位 N または下位 N の順序付けに使用します。
ホスト数	表示するホストの行数 (1 - 100)

列の設定

New column ✕

Name

Data

* Item

Time shift

Aggregation function

Display

History data

Base color

Thresholds

Threshold	Action
Add	

共通の列パラメーター :

名前	列の名前
取得データ	列に表示するデータ : アイテムの値 - 指定されたアイテムの値 ホスト名 - アイテム値で指定されたアイテムのホスト名 テキスト - 静的テキスト文字列
基本色	カラムの背景色。アイテム値データがバー/インジケータとして表示される場合は色を塗りつぶします。 アイテム値データの場合、アイテム値が指定された「しきい値」の1つを超えると、デフォルトの色をカスタム色で上書きできます。

アイテム値列の特定パラメーター :

アイテム	アイテムを選択します。 Zabbix6.0.4以降、数値以外のアイテム選択がサポートされています。
タイムシフト	必要に応じてタイムシフトを指定します。 このフィールドでは タイムサフィックス を使用できます。負の値が許可されます。

アグリゲーション関数	<p>使用する集計関数を指定します。</p> <p>最小 - 最小値を表示します</p> <p>最大 - 最大値を表示します</p> <p>平均 - 平均値を表示します</p> <p>合計 - 値の合計を表示します</p> <p>個数 - 値の数を表示します</p> <p>第 1 - 最初の値を表示します</p> <p>最新値 - 最後の値を表示します</p> <p>なし - すべての値を表示します (集計なし)</p> <p>集計を使用すると、すべての値ではなく、選択した間隔 (5 分、1 時間、1 日) の集計値を表示できます。</p> <p>注: この設定がなしでない場合、この列に表示できるのは数値項目のみです。</p>
アグリゲーション間隔	<p>値を集計する間隔を指定します。このフィールドでは時間サフィックスを使用できます。接尾辞のない数値は秒と見なされません。</p> <p>アグリゲーション関数がなしの場合、このフィールドは表示されません。</p>
表示	<p>値の表示方法を定義します。</p> <p>そのまま - 通常のテキストとして表示</p> <p>棒 - 実線の色で塗りつぶされたバーとして</p> <p>インジケーター - セグメント化された色で塗りつぶされたバーとして</p> <p>設定がそのままでない場合、この列に表示できるのは数値項目のみであることを注意してください。</p>
履歴データ	<p>履歴またはトレンドからデータを取得:</p> <p>自動 - 自動選択</p> <p>履歴 - 履歴データを取得</p> <p>トレンド - トレンドデータを取得</p> <p>この設定は数値データにのみ適用されます。非数値データは常に履歴から取得されます。</p>
最小値 最大値 閾値	<p>バー/インジケーターの最小値。</p> <p>バー/インジケーターの最大値。</p> <p>背景/塗りつぶしの色を変更する必要がある場合のしきい値を指定します。保存すると、リストは昇順で並べ替えられます。しきい値が使用されている場合、この列に表示できるのは数値項目のみであることを注意してください。</p>

テキスト列の特定パラメーター:

テキスト 表示する文字列を入力します。ホストとインベントリ**マクロ**が含まれる場合があります。

22 URL

概要

このウィジェットは、指定された URL から取得されたコンテンツを表示します。

設定

設定するにはタイプとして URL を選択します。

Add widget ✕

Type Show header

Name

Refresh interval

* URL

Dynamic item

すべてのウィジェットの共通パラメーターに加えて、次のオプションを設定できます。

URL	表示する URL を入力します。 Zabbix4.4.8 以降、相対パスが許可されています。 {HOST.*} マクロがサポートされています。
ダイナミックアイテム	選択したホストに応じて異なる URL コンテンツを表示するように設定します。 これは URL で {HOST.*} マクロが使用されている場合に機能します。

Attention:

Zabbix フロントエンドが HTTPS 経由でアクセスされる場合、ブラウザはウィジェットに含まれる HTTP ページをロードしない可能性があります。

23 Web 監視

概要

このウィジェットはアクティブな Web 監視シナリオのステータス概要を表示します。

設定

Add widget ✕

Type Show header

Name

Refresh interval

Host groups

Exclude host groups

Hosts

Tags And/Or Or

[Add](#)

Show hosts in maintenance

Note:

ユーザーが特定のウィジェット要素にアクセスする権限を持っていない場合、その要素の名前はウィジェットの構成中にアクセス不能として表示されます。これにより要素の実際の名前の代わりに、アクセスできないアイテム、アクセスできないホスト、アクセスできないグループ、アクセスできないマップ、およびアクセスできないグラフが表示されます。

すべてのウィジェットの**共通**パラメーターに加えて、次のオプションを設定できます。

パラメーター	詳細
ホストグループ	ウィジェットに表示するホストグループを入力します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択され、ホストデータがウィジェットに表示されます。ホストグループが入力されていない場合、すべてのホストグループが表示されます。
除外するホストグループ	ウィジェットから非表示にするホストグループを入力します。このフィールドはオートコンプリートであるため、グループ名を入力し始めると、一致するグループ名がドロップダウンで表示されます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。これらのホストグループからのホストデータはウィジェットに表示されますが、たとえばホスト 001、002、003 はグループ A にあり、ホスト 002、003 もグループ B にあるとします。その上でグループ A を表示し、グループ B を除外した場合、ホスト 001 からのデータのみがダッシュボードに表示されます。
ホスト	ウィジェットに表示するホストを入力します。このフィールドはオートコンプリートであるため、ホストの名前を入力し始めると、一致するホスト名がドロップダウンで表示されます。ホスト名が入力されていない場合は、すべてのホストが表示されません。

パラメーター	詳細
タグ	<p>タグを指定して、ウィジェットに表示される Web シナリオの数を制限します。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。</p> <p>各条件で使用できる演算子はいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名と値を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める (文字列の部分一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (文字列の部分一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります :</p> <p>And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる</p>
メンテナンス中のホストを表示	メンテナンス中のホストを統計に含めます。

Web monitoring widget

Once you have completed the configuration, you might like to see the widget with the data it displays. To do it, go to Monitoring → Dashboards, click on the name of a dashboard where you created the widget.

In this example, you can see the widget named "Zabbix frontend" displaying the status of the web monitoring for three host groups: "Internal network," "Linux servers," and "Web servers."

Zabbix frontend

Host group ▲	Ok	Failed	Unknown
Internal network	1		
Linux servers		1	
Web servers			1

A web monitoring widget displays the following information:

- a name of a widget; below it, there are four columns:
 - Host group - displays a list of host groups that contain hosts having web scenarios configured;
 - Ok - displays a number of web scenarios (in green color) when two conditions are observed:
 - * Zabbix has collected the latest data for a web scenario(s);
 - * all steps that were configured in a **web scenario** are in "Ok" Status.
 - Failed - displays a number of web scenarios (in red color), which have some failed steps:
 - * click on the host name, and it will open a new window; the Status column provides detailed information (in red color) on the step where Zabbix failed to collect the data; and also,
 - * gives a hint for the parameter that has to be corrected in the **configuration form**.

Host	Name ▲	Number of steps	Last check	Status
Internal documentation	Internal Wiki	2	38s	Step "Configuration page" [2 of 2] failed: required pattern "winter" was not found on http://localhost/index.php

- Unknown - displays a number of web scenarios (in grey color) for which Zabbix has neither collected data, nor has an information about the failed steps.

Host	Name	Number of steps	Last check	Status	Tags
Zabbix site	Zabbix site	1			

Displaying 1 of 1 found

Viewing the status and data

Clickable links in the widget allow to easily navigate and quickly acquire a full information on each web scenario. Thus, to view:

- the **Status** of a web scenario, click on the name of a host group.
- more detailed statistics, click on the scenario name. In this example it is "Zabbix frontend".
- the details in the case of Failed status, click on a host group name; in the window that opens, click on a web scenario name in the Name column; it will open more detailed information on the configured steps for which Zabbix failed to collect the data.

Step	Speed	Response time	Response code	Status
First page	95.94 KBps	256.75ms	200	OK
Configuration page	40.46 KBps	33.5ms	200	Error: required pattern "winter" was not found on http://localhost/index.php
TOTAL		290.25ms		Error: required pattern "winter" was not found on http://localhost/index.php

Now, you can return to the **web scenario configuration form** and correct your settings.

To view the details in the case of Unknown status, you can repeat the same steps as explained for Failed.

Attention:

At the first monitoring instance, a web scenario is always displayed in Unknown state, which is switched to Failed or Ok state right after the first check. In the case when a host is monitored by the proxy, the status change occurs in accordance with the data collection frequency configured on the proxy.

2 障害

概要

監視データ → 障害で現在発生している障害を確認できます。障害とはステータスが" 障害" となったトリガーです。

Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
10:39:56	Information		PROBLEM		Zabbix server	Zabbix server: Version has changed (new version: 6.4.0alpha1)	2m 52s	No		class: software component: system scope: notice ***
2022-07-29 17:16:51	Average		PROBLEM		Zabbix server	↑ Interface wlp3s0: Link down	2d 17h 25m	No		class: os component: network interface: wlp3s0 ***
2022-06-09 13:11:16	Warning		PROBLEM		Zabbix server	! Disk space is critically low (used > 90%)	1M 22d 21h	No		class: os component: storage filesystem: / ***

0 selected | Mass update | Displaying 3 of 3 found

列

詳細

時間

障害の発生時刻が表示されます。

深刻度



障害の深刻度が表示されます。
 障害の深刻度は障害トリガーの深刻度に基づいていますが、イベントが発生した後、障害の更新画面を使用して更新できます。障害の深刻度の色は障害時のセルの背景色として使用されます。








復旧時刻

障害解決時間が表示されます。

ステータス

障害のステータスが表示されます：
 障害 - 未解決の障害
 解決済み - 最近解決された障害。フィルタを使用して最近解決された障害を非表示にできます。
 新しい障害と最近解決された障害は 2 分間点減します。解決した障害は合計 5 分間表示されます。これらの表示は両方とも、管理 → 一般設定 → *イベントのステータスで設定できます。

<p>列</p> <p>情報</p> <p>ホスト 障害</p> <p>運用データ</p> <p>継続期間</p> <p>確認済み</p>	<p>詳細</p> <p>グローバル相関によって障害が解決された場合、または障害を更新するときに手動で解決された場合は、緑色の情報アイコンが表示されます。アイコンの上にマウスを置くと詳細が表示されます：</p>  <p>
</p> <p>> 抑制された障害が表示されている場合は、次のアイコンが表示されます（フィルターの抑制された問題を表示オプションを参照）。アイコンの上にマウスを置くと詳細が表示されます：</p> <p>障害が発生したホストが表示されます。 障害名が表示されます。 障害名は障害トリガーの名前に基づいています。 トリガー名のマクロは、障害が発生したときに解決され、解決された値は更新されなくなります。
 注意: 障害名に運用データを追加して最新の項目値を表示することができます。 障害名をクリックするとイベントメニューが表示されます。
 障害名の後にある  アイコンにカーソルを合わせると、トリガーの説明が表示されます（障害がある場合） （トリガーの説明の {ITEM.VALUE} および {ITEM.LASTVALUE} マクロの解決された値は 20 文字に切り捨てられることに注意してください。値全体を表示するには、回避策としてこれらのマクロで [マクロ関数] を使用します（例： {{ITEM.VALUE}.regsub("(.*)", \1)}, {{ITEM.LASTVALUE}.regsub("(.*)", \1)}) 運用データが表示され、最新のアイテム値が含まれます。 運用データは、トリガーレベルで構成されている場合、テキストマクロとアイテム値マクロの組み合わせにすることができます。トリガーレベルで運用データが構成されていない場合は、式のすべての項目の最新値が表示されます。 この列は、フィルターの [運用データの表示] で別々にが選択されている場合にのみ表示されます。 障害の期間が表示されます。 関連項目：障害の期間 障害の確認ステータスが表示されます。 はい - 障害が確認されたことを示す緑色のテキスト。障害のすべてのイベントが確認された場合、障害は確認されたと見なされます。 いいえ - 未確認のイベントを示す赤いリンク。 リンクをクリックすると障害へのコメントや確認など、問題に対してさまざまなアクションを実行できる障害の更新に移動します。</p>
---	---

列	詳細
アクション	<p>障害に関するアクティビティの履歴は、記号アイコンを使用して表示されます：</p> <ul style="list-style-type: none">  - コメントが作成されました。コメントの数も表示されます。  - 障害の深刻度が増加しました（例：情報 → 警告）  - 障害の深刻度が低下しました（例：警告 → 情報）  - 障害の深刻度は変更されましたが、元のレベルに戻りました（例：警告 → 情報 → 警告）  - アクションが実行されました。アクションの数も表示されます。  - アクションが実行されました。少なくとも 1 つは進行中です。アクションの数も表示されます。  - アクションが実行されました。少なくとも 1 つは失敗しました。アクションの数も表示されます。 <p>アイコンの上にマウスを置くと、アクティビティの詳細を示すポップアップが表示されます。実行されたアクションのポップアップで使用するアイコンの詳細については詳細の表示を参照してください。</p> <p>(存在する場合)タグが表示されます。</p> <p>さらに、外部チケットシステムからのタグも表示される場合があります (webhooksを設定する際のプロセスタグオプションを参照)</p>
タグ	

障害の運用データ

現在の障害の運用データ、つまり障害発生時のアイテム値ではなく最新のアイテム値を表示することができます。

運用データの表示は監視データ → 障害のフィルターまたはそれぞれのダッシュボードウィジェットの設定で構成できます。オプションは 3 つあります。

- なし - 運用データは表示されません
- 別々に - 運用データは別の列に表示されます

Time	Severity	Recovery time	Status	Info	Host	Problem	Operational data	Duration
09:28:35	Average		PROBLEM		Zabbix server	Zabbix discoverer processes more than 75% busy	Current value: 100 %	3h 32m 8s

- 障害名と一緒に - 運用データは障害名の括弧内に追加されます。トリガー構成で障害データフィールドが空でない場合にのみ運用データが障害名に追加されます。

Time	Severity	Recovery time	Status	Info	Host	Problem	Operational data	Duration
09:28:35	Average		PROBLEM		Zabbix server	Zabbix discoverer processes more than 75% busy	Current value: 100 %	3h 29m 34s

運用データの内容は [運用データ] フィールドの各トリガーで設定できます。このフィールドは {ITEM.LASTVALUE<1-9>} マクロを含む任意の文字列を受け入れます。

このフィールドの {ITEM.LASTVALUE<1-9>} は、トリガー式のアイテムの最新の値に常に解決されます。このフィールドの {ITEM.VALUE<1-9>} は、トリガーステータスの変更（つまり、障害への変更、OK への変更、ユーザーによる手動でのクローズ、または相関によるクローズ）の時点でアイテム値に解決されます。

負の障害期間

いくつかの状況では障害の解決時間が障害の作成時間よりも早いなど、障害の期間がマイナスになる可能性があります。

例： - If some host is monitored by proxy and a network error happens, leading to no data received from the proxy for a while, the nodata(/host/key) trigger will be fired by the server. When the connection is restored, the server will receive item data from the proxy having a time from the past. Then, the nodata(/host/key) problem will be resolved and it will have a negative problem duration; 一部のホストがプロキシによって監視されていて、ネットワークエラーが発生し、しばらくの間プロキシからデータを受信されない場合、サーバーによって nodata(/host/key) トリガーが起動されます。接続が回復すると、サーバーは過去の時間を持つプロキシからアイテムデータを受信します。次に nodata(/host/key) 障害が解決され、障害の期間がマイナスになります。 - 障害イベント

を解決するアイテムデータが Zabbix 送信者によって送信され、障害作成時刻より前のタイムスタンプが含まれている場合、負の障害期間が表示されます。

Note:

負の障害期間は、特定のトリガーのSLA 計算または稼働レポートに影響を与えません。障害期間を短縮も拡大もしません。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 一括更新 - 障害の更新画面に移動して、選択した障害を更新します

このオプションを使用するには、それぞれの障害の前あるにチェックボックスをオンにしてから一括更新ボタンをクリックします。

ボタン

右側のボタンには、次のオプションがあります。

Export to CSV

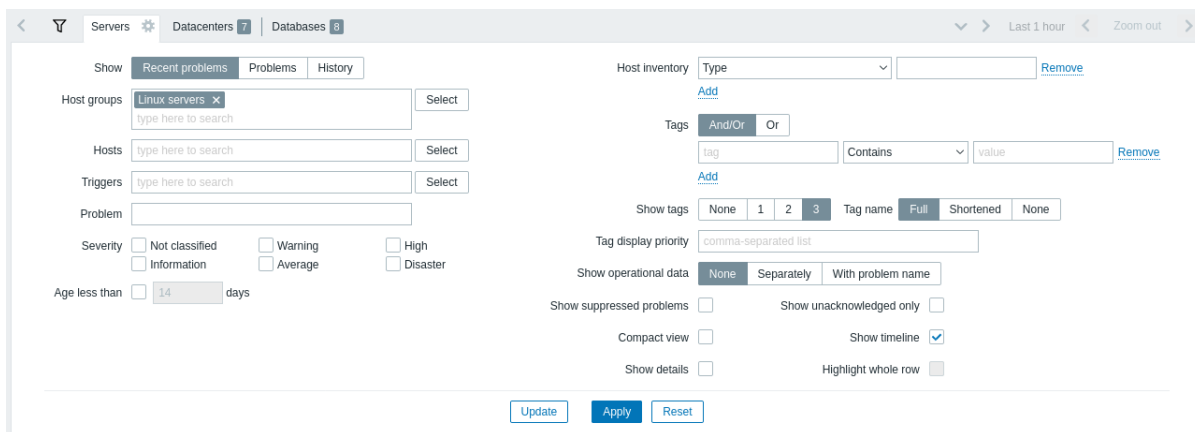
すべてのページコンテンツを CSV ファイルにエクスポートします。

すべてのセクション共通の表示モードボタンについては監視データページで説明されています。

フィルタの使用

フィルタを使用して任意の障害のみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタはテーブルの上にあります。お気に入りのフィルター設定をタブとして保存し **フィルターの上のタブ** をクリックしてすばやくアクセスできます。



パラメータ	詳細
表示	障害のステータスでフィルタリングします： 最近の障害 - 未解決の障害と最近解決された障害が表示されます（デフォルト） 障害 - 未解決の障害が表示されます
ホストグループ	履歴 - すべての履歴イベントが表示されます 1つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
ホスト	一つ以上のホスト名でフィルタリングします
トリガー	一つ以上のトリガー名でフィルタリングします
障害	障害名でフィルタリングします。
深刻度	トリガー（障害）の深刻度でフィルターします。
表示期間	指定された表示期間でフィルタリングします。
ホストインベントリ	ホストインベントリの種類と値でフィルタリングします。

パラメータ	詳細
タグ	<p>イベントタグの名前と値でフィルタリングします。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。条件ごとに使用できる演算子がいくつかあります。</p> <p>Exists - 指定されたタグ名を含める Equals - 指定されたタグ名と値を含める (大文字と小文字を区別) Contains - タグ値に入力された文字列が含まれる場合に指定されたタグ名を含める (文字列の部分一致、大文字と小文字を区別しない) Does not exist - 指定されたタグ名を除外する Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別) Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (文字列の部分一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります： And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or - 1 つの条件が満たされる</p> <p>フィルタリングするとタグ表示優先度 (以下を参照) リストで上書きされない限り、ここで指定されたタグが障害とともに最初に表示されます。</p>
タグを表示	<p>表示するタグの数を選択します。</p> <p>なし - 監視データ → 障害にタグ列を表示しません。</p> <p>1 - タグ列に 1 つのタグが表示されます 2 - タグ列には 2 つのタグが表示されます 3 - タグ列には 3 つのタグが表示されます</p> <p>障害のすべてのタグを表示するには、3 ドットアイコンの上にマウスを置きます。</p>
タグ名	<p>タグ名の表示モードを選択します。</p> <p>すべて - タグの名前と値はすべて表示されます 短縮 - タグ名は 3 つの記号に短縮されます。タグ値は完全に表示されません</p> <p>なし - タグ値のみが表示されます。タグ名は表示しません</p>
タグ表示優先度	<p>問題のタグ表示優先度を、タグのコンマ区切りリストとして入力します (例: Services、Applications、Application)。タグ名のみを使用し、値は使用しないでください。このリストのタグは常に最初に表示され、アルファベットによる自然順序を上書きします。</p>
運用データの表示	<p>[運用データ] を表示するモードを選択します (#operational_data_of_problems) :</p> <p>なし - 運用データは表示されません 別々に - 運用データは別の列に表示されます 障害名と一緒に - 括弧を使用して問題名に運用データを追加します</p>
メンテナンス中の障害を表示	<p>チェックボックスをオンにすると、ホストのメンテナンスに抑制される (表示されない) 障害が表示されます。</p>
コンパクト表示	<p>チェックボックスをオンにすると、コンパクト表示が有効になります。</p>
詳細を表示	<p>チェックボックスをオンにすると障害の根本的なトリガー式が表示されます。コンパクト表示チェックボックスがマークされている場合は無効になります。</p>
未確認のみ表示	<p>未確認の障害のみを表示するには、チェックボックスをオンにします。</p>
タイムラインを表示	<p>チェックボックスをオンにすると、視覚的なタイムラインとグループ化が表示されます。コンパクト表示チェックボックスがマークされている場合は無効になります。</p>
行を強調表示	<p>チェックボックスをオンにすると未解決の障害の全行が強調表示されます。障害の深刻度の色は強調表示に使用されます。</p> <p>標準の青と暗いテーマで [コンパクト表示] チェックボックスがオンになっている場合にのみ有効になります。行全体を強調表示は高コントラストのテーマでは使用できません。</p>

頻繁に使用するフィルターパラメーターのセットをタブに保存できます。

新しいフィルターパラメーターのセットを保存するには、メインタブを開きフィルター設定を構成してから保存するボタンを押し、ポップアップウィンドウでフィルタープロパティを定義します。

Filter properties

* Name

Show number of records

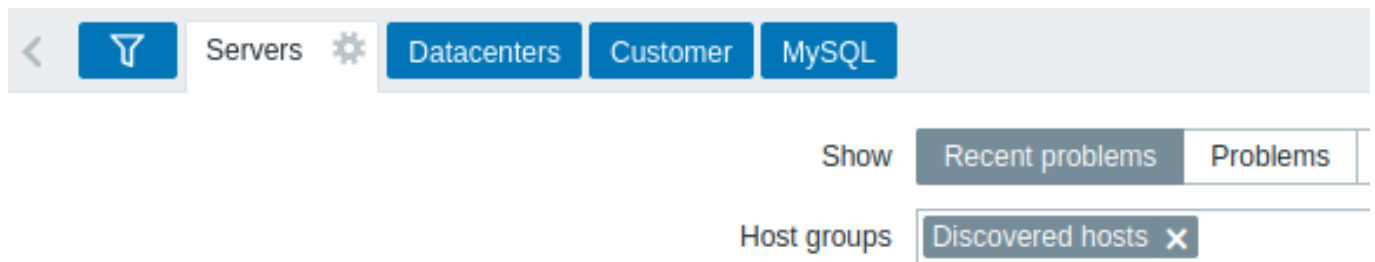
Set custom time period

From

To

パラメータ	詳細
名前	タブリストに表示するフィルターの名前
レコード数の表示	タブ名の横に障害数を表示するか
時間範囲のカスタマイズ	このフィルターセットに特定のデフォルト期間を設定する場合にチェックします。設定されている場合、フィルター設定を更新することによってのみ、このタブの期間を変更できます。カスタム期間のないタブの場合、右上隅にある時間選択ボタンを押すことで時間範囲を変更できます (ボタン名は選択した時間間隔によって異なります: 今週、過去 30 分、昨日など)。
開始/終了	このオプションは、[監視データ] → [障害] のフィルターでのみ使用できます。 期間セレクター 絶対 (Y-m-d H:i:s) または相対時間構文 (now-1d) で開始および終了を設定します。時間範囲のカスタマイズがチェックされている場合に使用できます。

既存フィルターのフィルタープロパティを編集するには、アクティブなタブ名の横にある歯車の記号を押します。







注:

- フィルタを非表示にするには現在のタブの名前を押します。フィルターを開くにはアクティブなタブ名をもう一度押します。
- フィルタタブは、ドラッグアンドドロップで移動できます。
- キーボードナビゲーションがサポートされています。矢印を使用してタブを切り替え、Enter を押して開きます。
- 右上隅にある下向き矢印アイコンを押すと、保存されているフィルタータブの完全なリストがドロップダウンメニューとして開きます。

Note:

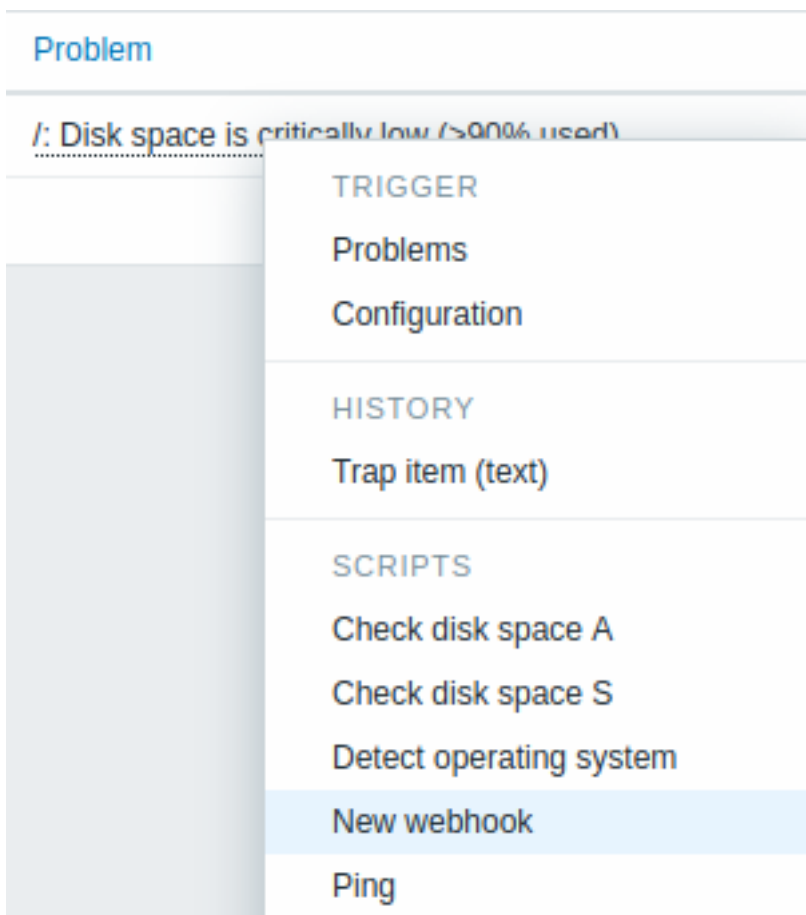
フィルタを共有するには、アクティブなフィルタの URL をコピーして他の人に送信します。この URL を開くと他のユーザーはこのパラメーターのセットを永続フィルターとして Zabbix アカウントに保存できるようになります。関連項目: [ページパラメーター](#)

フィルターのボタン

	指定されたフィルタリングを（保存せずに）適用します
	現在のフィルターをリセットし、現在のタブに保存されたパラメーターに戻ります。メインタブの場合はフィルターがクリアされます。
	現在のフィルタリングパラメーターを新しいタブに保存します。メインタブでのみ使用できます。
	タブに保存されているパラメータを現在指定されているパラメータに置き換えます。メインタブでは使用できません。

イベントメニュー

障害名をクリックするとイベントメニューが表示されます。



イベントメニューでは次のことができます。

- トリガーの障害をフィルタリングします
- トリガー設定にアクセスします
- 基となるアイテムの簡単なグラフ/アイテム履歴にアクセスします
- 問題の外部チケットにアクセスします ([webhooks](#)を設定するときにイベントメニューエントリを含めるオプションを参照してください)
- [グローバルスクリプト](#)を実行します (これらのスクリプトの範囲は手動イベントアクションとして定義されている必要があります)。この機能は外部システムで障害のあるチケットを管理するために使用されるスクリプトを実行する場合に便利です。

詳細表示












監視データ → 障害の障害開始時刻と復旧時刻のリンクをクリックするとイベントの詳細が開きます。

Event details

Trigger details		Actions	
Host	New host	Step	Time
Trigger	CPU load too high on "New host" for 3 minutes	User/Recipient	Action
Severity	Warning	Message/Command	Status
Problem expression	{New host:system.cpu.load.avg(3m)}>2	Info	
Recovery expression			
Event generation	Normal		
Allow manual close	No		
Enabled	Yes		
Event details		Event list [previous 20]	
Event	CPU load too high on "New host" for 3 minutes	Time	Recovery time
Operational data	1.99	Status	Age
Severity	Information	Duration	Ack
Time	2019-10-15 16:12:35	Actions	
Acknowledged	Yes		
Tags	Service: Operations		
Description			

トリガーと障害イベントで障害の深刻度がどのように異なるかに注意してください。障害イベントは障害の更新画面を使用して更新されています。

アクションリストでは、アクティビティタイプを示すために次のアイコンが使用されています。


-  - 障害イベントが生成されました
-  - メッセージが送信されました
-  - 障害のイベントが確認されました
-  - 未確認の問題イベント
-  - コメントが追加されました
-  - 障害の深刻度が高くなりました (例: 情報 → 警告)
-  - 障害の深刻度が低下しました (例: 警告 → 情報)
-  - 障害の深刻度は変更されましたが、元のレベルに戻りました (例: 警告 → 情報 → 警告)
-  - リモートコマンドが実行されました
-  - 障害イベントが回復しました
-  - 障害は手動でクローズされました

3 ホスト

概要

監視データ → ホストセクションには、監視対象ホストの完全なリストが表示され、ホストインターフェイス、エージェントの状態、タグ、現在の問題、ステータス (有効/無効) の詳細情報およびホストの最新データ、障害、履歴、グラフ、ダッシュボードおよび Web シナリオに簡単に移動するためのリンクが表示されます、

Hosts Create host									
Name	Interface	Availability	Tags	Status	Latest data	Problems	Graphs	Dashboards	Web
Apache server DC1	127.0.0.1:10050	ZBX		Enabled	Latest data	Problems	Graphs	Dashboards	Web
Zabbix NYC	127.0.0.1:10050	ZBX	Apache	Enabled	Latest data 2	1	Graphs 27	Dashboards 3	Web
Zabbix server	127.0.0.1:10050	ZBX		Enabled	Latest data 163	1 2 1 1	Graphs 27	Dashboards 3	Web
Zabbix Tokyo	127.0.0.1:10050	ZBX		Enabled	Latest data 26	1	Graphs 5	Dashboards 2	Web

列	詳細
名前	表示されるホスト名。名前をクリックすると、 ホストメニュー が表示されます。 名前の後にオレンジ色のレンチアイコン  が表示されている場合は、このホストがメンテナンス中であることを示します。
インターフェース エージェントの状態	列ヘッダーをクリックして、ホストを名前の昇順または降順で並べ替えます ホストのメインインターフェイスが表示されます 構成されたインターフェースごとのホストの 状態 アイコンは、構成されたインターフェースタイプ (Zabbix エージェント、SNMP、IPMI、JMX) のみを表します。アイコンにマウスオーバーすると、このタイプのすべてのインターフェースのポップアップリストが表示され、各インターフェースの詳細、ステータス、エラーが示されます。 インターフェースのないホストの場合、列は空です。 すべてのインターフェースの現在のステータス 1 つのタイプのアイコンがそれぞれのアイコンの色で表示されます。 緑 - すべてのインターフェースが使用可能 黄色 - 少なくとも 1 つのインターフェースが使用可能で、少なくとも 1 つが使用不可、その他に「不明」を含む任意の値を設定できます。 赤 - 使用可能なインターフェースがありません 灰色 - 少なくとも 1 つのインターフェースが不明 (使用不可) アクティブな Zabbix エージェントアイテムは使用できないことに注意してください。ホストの可用性に影響します。
タグ ステータス	ホストとすべてのリンクされたテンプレートの タグ 。マクロは未解決。 ホストステータス - 有効または無効。 列ヘッダーをクリックして、ステータスの昇順または降順でホストを並べ替えます
最新データ	リンクをクリックすると、ホストから収集されたすべての最新データを含む監視データ - 最新データページが開きます。 最新データを含むアイテムの数が灰色で表示されます (Zabbix 6.0.5 以降で表示されます)
障害	深刻度でソートされたオープンホストの障害の数。四角の色は障害の深刻度を示します。四角の数字は、特定の深刻度の障害数を意味します。 アイコンをクリックすると、現在のホストの監視データ - 障害ページが開きます。 ホストに障害がない場合は、このホストの障害セクションへのリンクがテキストとして表示されます。 フィルターを使用して抑制された障害を含めるかどうかを選択します (デフォルトでは含まれません)
グラフ	リンクをクリックすると、ホスト用に設定されたグラフが表示されます。グラフの数は灰色で表示されます。 ホストにグラフがない場合、リンクは無効になり (灰色のテキスト) 数字は表示されません。
ダッシュボード	リンクをクリックすると、ホスト用に設定されたダッシュボードが表示されます。ダッシュボードの数は灰色で表示されます。 ホストにダッシュボードがない場合、リンクは無効になり (灰色のテキスト) 数字は表示されません。
Web	リンクをクリックすると、ホスト用に設定された Web シナリオが表示されます。Web シナリオの数は灰色で表示されます。 ホストに Web シナリオがない場合、リンクは無効になり (灰色のテキスト) 数字は表示されません。

ボタン

ホストの作成では、**新しいホスト**を作成できます。このボタンは、管理者およびスーパー管理者ユーザーのみが使用できます。

すべてのセクションに共通の表示モードボタンについては**監視データ**ページで説明されています。

フィルタの使用

フィルタを使用して、任意のホストのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタはテーブルの上にあります。名前、ホストグループ、IP または DNS、ポート、タグ、障害の深刻度、ステータス (有効/無効/すべて) でホストをフィルタリングできます。メンテナンス中の障害とメンテナンス中のホストを表示するかどうかを選択することもできます。お気に入りのフィルター設定をタブとして保存し、**フィルターの上のタブ**をクリックしてすばやくアクセスできます。

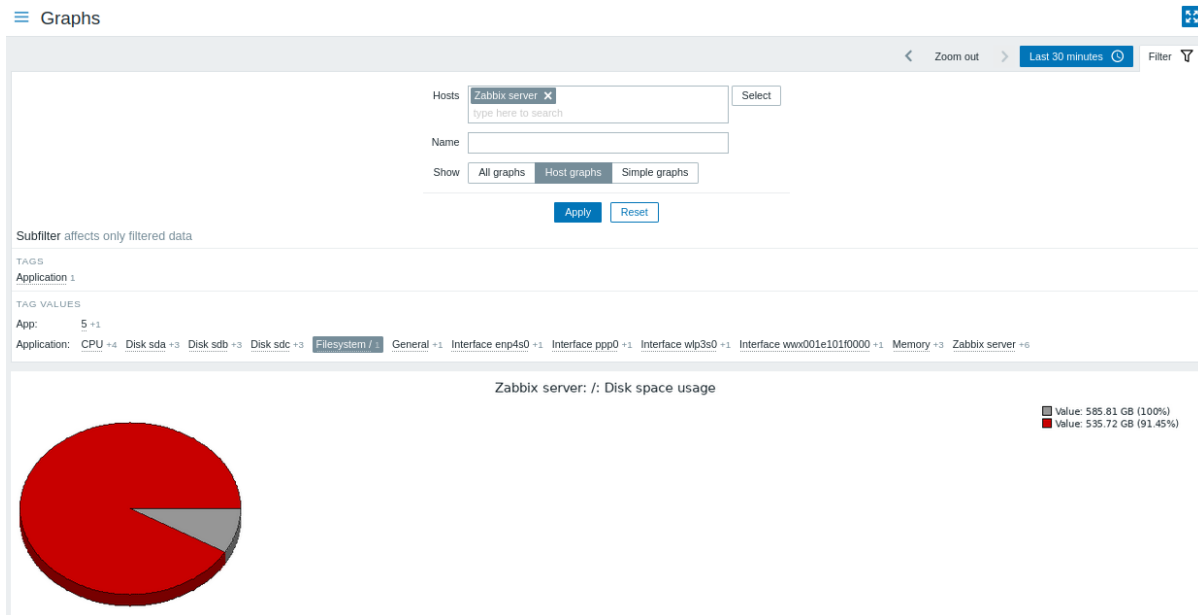
パラメータ	詳細
名前	表示ホスト名でフィルタリングします
ホストグループ	1つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます
IP アドレス	IP アドレスでフィルタリングします
DNS	DNS 名でフィルタリングします
ポート	ポート番号でフィルタリングします
深刻度	障害の深刻度でフィルタリングします。デフォルトでは、すべての深刻度のメンテナンスされていない障害が表示されます
ステータス	ステータスでフィルタリングします
タグ	ホストタグの名前と値でフィルタリングします。ホストは、ホストレベルのタグと、親テンプレートを含むすべてのリンクされたテンプレートのタグでフィルタリングできます。 特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。 条件ごとに使用できる演算子がいくつかあります。 Exists - 指定されたタグ名を含める Equals - 指定されたタグ名を含める値 (大文字と小文字を区別) Contains - 指定されたタグ名を含み、タグ値に入力された文字列が含まれる (部分文字列の一致、大文字と小文字を区別しない) Does not exist - 指定されたタグ名を含めない Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別) Does not contain - タグ値に入力された文字列が含まれる指定されたタグ名を除外する (サブ文字列の一致、大文字と小文字を区別しない) 条件には 2 つの計算タイプがあります： And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます Or - 1 つの条件が満たされる
メンテナンス中のホストを表示	チェックボックスをオンにすると、メンテナンス中のホストも表示されます (デフォルトはオン)
メンテナンス中の障害を表示	チェックボックスをオンにすると、メンテナンス中の障害も表示されます。(デフォルトはオフ)

1 グラフ

概要

ホストグラフには監視データ → ホストのそれぞれのホストのグラフをクリックしてアクセスできます。

単純なグラフだけでなく、ホスト用に構成された**カスタムグラフ**も表示できます。



グラフは次の順序で並べ替えられます。

- グラフ名 (カスタムグラフ)
- アイテム名 (シンプルグラフ)

無効になっているホストのグラフにもアクセスできます。

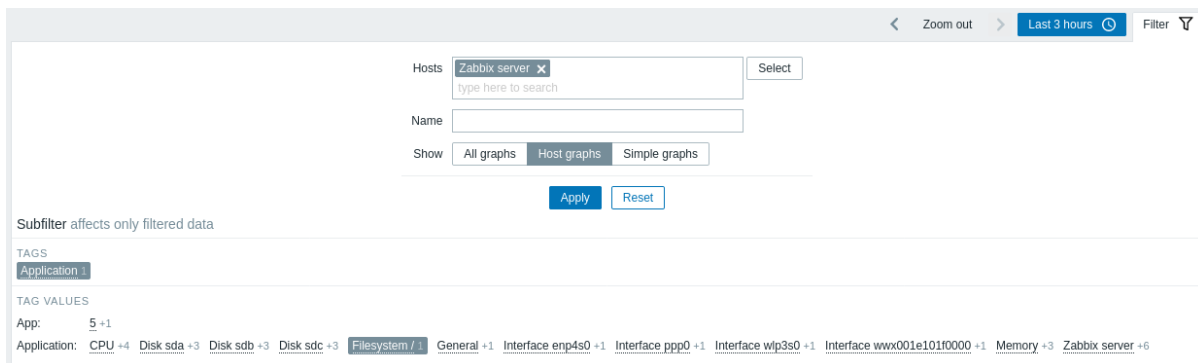
期間セレクトター

グラフの上にある期間セレクトターを使用すると、マウスを 1 回クリックするだけで期間を設定できます。

参照：[期間セレクトター](#)

フィルタの使用

特定のグラフを表示するには、フィルターでそのグラフを選択します。フィルターを使用すると、ホスト、グラフ名、および表示オプション (すべて/ホストグラフ/単純グラフ) を指定できます。



フィルタでホストが選択されていない場合、グラフは表示されません。

サブフィルタの使用

サブフィルターは、関連するグラフにワンクリックですばやくアクセスするのに役立ちます。サブフィルターはメインフィルターから自律的に動作します。結果はすぐにフィルターされ、メインフィルターの [適用] をクリックする必要はありません。

サブフィルターでは、メインフィルターからのフィルターのみをさらに変更できることに注意してください。

メインフィルターとは異なり、サブフィルターは各テーブル更新要求とともに更新され、使用可能なフィルターオプションとそのカウンター番号の最新情報を常に取得します。

サブフィルターにはクリック可能なリンクが表示され、共通のエンティティ (タグ名またはタグ値) に基づいてグラフをフィルター処理できます。エンティティがクリックされるとすぐに、グラフはすぐにフィルタリングされます。選択したエンティティは灰色の背景で強調表示されます。フィルタリングを削除するには、エンティティをもう一度クリックします。フィルタリングされた結果に別のエンティティを追加するには、別のエンティティをクリックします。

表示されるエンティティの数は、水平方向に 100 に制限されています。それ以上ある場合は、最後に 3 つのドットのアイコンが表示されますが、クリックしても反応しません。垂直リスト (値付きのタグなど) は、20 エントリに制限されています。それ以上ある場合は、3 つのドットのアイコンが表示されますがクリックしても反応しません。

クリック可能な各エンティティの横の数字は、メインフィルターの結果に含まれるグラフの数を示します。

1つのエンティティが選択されると、他の使用可能なエンティティの番号が、現在の選択に追加できるグラフの数を示すプラス記号とともに表示されます。

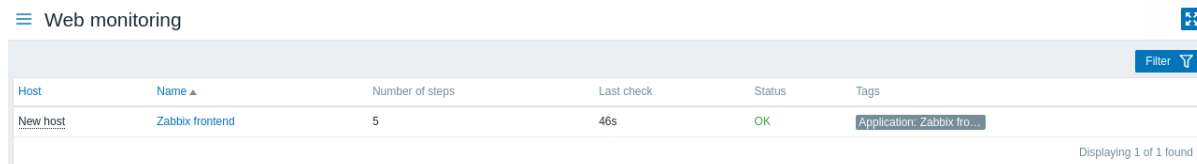
ボタン

すべてのセクションに共通の表示モードボタンについては[監視データ](#)ページで説明されています。

2 Web シナリオ

概要

ホストの[Web シナリオ](#)情報には、監視データ → ホストでそれぞれのホストの Web をクリックしてアクセスできます。



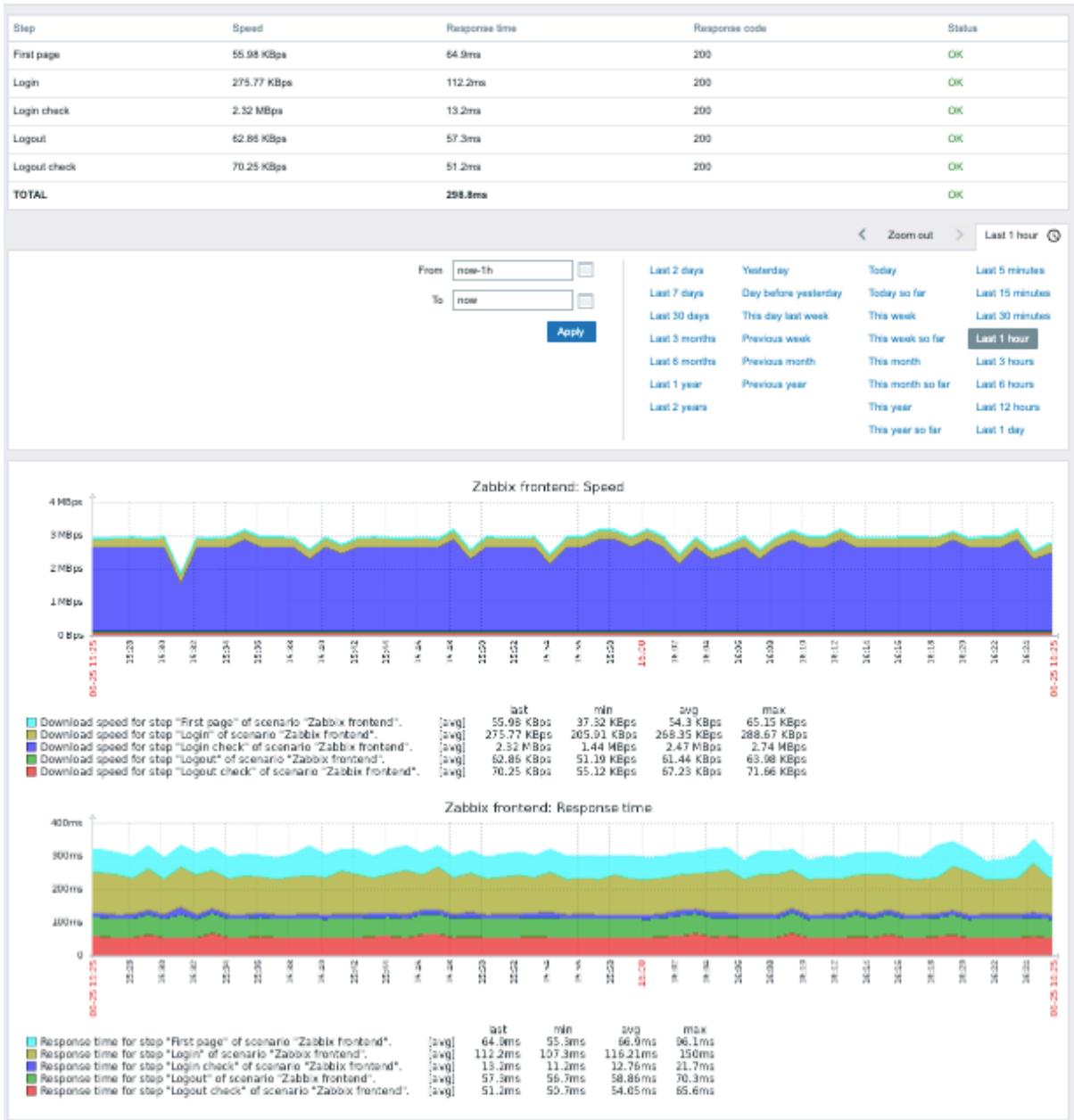
Host	Name ▲	Number of steps	Last check	Status	Tags
New host	Zabbix frontend	5	46s	OK	Application: Zabbix fro...

無効になっているホストのデータにもアクセスできます。無効になっているホストの名前は赤で表示されます。

ページごとに表示されるシナリオの最大数は、ユーザープロフィール[設定](#)のページあたりの表示行数によって異なります。

デフォルトでは、過去 24 時間以内の値のみが表示されます。この制限は、最新データの大きなページの初期読み込み時間を改善することを目的として導入されました。この期間は、管理 → [一般設定](#)の履歴の最大表示期間パラメーターの値を変更することで延長できます。

シナリオ名は、それに関するより詳細な統計へのリンクです。



フィルタの使用

このページには、選択したホストのすべての Web シナリオのリストが表示されます。監視データ → ホストページに戻らずに別のホストまたはホストグループの Web シナリオを表示するには、フィルターでそのホストまたはグループを選択します。タグに基づいてシナリオをフィルタリングすることもできます。

ボタン

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

4 最新データ

概要

このセクションでは、アイテムによって収集された最新の値を表示できます。

アイテム値のグラフも利用できます。

Latest data

Memory | CPU | Server | Web checks

Subfilter affects only filtered data

HOSTS
Zabbix server 2

TAG VALUES
Application: Interface enp4s0 Interface ppp0 +2 Interface wlp3s0 +2

Host	Name	Last check	Last value	Change	Tags	Info
Zabbix server	Interface enp4s0: Bits received	3s	5.35 Kbps	-496 bps	Application: Interface ...	Graph
Zabbix server	Interface enp4s0: Bits sent	3s	992 bps	-144 bps	Application: Interface ...	Graph

0 selected | Display stacked graph | Display graph | Execute now

Displaying 2 of 2 found

このセクションの内容は次のとおりです。

- **フィルタ** (デフォルトでは折りたたまれています)
- **サブフィルタ** (折りたためません)
- **アイテムリスト**

アイテムは、名前、最新のチェック時刻、最新の値、変化、タグ、およびアイテム値の単純なグラフ/履歴へのリンクとともに表示されます。

アイテム名をクリックすると、使用可能なグラフとアイテム設定へのリンクを含むアイテムメニューが開きます。

最新の値列の値は単位変換と値マッピングが適用された状態で表示されます。元データを表示するには値にカーソルを合わせます。

アイテムリストのタグはクリック可能です。タグをクリックすると、**サブフィルター**でこのタグが有効になります。アイテムリストに、このタグと、サブフィルターで以前に選択した他のタグに対応するアイテムが表示されます。アイテムがこのようにフィルタリングされると、リスト内のタグはクリックできなくなることに注意してください。タグに基づくさらなる変更（たとえば、削除、別のフィルターの追加）は、サブフィルターで実行する必要があります。

アイテムにエラーがある場合、たとえば、サポートされなくなった場合、情報アイコンが情報列に表示されます。詳細については、アイコンにカーソルを合わせてください。

疑問符の付いたアイコンには説明があります。このアイコンにカーソルを合わせると、アイテムの説明が記載されたツールチップが表示されます。

アイテムが属するホストがメンテナンス中の場合、オレンジ色のレンチアイコンがホスト名の後に表示されます。

注：無効になっているホストの名前は赤で表示されます。グラフやアイテム値リストなど、無効になっているホストのデータにも、最新の値でアクセスできます。

デフォルトでは、過去 24 時間以内の値のみが表示されます。この制限は、最新データの大きなページの初期読み込み時間を改善することを目的として導入されました。この期間は、管理 → **一般設定**の履歴の最大表示期間パラメーターの値を変更することで延長できます。

Attention:

更新頻度が 1 日以上アイテムの場合、変更量は表示されません（デフォルト設定の場合）。また 24 時間以上前に受信した最後の値は表示されません。

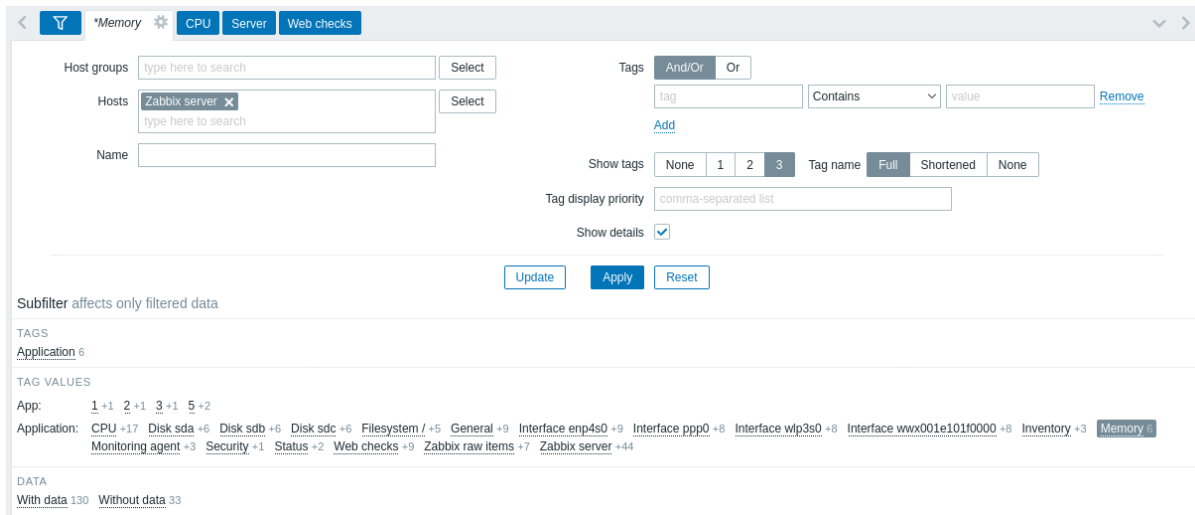
ボタン

すべてのセクションに共通の表示モードボタンについては**監視データ**ページで説明されています。

フィルタの使用

フィルタを使用して、任意のアイテムのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタアイコンは、テーブルとサブフィルタの上にあります。クリックしてフィルターを展開します。



フィルタを使用すると、ホストグループ、ホスト、アイテム名、タグ、およびその他の設定でリストを絞り込むことができます。フィルタで親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。タグによるフィルタリングの詳細については、監視データ -> 障害を参照してください。

詳細を表示を使用すると、アイテムに表示される情報を拡張できます。更新間隔、履歴とトレンドの設定、アイテムの種類、アイテムのエラー（細かいサポートされていない）などの詳細が表示されます。

フィルタの保存

フィルタ設定は、後ですばやくアクセスできるように保存できます。フィルタを保存するには保存するボタンをクリックし、フィルタ名と、保存したフィルタのレコード数を表示または非表示にするオプションを入力します。保存すると、フィルターは名前付きフィルタータブとして作成され、すぐにアクティブになります。

フィルタの上にある左/右ボタンを使用して、保存されたフィルタ間を移動できます。または、下向きのボタンをクリックすると、保存されているすべてのフィルターを含むドロップダウンメニューが開き、必要なフィルターをクリックできます。

保存したフィルターの設定を変更した場合は、フィルター名の前にアスタリスクが斜体で表示されます。新しい設定でフィルターを更新するには保存するボタンの代わりに表示される更新ボタンをクリックします。

リセットボタンを使用すると、メインフィルターとサブフィルターの未保存のフィルター設定を無効にすることができます。

サブフィルターの使用

サブフィルターは、関連するアイテムのグループにワンクリックですばやくアクセスするのに役立ちます。サブフィルターはメインフィルターから自律的に動作します。結果はすぐにフィルターされ、メインフィルターの適用をクリックする必要はありません。

サブフィルターでは、メインフィルターからのフィルターのみをさらに変更できることに注意してください。

メインフィルターとは異なり、サブフィルターは各テーブル更新要求とともに更新され、使用可能なフィルターオプションとそのカウンター番号の最新情報を常に取得します。

サブフィルターにはクリック可能なリンクが表示され、ホスト、タグ名、タグ値などの共通のエンティティに基づいてアイテムをフィルター処理できます。エンティティがクリックされるとすぐに、アイテムはすぐにフィルタリングされます。選択したエンティティは灰色の背景で強調表示されます。フィルタリングを削除するには、エンティティをもう一度クリックします。フィルタリングされた結果に別のエンティティを追加するには、別のエンティティをクリックします。

エンティティグループ（タグ、ホストなど）ごとに、最大 10 行のエンティティが表示されます。さらにエンティティがある場合は、最後に表示される 3 つのドットのアイコンをクリックすることで、このリストを最大 1000 エントリ（フロントエンド定義の SUBFILTER_VALUES_PER_GROUP の値）に拡張できます。最大に拡張すると、リストを折りたたむことはできません。（注拡張不可能な最大値 100 は、Zabbix 6.0.5 より前の制限であることに注意してください。）

タグ値のリストには、最大 10 行のタグ名が表示されます。値の付いたタグ名がさらにある場合は、下部に表示されている 3 つのドットアイコンをクリックすると、このリストを最大 200 個のタグ名に展開できます。最大に拡張すると、リストを折りたたむことはできません。（注拡張不可能な最大行 20 は Zabbix 6.0.5 より前の制限であることに注意してください。）

タグ名ごとに最大 10 行の値が表示されます（1000 エントリまで拡張可能（フロントエンド定義の SUBFILTER_VALUES_PER_GROUP の値））。

サブフィルターのホストオプションは、メインフィルターでホストが選択されていないか、複数のホストが選択されている場合のみ使用できます。

デフォルトでは、データのあるアイテムとないアイテムがアイテムリストに表示されます。メインフィルターでホストが 1 つだけ選択されている場合、サブフィルターには、このホストのデータを含むアイテムのみ、データを含まないアイテムのみ、またはその両方をフィルタリングするオプションがあります。

クリック可能な各エンティティの横の数字は、メインフィルターの結果に含まれるアイテムの数を示します。以前にサブフィルターで選択されていない限り、アイテムのないエンティティは表示されません。

1つのエンティティが選択されると、他の使用可能なエンティティの番号が、現在の選択に追加できるアイテムの数を示すプラス記号とともに表示されます。

グラフ

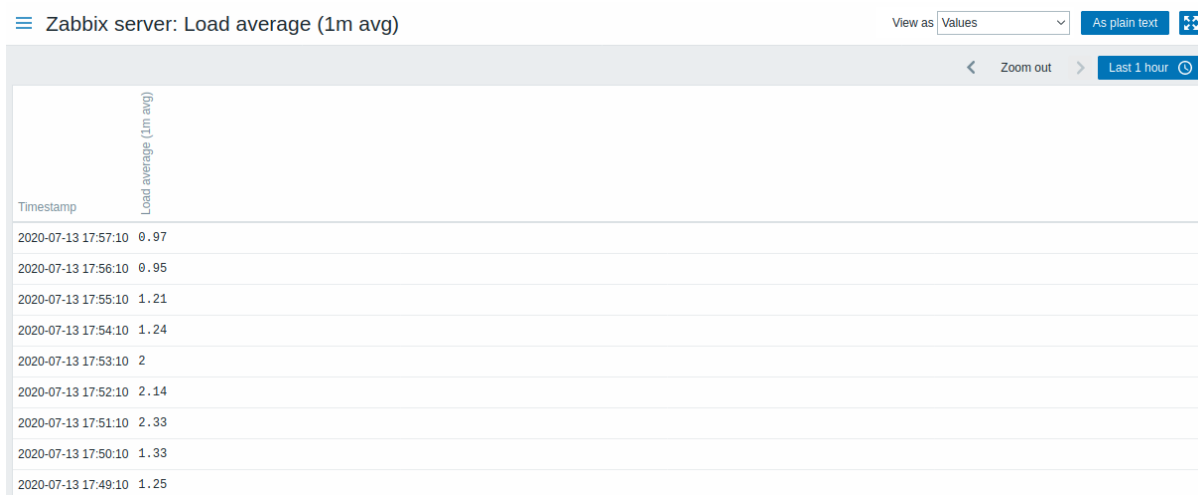
アイテムを比較するためのアドホックグラフ

最初の列のチェックボックスを使用して複数のアイテムを選択し、それらのデータを単純なアドホックグラフで比較できます。これを行うには任意の項目を選択し、表の下にあるグラフボタンをクリックします。

値の履歴/単純なグラフへのリンク

最新の値リストの最後の列は次のとおりです。

- 履歴リンク (すべてのテキストアイテム用) - 以前のアイテム値の履歴を表示するリスト (値/最新のデータ 500) につながります。
- グラフリンク (すべての数値項目用) - **単純なグラフ**につながります。ただし、グラフが表示されると、右上のドロップダウンで値/最新のデータ 500 に切り替えることもできます。



このリストに表示される値は"raw"です。つまり、後処理は適用されません。

Note:

表示される値の合計は**管理** → **一般設定**で設定された検索/フィルターの項目の上限値パラメーターの値によって定義されます。

5 マップ

概要

監視データ → マップセクションでは**ネットワークマップ**を構成、管理、および表示できます。

このセクションを開くと最後にアクセスしたマップ、またはアクセスできるすべてのマップのリストが表示されます。

すべてのマップは、パブリックまたはプライベートのいずれかになります。パブリックマップはすべてのユーザーが利用できますが、プライベートマップは、所有者とマップが共有されているユーザーのみがアクセスできます。

マップリスト

The screenshot shows the 'Maps' section in Zabbix. It includes a 'Filter' button and a table with the following data:

Name ▲	Width	Height	Actions
<input type="checkbox"/> Local network	600	400	Properties Constructor
<input type="checkbox"/> Local network2	680	200	Properties Constructor

At the bottom, it shows '0 selected', 'Export', and 'Delete' buttons, and 'Displaying 2 of 2 found'.

表示情報

列	詳細
名前	マップの名前。名前をクリックして、マップを 表示 します。
幅	マップの表示幅
高さ	マップの表示高
アクション	次の2つのアクションを使用できます。 プロパティ - 一般的なマップの プロパティ 編集 変更 - マップ要素 グリッドにアクセス

新しいマップを**設定**するには、右上隅にあるマップの作成ボタンをクリックします。YAML、XML、またはJSONファイルからマップをインポートするには、右上隅にあるインポートボタンをクリックします。マップをインポートしたユーザーがその所有者として設定されます。

リストの下にある2つのボタンは、いくつかの一括編集オプションを提供します。

- エクスポート - マップをYAML,XML,JSONいずれかのファイルにエクスポートします
- 削除 - マップを削除します

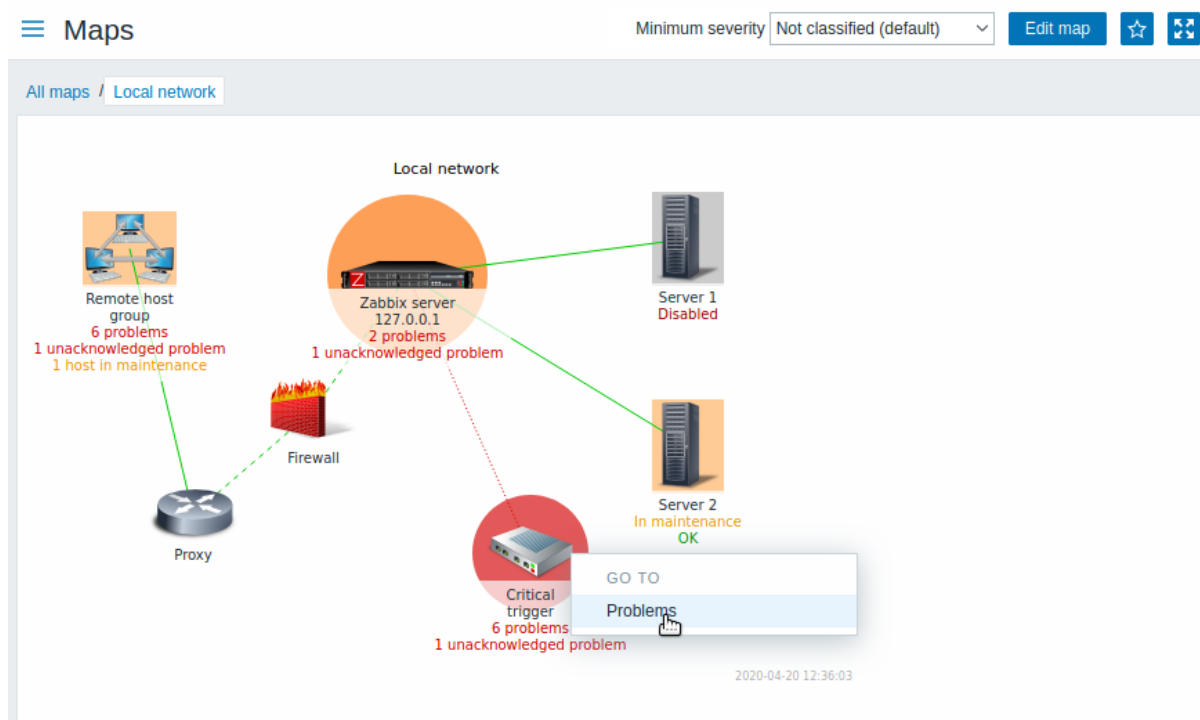
これらのオプションを使用するには、マップの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のマップのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

マップの表示

マップを表示するには、マップのリストで表示したいマップの名前をクリックします。



マップのタイトルバーのドロップダウンを使用して、表示する問題トリガーの最低の重大度レベルを選択できます。default としてマークされた重大度は、マップ設定で設定されたレベルです。マップにサブマップが含まれている場合は、サブマップに移動すると上位レベルのマップの深刻度が保持されます（ただし、分類されていない場合を除き、深刻度はサブマップに渡されません）。

アイコンの強調表示

マップ要素に障害がある場合、丸い円で強調表示されます。円の塗りつぶしの色は、問題の深刻度の色に対応しています。選択した深刻度レベル以上の問題のみが要素とともに表示されます。すべての障害が確認されると、円の周りに太い緑色の境界線が表示されます。

さらに：

- **メンテナンス**のホストは、オレンジ色の塗りつぶされた四角で強調表示されます。メンテナンスの強調表示は、問題の深刻度の強調表示よりも優先されることに注意してください（Zabbix 6.0.5 以降、マップ要素がホストの場合のみ）
- 無効になっている（監視されていない）ホストは、灰色の塗りつぶされた四角で強調表示されます。

マップ設定でアイコンの強調表示チェックボックスがマークされている場合、強調表示されます。

最近の変更マーカー




要素の周りを内側に向けた赤い三角形は、最近のトリガーステータスの変更を示します。これは過去 30 分以内に発生したものです。これらの三角形は、マップ設定でトリガーステータス変更時に要素をマークするチェックボックスがマークされている場合に表示されます。

リンク

マップ要素をクリックすると、いくつかの利用可能なリンクを含むメニューが開きます。

ボタン

右側のボタンには、次のオプションがあります。

	マップコンストラクターに移動して、マップコンテンツを編集します
	ダッシュボードのお気に入りウィジェットにマップを追加します
	マップはダッシュボードのお気に入りウィジェットに登録済みです。クリックでお気に入りウィジェットからマップを削除します

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

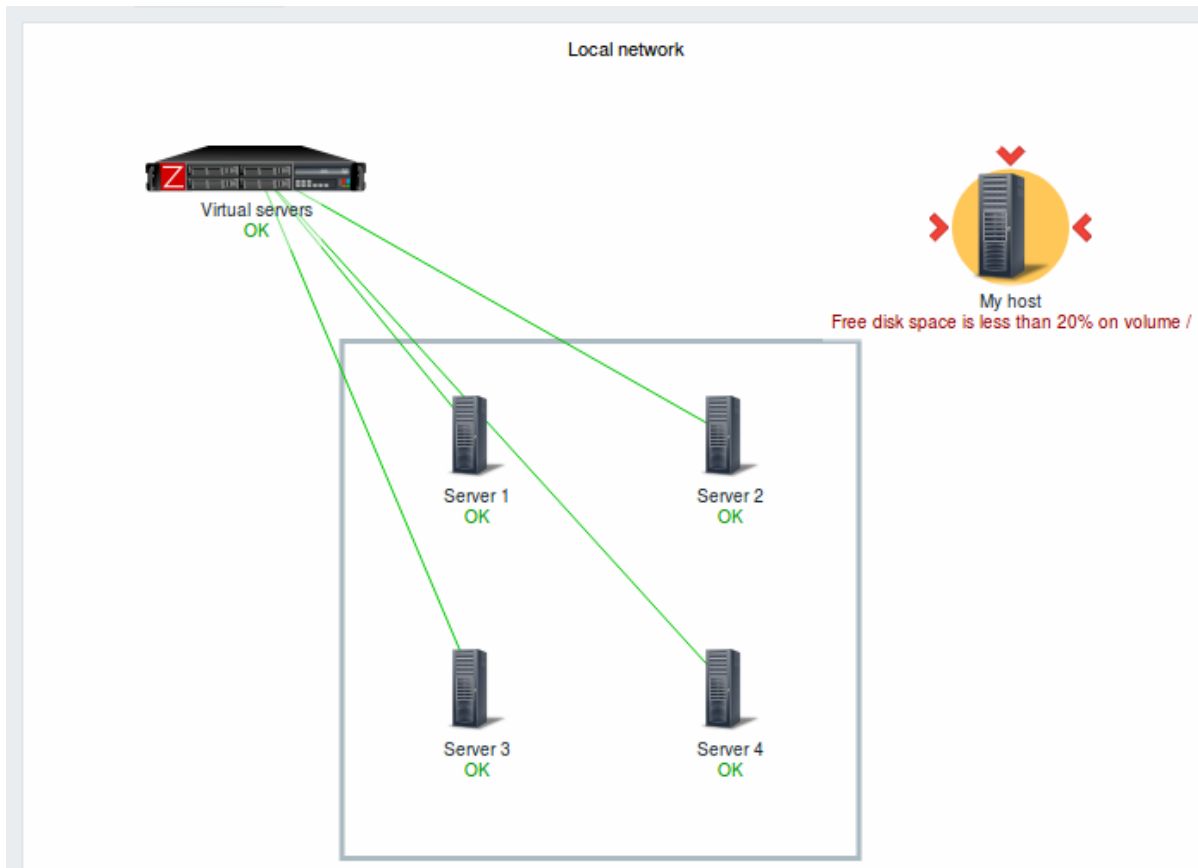
読みやすい要約地図

非表示の「aria-label」プロパティを使用すると、スクリーンリーダーで地図情報を読み取ることができます。一般的なマップの説明と個々の要素の説明の両方が、次の形式で利用できます。

- マップの説明: <Map name>, <* of * items in problem state>, <* problems in total>.
- 1 つの障害で 1 つの要素を記述する: <Element type>, Status <Element status>, <Element name>, <Problem description>.
- 複数の障害がある 1 つの要素を記述する: <Element type>, Status <Element status>, <Element name>, <* problems>.
- 障害のない 1 つの要素を記述する: <Element type>, Status <Element status>, <Element name>.

たとえば、次のマップの場合以下のような説明が利用可能です。

```
'Local network, 1 of 6 elements in problem state, 1 problem in total. Host, Status problem, My host, Free
```



ネットワークマップの参照

ネットワークマップは `sysmapid` と `mapname` の両方の GET パラメータで参照できます。例えば、
<http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network>
 指定された名前のマップを開きます。(Local network)

`sysmapid` (マップ ID) と `mapname` (マップ名) の両方が指定されている場合、`mapname` の優先度が高くなります。

6 ディスカバリ

概要

監視データ → ディスカバリセクションにはネットワークディスカバリ (/manual/discovery/network_discovery) の結果が表示されます。検出されたデバイスは、ディスカバリルールによって並べ替えられます。

☰ Status of discovery Filter ▼

Discovery rule

Discovered device ▼	Monitored host	Uptime/Downtime	SNMPv2 agent: iso-3.6.1.2.1.1.1.0
Local network (14 devices)			
192.168.3.114 (radix-ilo.zabbix.lan)	Integrated Lights-Out 4 2.61 Jul 27 2018	1d 2h 47m	
192.168.3.72 (winxp.zabbix.lan)	Linux zeus 4.8.6.5-smp_2 SMP Sun Nov 13 14_58_11 CDT 2016 i686	7 days, 20:37:53	7d 20h 37m
192.168.3.70 (win2008i386.zabbix.lan)	Hardware_ x86 Family 6 Model 23 Stepping 6 AT_AT COMPATIBLE - Software_ Windows Version 6.0_Build 6001 Multiprocessor Free_	2 days, 02:23:47	2d 2h 23m

デバイスがすでに監視されている場合は、ホスト名が監視中のホスト列に表示され、前回の検出後にデバイスが検出または失われた期間がアップタイム/ダウンタイム列に表示されます。

検出された各デバイスの個々のサービスの状態が表示されます。(赤いセルはダウンしているサービスを示します) サービスのアップタイムまたはダウンタイムはセルに含まれます。

Attention:

少なくとも 1 つのデバイスで検出されたサービスに、状態を示す列が表示されます。

ボタン

すべてのセクションに共通の表示モードボタンについては[監視データ](#)ページで説明されています。

フィルタの使用

フィルタを使用して、任意のディスカバリルールのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタで何も選択されていない場合、有効なすべてのディスカバリルールが表示されます。表示する特定のディスカバリルールを選択するには、フィルターにその名前を入力します。一致するすべての有効なディスカバリルールが選択のために一覧表示されます。複数のディスカバリルールを選択できます。

2 サービス

概要

サービスメニューは、Zabbix の[サービスモニタリング](#)機能です。

1 Services

Overview



In this section you can see a high-level status of whole services that have been configured in Zabbix, based on your infrastructure.

A service may be a hierarchy consisting of several levels of other services, called "child" services, which are attributes to the overall status of the service (see also an overview of the [service monitoring](#) functionality.)

The main categories of service status are OK or Problem, where the Problem status is expressed by the corresponding problem severity name and color.

While the view mode allows to monitor services with their status and other details, you can also [configure](#) the service hierarchy in this section (add/edit services, child services) by switching to the edit mode.

To switch from the view to the edit mode (and back) click on the respective button in the upper right corner:

-  - view services
-  - add/edit services, and child services

Note that access to editing depends on [user role](#) settings.

Viewing services

Services				
Name	Status	Root cause	Created at	Tags
Availability 2	High	Nodata trigger, Nodata trigger 1h	2000-01-01	SLA: 3
Disc space	OK		2000-01-01	SLA: 1
Example service	OK		2000-01-01	SLA: 5

Displaying 3 of 3 found

A list of the existing services is displayed.

Displayed data:

Parameter	Description
Name	Service name. The service name is a link to service details . The number after the name indicates how many child services the service has.

Parameter	Description
Status	Service status: OK - no problems (trigger color and severity) - indicates a problem and its severity. If there are multiple problems, the color and severity of the problem with highest severity is displayed.
Root cause	Underlying problems that directly or indirectly affect the service status are listed. The same problems are listed as returned by the <code>{SERVICE.ROOTCAUSE}</code> macro. Click on the problem name to see more details about it in Monitoring → Problems. Problems that do not affect the service status are not in the list.
Created at	The time when the service was created is displayed.
Tags	Tags of the service are displayed. Tags are used to identify a service in service actions and SLAs.

Buttons




View mode buttons being common for all sections are described on the [Monitoring](#) page.

Using filter

You can use the filter to display only the services you are interested in.

Editing services

Click on the Edit button to access the edit mode. When in edit mode, the listing is complemented with checkboxes before the entries and also these additional options:

-  - add a child service to this service
-  - edit this service
-  - delete this service

Services Create service View Edit

<input type="checkbox"/>	Name	Status	Root cause	Created at	Tags	Filter
<input type="checkbox"/>	Availability 2	High	Nodata trigger, Nodata trigger 1h, Temperature is too high	2000-01-01	SLA: 3	+ ↙ ×
<input type="checkbox"/>	Disc space	OK		2000-01-01	SLA: 1	+ ↙ ×
<input type="checkbox"/>	Example service	OK		2000-01-01	SLA: 5	+ ↙ ×

Displaying 3 of 3 found

To **configure** a new service, click on the Create service button in the top right-hand corner.

Service details

To access service details, click on the service name. To return to the list of all services, click on All services.

Service details include the info box and the list of child services.

All services / Availability Info Filter

Availability

Parent services:

Status: **High**

SLA: SLA:3: 0

Tags: SLA: 3

<input type="checkbox"/>	Name	Reporting period	SLO	SLI	Uptime	Downtime	Error budget	Created at	Tags
<input type="checkbox"/>	Connections	2022-01-09 – 01-15	100%	0	0	4d 10h 48m	-4d 10h 48m	2000-01-01	SLA: 2 Type: Connection
<input type="checkbox"/>	Servers							2000-01-01	SLA: 4 Type: CPU

Displaying 2 of 2 found

0 selected Mass update Delete




To access the info box, click on the Info tab. The info box contains the following entries:

- Names of parent services (if any)
- Current status of this service
- Current SLA(s) of this service, in the format `SLA name:service level indicator`. 'SLA name' is also a link to the SLA report for this service. If you position the mouse on the info box next to the service-level indicator (SLI), a pop-up info list is displayed with SLI details. The service-level indicator displays the current service level, in percentage.
- Service tags

The info box also contains a link to the [service configuration](#).

To use the filter for child services, click on the Filter tab.

When in edit mode, the child service listing is complemented with additional editing options:

-  - add a child service to this service
-  - edit this service
-  - delete this service

2 サービスアクション

概要

サービス → サービスアクションセクションで、サービスアクションを**設定**および**維持**できます。

ユーザーロール権限で設定されたアクションがリストに表示されます。ユーザーには、ユーザーロールがアクセスを許可するサービスのアクションのみが表示されます。

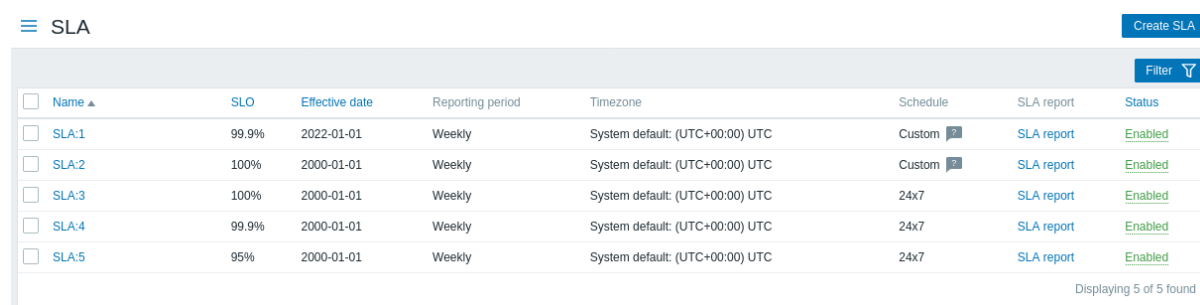
表示されるデータ、フィルター、および使用可能な一括編集オプションは、**アクション**と同じです。

3 SLA

概要

このセクションでは SLA を表示および**設定**することができます。

SLAs



The screenshot shows a table with columns: Name, SLO, Effective date, Reporting period, Timezone, Schedule, SLA report, and Status. There are 5 rows of SLA data. A 'Create SLA' button is in the top right, and a 'Filter' button is in the top right of the table area. The table footer says 'Displaying 5 of 5 found'.

Name	SLO	Effective date	Reporting period	Timezone	Schedule	SLA report	Status
SLA.1	99.9%	2022-01-01	Weekly	System default: (UTC+00:00) UTC	Custom	SLA report	Enabled
SLA.2	100%	2000-01-01	Weekly	System default: (UTC+00:00) UTC	Custom	SLA report	Enabled
SLA.3	100%	2000-01-01	Weekly	System default: (UTC+00:00) UTC	24x7	SLA report	Enabled
SLA.4	99.9%	2000-01-01	Weekly	System default: (UTC+00:00) UTC	24x7	SLA report	Enabled
SLA.5	95%	2000-01-01	Weekly	System default: (UTC+00:00) UTC	24x7	SLA report	Enabled

設定された SLAs のリストが表示されます。注ユーザーがアクセスできるサービスに関連する SLAs のみが表示されることに注意してください。(ユーザーロールに対して * SLA の管理 * が有効になっていない限り、読み取り専用として表示されます)

表示情報:

パラメータ	説明
名前	SLA 名が表示されます。 名前は SLA の設定 へのリンクです。
SLO	サービスレベル目標 (SLO) が表示されます。
発行日	SLA 計算の開始日が表示されます
レポート期間	SLA レポートで使用される期間が表示されます - 毎日、毎週、毎月、四半期、または毎年
タイムゾーン	SLA タイムゾーンが表示されます
スケジュール	SLA スケジュールが表示されます - 24 時間 365 日またはカスタム
SLA レポート	リンクをクリックして、この SLA の SLA レポートを表示します
ステータス	SLA ステータスが表示されます - 有効または無効

4 SLA レポート

概要

このセクションではフィルターで選択された基準に基づいて、SLA レポートを表示できます。

SLA レポートは**ダッシュボードウィジェット**として表示することもできます。

レポート

フィルタを使用すると SLA 名とサービス名に基づいてレポートを選択できます。表示期間を絞り込む事も可能です。

≡ SLA report

Service	SLO	2020-06	2020-07	2020-08	2020-09	2020-10	2020-11	2020-12	2021-01	2021-02	2021-03	2021-04	2021-05	2021-06	2021-07	2021-08	2021-09	2021-10	2021-11	2021-12	2022-01
Availability	100%	100	100	100	100	100	100	100	100	100	100	100	100	100	72.5434	0.0028	28.8072	17.049	0	0	0

Displaying 1 of 1 found

各列（期間）には、その期間の SLI が表示されます。セット SLO に違反している SLI は、赤で強調表示されます。レポートには 20 の期間が表示されます。開始日と終了日の両方を指定した場合、最大 100 の期間を表示できます。

レポートの詳細

レポートのサービス名をクリックすると、より詳細なビューを表示する別のレポートにアクセスできます。

≡ SLA report

Month	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2022-01	100%	0	0	12d 16h 16m	-12d 16h 16m	
2021-12	100%	0	0	1m 1d	-1m 1d	
2021-11	100%	0	0	1m	-1m	
2021-10	100%	17.049	5d 6h 50m	25d 17h 9m	-25d 17h 9m	
2021-09	100%	28.8072	8d 15h 24m	21d 8h 35m	-21d 8h 35m	
2021-08	100%	0.0028	1m 15s	1m 23h	-1m 23h	
2021-07	100%	72.5434	22d 11h 43m	8d 12h 16m	-8d 12h 16m	
2021-06	100%	100	1m	0	0	
2021-05	100%	100	1m 1d	0	0	
2021-04	100%	100	1m	0	0	
2021-03	100%	100	1m 1d	0	0	
2021-02	100%	100	28d	0	0	

3 インベントリ

概要

インベントリメニューには、選択したパラメータごとのホストインベントリデータの概要と、ホストインベントリの詳細を表示するセクションがあります。

1 ホストインベントリの概要

概要

インベントリ → 概要セクションではホストインベントリデータの概要を確認する方法を提供します。

表示する概要については、ホストグループ（またはなし）とデータを表示するインベントリフィールドを選択します。選択したフィールドの各エントリに対応するホストの数が表示されます。

Host inventory overview

Type	Host count
Server	4
Zabbix server	1

概要の完全性は、ホストで維持されるインベントリ情報の量によって異なります。

ホスト数列の数字はリンクです。これらのホストは Host Inventory テーブルで除外されます。

Host inventory

Host	Group	Name	Type	OS	Serial number	Tag	MAC address
Zabbix server	Zabbix servers	martins-hp	Zabbix server	Linux version 5.3.0-46-generic (buildd@lcy01-amd64-013) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1-18.04)) #38~18.04.1-Ubuntu SMP			

2 ホスト

概要

インベントリ → ホストセクションでは、ホストのインベントリデータが表示されます。

ホストグループおよびインベントリフィールドでホストをフィルタリングして、任意のホストのみを表示できます。

Host inventory

Host	Group	Name	Type	OS	Serial number	Tag	MAC address
Zabbix server	Zabbix servers	martins-hp	Zabbix server	Linux version 5.3.0-46-generic (buildd@lcy01-amd64-013) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1-18.04)) #38~18.04.1-Ubuntu SMP			

すべてのホストインベントリを表示するには、フィルターでホストグループを選択せず、フィルターの比較フィールドをクリアにしてフィルターを押します。

一部の主要なインベントリフィールドのみがテーブルに表示されますが、そのホストで利用可能なすべてのインベントリ情報を表示することもできます。これを行うには、最初の列のホスト名をクリックします。

インベントリの詳細

概要タブには、事前定義されたスクリプト、最新の監視データ、およびホスト設定オプションへのリンクとともに、ホストに関する一般的な情報が含まれています。

Host inventory

Overview **Details**

Host name **Zabbix server**

Agent interfaces

IP address	DNS name	Connect to	Port
127.0.0.1		IP DNS	10050

SNMP interfaces

127.0.0.1		IP DNS	161
-----------	--	--------	-----

OS Linux version 5.3.0-46-generic (buildd@lcy01-amd64-013) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)) #38~18.04.1-Ubuntu SMP

Monitoring [Web](#) [Latest data](#) [Problems](#) [Graphs](#) [Dashboards](#)

Configuration [Host](#) [Items 148](#) [Triggers 67](#) [Graphs 28](#) [Discovery 4](#) [Web 1](#)

[Cancel](#)

詳細タブには、ホストで利用可能なすべてのインベントリの詳細が含まれています。

Overview **Details**

Type **Zabbix server**

Name **martins-hp**

OS Linux version 5.3.0-46-generic (buildd@lcy01-amd64-013) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)) #38~18.04.1-Ubuntu SMP

[Cancel](#)

インベントリデータの完全性は、ホストで維持されるインベントリ情報の量によって異なります。情報が保持されていない場合、詳細タブは無効になります。

4 レポート

概要

レポートメニューには、システム情報、トリガー、収集されたデータなどのパラメーターの表示に焦点を当てた、事前定義されたユーザーがカスタマイズ可能なさまざまなレポートを含むいくつかのセクションがあります。

1 システム情報

概要

レポート → システム情報には、主要な Zabbix サーバーとシステムデータの概要が表示されます。

高可用性セットアップでは、ui/conf/zabbix.conf.php ファイルを編集してシステム情報ソース（サーバーインスタンス）をリダイレクトすることに注意してください - コメントを外して set \$ZBX_SERVER, \$ZBX_SERVER_PORT をアクティブに示されているサーバー以外のサーバーに設定します。

高可用性セットアップを有効にすると、システム統計の下に高可用性ノードの詳細を示す別のブロックが表示されます。このブロックは、Zabbix スーパー管理者ユーザーにのみ表示されます。

システム情報はダッシュボードウィジェットとしても利用できます。

システム情報

System information

Parameter	Value	Details
Zabbix server is running	Yes	192.168.8.103:10051
Number of hosts (enabled/disabled)	5	4 / 1
Number of templates	140	
Number of items (enabled/disabled/not supported)	199	155 / 29 / 15
Number of triggers (enabled/disabled [problem/ok])	89	87 / 2 [8 / 79]
Number of users (online)	3	1
Required server performance, new values per second	1.96	
High availability cluster	Enabled	Fail-over delay: 1 minute

Name	Address	Last access	Status
base	192.168.8.103:10051	2s	Active
base2	localhost:10051	5m 11s	Stopped

表示情報

パラメータ	値	詳細
Zabbix サーバーの起動	Zabbix サーバーのステータス： はい - サーバーは実行中です いいえ - サーバーは実行されていません 注：残りの情報を表示するには、サーバーの StartTrappers パラメーターが <code>zabbix_server.conf</code> ファイルにて >0 であり、少なくとも 1 つのトラッパープロセスが開始されている必要があります。	Zabbix サーバーの場所とポート
ホスト数 (有効/無効)	構成されているホストの総数が表示されます	監視対象ホスト/監視対象外ホストの数
テンプレート数	テンプレートの総数が表示されます	
アイテム数 (有効/無効/取得不可)	アイテムの総数が表示されます	有効/無効/取得不可のアイテム数 無効なホスト上のアイテムは無効としてカウントされます
トリガー数 (有効/無効 [障害/正常])	トリガーの総数が表示されます	有効/無効なトリガーの数 [障害/正常のトリガー] 無効なホストに割り当てられたトリガー、または無効なアイテムに依存するトリガーは、無効としてカウントされます
ユーザー数 (オンライン)	ユーザーの総数が表示されます	オンラインユーザー数
1 秒あたりの監視項目数 (Zabbix サーバーの要求パフォーマンス)	Zabbix サーバーによって 1 秒あたりに処理される監視項目数が表示されます	サーバーの要求パフォーマンスは推定値であり、ガイドラインとして役立ちます。処理される値の正確な数については、 <code>zabbix[ocache,values,all]</code> <code>internal item</code> を参照してください。 監視対象ホストからの有効なアイテムが値に含まれます。ログアイテムは、アイテムの更新間隔ごとに 1 つの値としてカウントされます。通常の間隔値はカウントされますが、柔軟なスケジューリング間隔の値はカウントされません。"nodata" メンテナンス期間中は計算されません。トラッパーアイテムはカウントされません。

パラメータ	値	詳細
データベース履歴テーブルのアップグレード	データベースのアップグレードステータス： いいえ - データベース履歴テーブルはアップグレードされていません	このフィールドは、数値（浮動小数点）値の拡張範囲へのデータベースのアップグレードが完了していない場合に表示されます。 手順 を参照して、数値（浮動小数点）の範囲を拡張できるようにします。
HA クラスター	Zabbix サーバーの 高可用性クラスター のステータス： 無効 - スタンドアロンサーバー 有効 - 少なくとも 1 つの高可用性ノードが存在する	有効の場合、フェイルオーバーディレイが表示されます

システム情報は、次の場合にもエラーメッセージを表示します。

- 使用されているデータベースに必要な文字セットまたは照合（UTF-8）がありません。
- データベースのバージョンが**サポートされている範囲外**（スーパー管理者ユーザーのみが使用できます）

HA ノード

HA クラスターが有効になっている場合、各 HA ノードのステータスとともに別のデータブロックが表示されます。

Name	Address	Last access	Status
node-active	192.168.1.13:10051	12s	Active
node6	192.168.1.10:10053	1h 2m 40s	Unavailable
node7	192.168.1.11:10053	3m 40s	Unavailable
node4	192.168.1.8:10052	1h 34m 29s	Stopped
node5	192.168.1.9:10053	1h 9m 51s	Stopped
node8	192.168.1.12:10051	21m 16s	Stopped
node1	192.168.1.5:10051	17s	Standby
node2	192.168.1.6:10051	16s	Standby
node3	192.168.1.7:10052	16s 2021-10-20 17:58:47	Standby

表示情報

列	詳細
名前	サーバー設定で定義されているノード名
アドレス	ノードの IP アドレスとポート
最終アクセスからの時間	ノードの最終アクセス時刻。 カーソルを合わせると、最終アクセスのタイムスタンプが長い形式で表示されません。
ステータス	ノードのステータス： アクティブ - ノードは稼働中です 使用不可 - ノードはフェイルオーバーディレイを超えて確認されていません（原因を確認することをお勧めします） ** 停止 - ノードが停止しているか、起動できませんでした（起動または削除することをお勧めします） スタンバイ ** - ノードが起動して待機しています

2 定期レポート

概要

レポート → 定期レポートでは、十分な権限を持つユーザーが、指定された受信者に電子メールで送信される PDF バージョンのダッシュ

Scheduled reports

Name Show **All** Created by me Status **Any** Enabled Disabled Expired

<input type="checkbox"/>	Name ▲	Owner	Repeats	Period	Last sent	St
<input type="checkbox"/>	Global view daily	Admin (Zabbix Administrator)	Daily	Previous day	Never	Er

ボードの生成スケジュールを設定できます。

開始画面には、スケジュールされたレポートに関する情報が表示されます。これらの情報は、ナビゲーションを容易にするためにフィルタリングできます。[フィルターの使用](#)セクションを参照してください。

表示情報

列	詳細
名前	レポート名
所有者	レポートを作成したユーザー
繰り返し	レポート生成頻度 (日/週/月/年)
期間	レポート作成期間
最終送信日時	最新レポートが送信された日時
ステータス	レポートの現在のステータス (有効/無効/期限切れ) 十分な権限を持つユーザーは、ステータスをクリックして [有効] から [無効] (およびその逆) または期限切れから無効 (およびその逆) へ変更できます。捜査権限の無いユーザーにはテキストで表示されます。
情報	インフォメーションアイコンを表示します。 赤いアイコンは、レポートの生成に失敗したことを示します。カーソルを合わせると、エラー情報が記載されたツールチップが表示されます。 黄色のアイコンは、レポートが生成されたが、一部 (またはすべて) の受信者への送信に失敗したか、レポートの有効期限が切れていることを示します。カーソルを合わせると、追加情報を含むツールチップが表示されます

フィルタの使用

フィルタを使用して、レポートのリストを絞り込むことができます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

次のフィルタリングオプションを使用できます。

- 名前 - 名前の部分的一致でも可能です
- 表示 - すべてまたはカレントユーザーが作成したもの
- ステータス - すべて、有効、無効、または期限切れ

フィルタは、定期レポートバーの下にあります。右上隅のフィルタータブをクリックすると、開いたり折りたたんだりできます。

一括更新

一度に複数のレポートのステータス変更したり、削除したりしたい場合があります。その場合は個々のレポートを開いて編集する代わりに、一括更新機能を使用することもできます。

レポートを一括更新するには、次の手順を実行します。

- リストで更新するレポートのチェックボックスをオンにします
- リストの下にある必要なボタンをクリックして、変更 (有効、無効、または削除) を行います。

3 稼働レポート

概要

レポート → 稼働レポートで、各トリガーが障害/正常状態にあった時間の割合を確認できます。各状態の時間がパーセンテージで表示されます。

システム上のさまざまな要素の可用性状況を簡単に判断できます。

Availability report

Mode By host

Zoom out Last 1 hour Filter

Host groups Select

Hosts Select

Apply Reset

Host	Name	Problems	Ok	Graph
Zabbix server	/: Disk space is critically low (used > 90%)		100.0000%	Show
Zabbix server	/: Disk space is low (used > 80%)	0.0556%	99.9444%	Show
Zabbix server	/: Running out of free inodes (free < 10%)		100.0000%	Show
Zabbix server	/: Running out of free inodes (free < 20%)		100.0000%	Show
Zabbix server	/etc/passwd has been changed		100.0000%	Show
Zabbix server	Configured max number of open filedescriptors is too low (< 256)		100.0000%	Show

右上隅のドロップダウンから、選択モードを選択できます。トリガーをホストごとに表示するか、トリガーテンプレートごとに表示するかを選択できます。

Availability report

Mode By trigger template

Zoom out Last 1 hour Filter

Template group

Template

Template trigger

Host group

Apply Reset

Host	Name	Problems	Ok	Graph
My host	/etc/passwd has been changed		100.0000%	Show
My host	Configured max number of open filedescriptors is too low (< 256)		100.0000%	Show
My host	Configured max number of processes is too low (< 1024)		100.0000%	Show
My host	Getting closer to process limit (over 80% used)		100.0000%	Show
My host	High CPU utilization (over 90% for 5m)		100.0000%	Show
My host	High memory utilization (>90% for 5m)		100.0000%	Show
My host	High swap space usage (less than 50% free)	100.0000%		Show
My host	Lack of available memory (< 20M of 15.54 GB)		100.0000%	Show
My host	Load average is too high (per CPU load over 1.5 for 5m)		100.0000%	Show

トリガーの名前は、そのトリガーの最新イベントへのリンクです。

フィルタの使用

フィルタは、表示されるホストやトリガーの数を絞り込むのに役立ちます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタは稼働レポートバーの下にあります。フィルタータブをクリックすると、開いたり折りたたんだりできます。

トリガーテンプレートによるフィルタリング

トリガーテンプレートモードでは、結果は以下にリストされている 1 つまたは複数のパラメーターでフィルター処理できます。

パラメータ	詳細
テンプレートグループ	そのグループに属するテンプレートからトリガーを持つすべてのホストを選択します。少なくとも 1 つのテンプレートをを含む任意のホストグループを選択できます。
テンプレート	選択したテンプレートとすべてのネストされたテンプレートから、トリガーのあるホストを選択します。選択したテンプレートから継承されたトリガーのみが表示されます。ネストされたテンプレートに追加の独自のトリガーがある場合、それらのトリガーは表示されません。
//テンプレートトリガー//	選択したトリガーを持つホストを選択します。選択したホストの他のトリガーは表示されません。
ホストグループ	グループに属するホストを選択します。

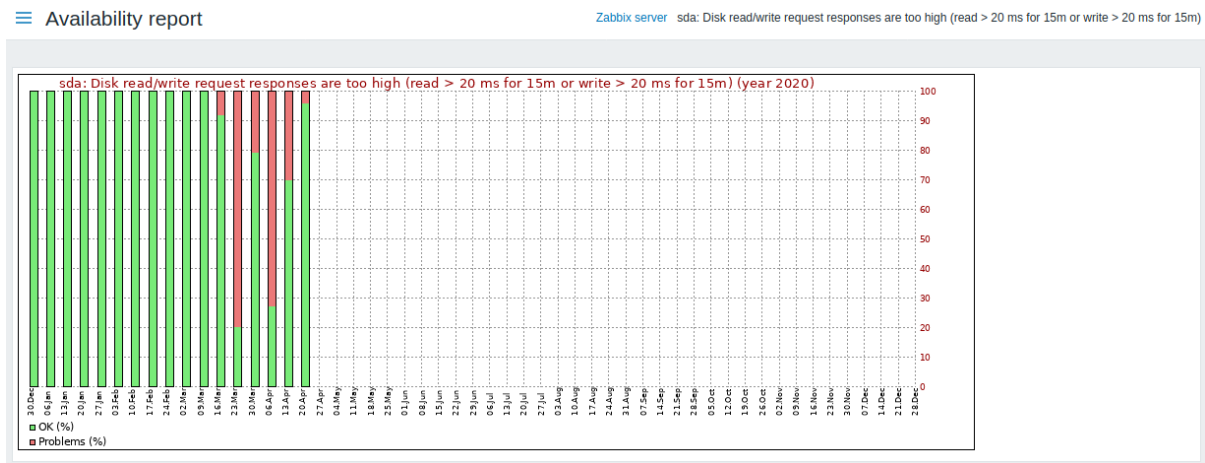
ホストによるフィルタリング

ホストモードでは、結果をホストまたはホストグループでフィルタリングできます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。

期間セクター

期間セクターを使用すると、マウスを1回クリックするだけでフィルタ期間を選択できます。期間セクターは、フィルターのある期間タブをクリックして開くことができます。

グラフ列の表示をクリックすると、棒グラフが表示されます。この棒グラフは現在年を各1週間ごとに表しています。



バーの緑色の部分は正常期間を表し、赤色は障害期間を表します。

4 障害発生数上位 100 項目

概要

レポート → 障害発生数上位 100 項目で、評価期間内に最も頻繁に状態を変更したトリガーを、ステータス変更の数でソートして確認できます。

100 busiest triggers

Host groups: type here to search [Select]

Hosts: type here to search [Select]

Severity: Not classified Warning High Information Average Disaster

[Apply] [Reset]

Host	Trigger	Severity	Number of status changes
New host	CPU load too high on New host for 3 minutes	Warning	92
Zabbix server	Disk I/O is overloaded on Zabbix server	Warning	88
New host	Disk I/O is overloaded on New host	Warning	82
New host	New host has just been restarted	Information	19
Zabbix server	Zabbix server has just been restarted	Information	19
Zabbix server	Lack of free swap space on Zabbix server	Warning	16
New host	Lack of free swap space on New host	Warning	12
New host	Zabbix agent on New host is unreachable for 5 minutes	Average	8
Zabbix server	Zabbix agent on Zabbix server is unreachable for 5 minutes	Average	8
New host	/etc/passwd has been changed on New host	Warning	4

ホスト列とトリガー列のエントリは、いくつかの便利なオプションを提供するリンクです。

- ホストの場合 - ホストのユーザー定義スクリプト、最新データ、インベントリ、グラフ、およびダッシュボードへのリンク
- トリガーの場合 - 最新のイベント、トリガー構成フォーム、および単純なグラフへのリンク

フィルタの使用

フィルタを使用して、ホストグループ、ホスト、またはトリガーの重大度ごとにトリガーを表示できます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタは障害発生数上位 100 項目バーの下にあります。フィルタータブをクリックすると、開いたり折りたたんだりできます。

期間セクター

期間セクターを使用すると、マウスを 1 回クリックするだけでフィルタ期間を選択できます。期間セクターは、フィルターの横にある期間タブをクリックして開くことができます。

5 監査

概要

レポート → 監査セクションで、フロントエンドで行われた変更の記録を表示できます。

Note:

監査レコードを表示するには、管理設定で監査ログを有効にする必要があります。ロギングが無効になっている場合、フロントエンドの変更の履歴はデータベースに記録されず、監査レコードを表示できません。

Audit log

Time	User	IP	Resource	ID	Action	Recordset ID	Details
2022-05-30 12:07:34	Admin	127.0.0.1	User	4	Update	cl3sicbqq0000z8ep87xz41zs	Description: Database manager user.lang: default => en_GB
2022-05-30 12:07:13	Admin	127.0.0.1	User	1	Login	cl3sibvqn0000z8ep40q8w1k	
2022-05-30 12:07:13	guest	127.0.0.1	User	2	Failed login	cl3sibvqn0000z8ep40q8w1k	
2022-05-30 12:07:12	guest	127.0.0.1	User	2	Failed login	cl3sibvem0000z8epvfm1xtzi	

監査ログには、次のデータが表示されます。

列	詳細
時間	監査レコードのタイムスタンプ
ユーザー	アクティビティを実行したユーザー
IP アドレス	アクティビティが開始された IP
リソース	影響を受けるリソースタイプ (ホスト、ホストグループなど)
アクション	アクティビティタイプ: ログイン、ログアウト、追加、更新、削除、有効、無効
ID	リソース ID。ハイパーリンクをクリックすると、このリソース ID で監査ログレコードがフィルタリングされます。
記録 ID	同じフロントエンド操作の結果として作成されたすべての監査ログレコードの共有 ID。たとえば、テンプレートをホストにリンクすると、継承されたテンプレートエンティティ (アイテム、トリガーなど) ごとに個別の監査ログレコードが作成されます。これらのレコードはすべて同じレコードセット ID を持ちます。ハイパーリンクをクリックするとこのレコードセット ID で監査ログレコードがフィルタリングされます。
詳細	リソースの説明と実行されたアクティビティに関する詳細情報。レコードに 3 つ以上の行が含まれている場合は、追加のリンク「詳細」が表示されます。変更の完全なリストを表示するには、このリンクをクリックしてください。

フィルタの使用

フィルタは監査ログバーの下にあります。右上隅のフィルタータブをクリックすると、開いたり折りたたんだりできます。

フィルタを使用して、ユーザー、影響を受けるリソース、リソース ID、およびフロントエンド操作 (レコードセット ID) ごとにレコードを絞り込むことができます。リソースのアクション (追加、更新、削除など) を選択することもできます。Zabbix 6.0.5 以降では複数のアクションを選択できます。

検索パフォーマンスを向上させるために、すべてのデータは未解決のマクロで検索されます。

期間セクター

期間セクターを使用すると、マウスを1回クリックするだけでフィルタ期間を選択できます。期間セクターは、フィルターのある期間タブをクリックして開くことができます。

6 アクションログ

概要

レポート → アクションログセクションで、ユーザーはアクション内で実行された操作（通知、リモートコマンド）の詳細を表示できます。

Time	Action	Type	Recipient	Message	Status	Info
2020-06-09 15:47:16	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) marina.generalova@zabbix.com	Subject: Resolved in 2m: High CPU utilization (over 75% for 5m) Message: Problem has been resolved at 15:47:13 on 2020.06.09 Problem name: High CPU utilization (over 75% for 5m) Problem duration: 2m Host: Zabbix server Severity: Warning Original problem ID: 1287	Sent	
2020-06-09 15:44:40	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) marina.generalova@zabbix.com	Subject: Resolved in 3m: Zabbix agent is not available (for 1m) Message: Problem has been resolved at 15:44:37 on 2020.06.09 Problem name: Zabbix agent is not available (for 1m) Problem duration: 3m Host: Zabbix server Severity: Average Original problem ID: 1286	Sent	

表示情報

列	詳細
時間	操作のタイムスタンプ
アクション	操作の原因となるアクションの名前が表示されます
タイプ	操作タイプが表示されます - E メールまたはコマンド
送信先	通知受信者のユーザー名、名前、苗字（括弧内）、および電子メールアドレスが表示されます。
メッセージ	メッセージ/リモートコマンドの内容が表示されます。 リモートコマンドは、コロン記号 <host>:<command> でターゲットホストから区切られます。リモートコマンドが Zabbix サーバーで実行される場合、情報の形式は次のとおりです。'Zabbix サーバー:< コマンド >
ステータス	動作ステータスが表示されます： 実行中 - アクションが実行中です 実行中のアクションの場合、サーバーが通知の送信を試行する残りの再試行回数 が表示されます。 送信済み - 通知が送信されました 実行済み - コマンドが実行されました 送信されていません - アクションが完了していません。
情報	アクションの実行に関するエラー情報（存在する場合）が表示されます。

フィルターの使用

フィルタを使用して、メッセージの受信者ごとにレコードを絞り込むことができます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタはアクションログバーの下にあります。フィルタータブをクリックすると、開いたり折りたたんだりできます。

期間セクター

期間セクターを使用すると、マウスを1回クリックするだけでフィルタ期間を選択できます。期間セクターは、フィルターのある期間タブをクリックして開くことができます。

7 通知レポート

概要

レポート → 通知レポートセクションには、各ユーザーに送信された通知の数に関するレポートが表示されます。

右上隅のドロップダウンから、送信される通知のメディアタイプ（またはすべて）、期間（各日/週/月/年のデータ）、および年を選択できます。

☰ Notifications Media type: all Period: Monthly Year: 2020

Month	Admin (Zabbix Administrator)	Database manager	guest	user (New User)
January				
February				
March				
April				48
May				568

各列にはシステムユーザーごとの合計が表示されます。

5 設定

概要

設定メニューには、ホストとホストグループ、データ収集、データしきい値、問題通知の送信、データの視覚化の作成など、Zabbixの主要な機能を設定するためのセクションが含まれています。

1 アイテム

概要

テンプレートのアイテムリストには、設定 → テンプレートからそれぞれのテンプレートのアイテムをクリックしてアクセスできます。

既存のアイテムのリストが表示されます。

☰ Items Create item

All templates / Template OS Linux by Zabbix agen... Items 41 Triggers 14 Graphs 8 Dashboards 1 Discovery rules 3 Web scenarios Filter

<input type="checkbox"/>	Name	Triggers	Key ▲	Interval	History	Trends	Type	Status	Tags
<input type="checkbox"/>	... Template Module Zabbix agent active: Host name of Zabbix agent running		agent.hostname	1h	7d		Zabbix agent (active)	Enabled	Application: Monitorin...
<input type="checkbox"/>	... Template Module Zabbix agent active: Zabbix agent ping	Triggers 1	agent.ping	1m	7d	365d	Zabbix agent (active)	Enabled	Application: Status
<input type="checkbox"/>	... Template Module Zabbix agent active: Version of Zabbix agent running		agent.version	1h	7d		Zabbix agent (active)	Enabled	Application: Monitorin...
<input type="checkbox"/>	... Template Module Linux generic by Zabbix agent active: Maximum number of open file descriptors	Triggers 1	kernel.maxfiles	1h	7d	365d	Zabbix agent (active)	Enabled	Application: General
<input type="checkbox"/>	... Template Module Linux generic by Zabbix agent active: Maximum number of processes	Triggers 2	kernel.maxproc	1h	7d	365d	Zabbix agent (active)	Enabled	Application: General
<input type="checkbox"/>	... Template Module Linux generic by Zabbix agent active: Number of processes	Triggers 1	proc.num	1m	7d	365d	Zabbix agent (active)	Enabled	Application: General
<input type="checkbox"/>	... Template Module Linux generic by Zabbix agent active: Number of running processes		proc.num[.,run]	1m	7d	365d	Zabbix agent (active)	Enabled	Application: General

表示情報

列	詳細
アイテムメニュー	<p>三点リーダーのアイコンをクリックして、このアイテムのオプションメニューを開きます。</p> <p>トリガーの作成 - このアイテムに基づいてトリガーを作成します</p> <p>トリガー - クリックしてこのアイテムの構成済みトリガーへのリンク付きリストを表示します</p> <p>依存アイテムの作成 - このアイテムの依存アイテムの作成</p> <p>依存するディスカバリルールの作成 - このアイテムの依存ディスカバリルールの作成</p>
テンプレート	<p>アイテムが属するテンプレート。</p> <p>この列は、フィルターで複数のテンプレートが選択されている場合にのみ表示されます。</p>
名前	<p>アイテムの詳細へのリンクとして表示されるアイテムの名前。</p> <p>アイテム名をクリックすると、アイテム設定フォームが開きます。</p> <p>アイテムが継承されている場合別のテンプレートからは、テンプレート名がアイテム名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのアイテムリストが開きます。</p>
トリガー	<p>トリガーにマウスオーバーすると、アイテムに関連付けられているトリガーを表示するインフォボックスが表示されます。</p> <p>トリガーの数は灰色で表示されます。</p>
キー	<p>アイテムキーの表示</p>
監視間隔	<p>監視間隔の表示</p>
履歴	<p>アイテムデータ履歴が保持される日数を表示</p>
トレンド	<p>アイテムのトレンド履歴が保持される日数を表示</p>
タイプ	<p>アイテムタイプの表示 (Zabbix エージェント, SNMP エージェント, シンプルチェック, etc).</p>
ステータス	<p>アイテムのステータスが表示されます - 有効または無効。ステータスをクリックすると、ステータスを有効から無効 (およびその逆) に変更できます。</p>
タグ	<p>アイテムタグが表示されます。</p> <p>最大 3 つのタグ (名前: 値のペア) を表示できます。タグがさらにある場合は、"..." リンクが表示され、マウスオーバーですべてのタグを表示できます。</p>

新しいアイテムを設定するには、右上隅にあるアイテムの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - アイテムのステータスを有効にします
- 無効 - アイテムのステータスを無効にします
- コピー - アイテムを他のホストまたはテンプレートにコピーします
- 一括更新 - 一度に複数のアイテムの複数のプロパティを更新
- 削除 - アイテムを削除します

これらのオプションを使用するには、それぞれの項目の先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

アイテムリストには多くのアイテムが含まれている場合があります。フィルタを使用すると、それらの一部を除外して、探しているアイテムをすばやく見つけることができます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターアイコンは右上隅にあります。クリックすると、フィルターが開き、目的のフィルター基準を指定できます。

All templates / Linux by Zabbix agent Items 42 Triggers 14 Graphs 3 Dashboards 2 Discovery rules 3 Web scenarios Filter

Template groups type here to search Select Type all

Templates Linux by Zabbix agent X Select Type of information all

Name type here to search History

Key Update interval

Value mapping type here to search Select

Tags And/Or Or tag Contains value Remove

Add

Status all Enabled Disabled

Triggers all Yes No

Inherited all Yes No

Apply Reset

Subfilter affects only filtered data

TAGS
component: application 1 component: cpu 17 component: environment 2 component: memory 7 component: os 3 component: storage 3 component: system 12

TYPES
Dependent item 2 Zabbix agent 39 Zabbix internal 1

TYPE OF INFORMATION
Character 7 Numeric (float) 19 Numeric (unsigned) 15 Text 1

WITH TRIGGERS
Without triggers 23 With triggers 19

HISTORY
1w 36 2w 6

TRENDS
0 1 52w 1d 33

INTERVAL
30s 1 1m 28 15m 3 1h 8

パラメータ

詳細

ホストグループ

1つ以上のホストグループでフィルタリングします。
少なくとも1つのテンプレートを含まないホストグループのみを選択できます。

テンプレート

名前

親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。

キー

1つ以上のテンプレートでフィルタリングします。

値のマッピング

アイテム名でフィルタリングします。

アイテムキーでフィルタリングします。

使用する値マップでフィルタリングします。

Templates オプションが空の場合、このパラメータは表示されません。

タイプ

アイテムタイプでフィルタリングします (Zabbix エージェント, SNMP エージェント, 他)。

データ型

データ型でフィルタリングします (Numeric unsigned, float, etc.)。

履歴

アイテム履歴が保持される期間でフィルタリングします。

トレンド

アイテムのトレンドが保持される期間でフィルタリングします。

監視間隔

アイテムの更新間隔でフィルタリングします。

タグ

表示項目を制限するタグを指定してください。特定のタグとタグ値を含めたり除外したりするためにいくつかの条件を設定できます。タグ名の照合では、常に大文字と小文字が区別されます。条件ごとに使用できる演算子がいくつかあります。

Exists - 指定されたタグ名を含める

Equals - 指定されたタグ名を含める (大文字と小文字を区別)

Contains - タグ値に入力された文字列が含まれるタグ名を含める (部分文字列の一致、大文字と小文字を区別しない)

Does not exist - 指定されたタグ名と値を除外する

Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)

Does not contain - タグ値に入力された文字列が含まれるタグ名を除外する (サブstringの一致、大文字と小文字を区別しない)

条件には2つの計算タイプがあります:

** And/Or - すべての条件が満たされる必要があります、同じタグ名を持つ条件は **Or** 条件によってグループ化されます **
Or**** - どれか1つの条件が満たされた場合

ステータス

アイテムのステータスでフィルタリングします - 有効または無効

トリガー

トリガーあり (またはなし) のアイテムをフィルタリングします。

継承したもの

リンクされたテンプレートから継承された (または継承されていない) アイテムをフィルタリングします。

フィルタの下サブフィルターは、さらにフィルタリングオプションを提供します (すでにフィルタリングされたデータの場合)。共

通常のパラメータ値を持つアイテムのグループを選択できます。グループをクリックすると、グループが強調表示され、このパラメータ値を持つアイテムのみがリストに残ります。

2 トリガー

概要

テンプレートのトリガーリストには、設定 → テンプレートからテンプレートのトリガーをクリックしてアクセスできます。

Severity	Name	Operational data	Expression	Status	Tags
Information	Template Module Linux generic by Zabbix agent: /etc/passwd has been changed Depends on: Linux OS agent: Operating system description has changed Linux OS agent: System name has changed (new name: {ITEM.VALUE})		(last(/Linux OS agent/vfs.file.cksum[/etc/passwd],#1)->last(/Linux OS agent/vfs.file.cksum[/etc/passwd],#2))>0	Enabled	
Information	Template Module Linux generic by Zabbix agent: Configured max number of open file descriptors is too low (< {SKERNEL.MAXFILES.MIN})		last(/Linux OS agent/kernel.maxfiles)<{SKERNEL.MAXFILES.MIN}	Enabled	
Information	Template Module Linux generic by Zabbix agent: Configured max number of processes is too low (< {SKERNEL.MAXPROC.MIN}) Depends on: Linux OS agent: Getting closer to process limit (over 80% used)		last(/Linux OS agent/kernel.maxproc)<{SKERNEL.MAXPROC.MIN}	Enabled	
Warning	Template Module Linux generic by Zabbix agent: Getting closer to process limit (over 80% used)	{ITEM.LASTVALUE1} active, {ITEM.LASTVALUE2} limit.	last(/Linux OS agent/proc.num)/last(/Linux OS agent/kernel.maxproc)*100>80	Enabled	
Warning	Template Module Linux CPU by Zabbix agent: High CPU utilization (over {SCPU.UTIL.CRIT}% for 5m) Depends on: Linux OS agent: Load average is too high (per CPU load over {SLOAD_AVG_PER_CPU.MAX.WARN} for 5m)	Current utilization: {ITEM.LASTVALUE1}	min(/Linux OS agent/system.cpu.util,5m)>{SCPU.UTIL.CRIT}	Enabled	

表示情報

列	詳細
深刻度 テンプレート	トリガーの深刻度は名前とセルの背景色の両方で表示されます。 トリガーが属するテンプレート。 この列は、フィルターで複数のテンプレートが選択されている場合のみ表示されます。
名前	トリガー詳細へのリンクとして表示されるトリガーの名前。 トリガー名をクリックすると、トリガー設定フォームが開きます。 トリガーが継承されている場合別のテンプレートからは、テンプレート名がトリガー名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルでトリガーリストが開きます。
運用データ	監視データ → 障害で動的に解決される任意の文字列とマクロを含む、トリガーの操作データ定義。
条件式	トリガー式が表示されます。式のテンプレートアイテム部分は、アイテム設定フォームにつながるリンクとして表示されます。
ステータス	トリガーステータスが表示されます - 有効または無効。ステータスをクリックすると、ステータスを有効から無効（およびその逆）に変更できます。
タグ	トリガーにタグが含まれている場合、タグの名前と値がこの列に表示されます。

新しいトリガーを構成するには、右上隅にあるトリガーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - トリガーのステータスを有効に
- 無効 - トリガーのステータスを無効に
- コピー - トリガーを他のホストまたはテンプレートにコピーします
- 一括更新 - 一度に複数のトリガーのいくつかのプロパティを更新します
- 削除 - トリガーの削除

これらのオプションを使用するには、トリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のトリガーのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターアイコンは右上隅にあります。クリックすると、フィルターが開き、目的のフィルター基準を設定できます。

パラメータ	詳細
ホストグループ	1つ以上のホストグループでフィルタリングします。 少なくとも1つのテンプレートを含むホストグループのみを選択できます。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
テンプレート	1つ以上のテンプレートでフィルタリングします。 上記でホストグループがすでに選択されている場合、テンプレートの選択はそれらのグループに限定されます。
名前	トリガー名でフィルタリングします
深刻度	1つまたは複数のトリガーの深刻度でフィルタリングします
ステータス	ステータスでフィルタリングします
タグ	トリガータグの名前と値でフィルタリングします。特定のタグとタグ値を含めたり除外したりするためにいくつかの条件を設定できます。タグ名の照合では、常に大文字と小文字が区別されます。条件ごとに使用できる演算子がいくつかあります。 Exists - 指定されたタグ名を含める Equals - 指定されたタグ名を含める（大文字と小文字を区別） Contains - タグ値に入力された文字列が含まれるタグ名を含める（部分文字列の一致、大文字と小文字を区別しない） Does not exist - 指定されたタグ名と値を除外する Does not equal - 指定されたタグ名と値を除外する（大文字と小文字を区別） Does not contain - タグ値に入力された文字列が含まれるタグ名を除外する（サブストリングの一致、大文字と小文字を区別しない） 条件には2つの計算タイプがあります： ** And/Or - すべての条件が満たされる必要があります、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or** - どれか1つの条件が満たされた場合 マクロと マクロ関数 は、タグ名とタグ値のフィールドでサポートされています。
継承したもの	リンクされたテンプレートから継承された（または継承されなかった）トリガーをフィルターします。
依存関係がある	依存関係がある（またはない）トリガーをフィルターします。

3 グラフ

概要

テンプレートのカスタムグラフリストには、設定 → テンプレートからテンプレートのグラフをクリックしてアクセスできます。既存のグラフのリストが表示されます。

All templates / Template App Zabbix Server Applications 1 Items 46 Triggers 34 Graphs 6 Dashboards 1 Discovery rules Web scenarios			
Name ▲	Width	Height	Graph type
<input type="checkbox"/> Value cache effectiveness	900	200	Stacked
<input type="checkbox"/> Zabbix cache usage, % used	900	200	Normal
<input type="checkbox"/> Zabbix data gathering process busy %	900	200	Normal
<input type="checkbox"/> Zabbix internal process busy %	900	200	Normal
<input type="checkbox"/> Zabbix internal queues	900	200	Normal
<input type="checkbox"/> Zabbix server performance	900	200	Normal

表示情報

列	詳細
テンプレート	グラフが属するテンプレート。 この列は、フィルターで複数のテンプレートが選択されている場合にのみ表示されます。
名前	カスタムグラフの名前。グラフの詳細へのリンクとして表示されます。 グラフ名をクリックすると、 グラフ設定フォーム が開きます。 グラフは別のテンプレートから継承され、テンプレート名はグラフ名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのグラフリストが開きます。
幅	グラフの表示幅
高さ	グラフの表示高さ
グラフのタイプ	グラフの種類が表示されます - ノーマル,* 積算グラフ, 円グラフまたは分解円グラフ *

新しいグラフを設定するには、右上隅にあるグラフの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- コピー - グラフを他のホストまたはテンプレートにコピーします
- 削除 - グラフの削除

これらのオプションを使用するには、それぞれのグラフ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

ホストグループとテンプレートでグラフをフィルタリングできます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

4 ディスカバリルール

概要

テンプレートの低レベルディスカバリルールのリストには、構成 → テンプレートからそれぞれのテンプレートのディスカバリをクリックしてアクセスできます。

既存の低レベルディスカバリルールのリストが表示されます。テンプレートとは別にすべてのディスカバリルールを表示したり、フィルター設定を変更することで特定のホストグループのすべてのディスカバリルールを表示したりすることもできます。

Template	Name	Items	Triggers	Graphs	Hosts	Key	Interval	Type	Status	
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Array Controller Cache Discovery	Item prototypes 1	Trigger prototypes 2	Graph prototypes	Host prototypes	array.cache.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Array Controller Discovery	Item prototypes 2	Trigger prototypes 3	Graph prototypes	Host prototypes	array.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	FAN Discovery	Item prototypes 1	Trigger prototypes 2	Graph prototypes	Host prototypes	fan.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Physical Disk Discovery	Item prototypes 4	Trigger prototypes 2	Graph prototypes	Host prototypes	physicalDisk.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	PSU Discovery	Item prototypes 1	Trigger prototypes 2	Graph prototypes	Host prototypes	psu.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Temperature CPU Discovery	Item prototypes 1	Trigger prototypes 3	Graph prototypes	Host prototypes	temp.cpu.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Temperature Discovery	Item prototypes 4	Trigger prototypes 12	Graph prototypes	Host prototypes	temp.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Unit Discovery	Item prototypes 3	Trigger prototypes 3	Graph prototypes	Host prototypes	unit.discovery	1h	SNMP agent	Enabled
<input type="checkbox"/>	Template Server Cisco UCS SNMPv2	Virtual Disk Discovery	Item prototypes 3	Trigger prototypes 1	Graph prototypes	Host prototypes	virtualdisk.discovery	1h	SNMP agent	Enabled

0 selected Enable Disable Delete

表示情報

列	詳細
テンプレート 名前	ディスカバリルールが属するテンプレート リンクとして表示されるルールの名前。 ルール名をクリックすると、低レベルのディスカバリルール設定フォームが開きます。 ディスカバリルールが継承されている場合。別のテンプレートからは、テンプレート名がルール名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルでディスカバリルールリストが開きます。
アイテム	アイテムのプロトタイプのリストへのリンク。 既存のアイテムのプロトタイプの数で灰色で表示されます。
トリガー	トリガープロトタイプのリストへのリンク。 既存のトリガープロトタイプの数で灰色で表示されます。
グラフ	グラフプロトタイプリストへのリンク。 既存のグラフプロトタイプの数で灰色で表示されます。
ホスト	ホストプロトタイプリストへのリンク。 既存のホストプロトタイプの数で灰色で表示されます。
キー 監視間隔 タイプ	ディスカバリに使用したアイテムキーが表示されます。 ディスカバリを実行する頻度が表示されます。 ディスカバリに使用したアイテムタイプ (Zabbix エージェント、SNMP エージェントなど) が表示されます。
ステータス	ディスカバリルールのステータスが表示されます - 有効または無効。ステータスをクリックすると、ステータスを有効から無効 (およびその逆) に変更できます。

新しい低レベルのディスカバリルールを構成するには、右上隅にあるディスカバリルールの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - 低レベルのディスカバリルールのステータスを有効に変更します
- 無効 - 低レベルのディスカバリルールのステータスを無効に変更します
- 削除 - 低レベルのディスカバリルールを削除します

これらのオプションを使用するには、ディスカバリルール先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のディスカバリルールのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターアイコンは右上隅にあります。それをクリックすると、フィルターが開き、テンプレート、ディスカバリルール名、アイテムキー、アイテムタイプなどの目的のフィルター基準を指定できます。

All templates / VMware Hypervisor Items 21 Triggers Graphs Dashboards Discovery rules 1 Web scenarios Filter

Host groups Select Type: all Status: all Enabled Disabled

Templates: VMware Hypervisor Select Update interval: Keep lost resources period:

Name: Key:

Apply Reset

パラメータ	詳細
ホストグループ	1つ以上のホストグループでフィルタリングします。 少なくとも1つのテンプレートを含まないホストグループのみを選択できます。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
テンプレート 名前 キー タイプ 監視間隔	1つ以上のテンプレートでフィルタリングします。 ディスカバリルール名でフィルタリングします ディスカバリアイテムキーでフィルタリングします。 ディスカバリアイテムタイプでフィルタリングします 更新間隔でフィルタリングします。 Zabbix トラッパーおよび依存アイテムでは使用できません。
存在しなくなったリソースの保持期間 ステータス	存在しなくなったリソースの保持期間でフィルタリングします。 ディスカバリルールのステータスを (すべて/有効/無効) でフィルタリングします。

1 アイテムのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスカバリルールの構成済みプロトタイプアイテムが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプアイテムは、低レベルディスカバリ中に実際のホスト items を作成するための基礎になります。

Item prototypes

[Create item prototype](#)

All templates / Template Module Linux filesystems... Discovery list / Mounted filesystem discovery

Item prototypes 4 Trigger prototypes 4 Graph prototypes 1 Host prototypes

<input type="checkbox"/>	Name ▲	Key	Interval	History	Trends	Type	Create enabled	Discover	Tags
<input type="checkbox"/>	... {#FSNAME}: Free inodes in %	vfs.fs.inode[{#FSNAME}.pfree]	1m	7d	365d	Zabbix agent	Yes	Yes	Application: Filesystem...
<input type="checkbox"/>	... {#FSNAME}: Space utilization	vfs.fs.size[{#FSNAME}.pused]	1m	7d	365d	Zabbix agent	Yes	Yes	Application: Filesystem...
<input type="checkbox"/>	... {#FSNAME}: Total space	vfs.fs.size[{#FSNAME}.total]	1m	7d	365d	Zabbix agent	Yes	Yes	Application: Filesystem...
<input type="checkbox"/>	... {#FSNAME}: Used space	vfs.fs.size[{#FSNAME}.used]	1m	7d	365d	Zabbix agent	Yes	Yes	Application: Filesystem...

Displaying 4 of 4 found

0 selected [Create enabled](#) [Create disabled](#) [Mass update](#) [Delete](#)

表示情報

列	詳細
名前	リンクとして表示されるプロトタイプアイテムの名前。 名前をクリックすると、アイテムのプロトタイプ設定フォームが開きます。 アイテムのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名は、アイテム名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでアイテムのプロトタイプリストが開きます。
キー	プロトタイプアイテムのキー
監視間隔	監視実行頻度
ヒストリ	アイテムデータ履歴を保持する日数
トレンド	アイテムのトレンド履歴を保持する日数
タイプ	プロトタイプアイテムのタイプ設定 (Zabbix エージェント、SNMP エージェント、簡易チェックなど)

列	詳細
自動生成時のステータス有効	プロトタイプアイテムのステータス設定 はい - 有効
ディスカバリ	いいえ - 無効。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプアイテムのディスカバリ設定 はい - ディスカバリする いいえ - ディスカバリしない。クリックで「はい」と「いいえ」を切り替えることができます。
タグ	プロトタイプアイテムのタグが表示されます。

新しいプロトタイプアイテムを構成するには、右上隅にあるアイテムのプロトタイプを作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - アイテムを有効として作成します
- 自動生成無効 - アイテムを無効として作成します
- 一括更新 - プロトタイプアイテムを一括更新します
- 削除 - プロトタイプアイテムを削除する

これらのオプションを使用するには、プロトタイプアイテムの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

2 トリガーのプロトタイプ

概要

このセクションでは、テンプレートに設定された低レベルディスカバリルールのプロトタイプトリガーが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプトリガーは、低レベルのディスカバリ中に実際のホストトリガーを作成するための基礎になります。

☰ Trigger prototypes
Create trigger prototype

All templates / Template Module Linux filesystems... Discovery list / Mounted filesystem discovery

Item prototypes 4 Trigger prototypes 4 Graph prototypes 1 Host prototypes

<input type="checkbox"/>	Severity Name ▲	Operational data	Expression	Create enabled	Discover	Tags
<input type="checkbox"/>	Average {#FSNAME}: Disk space is critically low (used > {SVFS.FS.PUSED.MAX.CRIT:"{#FSNAME}"}%)	Space used: (ITEM.LASTVALUE3) of (ITEM.LASTVALUE2) ((ITEM.LASTVALUE1))	last (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.pused])>{SVFS.FS.PUSED.MAX.CRIT:"{#FSNAME}"} and ((last (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.total]) last (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.used])<5G or timeleft (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.pused],1h,100)<1d)	Yes	Yes	
<input type="checkbox"/>	Warning {#FSNAME}: Disk space is low (used > {SVFS.FS.PUSED.MAX.WARN:"{#FSNAME}"}%) Depends on: Template Module Linux filesystems by Zabbix agent: {#FSNAME}: Disk space is critically low (used > {SVFS.FS.PUSED.MAX.CRIT:"{#FSNAME}"}%)	Space used: (ITEM.LASTVALUE3) of (ITEM.LASTVALUE2) ((ITEM.LASTVALUE1))	last (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.pused])>{SVFS.FS.PUSED.MAX.WARN:"{#FSNAME}"} and ((last (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.total]) last (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.used])<10G or timeleft (/Template Module Linux filesystems by Zabbix agent/vfs.fs.size[{#FSNAME}.pused],1h,100)<1d)	Yes	Yes	
<input type="checkbox"/>	Average {#FSNAME}: Running out of free inodes (free < {SVFS.FS.INODE.PFREE.MIN.CRIT:"{#FSNAME}"}%)	Free inodes: (ITEM.LASTVALUE1)	min (/Template Module Linux filesystems by Zabbix agent/vfs.fs.inode[{#FSNAME}.pfree],5m)<{SVFS.FS.INODE.PFREE.MIN.CRIT:"{#FSNAME}"})	Yes	Yes	
<input type="checkbox"/>	Warning {#FSNAME}: Running out of free inodes (free < {SVFS.FS.INODE.PFREE.MIN.WARN:"{#FSNAME}"}%) Depends on: Template Module Linux filesystems by Zabbix agent: {#FSNAME}: Running out of free inodes (free < {SVFS.FS.INODE.PFREE.MIN.CRIT:"{#FSNAME}"}%)	Free inodes: (ITEM.LASTVALUE1)	min (/Template Module Linux filesystems by Zabbix agent/vfs.fs.inode[{#FSNAME}.pfree],5m)<{SVFS.FS.INODE.PFREE.MIN.WARN:"{#FSNAME}"})	Yes	Yes	

Displaying 4 of 4 found

0 selected Create enabled Create disabled Mass update Delete

表示情報

列	詳細
名前	リンクとして表示されるプロトタイプトリガーの名前。 名前をクリックすると、トリガーのプロトタイプ設定フォームが開きます。 トリガーのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名は、アイテム名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでトリガーのプロトタイプリストが開きます。

列	詳細
運用データ	監視データ → 障害で動的に解決される任意の文字列とマクロを含む、トリガーの操作データ形式が表示されます。
自動生成時のステータス有効	プロトタイプトリガーのステータス設定 はい - 有効
ディスカバリ	いいえ - 無効。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプトリガーのディスカバリ設定 はい - ディスカバリする
タグ	いいえ - ディスカバリしない。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプトリガーのタグが表示されます。

新しいプロトタイプトリガーを構成するには、右上隅にあるトリガーのプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - トリガーを有効として作成します
- 自動生成無効 - トリガーを無効として作成します
- 一括更新 - プロトタイプトリガーを一括更新します
- 削除 - プロトタイプトリガーを削除する

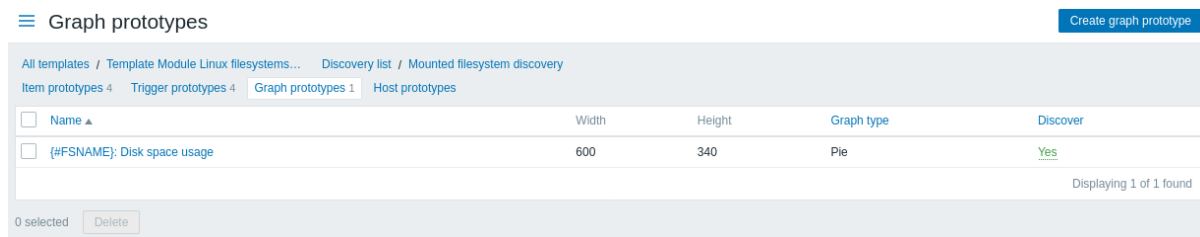
これらのオプションを使用するには、プロトタイプトリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

3 グラフのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスカバリルールの構成済みプロトタイプグラフが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプグラフは、低レベルディスカバリ中に実際のホスト **items** を作成するための基礎になります。



表示情報

列	詳細
名前	リンクとして表示されるプロトタイプグラフの名前。 名前をクリックすると、グラフのプロトタイプ設定フォームが開きます。 グラフのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名はグラフ名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでグラフのプロトタイプリストが開きます。
幅	プロトタイプグラフの表示幅
高さ	プロトタイプグラフの表示高
グラフのタイプ	プロトタイプグラフのタイプ設定 - ノーマル, 積算グラフ, 円グラフ, 分解円グラフ
ディスカバリ	プロトタイプグラフのディスカバリ設定 はい - ディスカバリする いいえ - ディスカバリしない。クリックで「はい」と「いいえ」を切り替えることができます。

新しいプロトタイプグラフを構成するには、右上隅にあるグラフのプロトタイプを作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 削除 - プロトタイプグラフを削除する

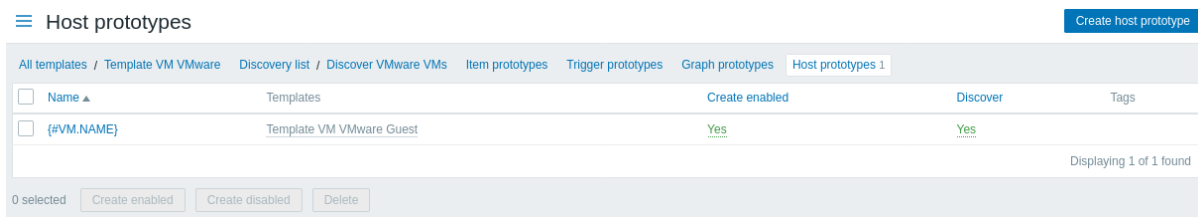
これらのオプションを使用するには、プロトタイプグラフの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

4 ホストのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスクカバリルールの構成済みプロトタイプホストが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプホストは、低レベルディスクカバリ中に実際のホスト **items** を作成するための基礎になります。



表示情報

列	詳細
名前	リンクとして表示されるプロトタイプホストの名前。 名前をクリックすると、ホストのプロトタイプ設定フォームが開きます。 ホストのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名は、ホスト名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでホストのプロトタイプリストが開きます。
テンプレート 自動生成時のステータス有効	プロトタイプホストのテンプレート表示 プロトタイプホストのステータス設定 はい - 有効
ディスクバリ	いいえ - 無効。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプホストのディスクバリ設定 はい - ディスカバリする いいえ - ディスカバリしない。クリックで「はい」と「いいえ」を切り替えることができます。
タグ	プロトタイプホストのタグ表示

新しいプロトタイプホストを構成するには、右上隅にあるホストのプロトタイプを作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - ホストを有効として作成します
- 自動生成無効 - ホストを無効として作成します
- 削除 - プロトタイプホストを削除する

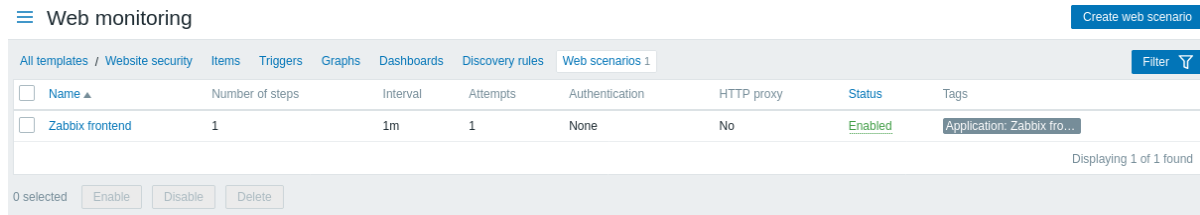
これらのオプションを使用するには、プロトタイプホストの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

5 Web シナリオ

概要

テンプレートの Web シナリオリストには、設定 → テンプレートからそれぞれのテンプレートの Web をクリックしてアクセスできます。

既存の Web シナリオのリストが表示されます。



表示情報

列	詳細
名前	Web シナリオの名前。Web シナリオ名をクリックすると、Web シナリオ設定フォームが開きます。 Web シナリオが別のテンプレートから継承されている場合、テンプレート名は Web シナリオ名の前に灰色で表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルで Web シナリオリストが開きます。
ステップ数	シナリオに含まれるステップ数
監視間隔	シナリオが実行される頻度
試行回数	Web シナリオステップを実行するための試行回数
認証	認証方法が表示されます - Basic, NTLM はなし
HTTP プロキシ	HTTP プロキシを表示します。使用しない場合は「いいえ」を表示します。
ステータス	Web シナリオのステータスが表示されます - 有効または無効。 ステータスをクリックすると、ステータスを変更できます。
タグ	Web シナリオタグが表示されます。 最大 3 つのタグ (名前: 値のペア) を表示できます。タグがさらにある場合は「...」リンクが表示され、マウスオーバーですべてのタグを表示できます。

新しい Web シナリオを構成するには、右上隅にある Web シナリオの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

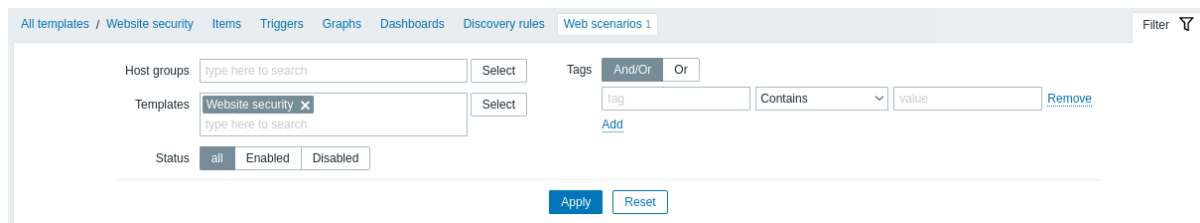
- 有効 - シナリオのステータスを有効にします
- 無効 - シナリオのステータスを無効にします
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、Web シナリオ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のシナリオのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、Web シナリオのリストの上にあります。それをクリックすると、ホストグループ、テンプレート、ステータス、タグでシナリオをフィルタリングできるフィルターが利用可能になります。



1 ホストグループ

概要

設定 → ホストグループセクションで、ホストグループを設定および保守できます。ホストグループには、テンプレートとホストの両方を含めることができます。

既存のホストグループとその詳細のリストが表示されます。ホストグループを名前で検索およびフィルタリングできます。

Name	Hosts	Templates	Members	Info
Discovered hosts	Hosts	Templates		
Hypervisors	Hosts	Templates		
Linux servers	Hosts 4	Templates	Server1, Server2, Server3, Server4	
Templates	Hosts	Templates		
Templates/Applications	Hosts	Templates 14	Template App Apache by HTTP, Template App Apache by Zabbix agent, Template App Apache Tomcat JMX, Template App Generic Java JMX, Template App Nginx by HTTP, Template App Nginx by Zabbix agent, Template App RabbitMQ cluster by HTTP, Template App RabbitMQ cluster by Zabbix agent, Template App RabbitMQ node by HTTP, Template App RabbitMQ node by Zabbix agent, Template App Remote Zabbix proxy, Template App Remote Zabbix server, Template App Zabbix Proxy, Template App Zabbix Server	
Templates/Databases	Hosts	Templates 2	Template DB MySQL, Template DB PostgreSQL	

表示情報

列	詳細
名前	ホストグループの名前。クリックするとホストグループ設定フォームが開きます。
ホスト	グループ内のホストの数 (灰色で表示)。“ホスト”をクリックすると、ホストのリストでグループに属するホスト一覧が表示されます。
テンプレート	グループ内のテンプレートの数 (灰色で表示)。“テンプレート”をクリックすると、テンプレートのリストでグループに属するテンプレート一覧が表示されます。
メンバー	グループメンバーの名前。テンプレート名は灰色で表示され、監視対象のホスト名は青色で表示され、監視対象外のホスト名は赤色で表示されます。名前をクリックすると、テンプレート/ホスト設定フォームが開きます。
情報	ホストグループに関するエラー情報が表示されます。(存在する場合)

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- ホストを有効 - グループに所属するホストのステータスを監視対象にします
- ホストを無効 - グループに所属するホストのステータスを監視対象外にします
- 削除 - ホストグループを削除します

これらのオプションを使用するには、それぞれのホストグループの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のホストグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

2 テンプレート

概要

設定 → テンプレートセクションで、ユーザーはテンプレートを設定および保守できます。

既存のテンプレートとその詳細のリストが表示されます。

Name	Hosts	Items	Triggers	Graphs	Dashboards	Discovery	Web	Linked templates	Linked to templates	Tags
Template OS Linux by Prom	Hosts	Items 34	Triggers 12	Graphs 7	Dashboards 2	Discovery 3	Web			
Template OS Linux by Zabbix agent	Hosts 1	Items 42	Triggers 14	Graphs 8	Dashboards 1	Discovery 3	Web	Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent		

表示情報

列	詳細
名前 ホスト	テンプレートの名前。クリックするとテンプレートの設定フォームが開きます テンプレートがリンクされている編集可能なホストの数。読み取り専用ホストは含まれていません。 ホストをクリックすると、テンプレート名でフィルターされたホストリストが開きます。
エンティティ (アイテム, トリガー, グラフ, ダッシュボード, ディスカバリ, Web)	テンプレート内のそれぞれのエンティティの数。(灰色で表示) エンティティ名をクリックすると、テンプレート名でフィルターされたそのエンティティのリストが表示されます。
テンプレートとのリンク	テンプレートにリンクされているテンプレート。ネストしてセットアップされたテンプレートは、リンクされたテンプレートのすべてのエンティティを継承します。
テンプレートへのリンク	テンプレートがリンクされているテンプレート (このテンプレートからすべてのエンティティを継承する「子」テンプレート)。 Zabbix 5.0.3 以降、この列にはホストが含まれなくなりました。
タグ	テンプレートのTags。マクロは未解決です。

新しいテンプレートを設定するには、右上隅にあるテンプレートの作成ボタンをクリックします。YAML、XML、または JSON ファイルからテンプレートをインポートするには、右上隅にあるインポートボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のテンプレートのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクはテンプレートの作成ボタンとインポートボタンの下にあります。クリックすると、ホストグループ、リンクされたテンプレート、名前、タグでテンプレートをフィルタリングできるフィルターが利用可能になります。

パラメータ	詳細
ホストグループ	1つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
テンプレートとのリンク	直接リンクされたテンプレートでフィルタリングします。
名前	テンプレート名でフィルタリングします。
タグ	テンプレートタグの名前と値でフィルタリングします。 フィルタリングは、テンプレートレベルのタグ (継承されたタグではない) のみ可能です。特定のタグとタグ値を含めたり除外したりすることができ、いくつかの条件も設定できます。タグ名のフィルタリングでは常に大文字と小文字が区別されます。 条件ごとに使用できる演算子がいくつかあります。 Exists - 指定されたタグ名を含める Equals - 指定されたタグ名を含める (大文字と小文字を区別) Contains - タグ値に入力された文字列が含まれるタグ名を含める (部分文字列の一致、大文字と小文字を区別しない) Does not exist - 指定されたタグ名と値を除外する Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別) Does not contain - タグ値に入力された文字列が含まれるタグ名を除外する (サブstringの一致、大文字と小文字を区別しない) 条件には2つの計算タイプがあります: ** And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or** - どれか1つの条件が満たされた場合

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- エクスポート - テンプレートを YAML、XML、または JSON ファイルにエクスポートします
- 一括更新 - 一度に複数のテンプレートのいくつかのプロパティを更新します
- 削除 - リンクされたエンティティ (アイテム、トリガーなど) をホストに残したまま、テンプレートを削除します
- 削除とクリア - テンプレートとそのリンクされたエンティティをホストから削除します

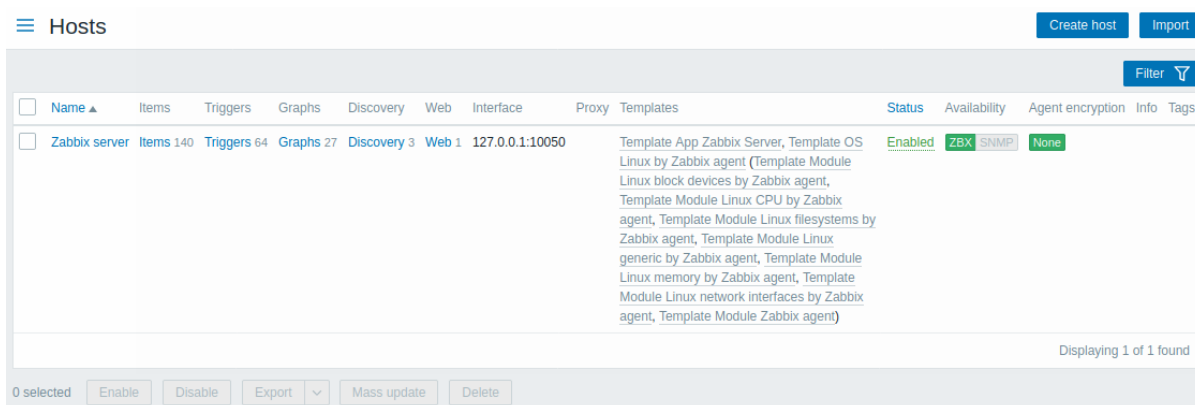
これらのオプションを使用するには、それぞれのテンプレートの前にあるチェックボックスをチェックしてから、必要なボタンをクリックします。

3 ホスト


概要

設定 → ホストセクションで、ユーザーはホストを設定および保守できます。

既存のホストとその詳細のリストが表示されます。



表示情報

列	詳細
名前 エンティティ (アイテム, トリガー, グラフ, ディスカバリ, Web) インターフェース プロキシ	ホストの名前。ホスト名をクリックするとホストの 設定フォーム が開きます。 エンティティ名をクリックすると、ホストのアイテム、トリガーなどが表示されます。それぞれのエンティティ数は灰色で表示されます。 ホストのメインインターフェースが表示されます。 ホストがプロキシによって監視されている場合、プロキシ名が表示されます。 この列は、監視するものフィルターオプションが 'すべて' または 'プロキシ' に設定されている場合にのみ表示されます。
テンプレート	ホストにリンクされているテンプレートが表示されます。リンクされたテンプレートに他のテンプレートが含まれている場合、それらはコマンドで区切られた括弧内に表示されます。テンプレート名をクリックすると設定フォームが開きます。
ステータス	ホストステータスが表示されます - 有効または無効。ステータスをクリックすると、ステータスを変更できます。 オレンジ色のレンチアイコン  はこのホストがメンテナンス中であることを示します。マウスポインタをアイコンの上に置くと、メンテナンスの詳細が表示されます。
エージェントの状態	設定されたインターフェースごとのホストの 状態 が表示されます。 アイコンは、設定されたインターフェースタイプ (Zabbix エージェント、SNMP、IPMI、JMX) のみを表します。アイコンにマウスを置くと、このタイプのすべてのインターフェースのポップアップリストが表示され、各インターフェースの詳細、ステータス、エラーが表示されます。 インターフェースのないホストの場合、列は空です。 すべてのインターフェースの現在のステータスがそれぞれのアイコンの色で表示されます。 緑 - すべてのインターフェースが使用可能 黄色 - 少なくとも 1 つのインターフェースが使用可能で、少なくとも 1 つが使用不可、他に「不明」を含む任意の値を設定できます。 赤 - 使用可能なインターフェースがありません 灰色 - 少なくとも 1 つのインターフェースが不明 (使用不可) アクティブな Zabbix エージェントアイテムは使用できないことに注意してください。ホストの可用性に影響します。

列	詳細
エージェント暗号化	ホスト接続の暗号化ステータスが表示されます： なし - 暗号化なし PSK - 事前共有キーを使用 Cert - 証明書を使用
情報 タグ	ホストに関するエラー情報（有る場合）が表示されます。 ホストの タグ 。マクロは未解決。

新しいホストを構成するには、右上隅にあるホストの作成ボタンをクリックします。YAML、XML、または JSON ファイルからホストをインポートするには、右上隅にあるインポートボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - ホストのステータスを有効に変更します
- 無効 - ホストのステータスを無効に変更します
- エクスポート - ホストを YAML、XML、または JSON ファイルにエクスポートします
- 一括更新 - 複数ホストの**複数のプロパティ一括更新**
- 削除 - ホストを削除します

これらのオプションを使用するには、ホスト名先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のホストのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、ホストリストの上にあります。クリックするとホストがサーバーまたはプロキシ、プロキシ名、タグ、ホストグループ、リンクされたテンプレート、名前、DNS、IP、ポート番号等でホストをフィルタリングできるフィルターが使用可能になります。

パラメータ	詳細
ホストグループ	1つ以上のホストグループでフィルタリングします。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
テンプレート	リンクされたテンプレートでフィルタリングします
名前	表示用のホスト名でフィルタリングします
DNS	DNS名でフィルタリングします
IP アドレス	IPアドレスでフィルタリングします
ポート	ポート番号でフィルタリングします
監視するもの	サーバーのみ、プロキシのみ、またはその両方によって監視されているホストをフィルタリングします。
プロキシ	ここで指定したプロキシによって監視されているホストをフィルタリングします。

パラメータ	詳細
タグ	<p>ホストタグの名前と値でフィルタリングします。特定のタグとタグ値を含めたり除外したりすることができます。いくつかの条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。</p> <p>各条件で使用できる演算子はいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名と値を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれるタグ名を含める (部分文字列の一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれるタグ名を除外する (サブストリングの一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります：</p> <p>** And/Or - すべての条件が満たされる必要があります、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or** - どれか 1 つの条件が満たされた場合</p>

ホストの状態確認

ホストの状態アイコンは、Zabbix サーバーの現在のホストインターフェースのステータスを反映しています。したがって、フロントエンドでは：

- ホストを無効にした場合、サーバーは最初に設定変更を同期する必要があるため、状態アイコンはすぐに灰色 (不明なステータス) になりません。
- ホストを有効にした場合、サーバーは設定変更を同期し、最初にホストのポーリングを開始する必要があるため、状態アイコンがすぐに緑色 (使用可能) に変わりません。

不明なインターフェースのステータス

Zabbix サーバーは、次の場合に対応するエージェントインターフェース (Zabbix、SNMP、IPMI、JMX) のステータスを判別します。

- インターフェイスに有効なアイテムが無い (削除または無効化されている)
- アクティブな Zabbix エージェントアイテムのみがある
- インターフェイスにポーラーが無い (例：StartPollers=0)
- ホストが無効
- ホストが、プロキシ、別のプロキシ、またはプロキシによってサーバーごとに監視されるように設定されている
- ホストがオフラインのように見えるプロキシによって監視されている (最大ハートビート間隔 - 1 時間の間にプロキシから更新を受信していない)

インターフェイスの可用性を不明に設定するのは、サーバー設定キャッシュの同期後に行われます。プロキシによって監視されているホストでのインターフェイスの可用性 (使用可能/使用不可) の復元は、プロキシ設定キャッシュの同期後に行われます。

ホストインターフェース **到達不能** の詳細も参照してください。

1 アイテム

概要

ホストのアイテムリストには、設定 → ホストからそれぞれのホストのアイテムをクリックしてアクセスできます。

既存のアイテムのリストが表示されます。

Name	Triggers	Key	Interval	History	Trends	Type	Status	Tags	Info
Template Module Zabbix agent: Host name of Zabbix agent running		agent.hostname	1h	7d		Zabbix agent (active)	Enabled	App: 1 App: 2 App: 3	
Template Module Zabbix agent: Zabbix agent ping		agent.ping	1m	1d	365d	Zabbix agent	Enabled	Application: Monitorin...	
Template Module Zabbix agent: Version of Zabbix agent running		agent.version	1h	7d		Zabbix agent	Enabled	Application: Monitorin...	
Template Module Linux generic by Zabbix agent: Maximum number of open file descriptors	Triggers 1	kernel.maxfiles	1h	7d	365d	Zabbix agent	Enabled	Application: General	
Template Module Linux generic by Zabbix agent: Maximum number of processes	Triggers 2	kernel.maxproc	1h	7d	365d	Zabbix agent	Enabled	Application: General	
A Interface \$1: Inbound packets, compressed		net.if.in["enp4s0",compressed]	3m	7d	365d	Zabbix agent	Enabled	Application: Interface ...	
Network interface discovery: Interface enp4s0: Inbound packets discarded		net.if.in["enp4s0",discarded]	3m	7d	365d	Zabbix agent	Enabled	Application: Interface ...	

表示情報

列	詳細
アイテムメニュー ホスト	アイテムメニュー アイテムのホスト。 この列は、フィルターで複数のホストが選択されている場合にのみ表示されます。
名前	アイテムの詳細へのリンクとして表示されるアイテムの名前。 アイテム名をクリックすると、アイテム設定フォームが開きます。 アイテムが継承されている場合別のテンプレートからは、テンプレート名がアイテム名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのアイテムリストが開きます。 アイテムがプロトタイプアイテムから作成されている場合、その名前の前には、オレンジ色の低レベルディスカバリルール名が付いています。ディスカバリルール名をクリックすると、プロトタイプアイテムリストが開きます。
トリガー	トリガーにマウスオーバーすると、アイテムに関連付けられているトリガーを表示するインフォボックスが表示されます。 トリガーの数は灰色で表示されます。
キー 監視間隔	アイテムキーの表示 監視間隔の表示 注パッシブアイテムは、監視データ取得ボタンを押してもすぐにチェックできません。
ヒストリ トレンド タイプ	アイテムデータ履歴が保持される日数を表示 アイテムのトレンド履歴が保持される日数を表示 アイテムタイプの表示 (Zabbix エージェント, SNMP エージェント, シンプルチェック等)
ステータス	アイテムのステータスが表示されます - 有効, 無効, 取得不可。ステータスをクリックすると、ステータスを有効から無効 (およびその逆) または無効から取得不可 (およびその逆) に変更できます。
タグ	アイテムタグが表示されます。 最大 3 つのタグ (名前: 値のペア) を表示できます。タグがさらにある場合は、"..." リンクが表示され、マウスオーバーですべてのタグを表示できます。
情報	アイテムが正常に機能している場合、この列にアイコンは表示されません。エラーが発生した場合は、「i」の文字が付いた正方形のアイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明が記載されたツールチップが表示されます。

新しいアイテムを設定するには、右上隅にあるアイテムの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - アイテムのステータスを有効にします
- 無効 - アイテムのステータスを無効にします
- 監視データ取得 - 新しいアイテム値のチェックをすぐに実行します。パッシブチェックのみでサポートされます (詳細を参照)。値をすぐにチェックする場合、構成キャッシュは更新されないため、値はアイテム構成に対するごく最近の変更を反映しないことに注意してください。
- 履歴のクリア - アイテムのヒストリとトレンドデータを削除します。
- コピー - アイテムを他のホストまたはテンプレートにコピーします

- 一括更新 - 一度に複数のアイテムの複数のプロパティを更新
- 削除 - アイテムを削除します

これらのオプションを使用するには、それぞれの項目の先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のアイテムのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターアイコンは右上隅にあります。クリックすると、フィルターが開き、目的のフィルター基準を指定できます。

パラメータ

詳細

ホストグループ

1つ以上のホストグループでフィルタリングします。
親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。

ホスト

テンプレートのみを含むホストグループは選択できません。

名前

一つ以上のホストでフィルタリングします

キー

アイテム名でフィルタリングします。

値のマッピング

アイテムキーでフィルタリングします。

使用する値マップでフィルタリングします。

タイプ

ホストオプションが空の場合、このパラメータは表示されません。
アイテムタイプでフィルタリングします (Zabbix エージェント, SNMP エージェント, 他).

データ型

データ型でフィルタリングします (Numeric unsigned, float, etc.).

ヒストリ

アイテム履歴が保持される期間でフィルタリングします。

トレンド

アイテムのトレンドが保持される期間でフィルタリングします。

監視間隔

アイテムの更新間隔でフィルタリングします。

パラメータ	詳細
タグ	<p>表示項目を制限するタグを指定してください。特定のタグとタグ値を含めたり除外したりするためにいくつかの条件を設定できます。タグ名の照合では、常に大文字と小文字が区別されます。</p> <p>条件ごとに使用できる演算子がいくつかあります。</p> <p>Exists - 指定されたタグ名を含める</p> <p>Equals - 指定されたタグ名を含める (大文字と小文字を区別)</p> <p>Contains - タグ値に入力された文字列が含まれるタグ名を含める (部分文字列の一致、大文字と小文字を区別しない)</p> <p>Does not exist - 指定されたタグ名と値を除外する</p> <p>Does not equal - 指定されたタグ名と値を除外する (大文字と小文字を区別)</p> <p>Does not contain - タグ値に入力された文字列が含まれるタグ名を除外する (サブstringの一致、大文字と小文字を区別しない)</p> <p>条件には 2 つの計算タイプがあります：</p> <p>** And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or** - どれか 1 つの条件が満たされた場合</p> <p>アイテムの状態でフィルタリングします - ノーマルまたは取得不可</p> <p>アイテムのステータスでフィルタリングします - 有効または無効</p> <p>トリガーあり (またはなし) のアイテムをフィルタリングします。</p> <p>テンプレートから継承された (または継承されていない) アイテムをフィルタリングします。</p> <p>低レベルディスクカバリによって検出された (または検出されなかった) アイテムをフィルタリングします。</p>
状態	
ステータス	
トリガー	
継承したもの	
Discovered	

フィルタの下サブフィルタは、さらにフィルタリングオプションを提供します。(すでにフィルタリングされたデータの場合) 共通のパラメータ値を持つアイテムのグループを選択できます。グループをクリックすると、グループが強調表示され、このパラメータ値を持つアイテムのみがリストに残ります。

2 トリガー

概要

ホストのトリガーリストには、設定 → ホストからホストのトリガーをクリックしてアクセスできます。

☰ Triggers
Create trigger

All hosts / Zabbix server
Enabled ZBX SNMP IPMI JMX
Items 142
Triggers 67
Graphs 27
Discovery rules 3
Web scenarios 1
Filter

Severity	Value	Name	Operational data	Expression	Status	Info	Tags
<input type="checkbox"/>	Average	OK	Mounted filesystem discovery: /: Disk space is critically low (used > {SVFS.FS.PUSED.MAX.CRIT:"7"}%)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})	<code>last({Zabbix server/vfs.fs.size[/,pused]})>{SVFS.FS.PUSED.MAX.CRIT:"7"} and ((last({Zabbix server/vfs.fs.size[/,total]})-last({Zabbix server/vfs.fs.size[/,used]})<5G or timeleft({Zabbix server/vfs.fs.size[/,pused],1h,100}<1d)</code>	Enabled	
<input type="checkbox"/>	Warning	OK	Mounted filesystem discovery: /: Disk space is low (used > {SVFS.FS.PUSED.MAX.WARN:"7"}%) Depends on: Zabbix server: /: Disk space is critically low (used > {SVFS.FS.PUSED.MAX.CRIT:"7"}%)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})	<code>last({Zabbix server/vfs.fs.size[/,pused]})>{SVFS.FS.PUSED.MAX.WARN:"7"} and ((last({Zabbix server/vfs.fs.size[/,total]})-last({Zabbix server/vfs.fs.size[/,used]})<10G or timeleft({Zabbix server/vfs.fs.size[/,pused],1h,100}<1d)</code>	Enabled	
<input type="checkbox"/>	Average	OK	Mounted filesystem discovery: /: Running out of free inodes (free < {SVFS.FS.INODE.PFREE.MIN.CRIT:"7"}%)	Free inodes: {ITEM.LASTVALUE1}	<code>min({Zabbix server/vfs.fs.inode[/,pfree],5m})<{SVFS.FS.INODE.PFREE.MIN.CRIT:"7"}</code>	Enabled	
<input type="checkbox"/>	Warning	OK	Mounted filesystem discovery: /: Running out of free inodes (free < {SVFS.FS.INODE.PFREE.MIN.WARN:"7"}%) Depends on: Zabbix server: /: Running out of free inodes (free < {SVFS.FS.INODE.PFREE.MIN.CRIT:"7"}%)	Free inodes: {ITEM.LASTVALUE1}	<code>min({Zabbix server/vfs.fs.inode[/,pfree],5m})<{SVFS.FS.INODE.PFREE.MIN.WARN:"7"}</code>	Enabled	
<input type="checkbox"/>	Information	OK	Template Module Linux generic by Zabbix agent: /etc/passwd has been changed Depends on: Zabbix server: Operating system description has changed Zabbix server: System name has changed (new name: {ITEM.VALUE})		<code>(last({Zabbix server/vfs.file.cksum[/etc/passwd],#1})<last({Zabbix server/vfs.file.cksum[/etc/passwd],#2})>0</code>	Enabled	

表示情報

列	詳細
深刻度	トリガーの深刻度は名前とセルの背景色の両方で表示されます。

列	詳細
値	トリガー値が表示されます。 OK - トリガーのステータスは OK です 障害 - トリガーのステータスは障害です。
ホスト	トリガーのホスト。 この列は、フィルターで複数のホストが選択されている場合にのみ表示されます。
名前	トリガーの名前。トリガー詳細へのリンクとして表示されます。 トリガー名のリンクをクリックすると、トリガー 設定フォーム が開きます。 ホストトリガーのテンプレートに属している場合、テンプレート名はトリガー名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、テンプレートレベルでトリガーリストが開きます。 トリガーがプロトタイプトリガーから作成されている場合、その名前の前に、オレンジ色の低レベルディスカバリールール名が表示されます。ディスカバリールール名をクリックすると、プロトタイプトリガーリストが開きます。
運用データ	監視データ → 障害で動的に解決される任意の文字列とマクロを含む、トリガーの操作データ定義。
条件式	トリガー式が表示されます。式のホスト-アイテム部分はアイテム設定フォームにつながるリンクとして表示されます。
ステータス	トリガーステータスが表示されます - 有効, 無効, 不明。ステータスをクリックすると、ステータスを有効から無効（およびその逆）または不明から無効（およびその逆）に変更できます。 無効なトリガーの障害はフロントエンドに表示されなくなりましたが、削除されません。
情報	すべてが正常に機能している場合、この列にアイコンは表示されません。エラーが発生した場合は、i の文字が付いた正方形のアイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明が記載されたツールチップが表示されます。
タグ	トリガーにタグが含まれている場合、タグの名前と値がこの列に表示されます。

新しいトリガーを構成するには、右上隅にあるトリガーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - トリガーのステータスを有効に
- 無効 - トリガーのステータスを無効に
- コピー - トリガーを他のホストまたはテンプレートにコピーします
- 一括更新 - 一度に複数のトリガーのいくつかのプロパティを更新します
- 削除 - トリガーの削除

これらのオプションを使用するには、トリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のトリガーのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターアイコンは右上隅にあります。クリックすると、フィルターが開き、目的のフィルター基準を設定できます。

パラメータ	詳細
ホストグループ	1つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
ホスト	1つ以上のホストでフィルタリングします。 上記でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに限定されます。
名前	トリガー名でフィルタリングします
深刻度	1つまたは複数のトリガーの深刻度でフィルタリングします
State	状態でフィルタリングします
ステータス	ステータスでフィルタリングします
値	値でフィルタリングします
Tags	トリガータグの名前と値でフィルタリングします。特定のタグとタグ値を含めたり除外したりするためにいくつかの条件を設定できます。タグ名の照合では、常に大文字と小文字が区別されます。条件ごとに使用できる演算子がいくつかあります。 Exists - 指定されたタグ名を含める Equals - 指定されたタグ名を含める（大文字と小文字を区別） Contains - タグ値に入力された文字列が含まれるタグ名を含める（部分文字列の一致、大文字と小文字を区別しない） Does not exist - 指定されたタグ名と値を除外する Does not equal - 指定されたタグ名と値を除外する（大文字と小文字を区別） Does not contain - タグ値に入力された文字列が含まれるタグ名を除外する（サブストリングの一致、大文字と小文字を区別しない） 条件には2つの計算タイプがあります： ** And/Or - すべての条件が満たされる必要があり、同じタグ名を持つ条件は Or 条件によってグループ化されます
Or** - どれか1つの条件が満たされた場合 マクロと マクロ関数 は、タグ名とタグ値のフィールドでサポートされています。
継承したもの	リンクされたテンプレートから継承された（または継承されなかった）トリガーをフィルターします。
Discovered	低レベルのディスカバリによって検出された（または検出されなかった）トリガーをフィルターします。
依存関係がある	依存関係がある（またはない）トリガーをフィルターします。

3 グラフ

概要

ホストのカスタムグラフリストには、設定 → ホストからホストのグラフをクリックしてアクセスできます。

既存のグラフリストが表示されます。

The screenshot shows the Zabbix 'Graphs' page. At the top, there are navigation links for 'All hosts / Zabbix server', status indicators for 'Enabled', 'ZBX', 'SNMP', 'IPMI', 'JMX', and counts for 'Items 151', 'Triggers 68', 'Graphs 30', 'Discovery rules 3', and 'Web scenarios 1'. A 'Create graph' button is visible in the top right. Below the navigation is a table with the following columns: Name, Width, Height, Graph type, and Info. The table lists various graphs such as 'Mounted filesystem discovery: /: Disk space usage', 'Template Module Linux CPU by Zabbix agent: CPU jumps', and 'Network interface discovery: Interface enp4s0: Network traffic'. Each row has a checkbox in the 'Name' column and an 'Info' icon in the 'Info' column.

表示情報

列	詳細
名前	カスタムグラフの名前。グラフ詳細へのリンクとして表示されます。グラフ名のリンクをクリックすると、グラフ設定フォームが開きます。ホストグラフはテンプレートに属しており、テンプレート名はグラフ名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、テンプレートレベルでグラフリストが開きます。グラフがプロトタイプグラフから作成されている場合、その名前の前に、オレンジ色の低レベルのディスカバリルール名が付きます。ディスカバリルール名をクリックすると、プロトタイプグラフのリストが開きます。
幅	グラフの表示幅
高さ	グラフの表示高さ
グラフのタイプ	グラフの種類が表示されます - ノーマル,* 積算グラフ, 円グラフまたは分解円グラフ *
情報	グラフが正しく機能している場合、この列にはアイコンは表示されません。エラーが発生した場合は、「i」の文字が付いた正方形のアイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明が記載されたツールチップが表示されます。

新しいグラフを設定するには、右上隅にあるグラフの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- コピー - グラフを他のホストまたはテンプレートにコピーします
- 削除 - グラフの削除

これらのオプションを使用するには、それぞれのグラフ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

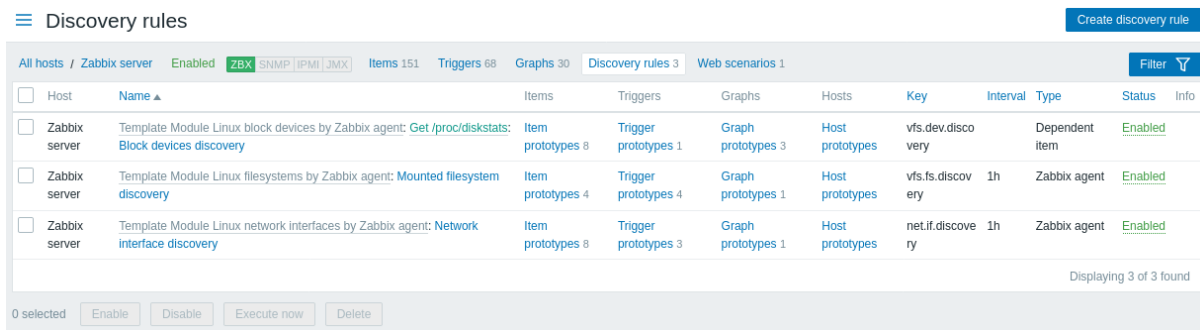
ホストグループとテンプレートでグラフをフィルタリングできます。検索パフォーマンスを向上させるために、データは未解決のマイクロで検索されます。

4 ディスカバリルール

概要

ホストの低レベルのディスカバリルールのリストには、構成 → ホストからそれぞれのテンプレートのディスカバリをクリックしてアクセスできます。

既存の低レベルディスカバリルールのリストが表示されます。フィルタ設定を変更することで、ホストとは関係なくすべてのディスカバリルールを表示したり、特定のホストグループのすべてのディスカバリルールを表示したりすることもできます。



表示情報

列	詳細
ホスト	現在表示中のホスト名が表示されます。表示されているホスト名がない場合は、技術的なホスト名が表示されます。

列	詳細
名前	リンクとして表示されるルールの名前。 ルール名をクリックすると、低レベルのディスカバリルール設定フォームが開きます。
アイテム	ディスカバリルールがテンプレートに属している場合、灰色のリンクとして、テンプレート名がルール名の前に再生されます。テンプレートリンクをクリックすると、テンプレートレベルでルールリストが開きます。
トリガー	プロトタイプアイテムリストへのリンクが表示されます。 既存のプロトタイプアイテム数が灰色で表示されます。
グラフ	プロトタイプトリガーリストへのリンクが表示されます。 既存のプロトタイプトリガー数が灰色で表示されます。
ホスト	プロトタイプグラフリストへのリンクが表示されます。 既存のプロトタイプグラフ数が灰色で表示されます。
キー	プロトタイプホストリストへのリンクが表示されます。 既存のプロトタイプホスト数が灰色で表示されます。
監視間隔	ディスカバリに使用したアイテムキーが表示されます。 ディスカバリを実行する頻度が表示されます。 注リストの下にある監視データ取得ボタンを押して検出をすぐに行うこともできます。
タイプ	ディスカバリに使用したアイテムタイプ (Zabbix エージェント、SNMP エージェントなど) が表示されます。
ステータス	ディスカバリルールのステータスが表示されます - 有効, 無効, 取得不可。ステータスをクリックすると、ステータスを有効から無効 (およびその逆) に変更できます。[取得不可]からは[無効] (およびその逆) となります。
情報	すべて問題がなければ、この列にアイコンは表示されません。エラーが発生した場合は、「i」の文字が付いた正方形のアイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明が記載されたツールチップが表示されます。

新しい低レベルのディスカバリルールを構成するには、右上隅にあるディスカバリルールの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - 低レベルのディスカバリルールのステータスを有効に変更します
- 無効 - 低レベルのディスカバリルールのステータスを無効に変更します
- 監視データ取得 - ディスカバリルールに基づいてディスカバリをすぐに行います。詳しくは[詳細](#)を参照してください。ディスカバリをすぐに行う場合、設定キャッシュは更新されないため、結果にはディスカバリルール設定に対する最近の変更が反映されないことに注意してください。
- 削除 - 低レベルのディスカバリルールを削除します

これらのオプションを使用するには、ディスカバリルール先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意の検出ルールのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、検出ルールのリストの上にあります。クリックすると、フィルターが使用可能になり、ホストグループ、ホスト、名前、アイテムキー、アイテムタイプ、およびその他のパラメーターで検出ルールをフィルター処理できます。

パラメータ	詳細
ホストグループ	1つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
ホスト名前	1つ以上のホストでフィルタリングします。
キー	ディスカバリルール名でフィルタリングします。
タイプ	ディスカバリアイテムキーでフィルタリングします。
監視間隔	ディスカバリアイテムタイプでフィルタリングします。
存在しなくなったリソースの保持期間	監視間隔でフィルタリングします。
SNMP OID	Zabbix トラッパおよび依存アイテムでは使用できません。 失われたリソースの保持期間でフィルタリング SNMP OID でフィルタリング タイプが SNMP エージェントを選択している場合にのみ使用できません。
状態	ディスカバリルールの状態でフィルタリング (すべて/ノーマル/取得負荷)
ステータス	ディスカバリルールのステータスでフィルタリング (すべて/有効/無効)

1 アイテムのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスカバリルールの構成済みプロトタイプアイテムが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプアイテムは、低レベルディスカバリ中に実際のホスト **items** を作成するための基礎になります。

☰ Item prototypes
Create item prototype

All hosts / Zabbix server Enabled ZBX SNMP IPMI Discovery list / Network interface discovery

Item prototypes 8 Trigger prototypes 3 Graph prototypes 1 Host prototypes

<input type="checkbox"/>	Name ▲	Key	Interval	History	Trends	Type	Create enabled	Discover	Tags
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Bits received	net.if.in["{#FNAME}"]	3m	7d	365d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Bits sent	net.if.out["{#FNAME}"]	3m	7d	365d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Inbound packets discarded	net.if.in["{#FNAME}",dropped]	3m	7d	365d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Inbound packets with errors	net.if.in["{#FNAME}",errors]	3m	7d	365d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Interface type	vfs.file.contents["/sys/class/net/{#FNAME}/type"]	1h	7d	0d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Operational status	vfs.file.contents["/sys/class/net/{#FNAME}/operstate"]	1m	7d	0	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Outbound packets discarded	net.if.out["{#FNAME}",dropped]	3m	7d	365d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}
<input type="checkbox"/>	... Template Module Linux network interfaces by Zabbix agent: Interface {#FNAME}: Outbound packets with errors	net.if.out["{#FNAME}",errors]	3m	7d	365d	Zabbix agent	Yes	Yes	Application: Interface {#FNAME}

Displaying 8 of 8 found

0 selected
Create enabled
Create disabled
Mass update
Delete

表示情報

列	詳細
名前	リンクとして表示されるプロトタイプアイテムの名前。 名前をクリックすると、アイテムのプロトタイプ設定フォームが開きます。 アイテムのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名は、アイテム名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでアイテムのプロトタイプリストが開きます。
キー	プロトタイプアイテムのキー
監視間隔	監視実行頻度
ヒストリ	アイテムデータ履歴を保持する日数
トレンド	アイテムのトレンド履歴を保持する日数

列	詳細
タイプ	プロトタイプアイテムのタイプ設定 (Zabbix エージェント、SNMP エージェント、簡易チェックなど)
自動生成時のステータス有効	プロトタイプアイテムのステータス設定 はい - 有効
ディスカバリ	いいえ - 無効。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプアイテムのディスカバリ設定 はい - ディスカバリする いいえ - ディスカバリしない。クリックで「はい」と「いいえ」を切り替えることができます。
タグ	プロトタイプアイテムのタグが表示されます。

新しいプロトタイプアイテムを構成するには、右上隅にあるアイテムのプロトタイプを作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - アイテムを有効として作成します
- 自動生成無効 - アイテムを無効として作成します
- 一括更新 - プロトタイプアイテムを一括更新します
- 削除 - プロトタイプアイテムを削除する

これらのオプションを使用するには、プロトタイプアイテムの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

2 トリガーのプロトタイプ

概要

このセクションでは、テンプレートに設定された低レベルディスカバリルールのプロトタイプトリガーが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプトリガーは、低レベルのディスカバリ中に実際のホストトリガーを作成するための基礎になります。

☰ Trigger prototypes
Create trigger prototype

All hosts / Zabbix server
Enabled ZBX SNMP IPMI
Discovery list / Network interface discovery

Item prototypes 8
Trigger prototypes 3
Graph prototypes 1
Host prototypes

	Severity	Name ▲	Operational data	Expression		Create enabled	Discover	Tags
<input type="checkbox"/>	Information	Template Module Linux network interfaces by Zabbix agent: Interface {#IFNAME}: Ethernet has changed to lower speed than it was before Depends on: Zabbix server: Interface {#IFNAME}: Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: <code>change(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/type"]<0 and last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/type"]>0 and (last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/type"])=6 or last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/type"])=1) and (last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/operstate"]<->2)</code> Recovery: <code>(change(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/type"]>0 and last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/type"]#2)>0) or (last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/operstate"])=2)</code>	Yes	Yes		
<input type="checkbox"/>	Warning	Template Module Linux network interfaces by Zabbix agent: Interface {#IFNAME}: High error rate (> {#IFERRORS.WARN:"{#IFNAME}"}) for 5m Depends on: Zabbix server: Interface {#IFNAME}: Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: <code>min(Zabbix server/net.if.in["{#IFNAME}",errors],5m)>{#IFERRORS.WARN:"{#IFNAME}"}) or min(Zabbix server/net.if.out["{#IFNAME}",errors],5m)>{#IFERRORS.WARN:"{#IFNAME}"})</code> Recovery: <code>max(Zabbix server/net.if.in["{#IFNAME}",errors],5m)<{#IFERRORS.WARN:"{#IFNAME}"})*0.8 and max(Zabbix server/net.if.out["{#IFNAME}",errors],5m)<{#IFERRORS.WARN:"{#IFNAME}"})*0.8</code>	Yes	Yes		
<input type="checkbox"/>	Average	Template Module Linux network interfaces by Zabbix agent: Interface {#IFNAME}: Link down	Current state: {ITEM.LASTVALUE1}	Problem: <code>{#IFCONTROL:"{#IFNAME}"}=1 and (last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/operstate"])=2 and (last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/operstate"],#1)->last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/operstate"],#2)=1)</code> Recovery: <code>last(Zabbix server/vfs.file.contents["/sys/class/net/{#IFNAME}/operstate"]<->2</code>	Yes	Yes		

Displaying 3 of 3 found

0 selected
Create enabled
Create disabled
Mass update
Delete

表示情報

列	詳細
名前	リンクとして表示されるプロトタイプトリガーの名前。 名前をクリックすると、トリガーのプロトタイプ設定フォームが開きます。 トリガーのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名は、アイテム名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでトリガーのプロトタイプリストが開きます。
運用データ	監視データ → 障害で動的に解決される任意の文字列とマクロを含む、トリガーの操作データ形式が表示されます。
自動生成時のステータス有効	プロトタイプトリガーのステータス設定 はい - 有効
ディスカバリ	いいえ - 無効。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプトリガーのディスカバリ設定 はい - ディスカバリする いいえ - ディスカバリしない。クリックで「はい」と「いいえ」を切り替えることができます。
タグ	プロトタイプトリガーのタグが表示されます。

新しいプロトタイプトリガーを構成するには、右上隅にあるトリガーのプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - トリガーを有効として作成します
- 自動生成無効 - トリガーを無効として作成します
- 一括更新 - プロトタイプトリガーを一括更新します
- 削除 - プロトタイプトリガーを削除する

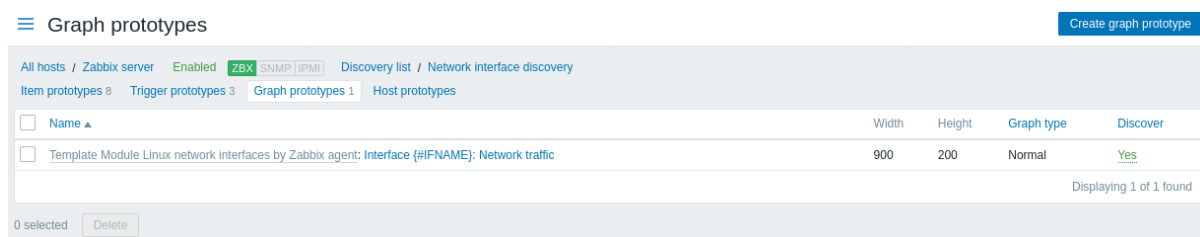
これらのオプションを使用するには、プロトタイプトリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

3 グラフのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスカバリルールの構成済みプロトタイプグラフが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプグラフは、低レベルディスカバリ中に実際のホスト items を作成するための基礎になります。



表示情報

列	詳細
名前	リンクとして表示されるプロトタイプグラフの名前。 名前をクリックすると、グラフのプロトタイプ設定フォームが開きます。 グラフのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名はグラフ名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでグラフのプロトタイプリストが開きます。
幅	プロトタイプグラフの表示幅
高さ	プロトタイプグラフの表示高
グラフのタイプ	プロトタイプグラフのタイプ設定 - ノーマル, 積算グラフ, 円グラフ, 分解円グラフ

列	詳細
ディスクバリ	プロトタイプグラフのディスクバリ設定 はい - ディスクバリする いいえ - ディスクバリしない。クリックで「はい」と「いいえ」を切り替えることができます。

新しいプロトタイプグラフを構成するには、右上隅にあるグラフのプロトタイプを作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 削除 - プロトタイプグラフを削除する

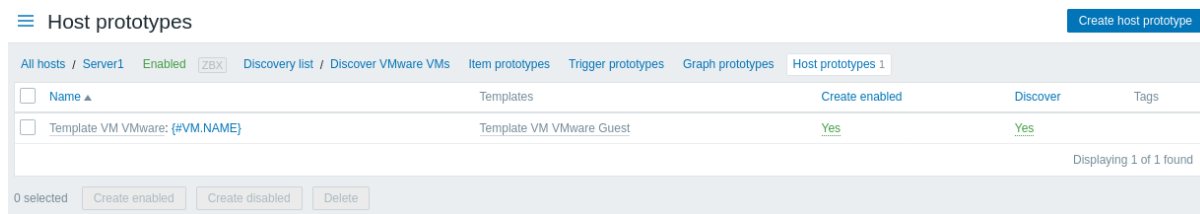
これらのオプションを使用するには、プロトタイプグラフの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

4 ホストのプロトタイプ

概要

このセクションでは、テンプレートの低レベルディスクバリルールの構成済みプロトタイプホストが表示されます。

テンプレートがホストにリンクされている場合、プロトタイプホストは、低レベルディスクバリ中に実際のホスト **items** を作成するための基礎になります。



表示情報

列	詳細
名前	リンクとして表示されるプロトタイプホストの名前。 名前をクリックすると、ホストのプロトタイプ設定フォームが開きます。 ホストのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名は、ホスト名の前に灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルでホストのプロトタイプリストが開きます。
テンプレート 自動生成時のステータス有効	プロトタイプホストのテンプレート表示 プロトタイプホストのステータス設定 はい - 有効
ディスクバリ	いいえ - 無効。クリックで「はい」と「いいえ」を切り替えることができます。 プロトタイプホストのディスクバリ設定 はい - ディスクバリする いいえ - ディスクバリしない。クリックで「はい」と「いいえ」を切り替えることができます。
タグ	プロトタイプホストのタグ表示

新しいプロトタイプホストを構成するには、右上隅にあるホストのプロトタイプを作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - ホストを有効として作成します
- 自動生成無効 - ホストを無効として作成します
- 削除 - プロトタイプホストを削除する

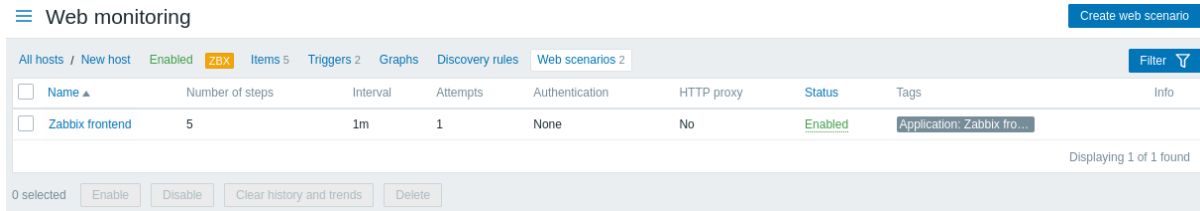
これらのオプションを使用するには、プロトタイプホストの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

5 Web シナリオ

概要

テンプレートの Web シナリオリストには、設定 → テンプレートからそれぞれのテンプレートの Web をクリックしてアクセスできます。

既存の Web シナリオのリストが表示されます。



表示情報

列	詳細
名前	Web シナリオの名前。Web シナリオ名をクリックすると、Web シナリオ設定フォームが開きます。 Web シナリオが別のテンプレートから継承されている場合、テンプレート名は Web シナリオ名の前に灰色で表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルで Web シナリオリストが開きます。
ステップ数	シナリオに含まれるステップ数
監視間隔	シナリオが実行される頻度
試行回数	Web シナリオステップを実行するための試行回数
認証	認証方法が表示されます - Basic, NTLM はなし
HTTP プロキシステータス	HTTP プロキシを表示します。使用しない場合は「いいえ」を表示します。 Web シナリオのステータスが表示されます - 有効または無効。 ステータスをクリックすると、ステータスを変更できます。
タグ	Web シナリオタグが表示されます。 最大 3 つのタグ (名前: 値のペア) を表示できます。タグがさらにある場合は「...」リンクが表示され、マウスオーバーですべてのタグを表示できます。
情報	すべてが正常に機能している場合、この列にアイコンは表示されません。エラーが発生した場合は、「i」の文字が付いた正方形のアイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明が記載されたツールチップが表示されます。

新しい Web シナリオを構成するには、右上隅にある Web シナリオの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - シナリオのステータスを有効にします
- 無効 - シナリオのステータスを無効にします
- ヒストリを削除 - シナリオのヒストリとトレンドを削除します
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、Web シナリオ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のシナリオのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、Web シナリオのリストの上にあります。クリックすると、ホストグループ、ホスト、ステータス、タグでシナリオをフィルタリングできるフィルターが利用可能になります。

All hosts / New host Enabled ZBX Items 5 Triggers 2 Graphs Discovery rules Web scenarios 2 Filter

Host groups: type here to search [Select] Tags: And/Or Or

Hosts: New host x type here to search [Select] tag Contains value [Remove] [Add]

Status: all Enabled Disabled

[Apply] [Reset]

4 メンテナンス

概要

設定 → メンテナンスセクションで、ホストのメンテナンス期間を設定および維持できます。

既存のメンテナンス期間とその詳細のリストが表示されます。

☰ Maintenance periods Create maintenance period Filter

<input type="checkbox"/>	Name ▲	Type	Active since	Active till	State	Description
<input type="checkbox"/>	Server regular	With data collection	2020-04-17 00:00	2021-04-18 00:00	Active	We break and fix things at this time.

0 selected [Delete] Displaying 1 of 1 found

表示情報

列	詳細
名前	メンテナンス期間の名前。メンテナンス期間名をクリックすると、メンテナンス期間設定フォームが開きます。
タイプ	メンテナンスのタイプが表示されます：データ収集ありまたはデータ収集なし
設定有効期間の開始日時	メンテナンス期間の実行がアクティブになる日時。 注：この時間はメンテナンス期間の設定ではありません。メンテナンス期間は別途設定する必要があります。
設定有効期間の終了日時	メンテナンス期間の実行がアクティブでなくなる日時。
状態	メンテナンス期間の状態： メンテナンス待ち - まもなくアクティブになります 有効 - アクティブです メンテナンス機関の終了 - もうアクティブではありません
説明	メンテナンス期間の説明が表示されます。

新しいメンテナンス期間を設定するには、右上隅にあるメンテナンス期間の作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 削除 - メンテナンス期間を削除します

これらのオプションを使用するには、それぞれのメンテナンス期間の先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のメンテナンス期間のみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、メンテナンス期間のリストの上にあります。それをクリックすると、ホストグループ、名前、および状態でメンテナンス期間をフィルタリングできるフィルターが使用可能になります。

Filter

Host groups: type here to search [Select] State: Any Active Approaching Expired

Name: []

[Apply] [Reset]

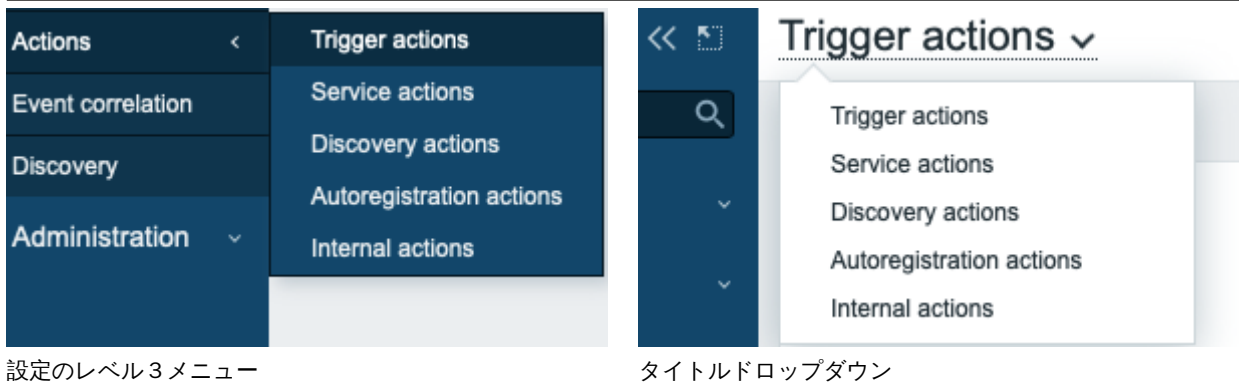
5 アクション

概要

設定 → アクションセクションで、ユーザーはアクションを設定および保守できます。

表示されるアクションは、選択したイベントソースに割り当てられたアクション（トリガー、サービス、ディスカバリ、自動登録、内部イベントアクション）です。

アクションは、イベントソース（トリガー、サービス、ディスカバリ、自動登録、内部イベントアクション）ごとのサブセクションにグループ化されます。使用可能なサブセクションのリストは、設定メニューセクションのアクションを押すと表示されます。左上隅のタイトルドロップダウンを使用して、サブセクションを切り替えることもできます。



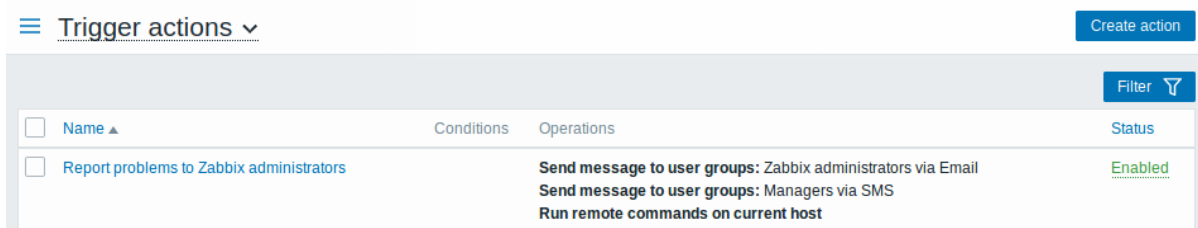
サブセクションを選択すると、既存のアクションのリストとその詳細が記載されたページが表示されます。

スーパー管理者権限を持たないユーザーの場合、権限設定に従ってアクションが表示されます。つまり、特定の権限制限のために、スーパー管理者権限を持たないユーザーは完全なアクションリストを表示できない場合があります。スーパー管理者権限のないユーザーの表示アクション条件は以下の通りです。

- ユーザーは、アクション条件でホストグループ、ホスト、テンプレート、およびトリガーへの読み取り/書き込みアクセス権を持っている
- ユーザーは、アクション操作、回復操作、および更新操作で、ホストグループ、ホスト、およびテンプレートへの読み取り/書き込みアクセス権を持っている
- ユーザーは、アクション操作、回復操作、および更新操作でユーザーグループおよびユーザーへの読み取りアクセス権を持っている

Note:

サービスのアクションは、サービス -> サービスアクションメニューセクションでも同様の方法で維持されます。特定のサービスアクションへのユーザーのアクセスは、サービスへのアクセスメニューセクションで設定されたユーザーロールのアクセス許可によって異なります。



表示情報

列	詳細
名前	アクションの名前。アクション名をクリックすると、アクション設定フォームが開きます
実行条件	実行条件が表示されます

列	詳細
実行内容	実行内容が表示されます。 Zabbix2.2以降、リストには、通知に使用されるメディアタイプ（電子メール、SMS、またはスクリプト）と、通知受信者の名前とユーザー名の後の括弧内も表示されます。 実行内容は、選択した操作のタイプに応じて通知またはリモートコマンドとなります。
ステータス	アクションのステータスが表示されます - 有効または無効 ステータスをクリックすると、ステータスを変更できます。 エスカレーション進行中にアクションが無効になった場合の詳細については エスカレーションセクション を参照してください。

新しいアクションを設定するには、右上隅にある[アクションの作成]ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - アクションステータスを有効に変更します
- 無効 - アクションステータスを無効に変更します
- 削除 - アクションを削除します

これらのオプションを使用するには、それぞれのアクションの前あるにチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のアクションのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクはアクションリストの上にあります。クリックすると名前とステータスでアクションをフィルタリングできるフィルターが使用可能になります。

6 イベント相関関係

概要

設定 → イベント相関セクションで、ユーザーは Zabbix イベントのグローバル相関ルールを設定および維持できます。

表示情報

列	詳細
名前	相関関係の名前。相関関係名をクリックすると、相関関係 [設定フォーム] が開きます。(/manual/config/event_correlation/global#configuration)
実行条件	相関関係の条件が表示されます。
実行内容	相関関係の操作が表示されます。
ステータス	相関関係のステータスが表示されます - 有効または無効。 ステータスをクリックすると、ステータスを変更できます。

新しい相関関係を設定するには、右上隅にある [相関関係の作成] ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - 相関関係のステータスを有効に変更します
- 無効 - 相関関係のステータスを無効に変更します
- 削除 - 相関関係を削除します

これらのオプションを使用するには、それぞれの相関ルールの前にチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意の相関関係のみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、相関関係リストの上にあります。クリックすると名前とステータスで相関関係をフィルタリングできるフィルターが使用可能になります。

7 ディスカバリ

概要

設定 → ディスカバリセクションで、ユーザーはディスカバリルールを構成および維持できます。

既存のディスカバリルールとその詳細のリストが表示されます。

表示情報:

列	詳細
名前	ディスカバリルールの名前。ルール名をクリックすると、ディスカバリルール設定フォームが開きます。
IP アドレスの範囲	ネットワークスキャンに使用する IP アドレスの範囲が表示されます。
プロキシ	プロキシによってディスカバリが実行される場合は、プロキシ名が表示されます。
監視間隔	ディスカバリを実行する頻度が表示されます。
チェック	ディスカバリに使用されたチェックのタイプが表示されます。
ステータス	アクションステータスが表示されます-有効または無効。 ステータスをクリックするとステータスを変更できます。

新しいディスカバリルールを設定するには、右上隅にある [ディスカバリルールの作成] ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

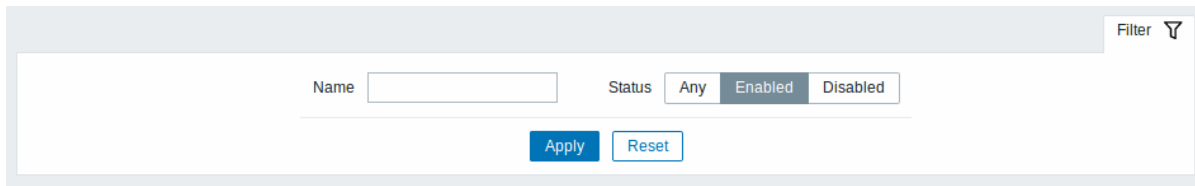
- 有効 - ディスカバリルールのステータスを有効にします
- 無効 - ディスカバリルールのステータスを無効にします
- 削除 - ディスカバリルールを削除します

これらのオプションを使用するには、それぞれのディスカバリルールの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のディスカバリルールのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、ディスカバリルールリストの上にあります。クリックすると、名前とステータスでディスカバリルールをフィルタリングできるフィルターが使用可能になります。



6 管理

概要

管理メニューは Zabbix の管理機能用です。このメニューは **Super Administrators** タイプのユーザーのみが使用できます。

1 一般設定

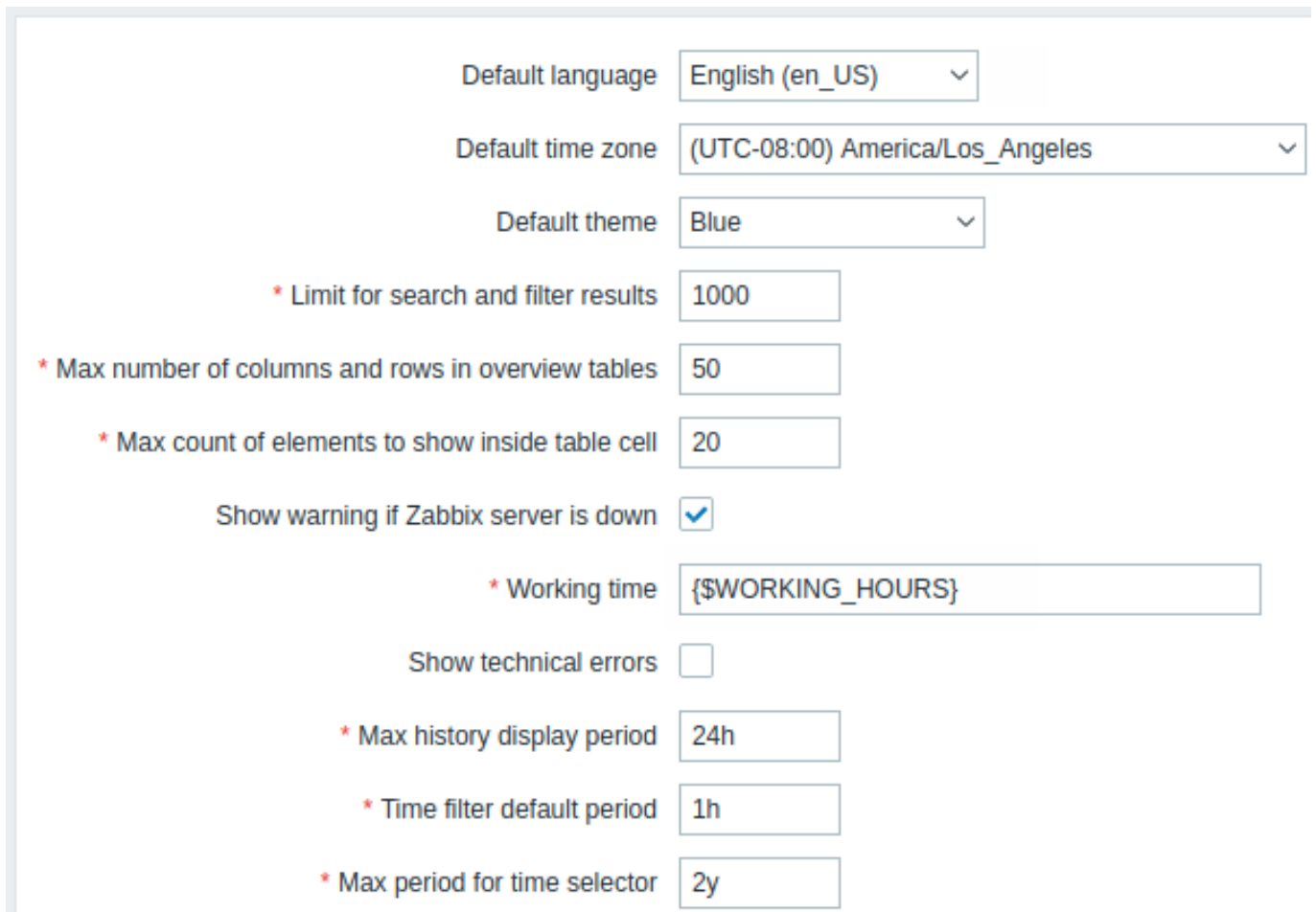
概要

管理 → 一般設定セクションには、フロントエンド関連のデフォルト設定と Zabbix をカスタマイズするためのいくつかのサブセクションが含まれています。

使用可能なサブセクションのリストは、管理メニューセクションの一般設定を押すと表示されます。左上隅のタイトルドロップダウンを使用して、サブセクションを切り替えることもできます。

1 表示設定

このセクションでは、フロントエンドに関連するデフォルトのカスタマイズについて説明します。



設定値:

パラメータ	詳細
デフォルトの言語	<p>プロファイルで言語を指定していないユーザーおよびゲストユーザーのデフォルト言語を設定します。</p> <p>詳細については、追加のフロントエンド言語のインストールを参照してください。</p>
デフォルトのタイムゾーン	<p>プロファイルでデフォルトのタイムゾーンを指定していないユーザーおよびゲストユーザーのデフォルトタイムゾーンを設定します。</p>
デフォルトのテーマ	<p>プロファイルでテーマを指定していないユーザーおよびゲストユーザーのデフォルトテーマを設定します。</p>
検索/フィルターの項目の上限値	<p>Web インターフェースリスト (たとえば、構成 → ホスト) に表示される要素 (行) の最大数。</p> <p>注:たとえば「50」に設定されている場合、最初の 50 要素のみ影響を受けるすべてのフロントエンドリストに表示されます。一部のリストに 50 を超える要素が含まれている場合、「+」サイン「50+」の 1 から 50 が見つかりました」で確認できます。またフィルタリングが使用されていても 50 を超える一致がある場合は、最初の 50 のみが表示されます。</p>
概要テーブル内の行と列の最大値	<p>データ概要およびトリガー概要ダッシュボードウィジェットに表示する列と行の最大数。同じ制限が列と行の両方に適用されます。表示されているよりも多くの行や列が存在する場合、システムはテーブルの下部に「すべての結果が表示されるわけではありません。より具体的な検索条件を指定してください」という警告を表示します。</p>
テーブルセル内のリストの最大項目数	<p>単一のテーブルセルに表示されるエントリの場合、ここで設定した数だけ表示されます。</p>
Zabbix サーバー停止時にワーニングを表示	<p>このパラメータを使用すると、Zabbix サーバーに到達できない (場合によってはダウンしている) 場合に、ブラウザウィンドウに警告メッセージを表示できます。ユーザーがページを下にスクロールしても、メッセージは表示されたままになります。カーソルを合わせると、メッセージが一時的に非表示になり、その下の内容が表示されず。</p> <p>このパラメータは Zabbix2.0.1 以降でサポートされています。</p>
ワーキングタイム	<p>このシステム全体のパラメーターは、労働時間を定義します。グラフでは、稼働時間は白地に表示され、非稼働時間は灰色で表示されます。時間形式の説明については、期間指定ページを参照してください。</p> <p>ユーザーマクロがサポートされています (Zabbix 3.4.0 以降)。</p>
技術的エラーを表示	<p>オンにすると、登録されているすべてのユーザーが技術的エラー (PHP / SQL) を確認できるようになります。チェックを外すと Zabbix スーパー管理者とデバッグモードが有効になっているユーザーグループに属するユーザーのみが情報を利用できます。</p>
履歴の最大表示期間	<p>監視サブセクションに履歴データを表示する最大期間: 最新データ、* Web、およびデータ概要 * ダッシュボードウィジェット。</p> <p>許容範囲: 24 時間 (デフォルト) -1 週間タイムサフィックス例: 1w (1 週間)、36h (36 時間) がサポートされています。</p>
フィルタリングするデフォルトの期間	<p>デフォルトでグラフとダッシュボードで使用される期間。許容範囲: 1 分-10 年 (デフォルト: 1 時間)</p> <p>タイムサフィックス例: 10m (10 分)、5w (5 週間) がサポートされています。</p> <p>注:ユーザーがグラフの表示中に期間を変更すると、この期間はユーザー設定として保存され、グローバルデフォルトまたは以前のユーザー選択が置き換えられます。</p>
タイムセレクターの最大期間	<p>グラフとダッシュボードで利用可能な最大期間。超過するとユーザーは古いデータを参照できなくなります。許容範囲: 1 年-10 年 (デフォルト: 2 年)。</p> <p>タイムサフィックス例: 1 年 (1 年)、365w (365 週間) がサポートされています。</p>

2 自動登録

このセクションでは、アクティブエージェントの自動登録の暗号化レベルを設定できます。

Encryption level No encryption
 PSK

* PSK identity

* PSK

アスタリスクでマークされたパラメーターは必須です。

設定値：

パラメータ	詳細
暗号化レベル	暗号化レベルオプションの 1 つまたは両方を選択します。 暗号無し - 暗号化されていない接続が許可されます PSK - 事前共有キーを使用した TLS 暗号化接続が許可されます 事前共有キー ID 文字列を入力します。
PSK ID	このフィールドは「暗号化レベル」の「PSK」が選択されている場合にのみ使用できます。 機密情報を PSKID に入れないでください。機密情報は暗号化されずにネットワーク経由で送信され、どの PSK を使用するかを受信者に通知します。
PSK	事前共有キー（偶数の 16 進文字）を入力します。 最大長：Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512。hex-digits (256 バイト PSK) mbed TLS (PolarSSL) ライブラリを使用する場合は 64 hex-digits (32-byte PSK)。 例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 このフィールドは、「暗号化レベル」の「PSK」が選択されている場合にのみ使用できます。

[安全な自動登録も参照](#)

3 データの保存期間

ハウスキーピングは定期的なプロセスで、Zabbix サーバーによって実行されます。このプロセスでは、古い情報とユーザーによって削除された情報が削除されます。

Events and alerts

Enable internal housekeeping

- * Trigger data storage period
- * Service data storage period
- * Internal data storage period
- * Network discovery data storage period
- * Autoregistration data storage period

Services

Enable internal housekeeping

- * Data storage period

Audit

Enable internal housekeeping

- * Data storage period

User sessions

Enable internal housekeeping

- * Data storage period

History

Enable internal housekeeping

Override item history period

- * Data storage period

Trends

Enable internal housekeeping

Override item trend period

- * Data storage period

このセクションでハウスキーピングタスクをタスクごと個別に有効または無効にできます。イベントとアラート/IT サービス/ユーザーセッション/履歴/トレンド。監査データ保存設定は、別のメニューセクションで利用できます。

データの保存が有効になっている場合、削除されるまでにデータレコードが保持される日数を設定できます。

アイテム/トリガーを削除すると、そのアイテム/トリガーによって生成された問題も削除されます。

またイベントは、障害に関連付けられていない場合にのみ、ハウスキーパーによって削除されます。これは、イベントが障害または回復イベントのいずれかである場合、関連する障害レコードが削除されるまで削除されないことを意味します。ハウスキーパーは、古いイベントや障害レコードに関する潜在的な問題を回避するために、最初に障害を削除し、その後にイベントを削除します。

履歴とトレンドについては、追加のオプションを使用できます：アイテムの履歴の保存期間設定を上書きおよびアイテムのヒス

トリの保存期間設定を上書きです。このオプションを使用すると、アイテムの履歴/トレンドを保持する日数（1 時間から 25 年、つまり「0」）をグローバルに設定できます。この場合、履歴保存期間/トレンド保存期間の個々のアイテムに設定された値が上書きされます。注意：アイテム構成のフィールド設定オプション履歴を保持しないおよび/またはトレンドを保持しないが有効になっているアイテムの保存期間は上書きされません。

内部ハウスキーピングが無効になっている場合でも、履歴/トレンド保存期間を上書きすることができます。したがって、外部ハウスキーパーを使用する場合、履歴保存期間は、履歴データ保存期間フィールドを使用して設定できます。

Attention:

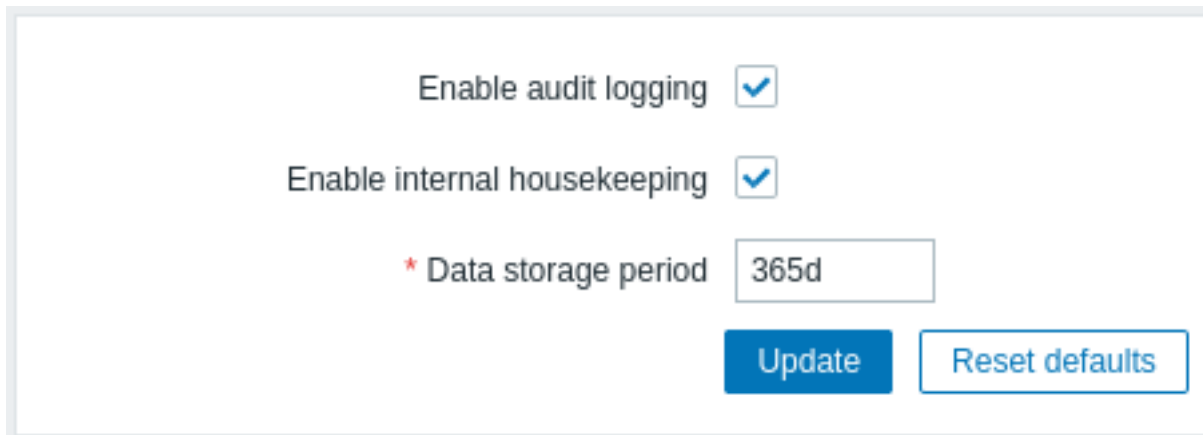
TimescaleDB を使用する場合、履歴とトレンドテーブルの TimescaleDB 自動パーティション化を最大限に活用するには、アイテムの履歴の保存期間設定を上書きおよびアイテムの履歴の保存期間設定を上書きオプションを有効にする必要があります。また、履歴とトレンドの内部ハウスキーピング* オプションを有効にする必要があります。それ以外の場合これらのテーブルに保持されているデータは引き続きパーティションに保存されますが、ハウスキーパーは、古いパーティションを削除するのではなく、個々のレコードを削除することで履歴とトレンドをクリーンアップします。古いパーティションのドロップが有効になっている場合、Zabbix サーバーとフロントエンドは削除されたアイテムを追跡しなくなり、古いパーティションが削除されると削除されたアイテムの履歴がクリアされます。

タイムサフィックスは期間フィールドでサポートされています。1d (1 日)、1w (1 週間)。最短は 1 日 (履歴は 1 時間)、最長は 25 年です。

デフォルトにリセットボタンを使用すると、行った変更を元に戻すことができます。

4 監査ログ

このセクションでは監査ログの設定ができます。

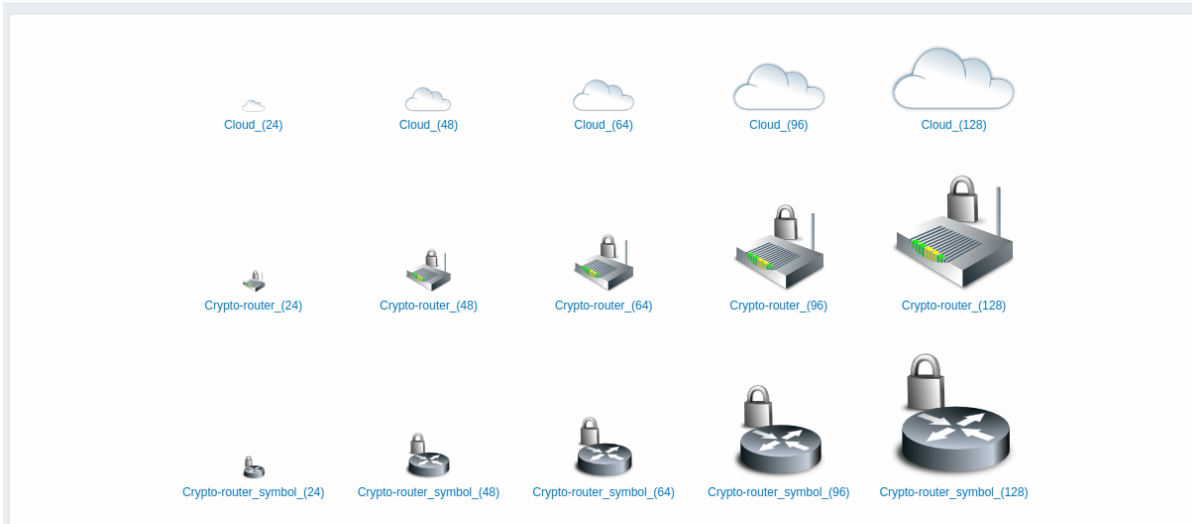


次のパラメータを使用できます：

パラメータ	詳細
監査ログの有効化	監査ログを有効/無効にします。デフォルトでマークされています。
削除処理を有効	監査ログのための内部的な削除処理を有効/無効にします。デフォルトでマークされています。
データ保存期間	内部的な削除処理によって削除される前に、監査記録を保持する必要がある日数。削除処理が有効になっている場合は必須です。デフォルト：365 日

5 イメージ

イメージセクションには、Zabbix で利用可能なすべての画像が表示されます。画像はデータベースに保存されます。



タイプドロップダウンを使用すると、アイコンと背景画像を切り替えることができます。

- アイコンはネットワークマップ要素を表示するために使用されます
- 背景はネットワークマップの背景画像として使用されます

イメージの追加

右上隅にある [アイコンの作成] または [背景の作成] ボタンをクリックして、独自の画像を追加できます。

* Name

* Upload No file selected.

画像属性 :

パラメータ	詳細
名前	画像の一意の名前
アップロード	Zabbix にアップロードするファイル (PNG、JPEG、GIF) をローカルシステムから選択します。 注意アップロード中に PNG に変換される他の形式をアップロードできる場合があります。GD ライブラリが画像処理に使用されるため、サポートされるフォーマットは使用するライブラリのバージョンによって異なります (Zabbix 2.0.28 以降が必要です)

Note:

アップロードファイルの最大サイズは、1024x1024 バイトまたは 1MB の ZBX_MAX_IMAGE_SIZE の値によって制限されます。画像サイズが 1MB に近く、max_allowed_packet の場合、画像のアップロードが失敗することがあります。MySQL 設定パラメーターのデフォルトは 1MB です。この場合max_allowed_packetパラメーターを増やします。

6 アイコンのマッピング

このセクションでは、特定のアイコンを使用してホストのマッピングを作成できます。ホストインベントリフィールド情報は、マッピングの作成に使用されます。

マッピングをネットワークマップ設定に使用して、一致するホストに適切なアイコンを自動的に割り当てることもできます。

新しいアイコンマップを作成するには、右上隅にあるアイコンのマッピング作成をクリックします。

* Name

* Mappings

Inventory field	Expression	Icon	Action
1: Type	server	Server_(96)	Remove
2: Type	router	Router_(96)	Remove
3: Type	workstation	Workstation_(96)	Remove
Add			
Default		Cloud_(24)	

設定値

パラメータ	詳細
名前	一意なアイコンマップ名
マッピング	マッピングのリスト。順序によってどちらが優先されるかが決まります。ドラッグアンドドロップでマッピングをリ
インベントリフィールド	一致するものを探すために検索されるホストインベントリフィールド
条件式	一致する正規表現
アイコン	式に一致するものが見つかった場合に使用するアイコン
標準	デフォルトで使われるアイコン

7 正規表現

このセクションでは、フロントエンドのいくつかの場所で使用できるカスタム正規表現を作成できます。詳細については[正規表現セクション](#)を参照してください。

8 マクロ

このセクションでは、システム全体の**ユーザーマクロ**を名前と値のペアとして定義できます。マクロ値は、プレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できることに注意してください。説明の追加もサポートされています。

Macro	Value	Description
<input style="background-color: #f0f0f0;" type="text" value="{MYSQL_PASSWORD}"/>	<input style="background-color: #f0f0f0;" type="text" value="*****"/>	<input type="text" value="description"/>
<input style="background-color: #f0f0f0;" type="text" value="{MYSQL_USERNAME}"/>	<input style="background-color: #f0f0f0;" type="text" value="*****"/>	<input type="text" value="description"/>
<input "{\$secret_password}"="" style="background-color: #f0f0f0;" type="text" value=""/>	<input type="text" value="path/to/secret:password"/>	<input type="text" value="description"/>
<input "{\$secret_username}"="" style="background-color: #f0f0f0;" type="text" value=""/>	<input type="text" value="path/to/secret:username"/>	<input type="text" value="description"/>
<input "{\$snmp_community}"="" style="background-color: #f0f0f0;" type="text" value=""/>	<input type="text" value="public"/>	<input type="text" value="description"/>
<input "{\$working_hours}"="" style="background-color: #f0f0f0;" type="text" value=""/>	<input type="text" value="1-5,09:00-18:00"/>	<input type="text" value="description"/>

[Add](#)

9 イベントのステータス

このセクションではフロントエンドのイベントステータスの表示方法と**イベントの重要度**の名前と色をカスタマイズできます。

Use custom event status colors

* Unacknowledged PROBLEM events blinking

* Acknowledged PROBLEM events blinking

* Unacknowledged RESOLVED events blinking

* Acknowledged RESOLVED events blinking

* Display OK triggers for

* On status change triggers blink for

* Not classified


* Information

* Warning

* Average

* High

* Disaster



パラメータ	詳細
イベントのステータスの文字色を変更	チェックすると確認済み/未確認の問題の色のカスタマイズがオンになります
未確認の障害イベント, 確認済障害イベント, 未確認の解決済イベント, 確認済みの解決済イベント 正常イベントの表示期間	新しいカラーコードを入力するか、色をクリックして、表示されたパレットから新しいカラーコードを選択します。 点滅チェックボックスがオンになっている場合、ステータスが変わるとトリガーがしばらく点滅して発見しやすくなります。
ステータスが変わった際の点滅期間	OK トリガーを表示する期間。範囲：0～24 時間。タイムサフィックス 例:5m,2h,1d がサポートされています。
未分類, 情報, 警告, 軽度の障害, 重度の障害, 致命的な障害	点滅期間の設定。許容範囲：0～24 時間。タイムサフィックス 例: 5m,2h,1d がサポートされています。 システムデフォルトの代わりに表示するカスタムの重大度の名前や色。 新しいカラーコードを入力するか、色をクリックして、提供されたパレットから新しいものを選択します。
	ここで設定したカスタマイズされた重大度の名前はすべてのローカルで使用されます。特定のユーザー向けに他の言語に翻訳する必要がある場合は、 イベントの重大度のカスタマイズ ページを参照してください。

10 地理マップ

このセクションでは、地理マップタイルサービスプロバイダーを選択し、Geomapダッシュボードウィジェットのサービスプロバイ

ダー設定を構成できます。地理マップを使用した視覚化を提供するために、Zabbix はオープンソースの JavaScript インタラクティブ マップライブラリリーフレットを使用します。Zabbix は、事前定義されたタイルプロバイダーを含むサードパーティのタイルプロバイダーによって提供される画像の品質を制御できないことに注意してください。

* Tile provider

* Tile URL

* Max zoom level

パラメータ	詳細
タイルプロバイダー	使用可能なタイルサービスプロバイダーの 1 つを選択するか、 その他 を選択して別のタイルプロバイダーまたはセルフホストタイルを追加します (カスタムタイルサービスプロバイダーの使用を参照)
タイル URL	タイルレイヤーをロードして地理マップに表示するための URL テンプレート。このフィールドは、タイルプロバイダーがその他に設定されている場合にのみ編集可能です。 次のプレースホルダーがサポートされています: {s} は使用可能なサブドメインの 1 つを表します {z} は URL のズームレベルパラメータを表します {x} と {y} はタイル座標を表します {r} は URL に"@2x" を追加し、retina タイルをロードできます。 例: <code>https://{s}.example.com/{z}/{x}/{y}{r}.png</code>
帰属表示	マップ上の小さなテキストボックスに表示されるタイルプロバイダーの帰属情報。このフィールドはタイルプロバイダーがその他に設定されている場合にのみ編集可能です。
最大ズームレベル	マップの最大ズームレベル。このフィールドはタイルプロバイダーがその他に設定されている場合にのみ編集可能です。

カスタムタイルサービスプロバイダーの使用

Geomap ウィジェットは、カスタムのセルフホストまたはサードパーティのタイルプロバイダーサービスからラスタータイル画像を読み込むことができます。カスタムサードパーティタイルプロバイダーサービスまたはセルフホストタイルフォルダーまたはサーバーを使用するにはタイルプロバイダーフィールドでその他を選択し、適切なプレースホルダーを使用してタイル URL フィールドでカスタム URL を指定します。

11 モジュール

このセクションでは、カスタムフロントエンドモジュールを管理できます。

☰ Modules ▾

<input type="checkbox"/> Name ▲	Version	Author	Description	Status
<input type="checkbox"/> Example module	1.0	John Smith	Short description of the module.	Enabled

Filter

Displaying 1 of 1 found

0 selected

ディレクトリスキャンをクリックして、カスタムモジュールを登録/登録解除します。登録済みのモジュールが詳細とともにリストに表示されます。未登録のモジュールはリストから削除されます。

モジュールを名前またはステータス (有効/無効) でフィルタリングできます。リスト内のモジュールステータスをクリックして、モジュールを有効/無効にします。また、リストでモジュールを選択し、リストの下にある [有効化/無効化] ボタンをクリックして、モジュールを一括有効化/無効化することもできます。

12 API トークン

このセクションでは、API トークンを作成および管理できます。

<input type="checkbox"/>	Name ▲	User	Expires at	Created at	Created by user	Last accessed at	Status
<input type="checkbox"/>	Token	Admin (Zabbix Administrator)	2022-01-26 00:00:00	2021-01-22 15:51:02	Admin (Zabbix Administrator)	Never	Enabled
<input type="checkbox"/>	Token 2	new_user	2021-01-26 00:00:00	2021-01-22 16:13:03	Admin (Zabbix Administrator)	Never	Enabled
<input type="checkbox"/>	Token 3	guest1	Never	2021-01-22 16:08:49	Admin (Zabbix Administrator)	Never	Enabled

API トークンは、名前、トークンが割り当てられているユーザー、有効期限、トークンを作成したユーザー、またはステータス（有効/無効）でフィルタリングできます。リスト内のトークンステータスをクリックして、トークンを有効/無効に設定できます。リストでトークンを選択し、リストの下にある有効化/無効化ボタンをクリックして、トークンを一括有効化/無効化することもできます。

新しいトークンを作成するには、右上隅にある API トークンの作成ボタンを押してから、トークン設定画面の必須フィールドに入力します。

API tokens ▾

* Name

* User Select

Description

Set expiration date and time

* Expires at 📅

Enabled

Add Cancel

パラメータ	詳細
名前	トークンの表示名
ユーザー	トークンを割り当てる必要があるユーザー。ユーザーを選択するには、ユーザー名、名または姓の入力を開始し、オートコンプリートリストから必要なユーザーを選択します。または選択ボタンを押して、ユーザーリストからユーザーを選択することもできます。トークンは 1 人のユーザーにのみ割り当てることができます。
説明	トークンの説明
有効期限の設定	トークンに有効期限を設定しない場合は、このチェックボックスをオフにします。
有効期限	カレンダーアイコンをクリックしてトークンの有効期限を選択するか、YYYY-MM-DD hh:mm:ss の形式で日付を手動で入力します
有効	無効な状態でトークンを作成する必要がある場合は、このチェックボックスをオフにします。

追加を押してトークンを作成します。次の画面で * ページを閉じる前に ** 認証トークンの値をコピーして安全な場所に保存して閉じる * を押します。トークンがリストに表示されます。

Warning:

認証トークンの値は後で再度表示することはできません。トークンを作成した直後のみ表示できます。保存したトークンを紛失した場合は、トークンを再生成する必要があります。そうすると、新しい認証文字列が作成されます。

トークン名をクリックして、名前、説明、有効期限の設定、またはトークンのステータスを編集します。トークンが割り当てられているユーザーを変更することはできないことに注意してください。変更後に更新ボタンを押して変更を保存します。トークンが紛失ま

たは公開された場合は、再生成ボタンを押して新しいトークン値を生成できます。再生成時は確認ダイアログボックスが表示されます。続行すると以前に生成されたトークンが無効になるためです。

管理メニューセクションにアクセスできないユーザーは、ユーザープロフィール → API トークンセクションで、API トークンの管理がユーザーロール 権限で許可されます。

13 その他の設定パラメータ

このセクションでは、その他のフロントエンドパラメータを設定できます。

Frontend URL

* Group for discovered hosts

Default host inventory mode Disabled Manual Automatic

User group for database down message

Log unmatched SNMP traps

Authorization

* Login attempts

* Login blocking interval

Security

Validate URI schemes

* Use X-Frame-Options HTTP header

Use iframe sandboxing

Communication with Zabbix server

* Network timeout

* Connection timeout

* Network timeout for media type test

* Network timeout for script execution

* Network timeout for item test

* Network timeout for scheduled report test

パラメータ	詳細
Web インターフェース URL	Zabbix Web インターフェースへの URL。このパラメータは、フロントエンドとの通信のために Zabbix Web サービスによって使用され、スケジュールされたレポートを有効にするために指定する必要があります。
ディスカバリで発見されたホストのグループ	ネットワーク検出およびエージェント自動登録によって検出されたホストは、ここで選択されたホストグループに自動的に配置されます。
デフォルトのホストインベントリモード	ホストインベントリのデフォルトのモード。ホストインベントリモードの設定操作によるホストの検出/自動登録中にオーバーライドされない限り、サーバーまたはフロントエンドによって新しいホストまたはホストのプロトタイプが作成されるたびにここでの設定値が設定されます。

パラメータ	詳細
データベース停止メッセージの送信先グループ	<p>アラームメッセージまたは'なし'を送信するためのユーザーグループ。</p> <p>Zabbix サーバーは、バックエンドデータベースの可用性によって異なります。データベースなしでは機能しません。データベースがダウンした場合、選択したユーザーが Zabbix から通知を受け取ることができます。通知は、設定されたすべてのユーザーメディアエントリを使用して、ここで設定されたユーザーグループに送信されます。Zabbix サーバーは停止しません。データベースが再び稼働して処理を続行するまで待機します。</p> <p>通知は次の内容で構成されます。</p> <p>[MySQL\ PostgreSQL\ Oracle] データベース <DB 名 >[<DB ホスト >:<DB ポート >] は使用できません:<DBMS のタイプに応じたエラーメッセージ (データベース) ></p> <p><DB ホスト > は、空の値として定義されている場合はメッセージに追加されず、デフォルト値 ("0") の場合は <DB ポート > は追加されません。アラートマネージャ (特別な Zabbix サーバードプロセス) は、10 秒ごとにデータベースへの新しい接続を確立しようとします。データベースがまだダウンしている場合、アラートマネージャはアラートの送信を繰り返しますが、15 分毎までです。</p> <p>対応する SNMP インターフェースが見つからない場合は、SNMP トラップをログに記録します。</p>
マッチしない SNMP trap をログに記録	

認可

パラメータ	詳細
ログインの試行	ログインがブロックされるログイン失敗の数
ログインブロックの間隔	ログインの試行回数を越えた場合にログインが禁止される期間

セキュリティ

パラメータ	詳細
URL スキーマの検証	有効な URI スキームで定義されたホワイトリストに対する URI スキーマの検証を無効にするには、チェックボックスをオフにします。(デフォルトは有効)
有効な URL スキーマ	許可された URL スキーマのコンマ区切りリスト。URL が使用されるフロントエンドのすべてのフィールドに適用されます (例: マップ要素の URL)
X-Frame-Options HTTP ヘッダ	このフィールドは URL スキーマの検証がチェックされている場合にのみ編集できます。 HTTPX-Frame-options ヘッダーの値。サポートされている値: SAMEORIGIN (デフォルト) - ページは、ページ自体と同じ原点のフレームにのみ表示できます。 DENY - サイトが表示しようとしているかどうかに関係なく、ページはフレームに表示できません。 null - X-Frame-options ヘッダーを無効にします (非推奨)
iframe サンドボックスを使用する	コンマ区切りのホスト名のリスト (文字列) - リストされたホスト名が許可されていない場合は、SAMEORIGIN オプションが使用されます。
iframe サンドボックスの例外	このパラメーターは、取得した URL コンテンツをサンドボックスに入れるかどうかを決定します。サンドボックスをオフにすることはお勧めしません。 サンドボックスが有効で、このフィールドが空の場合、すべてのサンドボックス属性の制限が適用されます。一部の制限を無効にするには、このフィールドでそれらを指定します。ここにリストされている制限のみが無効になります。他の制限は引き続き適用されます。詳細については、 サンドボックス属性 の説明を参照してください。

パラメータ	詳細
ネットワークタイムアウト	アイドル状態のソケットを閉じるまでの待機秒数 (Zabbix サーバーへの接続が以前に確立されているが、この間にフロントエンドがデータの読み取り/送信操作を完了できない場合、接続は切断されます)。許容範囲：1~300 秒 (デフォルト：3 秒)。
接続タイムアウト	Zabbix サーバーへの接続試行を停止するまでに待機する秒数。許容範囲：1~30 秒 (デフォルト：3 秒)。
メディアテストのためのネットワークタイムアウト	メディアタイプをテストするときに応答を待機する秒数。許容範囲：1~300 秒 (デフォルト：65 秒)。
スクリプト実行のためのネットワークタイムアウト	スクリプトの実行時に応答を待機する秒数。許容範囲：1~300 秒 (デフォルト：60 秒)。
アイテムテストのためのネットワークタイムアウト	アイテムをテストするときに返されるデータを待機する秒数。許容範囲：1~300 秒 (デフォルト：60 秒)。
定期レポートテストのためのネットワークタイムアウト	定期レポートをテストするときに返されたデータを待機する秒数。許容範囲：1~300 秒 (デフォルト：60 秒)。

2 プロキシ

概要

管理 → プロキシセクションで分散モニタリングのプロキシを Zabbix フロントエンドで設定できます。

プロキシ

既存のプロキシとその詳細のリストが表示されます。

表示情報

列	詳細
名前	プロキシの名前。プロキシ名をクリックするとプロキシ設定フォームが開きます。
モード 暗号化	プロキシモードが表示されます。- アクティブまたはパッシブ プロキシからの接続暗号化ステータスが表示されます。 None - 暗号化無し PSK - 事前共有キー使用 Cert - 証明書使用
最新データ受信時刻 (経過時間) ホスト数 アイテム数	サーバーがプロキシを最後に確認した時刻が表示されます。 プロキシに割り当てられている有効なホストの数が表示されます。 プロキシに割り当てられた有効なホスト上の有効なアイテムの数が表示されます。
要求パフォーマンス (vps)	必要なプロキシパフォーマンスが表示されます (1 秒あたりに収集する必要がある値の数)
ホスト	プロキシによって監視されているすべてのホストが一覧表示されます。ホスト名をクリックすると、ホスト設定フォームが開きます。

To configure a new proxy, click on the Create proxy button in the top right-hand corner.

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- ホストを有効 - プロキシによって監視されているホストのステータスを監視対象に変更します
- ホストを無効 - プロキシによって監視されているホストのステータスを監視対象外に変更します

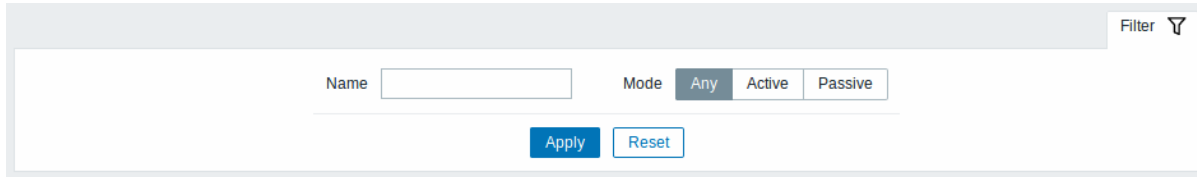
- 削除 - プロキシを削除します

これらのオプションを使用するには、それぞれのプロキシの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、関心のあるプロキシのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、プロキシのリストの上にあります。それをクリックすると、名前とモードでプロキシをフィルタリングできるフィルタが利用可能になります。



3 認証

概要

管理 → 認証セクションでは、Zabbix および内部パスワード要件に対するグローバルユーザー認証方法を指定できます。使用可能な方法は、内部、HTTP、LDAP および SAML 認証です。

デフォルト認証

デフォルトでは、Zabbix はすべてのユーザーに内部 Zabbix 認証を使用します。デフォルトの方法をシステム全体で **LDAP** に変更したり、特定のユーザーグループに対してのみ LDAP 認証を有効にしたりすることもできます。

LDAP をすべてのユーザーのデフォルトの認証方法として設定するには、LDAP タブに移動して認証パラメーターを構成してから、認証タブに戻り、デフォルト認証セレクターを LDAP に切り替えます。

認証方法は**ユーザーグループ**レベルで微調整できることに注意してください。LDAP 認証がグローバルに設定されている場合でも、一部のユーザーグループを Zabbix で認証することもできます。これらのグループでは**フロントエンドアクセス**を内部に設定する必要があります。逆に、内部認証をグローバルに使用する場合は、LDAP 認証の詳細を指定して、**フロントエンドアクセス**が LDAP に設定されている特定のユーザーグループに使用できます。ユーザーが少なくとも LDAP 認証を使用している 1 つのユーザーグループに含まれている場合、このユーザーは内部認証方式を使用できません。

デフォルトの認証方法に加えて**HTTP**および**SAML 2.0**認証方法を使用できます。

認証

認証タブでは、Zabbix 内部ユーザーのカスタムパスワード要件を定義できます。

Authentication

Authentication **HTTP settings** LDAP settings SAML settings

Default authentication **Internal** LDAP

Password policy

Minimum password length

Password must contain ? an uppercase and a lowercase Latin letter
 a digit
 a special character

Avoid easy-to-guess passwords ?

Update

次のパスワードポリシーオプションを設定できます。

パラメータ	説明
最小パスワード長	デフォルトではパスワードの最小長は 8 に設定されています。設定可能範囲:1~70。72 文字を超えるパスワードは切り捨てられます。
パスワード必須項目	1 つまたは複数のチェックボックスをオンにして、パスワードで指定した文字を使用するよう強制します。 -英字の大文字と小文字 -数字 -特殊記号
推測されやすいパスワードの回避	疑問符にカーソルを合わせると各オプションの文字リストを含むヒントを表示します。 チェックボックスがオンの場合、パスワードは次の要件に基づいてチェックされます。 -ユーザーの苗字、名前、またはユーザー名を含まないこと -共通またはコンテキスト固有のパスワードの 1 つではないこと。 共通およびコンテキスト固有のパスワードのリストは、NCSC の”トップ 100k パスワード”のリスト、SecList の”トップ 1M パスワード”のリスト、および Zabbix コンテキスト固有のパスワードのリストから自動的に生成されます。内部ユーザーは、このリストに含まれるパスワードを設定することはできません。このようなパスワードは、一般的に使用されているため、脆弱であると見なされるためです。

パスワードの複雑さの要件を変更しても、既存のユーザーパスワードには影響しませんが、既存のユーザーが管理 → ユーザーメニューからパスワードを変更する場合、新しいパスワードは現在の要件を満たす必要があります。要件のリストを含むヒントが、**ユーザープロフィール**および**ユーザー構成フォーム**の Password フィールドの横に表示されます。

HTTP 認証の設定

HTTP または Web サーバーベースの認証 (例: Basic 認証、NTLM / Kerberos) を使用して、ユーザー名とパスワードを確認できます。ユーザーは Zabbix にも存在する必要がありますが、その Zabbix パスワードは使用されないことに注意してください。

Attention:

オンにする前に、Web サーバー認証が構成され、正しく機能していることを確認してください。

Authentication HTTP settings ● LDAP settings SAML settings

Enable HTTP authentication

Default login form HTTP login form

Remove domain name comp, any

Case-sensitive login

Update

設定値:

パラメーター	説明
HTTP 認証の有効化 デフォルトのログイン画面	<p>チェックボックスをオンにすると、HTTP 認証が有効になります。認証されていないユーザーをどの画面に誘導するかを指定します： Zabbix のログイン画面 - 標準の Zabbix ログインページ HTTP のログイン画面 - HTTP ログインページ index_http.php ページに対してのみ Web サーバーベースの認証を有効にすることをお勧めします。デフォルトのログイン画面が HTTP ログインページに設定されている場合、Web サーバー認証モジュールが <code>\$_SERVER</code> 変数に有効なユーザーログインを設定すると、ユーザーは自動的にログインします。 サポートされている <code>\$_SERVER</code> キーは <code>PHP_AUTH_USER</code>, <code>REMOTE_USER</code>, <code>AUTH_USER</code> です。</p>
ドメイン名の削除	<p>ユーザー名から削除する必要があるドメイン名のカンマ区切りのリスト。 例：comp, any - ユーザー名が 'Admin@any', 'comp\Admin' の場合、ユーザーは 'Admin' としてログインします。ユーザー名が 'notacompany\Admin' の場合、ログインは拒否されます。</p>
ログイン時に大文字小文字を区別	<p>ユーザー名の大文字と小文字を区別するログイン (デフォルトで有効) を無効にするには、チェックボックスをオフにします。 例: 大文字と小文字を区別するログインを無効にすると、ユーザーが 'Admin' でログインした場合でも 'ADMIN' ユーザーでログインします。 注大文字と小文字を区別するログインを無効にすると、Zabbix データベースに同様のユーザー名を持つ複数のユーザーが存在する場合 (例:Admin, admin)、ログインが拒否されることに注意してください。</p>

Note:

Web サーバー認証の場合、すべてのユーザー (フロントエンドアクセスが LDAP/Internal に設定されている場合でも) は、Zabbix ではなく Web サーバーによって認証されます。

Note:

HTTP 資格情報 (HTTP ログインフォームがデフォルトとして設定されている) を使用してログインできず、401 エラーが発生する内部ユーザーは、Basic 認証ディレクトティブに `ErrorDocument 401/index.php?form=default` 行を追加すると、通常の Zabbix ログイン画面にリダイレクトされます。

LDAP 認証の設定

外部 LDAP 認証を使用して、ユーザー名とパスワードを確認できます。ユーザーは Zabbix にも存在する必要がありますが、その Zabbix パスワードは使用されないことに注意してください。

Zabbix LDAP 認証は、少なくとも Microsoft Active Directory と Open LDAP で機能します。

Enable LDAP authentication

* LDAP host

* Port

389

* Base DN

* Search attribute

Bind DN

Case sensitive login

Bind password

Test authentication [must be a valid LDAP user]

* Login

Admin

* User password

設定値:

パラメータ	説明
LDAP 認証の有効化	チェックボックスをオンにすると、LDAP 認証が有効になります。
LDAP ホスト	LDAP サーバーの名前。例：ldap://ldap.zabbix.com 安全な LDAP サーバーの場合は、ldaps プロトコルを使用します。 ldaps://ldap.zabbix.com OpenLDAP 2.x.x 以降では、フル ldap:// hostname:port または ldaps:// hostname:port 形式の LDAPURI を使用できます。
ポート	LDAP サーバーのポート。デフォルトは 389 です。 安全な LDAP 接続の場合、ポート番号は通常 636 です。 完全な LDAPURI を使用する場合は使用されません。
Base DN	検索アカウントへの基本パス： ou=Users、ou=system (OpenLDAP の場合)、 DC=company、DC=com (Microsoft Active Directory の場合)
検索の属性	検索に使用される LDAP アカウント属性： uid (OpenLDAP の場合) sAMAccountName (Microsoft Active Directory の場合)
Bind DN	LDAP サーバーを介してバインドおよび検索するための LDAP アカウント。例： uid=ldap_search,ou=system (OpenLDAP の場合)、 CN=ldap_search,OU=user_group,DC=company,DC=com (Microsoft Active Directory の場合) 匿名バインディングもサポートされています。

パラメータ	説明
ログイン時に大文字小文字を区別	ユーザー名の大文字と小文字を区別するログイン（デフォルトで有効）を無効にするには、チェックボックスをオフにします。 例：大文字と小文字を区別するログインを無効にすると、ユーザーが「Admin」でログインしたの場合でも「ADMIN」ユーザーでログインします。 注大文字と小文字を区別するログインを無効にすると、Zabbix データベースに同様のユーザー名を持つ複数のユーザーが存在する場合（例:Admin, admin）、ログインが拒否されることに注意してください。
Bind password	LDAP サーバーをバインドおよび検索するためのアカウントの LDAP パスワード。
認証のテスト ログイン	テスト用セクションのヘッダー テストユーザー（デフォルトでは現在ログイン中のユーザー）の名前。このユーザー名は LDAP サーバーに存在する必要があります。 Zabbix は、テストユーザーを認証できない場合、LDAP 認証をアクティブにしません。
ユーザーのパスワード	テストユーザーの LDAP パスワード。

Warning:

証明書に問題がある場合、安全な LDAP 接続 (ldaps) を機能させるには /etc/openldap/ldap.conf 設定ファイルに TLS_REQCERTallow 行を追加する必要がある場合がありますが、LDAP カタログへの接続のセキュリティが低下する可能性があります。

Note:

別の LDAP アカウント (Bind DN) を作成して、(Zabbix フロントエンドへのログインに使用している) 実際のユーザーアカウントを使用する代わりに、LDAP で最小限の権限で LDAP サーバー上でバインドと検索を実行することをお勧めします。このようなアプローチより高いセキュリティを提供し、ユーザーが LDAP サーバーで自分のパスワードを変更するときにバインドパスワードを変更する必要はありません。上の表では、*ldap_search* アカウント名です。

SAML 認証の設定

SAML 2.0 認証を使用して、Zabbix にサインインできます。ユーザーは Zabbix に存在する必要がありますが、その Zabbix パスワードは使用されないことに注意してください。認証が成功すると、Zabbix はローカルユーザー名を SAML によって返されたユーザー名属性と照合します。

Note:

SAML 認証が有効になっている場合、ユーザーはローカルでログインするか、SAML シングルサインオンを介してログインするかを選択できます。

ID プロバイダーの設定

SAML ID プロバイダーである Zabbix (onelogin.com, auth0.com, okta.com, など) を次の方法で設定する必要があります。

- アサーションコンシューマ URL は <path_to_zabbix_ui>/index_sso.php?acs に設定する必要があります
- シングルログアウト URL は <path_to_zabbix_ui>/index_sso.php?sls に設定する必要があります

<path_to_zabbix_ui> の例：%% <https://example.com/zabbix/ui>, <http://another.example.com/zabbix>, <http://> <any_public_ip_address> / zabbix %%

Zabbix のセッティング

Attention:

フロントエンドで SAML 認証を使用する場合は、php-openssl をインストールする必要があります。

SAML 認証を使用するには、Zabbix を次のように設定する必要があります。

1. **zabbix.conf.php** でカスタムパスが指定されていない限り、秘密鍵と証明書は ui/conf/certs/ に保存する必要があります。

デフォルトでは、Zabbix は次の場所を検索します。

- ui/conf/certs/sp.key - SP 秘密鍵ファイル
- ui/conf/certs/sp.crt - SP 証明書ファイル
- ui/conf/certs/idp.crt - IDP 証明書ファイル

2. 最も重要な設定はすべて、Zabbix フロントエンドで構成できます。ただし、**設定ファイル**で追加設定を指定することも可能です。

Authentication
HTTP settings
LDAP settings
SAML settings ●

Enable SAML authentication

* IdP entity ID

* SSO service URL

SLO service URL

* Username attribute

* SP entity ID

SP name ID format

Sign Messages
 Assertions
 AuthN requests
 Logout requests
 Logout responses

Encrypt Name ID
 Assertions

Case sensitive login

Zabbix フロントエンドで利用可能な設定パラメータ:

パラメータ	説明
SAML 認証の有効化	チェックボックスをマークして、SAML 認証を有効にします。
IdP エンティティ ID	SAML ID プロバイダーのユニーク識別子。
SSO サービス URL	ログイン時にユーザーがリダイレクトされる URL。
SLO サービス URL	ログアウト時にユーザーがリダイレクトされる URL。空のままにすると、SLO サービスは使用されません。
//ユーザー名の属性//	Zabbix へのログイン時にユーザー名として使用される SAML 属性。 サポートされている値のリストは ID プロバイダーによって決まります。 例: uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 <urn:oid:0.9.2342.19200300.100.1.44>
SP のエンティティ ID	SAML サービスプロバイダーのユニーク識別子。

パラメータ	説明
SP の NameID フォーマット	<p>使用する名前識別子の形式を定義します。</p> <p>例: urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:transient urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos urn:oasis:names:tc:SAML:2.0:nameid-format:entity</p>
サイン	<p>チェックボックスをマークして、SAML 署名を有効にする必要があるエンティティを選択します。</p> <p>メッセージ アサーション AuthN リクエスト ログアウトリクエスト ログアウトレスポンス</p>
暗号化	<p>チェックボックスをマークして、SAML 暗号化を有効にするエンティティを選択します:</p> <p>アサーション NameID</p>
ログイン時に大文字小文字を区別	<p>チェックボックスをマークして、ユーザー名の大文字と小文字を区別するログインを有効にします (デフォルトでは無効)。</p> <p>例: 大文字と小文字を区別するログインを無効にすると、たとえば、Zabbix ユーザーが 'Admin' であっても 'ADMIN' ユーザーでログインします。</p> <p>注大文字と小文字を区別するログインを無効にすると、Zabbix データベースに類似したユーザー名 (Admin、admin など) を持つ複数のユーザーが存在する場合、ログインが拒否されることに注意してください。</p>

高度な設定

追加の SAML パラメーターは、Zabbix フロントエンド設定ファイル (zabbix.conf.php) で設定できます。Additional SAML parameters can be configured in the Zabbix frontend configuration file (zabbix.conf.php):

- `$SSO['SP_KEY'] = '<SP 秘密鍵ファイルへのパス >';`
- `$SSO['SP_CERT'] = '<SP 証明書ファイルへのパス >';`
- `$SSO['IDP_CERT'] = '<IDP 証明書ファイルへのパス >';`
- `$SSO['SETTINGS']`

Note:

Zabbix は [OneLogin の SAMLPHPToolkit](#) ライブラリ (バージョン 3.4.1) を使用します。`$SSO['SETTINGS']` セクションの構造は、ライブラリで使用される構造と同様である必要があります。設定オプションの説明については、公式ライブラリ [ドキュメント](#) を参照してください。

`$SSO['SETTINGS']` の一部として設定できるのは、次のオプションです。

- `strict`
- `baseurl`
- `compress`
- `contactPerson`
- `organization`
- `sp` (only options specified in this list)
 - `attributeConsumingService`
 - `x509certNew`
- `idp` (only options specified in this list)
 - `singleLogoutService` (only one option)
 - * `responseUrl`
 - `certFingerprint`
 - `certFingerprintAlgorithm`
 - `x509certMulti`
- `security` (only options specified in this list)
 - `signMetadata`
 - `wantNameId`
 - `requestedAuthnContext`
 - `requestedAuthnContextComparison`
 - `wantXMLValidation`
 - `relaxDestinationValidation`

- destinationStrictlyMatches
- rejectUnsolicitedResponsesWithInResponseTo
- signatureAlgorithm
- digestAlgorithm
- lowercaseUrlencoding

他のすべてのオプションはデータベースから取得され、オーバーライドすることはできません。debug オプションは無視されます。

さらに、Zabbix UI がプロキシまたはロードバランサーの背後にある場合は、カスタムの use_proxy_headers オプションを使用できません。

- false (default) - オプションを無効に;
- true - use X-Forwarded-* ベース URL を構築するための HTTP ヘッダー。

ロードバランサーを使用して Zabbix インスタンスに接続する場合に、ロードバランサーは TLS/SSL を使用し、Zabbix は使用しない場合は、次のように「baseurl」「strict」および「use_proxy_headers」パラメーターを指定する必要があります。

```
$SSO['SETTINGS']=['strict' => false, 'baseurl' => "https://zabbix.example.com/zabbix/", 'use_proxy_headers'
```

設定例

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

4 ユーザーグループ

概要

管理 → ユーザーグループセクションでは、システムのユーザーグループをメンテナンスできます。

ユーザーグループ

既存のユーザーグループとその詳細のリストが表示されます。

Name ▲	#	Members	Frontend access	Debug mode	Status
<input type="checkbox"/> Disabled	Users 1	guest	System default	Disabled	Disabled
<input type="checkbox"/> Enabled debug mode	Users		System default	Enabled	Enabled
<input type="checkbox"/> Guests	Users 1	guest	Internal	Disabled	Enabled
<input type="checkbox"/> No access to the frontend	Users		Disabled	Disabled	Enabled
<input type="checkbox"/> Zabbix administrators	Users 1	Admin (Zabbix Administrator)	System default	Disabled	Enabled

表示情報

列	詳細
名前	ユーザーグループの名前。ユーザーグループ名をクリックすると、ユーザーグループの 設定フォーム が開きます。
#	グループ内のユーザーの数。ユーザーをクリックすると、ユーザーリストでフィルタされたそれぞれのユーザーが表示されます。
メンバー	ユーザーグループ内の個々のユーザー名。(括弧内にフルネームが表示されます) ユーザー名をクリックすると、ユーザー設定フォームが開きます。無効なグループのユーザーは赤で表示されます。

列	詳細
Web インターフェースへのアクセス	フロントエンドのアクセスレベルが表示されます。 システムのデフォルト - Zabbix、LDAP、または HTTP 認証。選択した認証方法に応じて Zabbix データベース内のユーザー情報 - システム設定に関係なくユーザーは Zabbix によって認証されます 無効 - このユーザーのフロントエンドアクセスは無効になります。 現在のレベルをクリックすると、レベルを変更できます。
デバッグモード	Debug mode のステータスが表示されます - 有効または無効。クリックでステータスを変更できます。
ステータス	ユーザーグループのステータスが表示されます - 有効または無効。クリックでステータスを変更できます。

新しいユーザーグループを構成するには、右上隅にあるユーザーグループの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - ユーザーグループのステータスを有効に
- 無効 - ユーザーグループのステータスを無効に
- デバッグモードを有効 - ユーザーグループのデバッグモードを有効に
- デバッグモードを無効 - ユーザーグループのデバッグモードを無効に
- 削除 - ユーザーグループを削除

これらのオプションを使用するには、それぞれのユーザーグループの前にチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、関心のあるユーザーグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、ユーザーグループのリストの上にあります。それをクリックすると、名前とステータスでユーザーグループをフィルタリングできます。

5 ユーザーの役割

概要

管理 → ユーザーの役割セクションでは、システムユーザーに割り当てることができる役割と各役割の権限を設定できます。

デフォルトユーザーロール

デフォルトでは Zabbix は事前定義された権限のセットを持つ 4 つのユーザーロールで設定されています。

- Admin role
- Guest role
- Super admin role
- User role

<input type="checkbox"/>	Name ▲	#	Users
<input type="checkbox"/>	Admin role	Users 1	db_manager (Database manager)
<input type="checkbox"/>	Guest role	Users	
<input checked="" type="checkbox"/>	Super admin role	Users 2	Admin (Zabbix Administrator), ljohnson (Lewis Johnson)
<input type="checkbox"/>	User role	Users 4	gslone (George Slone), guest (John Snow), test_admin, test_guest

Note:

Zabbix には無制限の権限を持つスーパー管理者ユーザーが少なくとも 1 人存在する必要があるため、デフォルトの Super admin role を変更または削除することはできません。

スーパー管理者権限を持つ Zabbix ユーザーは、既存のロールを変更または削除したり、新しいカスタムロールを作成したりできます。

新しい役割を作成するには、右上隅にあるユーザーの役割の作成ボタンをクリックします。既存の役割を更新するには、役割名を押して構成フォームを開きます。

* Name

User type

Access to UI elements

Monitoring	<input checked="" type="checkbox"/> Dashboard	<input checked="" type="checkbox"/> Hosts	<input checked="" type="checkbox"/> Maps
	<input checked="" type="checkbox"/> Problems	<input checked="" type="checkbox"/> Latest data	<input checked="" type="checkbox"/> Discovery
Services	<input checked="" type="checkbox"/> Services	<input checked="" type="checkbox"/> SLA	
	<input checked="" type="checkbox"/> Service actions	<input checked="" type="checkbox"/> SLA report	
Inventory	<input checked="" type="checkbox"/> Overview	<input checked="" type="checkbox"/> Hosts	
Reports	<input type="checkbox"/> System information	<input checked="" type="checkbox"/> Triggers top 100	<input checked="" type="checkbox"/> Notifications
	<input checked="" type="checkbox"/> Scheduled reports	<input type="checkbox"/> Audit	
	<input checked="" type="checkbox"/> Availability report	<input type="checkbox"/> Action log	
Configuration	<input checked="" type="checkbox"/> Host groups	<input checked="" type="checkbox"/> Maintenance	<input checked="" type="checkbox"/> Discovery
	<input checked="" type="checkbox"/> Templates	<input checked="" type="checkbox"/> Actions	
	<input checked="" type="checkbox"/> Hosts	<input type="checkbox"/> Event correlation	
Administration	<input type="checkbox"/> General	<input type="checkbox"/> User groups	<input type="checkbox"/> Media types
	<input type="checkbox"/> Proxies	<input type="checkbox"/> User roles	<input type="checkbox"/> Scripts
	<input type="checkbox"/> Authentication	<input type="checkbox"/> Users	<input type="checkbox"/> Queue

* At least one UI element must be checked.

Zabbix で事前設定されたユーザーロールのデフォルト権限セットとともに使用可能な権限オプションを以下に説明します。

パラメータ	説明	デフォルトユーザーロール
		Super admin role
名前	ロールの表示名	Super admin role

パラメータ	説明	デフォルトユーザーロール			
ユーザーの種類	選択したユーザータイプによって、使用可能な権限のリストが決まります。 ユーザータイプを選択すると、このユーザータイプで使用可能なすべての権限がデフォルトで付与されます。 チェックボックスをオフにすると、ユーザーロールの特定の権限が取り消されます。 < br> このユーザータイプで使用できない権限のチェックボックスはグレー表示されません。	Super admin		Admin	User User
UI 要素へのアクセス					
監視データ					
ダッシュボード	特定の監視メニューセクションと基になるページへのアクセスを有効/無効にします。	Yes	Yes	Yes	Yes
障害					
ホスト					
最新データ					
マップ					
ディスカバリ				No	No
サービス					
サービス	特定のサービスメニューセクションと基になるページへのアクセスを有効/無効にします。	Yes	Yes	Yes	Yes
サービスアクション				No	No
SLA					
SLA レポート				Yes	Yes
インベントリ					
概要	特定のインベントリメニューセクションと基になるページへのアクセスを有効/無効にします。	Yes	Yes	Yes	Yes
ホスト					
レポート					
システム情報	特定のレポートメニューセクションと基になるページへのアクセスを有効/無効にします。	Yes	No	No	No
稼働レポート			Yes	Yes	Yes
障害発生数上位 100 項目					
監査			No	No	No
アクションログ					
通知レポート			Yes		
定期レポート					
設定					
ホストグループ	特定の設定メニューセクションと基になるページへのアクセスを有効/無効にします。	Yes	Yes	No	No
テンプレート					
ホスト					
メンテナンス					
アクション					
イベント相関関係				No	
ディスカバリ				Yes	
管理					

パラメータ	説明	デフォルトユーザーロール			
一般設定	特定の管理メニューセクションと基になるページへのアクセスを有効/無効にします。	Yes	No	No	No
プロキシ 認証 ユーザーグループ ユーザーの役割 ユーザー メディアタイプ スクリプト キュー					
デフォルトで新しい UI 要素にアクセス	カスタム UI 要素へのアクセスを有効/無効にします。モジュールが存在する場合は、以下にリストされます。	Yes	Yes	Yes	Yes
サービスへのアクセス サービスの表示/設定	読み取り/書き込み可能なサービスを選択します： なし - アクセスなし すべて - すべてのサービスへ読み取り/書き込み可能 サービスのリスト - 読み取り/書き込みアクセス用のサービスを選択します	All	All	None	None
タグとサービスの表示/設定	読み取り/書き込みアクセスが許可されている場合、読み取り専用アクセス設定よりも優先され、子サービスによって動的に継承されます。 タグ名とオプションで値を指定して、タグに一致するサービスへの読み取り/書き込みアクセスを追加で許可します。 このオプションは、サービスへの読み取り/書き込みアクセスパラメーターでサービスリストが選択されている場合に使用できます。 読み取り/書き込みアクセスが許可されている場合、読み取り専用アクセス設定よりも優先され、子サービスによって動的に継承されます。				
サービスの表示のみ	表示のみのサービスを選択します。 なし - アクセスなし すべて - 全てのサービスは表示のみ サービスのリスト - 表示のみのサービスを選択する。 表示のみは読み取り/書き込みアクセスよりも優先されず、子サービスによって動的に継承されます。		All	All	

パラメータ	説明	デフォルトユーザーロール
タグとサービスの表示のみ	<p>タグ名とオプションで値を指定して、タグに一致する表示のみのサービスを追加します。</p> <p>このオプションは、サービスの表示のみパラメーターでサービスリストが選択されている場合に使用できます。</p> <p>表示のみは読み取り/書き込みアクセスよりも優先されず、子サービスによって動的に継承されます。</p>	
モジュールへのアクセス		
< モジュール名 >	<p>特定のモジュールへのアクセスを許可/拒否します。このセクションには、有効なモジュールのみが表示されます。現在無効になっているモジュールへのアクセスを許可または制限することはできません。</p> <p>追加されたモジュールへのアクセスをデフォルトで有効/無効にします。</p>	Yes
デフォルトで新しいモジュールに対してアクセス		Yes Yes Yes
API へのアクセス		
有効	<p>API へのアクセスを有効/無効にします。</p>	Yes
API メソッド	<p>リストを許可を選択して指定した API メソッドのみを許可するか、リストを拒否を選択して指定した API メソッドのみを制限します。</p> <p>検索フィールドでメソッド名の入力を開始し、オートコンプリートリストからメソッドを選択します。。</p> <p>または選択ボタンを押して、このユーザータイプで使用可能なリストからメソッドを選択することもできます。アクションへのアクセスブロックの特定のアクションがオフになっている場合、ユーザーはこのアクションに関連する API メソッドを使用できないことに注意してください。</p> <p>ワイルドカードがサポートされています。</p> <p>例:dashboard.* ('dashboard.'API サービスのすべてのメソッド) * (任意のメソッド)、*.export (すべての API サービスの'.export' 名のメソッド)</p> <p>
メソッドが指定されていない場合、許可/拒否リストルールは無視されます。</p>	Yes Yes No
アクションへのアクセス		

パラメータ	説明	デフォルトユーザーロール			
ダッシュボードの作成と変更	このチェックボックスをオフにすると、対応する要素に対して <code>.create</code> 、 <code>.update</code> および <code>.delete</code> API メソッドを使用する権限も取り消されます。	Yes	Yes	Yes	No
マップの作成と変更 メンテナンス機関の作成と変更					No
障害対応メッセージの追加	このチェックボックスをオフにすると、 <code>event.acknowledge</code> API メソッドを介して対応するアクションを実行する権限も取り消されます。			Yes	
深刻度の変更 障害の確認 障害のクローズ スクリプトの実行	このチェックボックスをオフにすると、 <code>script.execute</code> API メソッドを使用する権限も取り消されます。				
API トークンの管理	このチェックボックスをオフにすると、すべての <code>token.API</code> メソッドを使用する権限も取り消されます。				
定期レポートの管理	このチェックボックスをオフにすると、すべての <code>report.API</code> メソッドを使用する権限も取り消されます。				No
SLA の管理	SLA を管理する権限を有効/無効にします。				
デフォルトで新しいアクションに対してアクセス	新しいアクションへのアクセスをデフォルトで有効/無効にします。				Yes

情報:

- 各ユーザーには、1つの役割のみを割り当てることができます。
- 要素が制限されている場合、ユーザーはこの要素への直接 URL をブラウザに入力してもその要素にアクセスできません。
- タイプ `User` または `Admin` のユーザーは、自分の役割設定を変更できません。
- タイプ `スーパー管理者` のユーザーは、自分のロールの設定を変更できますがユーザータイプは変更できません。(ただしデフォルトのスーパー管理者ロールでは変更できません)
- すべてのレベルのユーザーは、自分のユーザータイプを変更することはできません。
- ユーザーの設定**も参照

6 ユーザー

概要

管理 → ユーザーセクションでシステムのユーザーを管理できます。

ユーザー

詳細を含む既存のユーザーのリストが表示されます。

Users User group: All [Create user](#)

<input type="checkbox"/>	Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access	API access	Debug mode	Status
<input type="checkbox"/>	Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (03/02/2021 02:35:19 PM)	Ok	System default	Enabled	Disabled	Enabled
<input type="checkbox"/>	Database manager	James	Hughes	Admin role	DB administrators	No	Ok	System default	Enabled	Disabled	Enabled
<input type="checkbox"/>	guest			User role	Disabled, Guests	No	Ok	Internal	Enabled	Disabled	Disabled

0 selected [Unblock](#) [Delete](#) Displaying 3 of 3 found

ユーザーバーの右側のドロップダウンから、すべてのユーザーを表示するか、特定のグループに属するユーザーを表示するかを選択できます。

表示データ

列	詳細
エイリアス	Zabbix にログインするためのユーザー名。ユーザー名をクリックすると、ユーザー設定フォームが開きます
名	ユーザーの名前です
姓	ユーザーの苗字です
ユーザーの種類グループ	ユーザーロールを表示します ユーザーがメンバーになっているグループが一覧表示されます。ユーザーグループ名をクリックすると、ユーザーグループ設定フォームが開きます。無効になっているグループは赤で表示されます。
ログイン状態	ユーザーのオンラインステータスが表示されます - はいまたはいいえ。最後のユーザーアクティビティの時刻は括弧内に表示されます。
ログイン	ユーザーのログインステータスが表示されます - OK または Blocked。管理 → 一般セクションで設定されたログイン試行の失敗回数（デフォルトでは 5 回）を超えると、ユーザーは一時的にブロックされます。ブロックをクリックすると、ユーザーのブロックを解除できます。
Web インターフェースへのアクセス	フロントエンドのアクセスレベルが表示されます - ユーザーグループ全体に設定されているものに応じてシステムデフォルト,Zabbix データベース内のユーザー情報、または無効が表示されます。
API アクセス	API アクセスステータスが表示されます - ユーザーロールに設定されているものに応じて有効または無効が表示されます。
デバッグモード	デバッグモードのステータスが表示されます - ユーザーグループ全体に設定されているものに応じて有効または無効が表示されます。
ステータス	ユーザーステータスが表示されます - ユーザーグループ全体に設定されているものに応じて有効または無効が表示されます。

新しいユーザーを構成するには、右上隅にあるユーザーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- ブロックを解除 - ブロックされたユーザーのシステムアクセスを再度有効にします
- 削除 - ユーザーを削除します

これらのオプションを使用するには、それぞれのユーザーの前にチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルタを使用して、関心のあるユーザーのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、ユーザーのリストの上にあります。それをクリックすると、エイリアス、名前、姓、およびユーザーの役割でユーザーをフィルターできるフィルターが使用可能になります。

Filter

Username <input style="width: 80%;" type="text"/>	Name <input style="width: 80%;" type="text"/>	Last name <input style="width: 80%;" type="text"/>	User roles <input style="width: 95%;" type="text" value="type here to search"/>	Select
Apply Reset				

7 メディアタイプ

概要

管理 → メディアタイプセクションで、ユーザーはメディアタイプ情報を設定および保守できます。

メディアタイプ情報には、通知の配信チャネルとしてメディアを使用するための一般的な手順が含まれています。通知を送信する個々の電子メールアドレスなどの特定の詳細は、個々のユーザーに保持されます。

既存のメディアタイプとその詳細のリストが表示されます。

<input type="checkbox"/>	Name ▲	Type	Status	Used in actions	Details	Action
<input type="checkbox"/>	Email	Email	Enabled		SMTP server: "mail.zabbix.com", SMTP helo: "zabbix.com", SMTP email: "zabbix-info@zabbix.com"	Test
<input type="checkbox"/>	Email (HTML)	Email	Enabled		SMTP server: "mail.example.com", SMTP helo: "example.com", SMTP email: "zabbix@example.com"	Test
<input type="checkbox"/>	Mattermost	Webhook	Enabled			Test
<input type="checkbox"/>	Notification script	Script	Enabled		Script name: "notification.sh"	Test
<input type="checkbox"/>	Opsgenie	Webhook	Enabled			Test
<input type="checkbox"/>	PagerDuty	Webhook	Enabled			Test
<input type="checkbox"/>	Pushover	Webhook	Enabled			Test
<input type="checkbox"/>	SMS	SMS	Enabled		GSM modem: "udev/ttyS0"	Test

表示データ

列	詳細
名前	メディアタイプの名前。名前をクリックすると、メディアタイプ構成フォームが開きます。
タイプ	メディアタイプを表示します。(e-mail, SMS, 他)
ステータス	メディアタイプのステータスが表示されます - 有効または無効 クリックでステータスを変更できます。
アクションで使用	[送信先] ドロップダウンで選択されているすべてのアクションが表示されます。 アクション名をクリックすると、アクション構成フォームが開きます。
詳細	メディアタイプの詳細情報が表示されます。
アクション	次のアクションを使用できます。 テスト - クリックしてテストフォームを開き、メディアタイプパラメータ（たとえば、テストの件名と本文を含む受信者アドレス）を入力し、テストメッセージを送信して、設定されたメディアタイプの動作確認をします。参照： メディアタイプテスト

新しいメディアタイプを設定するには、右上隅にあるメディアタイプの作成ボタンをクリックします。

XML からメディアタイプをインポートするには、右上隅にあるインポートボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。


- 有効 - メディアタイプのステータスを Enabled に変更します
- 無効 - メディアタイプのステータスを Disabled に変更します
- エクスポート - メディアタイプを YAML, XML または JSON ファイルとしてエクスポートします
- 削除 - メディアタイプを削除します

これらのオプションを使用するには、それぞれのメディアタイプの前のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

You can use the filter to display only the media types you are interested in. For better search performance, data is searched with macros unresolved.

フィルターリンクは、メディアタイプのリストの上にあります。それをクリックすると、名前とステータスでメディアタイプをフィルタリングできるフィルタが利用可能になります。

Filter 

Name
 Status Any Enabled Disabled

Apply Reset

8 スクリプト

概要

管理 → スクリプトセクションでは、ユーザー定義のグローバルスクリプトを設定および保守できます。

構成されたスコープとユーザー権限に応じて、グローバルスクリプトを実行できます。

- さまざまなフロントエンドの場所にある **ホストメニュー** から (ダッシュボード, 障害, 最新データ, マップ, 他)
- **イベントメニュー** から
- アクション操作として

スクリプトは Zabbix エージェント、Zabbix プロキシ、または Zabbix サーバーでのみ実行されます。 **コマンド実行** も参照してください。

Zabbix エージェントと Zabbix プロキシリモートスクリプトの両方がデフォルトで無効になっています。それらは次の方法で有効にできます。

- Zabbix エージェントで実行されるリモートコマンドの場合
 - エージェント構成で許可されたコマンドごとに AllowKey=system.run[< コマンド >,*] パラメーターを追加します。* は待機モードと非待機モードを表します。
- Zabbix プロキシで実行されるリモートコマンドの場合
 - 警告: **Zabbix** プロキシによって監視されている **Zabbix** エージェントでリモートコマンドを実行する場合は、**Zabbix** プロキシでリモートコマンドを有効にする必要はありません。ただし、Zabbix プロキシでリモートコマンドを実行する必要がある場合は、プロキシ設定の EnableRemoteCommands パラメーターを 1 に設定します。

既存のスクリプトとその詳細のリストが表示されます。

☰ Scripts
Create script

Name ▼	Scope	Used in actions	Type	Execute on	Commands	User group	Host group	Host access
<input type="checkbox"/> Traceroute	Manual host action		Script	Server (proxy)	/usr/bin/traceroute {HOST.CONN}	All	All	Read
<input type="checkbox"/> Restart webserver	Action operation		Script	Agent	sudo /etc/init.d/apache2 restart	All	All	Read
<input type="checkbox"/> Detect operating system	Manual host action		Script	Server (proxy)	sudo /usr/bin/nmap -0 {HOST.CONN}	Zabbix administrators	All	Read

Displaying 3 of 3 found

表示データ

列	説明
Name	スクリプトの名前。スクリプト名をクリックすると、スクリプト設定フォームが開きます
Scope	スクリプトの範囲。アクション操作、手動ホストアクション、または手動イベントアクション。この設定により、スクリプトを使用できる場所が決まります。
Used in actions	スクリプトが使用されているアクションが表示されます。
Type	スクリプトタイプが表示されます - Webhook、* Script、SSH、Telnet、または IPMI* コマンド。
Execute on	スクリプトが Zabbix エージェント、Zabbix プロキシ、または Zabbix サーバーの何処で実行されるかが表示されます。
Commands	スクリプト内で実行されるすべてのコマンドが表示されます。
User group	スクリプトを使用できるユーザーグループが表示されます (すべてのユーザーグループの場合はすべて)
Host group	スクリプトを使用できるホストグループが表示されます (すべてのホストグループの場合はすべて)
Host access	ホストグループのアクセス許可レベルが表示されます - 読み取りまたは書き込み。必要な権限レベルを持つユーザーのみが、スクリプトの実行にアクセスできます。

新しいスクリプトを構成するには、右上隅にあるスクリプトの作成ボタンをクリックします。

一括編集オプション

リストの下にあるボタンは、1つの一括編集オプションを提供します。

- 削除 - スクリプトを削除します

このオプションを使用するには、それぞれのスクリプトの前にあるチェックボックスをオンにして、削除をクリックします。

フィルタの使用

フィルタを使用して、関心のあるスクリプトのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

Filter リンクは、スクリプトのリストの上にあります。それをクリックすると、名前とスコープでスクリプトをフィルタリングできるフィルターが使用可能になります。

The screenshot shows a filter interface with a search box labeled 'Name', a 'Scope' section with buttons for 'Any', 'Action operation', 'Manual host action', and 'Manual event action', and 'Apply' and 'Reset' buttons at the bottom.

グローバルスクリプトの設定

The screenshot shows a configuration form for a global script. The fields are: Name (Restart webserver), Scope (Action operation, Manual host action, Manual event action), Menu path (<sub-menu/sub-menu/...>), Type (Webhook, Script, SSH, Telnet, IPMI), Execute on (Zabbix agent, Zabbix server (proxy), Zabbix server), Commands (sudo /etc/init.d/apache2 restart), Description (empty), Host group (All), User group (All), Required host permissions (Read, Write), Enable confirmation (checkbox), and Confirmation text (empty). 'Add' and 'Cancel' buttons are at the bottom.

スクリプト属性

パラメータ	詳細
名前	一意なスクリプト名 例：Clear /tmp filesystem
範囲	スクリプトの範囲 - アクション処理、手動ホスト処理、または手動イベント処理。この設定は、スクリプトを使用できる場所を決定します - アクション操作のリモートコマンドで、 ホストメニュー または イベントメニュー 範囲を'アクション処理'に設定すると、構成→アクションにアクセスできるすべてのユーザーがスクリプトを使用できるようになります。 スクリプトが実際にアクションの場合、その範囲を「アクション処理」から変更することはできません。 マクロのサポート 範囲は、使用可能なマクロの範囲に影響します。たとえば、ユーザー関連のマクロ ({USER.*}) はスクリプトでサポートされており、スクリプトを起動したユーザーに関する情報を渡すことができます。ただし、スクリプトスコープがアクション処理の場合は、アクション処理が自動的に実行されるため、サポートされていません。 サポートされているマクロを確認するには、「トリガーベースの通知とコマンド/トリガーベースのコマンド」を検索してください。 サポートされているマクロ テーブルの「手動ホスト処理スクリプト」および「手動イベント処理スクリプト」。マクロがスペースを含む値(ホスト名など)に解決される可能性がある場合は必要に応じて引用符で囲むことを忘れないでください。 スクリプトへの目的のメニューパス。たとえば Default または Default /は、それぞれのディレクトリにスクリプトを表示します。メニューはネストできます。たとえばメインメニュー/サブメニュー 1/サブメニュー 2。監視セクションのホスト/イベントメニューからスクリプトにアクセスすると、指定されたディレクトリに従ってスクリプトが整理されます。 このフィールドは、[手動ホスト処理]または[手動イベント処理]が範囲として選択されている場合にのみ表示されます。
メニューパス	ボタンをクリックして、スクリプトタイプを選択します： Webhook,Script,SSH,Telnet または **IPMI** コマンド
タイプ	
スクリプトタイプ: Webhookパラメータ	Webhook 変数を属性と値のペアとして指定します。 関連項目： Webhook メディア設定
スクリプト	マクロとカスタム ユーザーマクロ はパラメーター値でサポートされています。マクロのサポートはスクリプトの範囲によって異なります(上記の範囲を参照)パラメータフィールド(またはその横にある表示/編集ボタン)をクリックすると表示されるブロックにJavaScript コードを入力します。 マクロのサポートはスクリプトの範囲によって異なります(上記の範囲を参照) 関連項目： Webhook メディア設定, 追加の Javascript オブジェクト

パラメータ	詳細
タイムアウト	JavaScript 実行タイムアウト (1~60 秒、デフォルトは 30 秒) タイムサフィックスがサポートされています。例: 30s, 1m
スクリプトタイプ: スクリプト 次で実行	それぞれのボタンをクリックしてシェルスクリプトを実行します。 ** Zabbix エージェント ** - スクリプトはホスト上の Zabbix エージェントによって実行されます (system.run アイテムが許可されている場合) ** Zabbix サーバー (プロキシ) ** - スクリプトは Zabbix サーバーまたはプロキシによって実行されます (EnableRemoteCommandsによって有効になっている場合) - ホストはサーバーまたはプロキシによって監視されます Zabbix サーバー - スクリプトは Zabbix サーバーによってのみ実行されます。
コマンド	スクリプト内で実行するコマンドへのフルパスを入力します。 マクロのサポートはスクリプトの範囲によって異なります (上記の範囲を参照)。カスタムユーザーマクロがサポートされています。
スクリプトタイプ: SSH	認証方法を選択します - パスワードまたは公開鍵
認証方法 ユーザー名	ユーザー名を入力します
パスワード	パスワードを入力します このフィールドは、認証方法として 'パスワード' が選択されている場合に使用できます。
公開鍵ファイル	公開鍵ファイルへのパスを入力します。 このフィールドは、認証方法として '公開鍵' が選択されている場合に使用できます。
秘密鍵ファイル	秘密鍵ファイルへのパスを入力します。 このフィールドは、認証方法として '公開鍵' が選択されている場合に使用できます。
キーのパスフレーズ	パスフレーズを入力します。 このフィールドは認証方法として '公開鍵' が選択されている場合に使用できます。
ポート	ポート番号を入力します
コマンド	コマンドを入力します。 マクロのサポートはスクリプトの範囲によって異なります。(上記の範囲を参照) カスタムユーザーマクロがサポートされています。

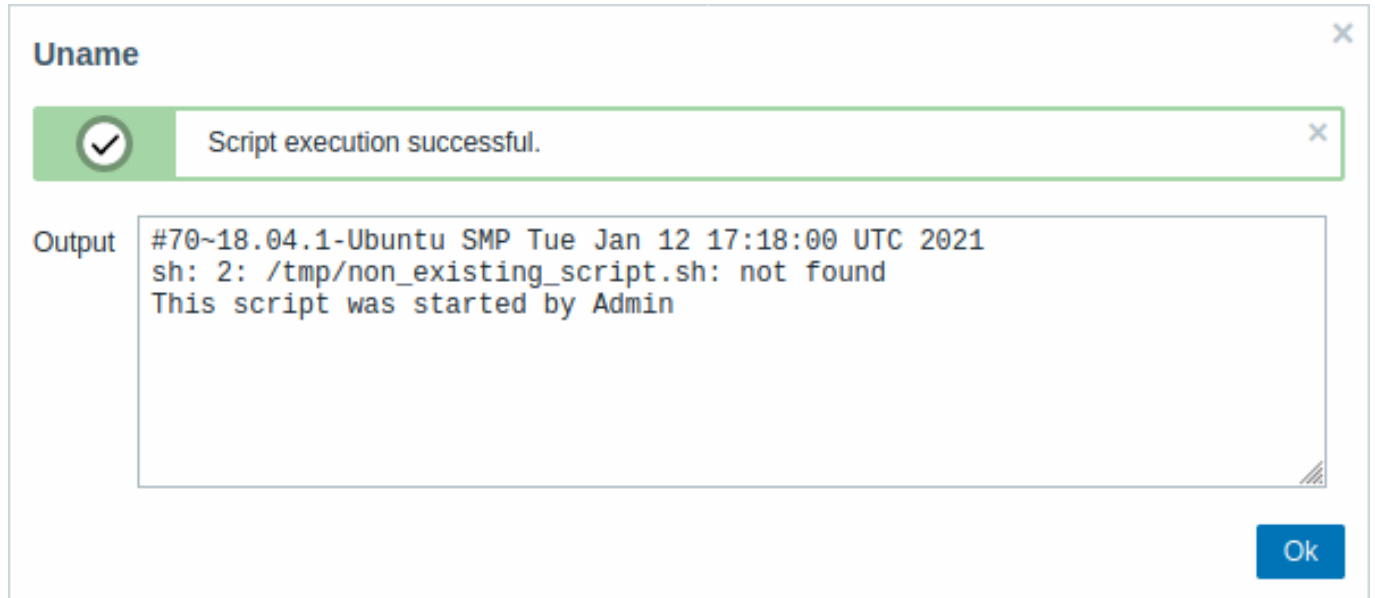
パラメータ	詳細
スクリプトタイプ: Telnet ユーザー名 パスワード ポート コマンド	ユーザー名を入力します パスワードを入力します ポート番号を入力します コマンドを入力します。 マクロのサポートはスクリプトの範囲によって異なります。(上記の範囲を参照) カスタムユーザーマクロがサポートされています。
スクリプトタイプ: IPMI コマンド	IPMI コマンドを入力します。 マクロのサポートはスクリプトの範囲によって異なります。(上記の範囲を参照) カスタムユーザーマクロがサポートされています。
説明	スクリプトの説明
ホストグループ	スクリプトを使用できるホストグループを選択します (すべてのホストグループの場合はすべて)
ユーザーグループ	スクリプトを使用できるユーザーグループを選択します。(すべてのユーザーグループの場合はすべて) このフィールドは範囲が「手動ホスト処理」または「手動イベント処理」の時に表示されます。
必要なホストへのアクセス権	ホストグループのアクセス許可レベルを選択します - 読み取りまたは書き込み。必要な権限レベルを持つユーザーのみがスクリプトの実行にアクセスできます。 このフィールドは範囲が「手動ホスト処理」または「手動イベント処理」の時に表示されます。
確認を有効	スクリプトを実行する前に確認メッセージを表示するには、チェックボックスをオンにします。この機能は、潜在的に危険な操作 (再起動スクリプトなど) や時間がかかる可能性のある操作で特に役立つ場合があります。 このオプションは範囲が「手動ホスト処理」または「手動イベント処理」の時に表示されます。
確認テキスト	上記のチェックボックスで有効にした確認ポップアップのカスタム確認テキストを入力します (例: リモートシステムが再起動されます。よろしいですか?)。テキストがどのように表示されるかを確認するには、フィールドの横にあるテストの確認をクリックします。{HOST.*} および {USER.*} マクロがサポートされています。カスタムユーザーマクロがサポートされています。 注: 確認メッセージのテスト時にマクロは展開されません。 このフィールドは範囲が「手動ホスト処理」または「手動イベント処理」の時に表示されます。

Zabbix サーバーで実行されるスクリプトは、終了コードの確認を含め**コマンド実行**セクションで説明されている順序で実行されます。スクリプトの結果は、スクリプトの実行後に表示されるポップアップウィンドウに表示されます。

Note: スクリプトの戻り値は標準エラーとともに標準出力です。

以下のスクリプトと結果ウィンドウの例を参照してください。

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



スクリプトの結果にスクリプト自体は表示されません。

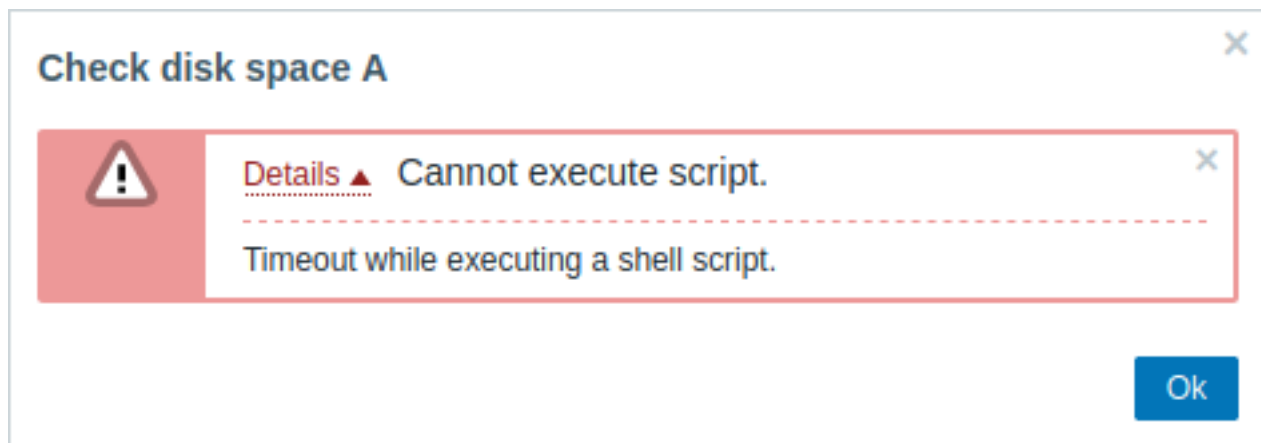
スクリプトのタイムアウト

Zabbix エージェント

スクリプトの実行中にタイムアウトが発生する場合があります。

Zabbix エージェントで実行されるスクリプトの例と以下の結果ウィンドウを参照してください。

```
sleep 5
df -h
```



この場合のエラーメッセージは次のとおりです。

Timeout while executing a shell script.

このような状況を回避するためにスクリプト自体を最適化することをお勧めします。(Zabbix エージェント設定とZabbix サーバー設定の Timeout パラメータを対応する値 (この場合は > '5') に調整するものではありません)

Zabbix エージェント設定でタイムアウトパラメータが変更された場合は、次のエラーメッセージが表示されます。

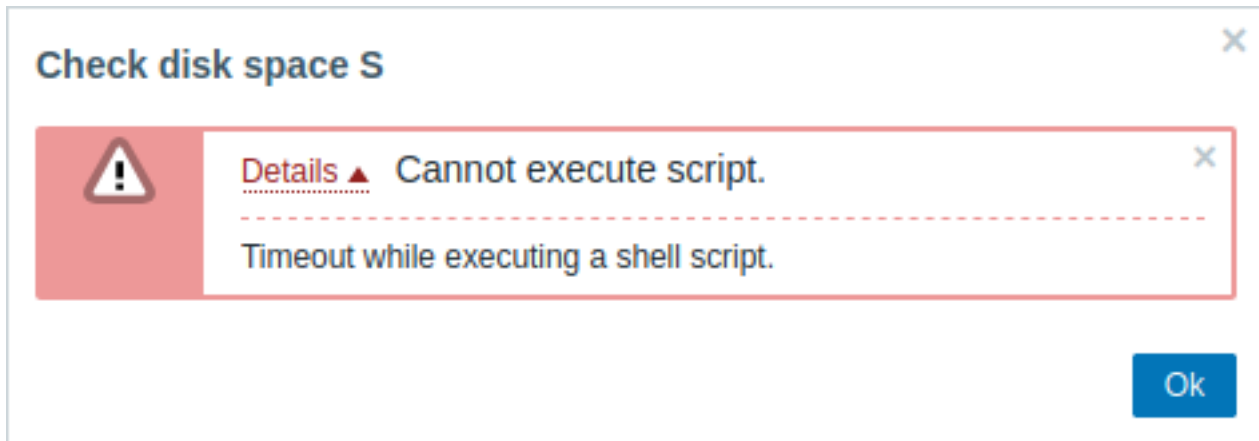
Get value from agent failed: ZBX_TCP_READ() timed out.

これは、Zabbix エージェント設定で変更が行われたことを意味し、Zabbix サーバー設定でもタイムアウト設定を変更する必要があります。

Zabbix サーバーとプロキシ

Zabbix サーバーで実行されているスクリプトの例と以下の結果ウィンドウを参照してください。

```
sleep 11
df -h
```



このような状況を回避するためにスクリプト自体を最適化することをお勧めします。(Zabbix サーバー設定の Timeout パラメーターを対応する値 (この場合は > '11') に調整するものではありません)

9 キュー

概要

管理 → キューセクションには、更新を待機しているアイテムが表示されます。

このセクションを開いた際に、キューにアイテムがないことを意味する「緑色」ですべて表示されているのが理想です。すべてのアイテムが遅滞なく更新されているのであれば待機しているものはありません。しかし、サーバーのパフォーマンスの低下、接続の問題、またはエージェントの問題により、一部の項目が遅延する可能性があります。それらの情報がこのセクションに表示されます。詳細については、[キューセクション](#)を参照してください。

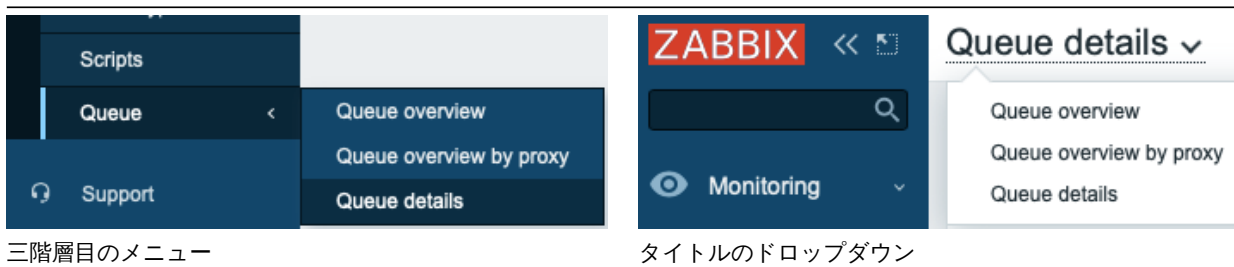
Note:

キューは、Zabbix サーバーが実行されている場合のみ使用できます。

管理 → キューセクションには、次のページが含まれています。

- キュー概要 — アイテムタイプごとにキューを表示します
- プロキシごとのキュー概要 — プロキシごとにキューを表示します
- キュー詳細 — 遅延アイテムのリストを表示します。

使用可能なページのリストは管理メニューセクションのキュー]を押すと表示されます。左上隅のタイトルドロップダウンを使用してページを切り替えることもできます。



三階層目のメニュー

タイトルのドロップダウン

キュー概要

この画面では、問題が1つまたは複数のアイテムタイプに関連しているかどうかを簡単に特定できます。

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

各行にはアイテムタイプが含まれています。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5~10 秒/10~30 秒/30~60 秒/1~5 分/5~10 分または 10 分以上です。

プロキシごとのキュー概要

この画面では、問題がプロキシの 1 つに関連しているか、サーバーに関連しているかを簡単に特定できます。

Queue overview by proxy

Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

Total: 2

各行にはプロキシが含まれ、サーバーはリストの最後にあります。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5~10 秒/10~30 秒/30~60 秒/1~5 分/5~10 分または 10 分以上です。

キュー詳細

この画面では、待機中の各アイテムが一覧表示されます。

Queue details

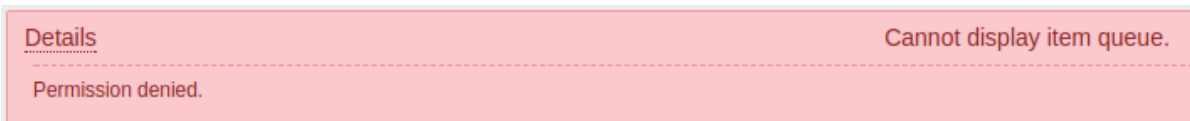
Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

表示データ

列	説明
Scheduled check	チェック日が表示されます
Delayed by	遅延期間が表示されます
Host	アイテムのホストが表示されます
Name	待機中のアイテム名が表示されます
Proxy	ホストがプロキシによって監視されている場合は、プロキシ名が表示されます。

表示可能性のあるエラーメッセージ

データが表示されず、次のエラーメッセージが表示される場合があります。



この場合のエラーメッセージは次のとおりです。

Cannot display item queue. Permission denied


これは、zabbix.conf.php の PHP 構成パラメーター \$ZBX_SERVER_PORT または \$ZBX_SERVER が、異なるデータベースを使用する既存の Zabbix サーバーを指している場合に発生します。

3 ユーザー設定

概要

ユーザーロールの権限によってはユーザー設定セクションに次のページが含まれる場合があります。

- ユーザープロフィール - 特定の Zabbix フロントエンド機能をカスタマイズ
- API トークン - 現在のユーザーに割り当てられた API トークンを管理

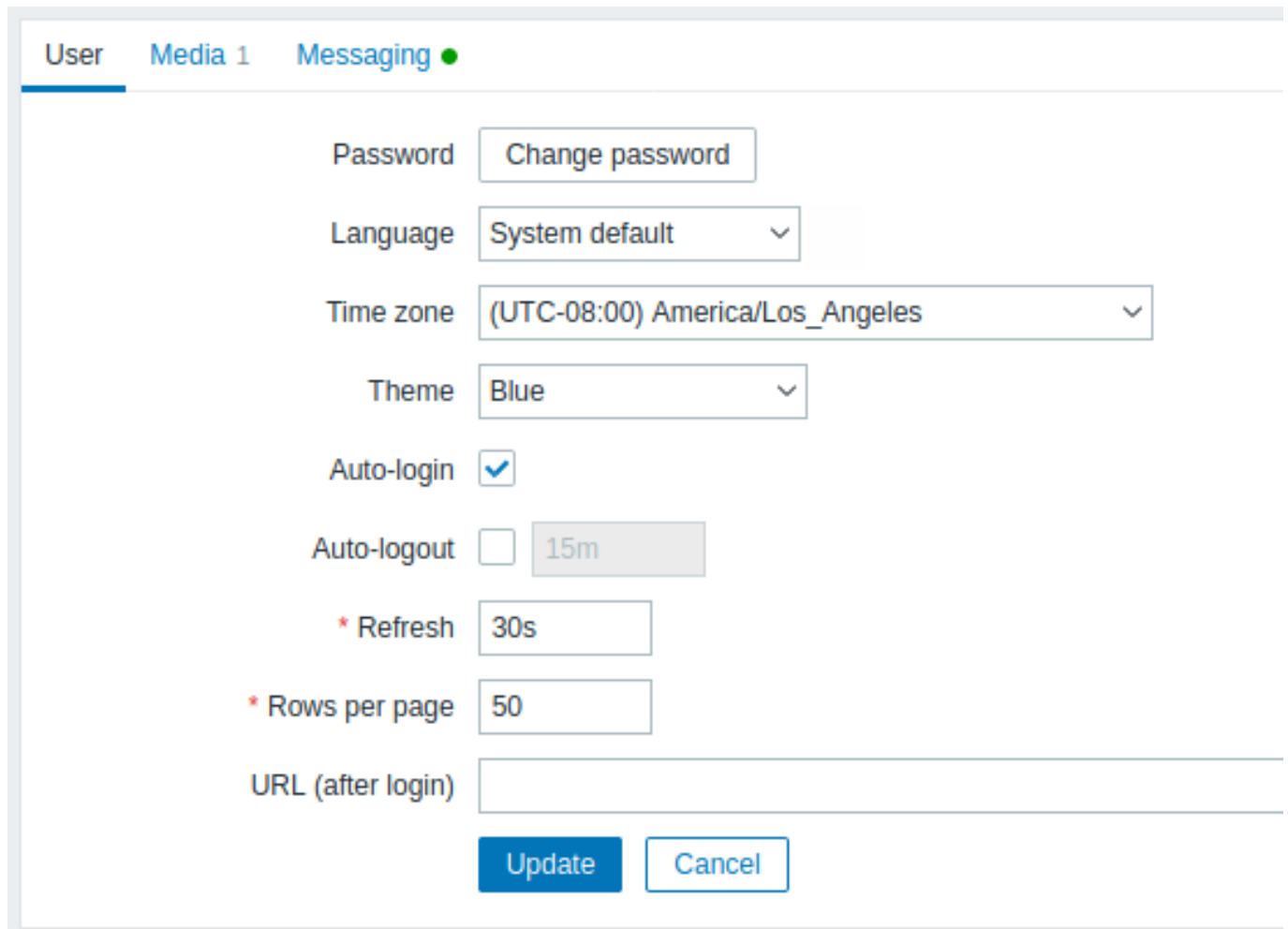
利用可能なページのリストは、Zabbix メニューの下部にある  ユーザーアイコンを押すと表示されます。(ゲストユーザーは利用できません) 左上隅のタイトルドロップダウンを使用してページを切り替えることもできます。



1 ユーザープロフィール

ユーザープロフィールセクションには、カスタムインターフェイス言語、カラーテーマ、リストに表示される行数などを設定するオプションがあります。ここで行った変更は、現在のユーザーにのみ適用されます。

ユーザータブでは、さまざまなユーザー設定を変更できます。



パラメータ	説明
パスワード	リンクをクリックすると新しいパスワードを入力するための2つのフィールドが表示されます。
言語設定	インターフェースの言語を選択するかシステムデフォルトを選択してデフォルトのシステム設定を使用します。 詳細については 追加のフロントエンド言語のインストール を参照してください。
タイムゾーン	ユーザーレベルでグローバルタイムゾーンを上書きするタイムゾーンを選択するか、システムデフォルトを選択してグローバルタイムゾーン設定を使用します。
テーマ	プロファイル専用のカラーテーマを選択します。 system default - システムのデフォルト設定を使用します Blue - 標準的な青のテーマ Dark - 代替のダークテーマ High-contrast light - コントラストの高い明るいテーマ High-contrast dark - コントラストの高いダークテーマ
自動ログイン	このチェックボックスをオンにすると、Zabbixにあなたを記憶させ、30日間自動的にログインさせます。これにはブラウザのCookieが使用されます。
自動ログアウト	このチェックボックスをオンにすると、設定した秒数（最小90秒、最大1日）が経過すると、自動的にログアウトされます。 タイムサフィックス がサポートされています。例：90s, 5m, 2h, 1d。 次の場合は自動ログアウトが機能しないことに注意してください。 * 監視メニューページが背景情報の更新を実行する場合、特定の時間間隔（ダッシュボード、グラフ、最新データなど）でデータを更新するページが開いたままになっているとセッションの有効期間が延長され、自動ログアウト機能が無効になります。 *30日間記憶する * オプションをオンにしてログインする場合。 自動ログアウトに0と設定するとプロファイル設定の更新後に自動ログアウトが無効になります。
更新間隔	ウィジェットごとに独自の更新パラメータを使用するダッシュボードを除いて、ページ内の情報が更新される頻度を監視メニューで設定できます。 タイムサフィックス がサポートされています。例：90s, 5m, 2h, 1d。
ページごとの行数	リストのページごとに表示される行数を設定できます。行が少ない（そして表示するレコードが少ない）ということは、ロード時間が短縮されることを意味します。
URL (ログイン後)	ログイン後に表示される特定のURLを設定できます。デフォルトのMonitoring→Dashboardの代わりに、たとえば、Monitoring→TriggersのURLにすることができます。

メディアタブでは、タイプ、使用するアドレス、通知の配信タイミングなど、[ユーザーメディアの詳細](#)を設定できます。

Note:
管理者レベルユーザー（管理者およびスーパー管理者）のみが、自分のメディアの詳細を変更できます。

メッセージタブでは[グローバル通知](#)を設定できます。

2 API トークン

API トークンセクションでは、ユーザーに割り当てられたトークンを表示したり、トークンの詳細を編集したり、[新しいトークンを作成](#)したりできます。このセクションは[ユーザーロール設定](#)でAPI トークンの管理アクションが許可されているユーザーのみ利用できます。

<input type="checkbox"/>	Name ▲	Expires at	Created at	Last accessed at	Status
<input type="checkbox"/>	Token 1	Never	2021-01-22 18:58:11	Never	Enabled
<input type="checkbox"/>	Token 2	2021-01-26 00:00:00	2021-01-22 16:13:03	Never	Enabled

Displaying 2 of 2 found

API トークンは、名前、有効期限、またはステータス（有効/無効）でフィルタリングできます。リスト内のトークンステータスをクリックして、トークンを即時、有効/無効にすることもできます。リストでトークンを選択し、リストの下にある有効化/無効化ボタンをクリックして、トークンを一括で有効化/無効化することもできます。

Attention:

ユーザーは、Zabbix で自分に割り当てられたトークンの認証トークン値を表示できません。認証トークン値は、トークンを作成した直後 1 回だけ表示されます。紛失した場合はトークンを再生成する必要があります。

1 グローバル通知

概要

グローバル通知は、Zabbix フロントエンドで現在発生している問題を画面に通知する方法です。

グローバル通知がない場合、問題またはダッシュボード以外の場所で作業しても、現在発生している問題に関する情報は表示されません。グローバル通知は、あなたがどこにいるかに関係なく、この情報を表示します。

グローバル通知には、メッセージの表示とサウンドの再生の両方が含まれます。

Attention:

最近のブラウザバージョンでは、サウンドの自動再生がデフォルトで無効になっている場合があります。この場合、この設定を手動で変更する必要があります。

設定

プロファイル設定のメッセージタブで、ユーザーごとにグローバル通知を有効にできます

Frontend messaging

Message timeout

Play sound

Trigger severity

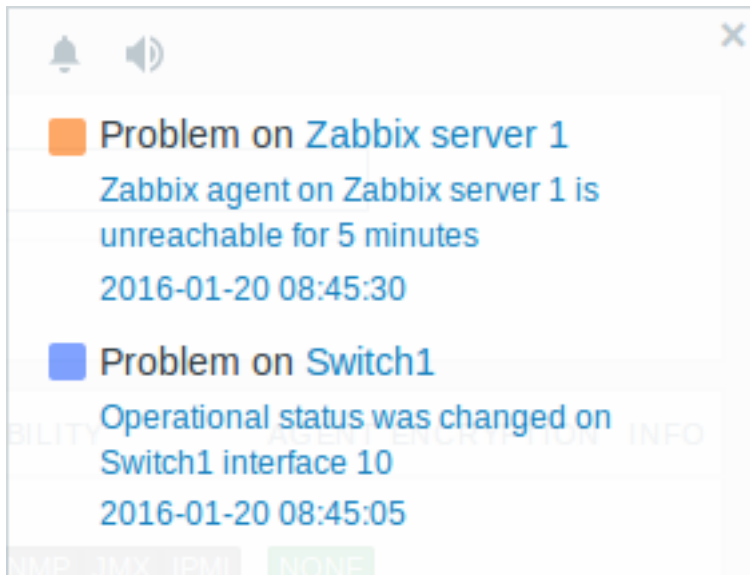
<input checked="" type="checkbox"/> Recovery	<input type="text" value="alarm_ok"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>
<input checked="" type="checkbox"/> Not classified	<input type="text" value="no_sound"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>
<input checked="" type="checkbox"/> Information	<input type="text" value="alarm_information"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>
<input checked="" type="checkbox"/> Warning	<input type="text" value="alarm_warning"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>
<input checked="" type="checkbox"/> Average	<input type="text" value="alarm_average"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>
<input checked="" type="checkbox"/> High	<input type="text" value="alarm_high"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>
<input checked="" type="checkbox"/> Disaster	<input type="text" value="alarm_disaster"/>	<input type="button" value="Play"/>	<input type="button" value="Stop"/>

Show suppressed problems



パラメーター	説明
Frontend messaging	チェックボックスをオンにすると、グローバル通知が有効になります。
Message timeout	メッセージ表示時間を設定できます。デフォルトでは 60 秒間画面に表示されたままになります。 タイムサフィックスを使用可能です。例：30s, 5m, 2h, 1d。
Play sound	サウンドの再生時間を設定できます。 Once - サウンドは一度だけ全て再生されます。 10 seconds - サウンドが 10 秒間繰り返されます。 Message timeout - メッセージが表示されている間、サウンドが繰り返されます。
Trigger severity	グローバル通知とサウンドがアクティブ化されるトリガーの重大度を設定できます。さまざまな重大度に対応したサウンドを選択することもできます。 重大度がマークされていない場合、メッセージは表示されません。また、回復メッセージは、マークされている重大度に対してのみ表示されます。したがって、回復と災害をマークすると、問題と災害重大度トリガーの回復に関するグローバル通知が表示されます。
Show suppressed problems	チェックボックスをオンにすると、ホストのメンテナンスのために抑制される（表示されない）問題の通知が表示されます。

表示されるグローバルメッセージ

メッセージが到着すると、右側のフローティングセクションに表示されます。このセクションは、セクションヘッダーをドラッグすることで自由に再配置できます。



このセクションでは、いくつかのコントロールを使用できます。



-  スヌーズボタンは、現在アクティブなアラーム音を消音します。
-  ミュート/ミュート解除ボタンは、アラーム音の再生と非再生を切り替えます。

2 ブラウザのサウンド

概要

グローバル通知でサウンドが使用されます。

Zabbix フロントエンドでサウンドを再生するには、ユーザープロファイルのメッセージタブでフロントエンドメッセージングを有効にし、すべてのトリガーの重大度をチェックする必要があります。また、グローバル通知ポップアップウィンドウでもサウンドを有効にする必要があります。

何らかの理由でデバイスでオーディオを再生できない場合、グローバル通知ポップアップウィンドウの  ボタンは永続的に「ミュート」状態のままになり、 ボタンにカーソルを合わせると、「このデバイスの通知オーディオをサポートできません」というメッセージが表示されます。

デフォルトのオーディオクリップを含むサウンドは、MP3 形式でのみサポートされています。


Zabbix フロントエンドのサウンドは、Linux および Chrome の最新 Firefox/Opera、Windows の Firefox、Microsoft Edge、Opera、Safari ブラウザーで正常にテストされています。

Attention:

最近のブラウザバージョンでは、サウンドの自動再生がデフォルトで無効になっている場合があります。この場合、この設定を手動で変更する必要があります。

4 グローバルサーチ

Zabbix フロントエンドでホスト、ホストグループ、テンプレートを検索することができます。

検索入力ボックスはメニューの Zabbix ロゴの下にあります。Enter を押すか、 検索アイコンをクリックすると、検索を開始できます。



名前のいずれかの部分に入力された文字列を含むホストがある場合、ドロップダウンが表示され、部分一致したホストが一覧表示されます（一致する部分がオレンジ色で強調表示されます）。ドロップダウンには、ホストの表示名が検索文字列として入力されたテクニカルネームと一致する場合にもホストが一覧表示されます。一致するホストに一覧表示されますが、強調表示はありません。

検索可能な属性

ホストは次のプロパティで検索できます。

- ホスト名
- 表示名
- IP アドレス
- DNS 名

ホストグループはグループ名で検索できます。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。

テンプレートは名前または表示されている名前を検索できます。（テンプレートまたはホストの）表示されている名前とは異なる名前を検索する場合、検索結果では、表示されている名前の下に括弧で囲まれて表示されます。

検索結果

検索結果は、ホスト、ホストグループ、およびテンプレートの 3 つの個別のブロックで構成されます。

☰ Search: Zabbix server

Hosts												
Host	IP	DNS	Monitoring				Configuration					
Zabbix server	127.0.0.1		Latest data	Problems	Graphs	Dashboards	Web	Items 141	Triggers 64	Graphs 27	Discovery 3	Web 1
Displaying 1 of 1 found												
Host groups												
Host group		Monitoring				Configuration						
Zabbix servers		Latest data		Problems		Web		Hosts 1		Templates		
Displaying 1 of 1 found												
Templates												
Template						Configuration						
Template App Remote Zabbix server						Items 47	Triggers 34	Graphs 6	Dashboards 1	Discovery	Web	
Template App Zabbix Server						Items 46	Triggers 34	Graphs 6	Dashboards 1	Discovery	Web	
Displaying 2 of 2 found												

個々のブロックを折りたたんだり展開したりすることができます。エントリ数は、各ブロックの下部に表示されます。例：13 個中 13 個が見つかりました。1 ブロック内に表示されるエントリの総数は 100 に制限されています。

各エントリは、監視および構成データへのリンクを提供します。リンクの**完全なリスト**を参照してください。

すべての構成データ（アイテム、トリガー、グラフなど）について、検出されたエンティティの量は、エンティティ名の横にある番号で灰色で表示されます。注エンティティがゼロの場合、番号は表示されません。

有効なホストは青で表示され、無効なホストは赤で表示されます。

利用可能なリンク

各エントリについて、次のリンクを利用できます。

- Hosts
 - Monitoring

- * Latest data
- * Problems
- * Graphs
- * Host dashboards
- * Web scenarios
- Configuration
 - * Items
 - * Triggers
 - * Graphs
 - * Discovery rules
 - * Web scenarios
- Host groups
 - Monitoring
 - * Latest data
 - * Problems
 - * Web scenarios
 - Configuration
 - * Hosts
 - * Templates
- Templates
 - Configuration
 - * Items
 - * Triggers
 - * Graphs
 - * Template dashboards
 - * Discovery rules
 - * Web scenarios

5 フロントエンドメンテナンスモード

概要

Zabbix Web フロントエンドへのアクセスを禁止するために、一時的に無効にすることができます。これは、ユーザーが開始した変更から Zabbix データベースを保護し、データベースの整合性を保護するのに役立ちます。

Zabbix フロントエンドがメンテナンスモードのときに、Zabbix データベースを停止してメンテナンスタスクを実行できます。

一部の許可された IP アドレスのユーザーは、メンテナンスモード中でも通常どおりフロントエンドを操作できます。

設定

メンテナンスモードを有効にするには `maintenance.inc.php` ファイル (Web サーバーの Zabbix HTML ドキュメントディレクトリの `/conf` にあります) を変更して、次の行のコメントを解除する必要があります：

```
// Maintenance mode.
define('ZBX_DENY_GUI_ACCESS', 1);

// Array of IP addresses, which are allowed to connect to frontend (optional).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Message shown on warning screen (optional).
$ZBX_GUI_ACCESS_MESSAGE = 'We are upgrading MySQL database till 15:00. Stay tuned...';
```

Note:

`maintenance.inc.php` ファイルは、大抵は Web サーバーの Zabbix HTML ドキュメントディレクトリの `/conf` にあります。ただし、使用するオペレーティングシステムと Web サーバーによって異なる場合があります。

たとえば、

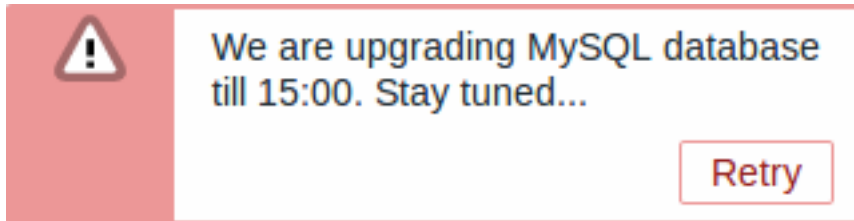
- SUSE または RedHat では `/etc/zabbix/web/maintenance.inc.php`.
- Debian-base のシステムでは `/usr/share/zabbix/conf/`.

PHP ファイルのコピーも参照してください。

パラメータ	詳細
ZBX_DENY_GUI_ACCESS	メンテナンスモードを有効にします。 1 - メンテナンスモード有効。それ以外は無効
ZBX_GUI_ACCESS_IP_RANGE	フロントエンドへの接続が許可される IP アドレスの配列 (オプション) 例: array('192.168.1.1', '192.168.1.2')
ZBX_GUI_ACCESS_MESSAGE	メンテナンスについてユーザーに通知するメッセージ (オプション)

表示

メンテナンスモードで Zabbix フロントエンドにアクセスしようとする時、次の画面が表示されます。メンテナンス終了後に、ユーザーの介入なしに通常の状態に戻るため、画面は 30 秒ごとに更新されます。



ZBX_GUI_ACCESS_IP_RANGE で定義された IP アドレスは、メンテナンス中でも通常どおりフロントエンドにアクセスできます。

6 ページパラメーター

概要

ほとんどの ZabbixWeb インターフェイスページは、表示される内容を制御するさまざまな HTTPGET パラメータをサポートしています。これらは、URL の後に parameter=value のペアで指定し、疑問符 (?) で URL から分離し、アンバサンド (&) でペアを分離することで渡すことができます。

モニタリング → 問題

次のパラメーターがサポートされています。

- show - フィルタオプション「表示」: 1 - 最近の問題 2 - すべて 3 - 問題の状態
- name - フィルタオプション「問題」: 自由形式の文字列
- severities - フィルタオプション「重大度」: 選択した重大度の配列' 重大度 [*] = *(* を重大度レベルに置き換えます): 0 - 未分類 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害
- inventory - フィルタオプション「ホストインベントリ」: インベントリフィールドの配列: [field],[value]
- evaltype - フィルタオプション「タグ」: タグフィルタリング方法: 0 - And/Or, 2 - Or
- tags - フィルタオプション「タグ」: 定義されたタグの配列: [tag],[operator],[value]
- show_tags - フィルタオプション「タグを表示」: 0 - なし, 1 - 1, 2 - 2, 3 - 3
- tag_name_format - フィルタオプション「タグ名」: 0 - フルネーム, 1 - 短縮, 2 - なし
- tag_priority - フィルタオプション「タグ表示優先度」: タグ表示優先度のカンマ区切り文字列
- show_suppressed - フィルタオプション「隠された問題を表示する」: 表示には「show_suppressed=1」を使用
- unacknowledged - フィルタオプション「未確認のみを表示」: 表示には「unacknowledged=1」を使用
- compact_view - フィルタオプション「コンパクトビュー」: 表示には「compact_view=1」を使用
- highlight_row - フィルタオプション「行全体を強調表示」(すべての問題行の背景色として問題の色を使用): 強調表示するには「1」にする必要があります。'compact_view' が設定されている場合にのみ使用できます。
- filter_name - フィルタプロパティオプション「名前」: 自由形式の文字列
- filter_show_counter - フィルタプロパティオプション「レコード数を表示」: 1 - 表示, 0 - 表示しない
- filter_custom_time - フィルタプロパティオプション「カスタム期間の設定」: 1 - 設定, 0 - 設定しない
- sort - 並べ替え列: 継続時間, ホスト, 重大度, 名前
- sortorder - 並べ替え順序または結果: DESC - 降順, ASC - 昇順
- age_state - フィルタオプション「表示期間」: 「age」を有効にするには、「age_state=1」にする必要があります。'show' が 3 の場合にのみ使用されます。
- age - フィルタオプション「表示期間」: 日数
- groupids - フィルタオプション「ホストグループ」: ホストグループ ID の配列
- hostids - フィルタオプション「ホスト」: ホスト ID の配列
- triggerids - フィルタオプション「トリガー」: トリガー ID の配列
- show_timeline - フィルタオプション「タイムラインを表示」: 表示するには「show_timeline=1」である必要があります
- details - フィルタオプション「詳細を表示」: 表示するには「details=1」にする必要があります

- from - 日付範囲の開始。「相対」にすることができます。(例: now-1m) 'filter_custom_time' が 1 の場合にのみ使用されます。
- to - 日付範囲の終了。「相対」にすることができます。(例: now-1m) 'filter_custom_time' が 1 の場合にのみ使用されます。

キオスクモード

フロントエンドページのキオスクモードは、そのページでサポートされていれば URL パラメータを使用してアクティブ化できます。たとえば、ダッシュボードでは次のようになります。

- /zabbix.php?action=dashboard.view&kiosk=1 - キオスクモードをアクティブに
- /zabbix.php?action=dashboard.view&kiosk=0 - ノーマルモードをアクティブに

スライドショー

ダッシュボードでスライドショーをアクティブにするには次の通りです。

- /zabbix.php?action=dashboard.view&slideshow=1 - スライドショーをアクティブに

7 定義

概要

フロントエンドの多くのものはフロントエンド自体を使用して設定できますが、一部のカスタマイズは現在、定義ファイルを編集することによってのみ可能です。

このファイルは、ZabbixHTML ドキュメントディレクトリの/include にある defines.inc.php です。

パラメータ

ユーザーが関心を持つ可能性のあるこのファイルのパラメーター

- ZBX_MIN_PERIOD

最小グラフ期間 (秒単位)。デフォルトは 1 分。

- GRAPH_YAXIS_SIDE_DEFAULT

単純なグラフの Y 軸のデフォルトの位置と、カスタムグラフにアイテムを追加するときのドロップダウンボックスのデフォルト値。
設定可能な値: 0 - 左 1 - 右デフォルト: 0

- ZBX_SESSION_NAME (4.0.0 以降で利用可能)

Zabbix フロントエンドセッションの Cookie 名として使用される文字列。デフォルト: zbx_sessionid

- ZBX_DATA_CACHE_TTL (5.2.0 以降で利用可能)

Vault response のデータキャッシュ無効化までの TTL タイムアウト (秒単位)。Vault response キャッシュを無効にするには、0 を設定します。デフォルト: 60

8 独自テーマ作成

概要

デフォルトで Zabbix はいくつか事前定義されたテーマを提供します。独自のテーマを作成するには、ここに記載されている手順に従ってください。何かいいものを作成した場合は、作業の結果を Zabbix コミュニティと自由に共有してください。

ステップ 1

独自のテーマを定義するには、CSS ファイルを作成し、それを assets/styles/フォルダー (たとえば、custom-theme.css) に保存します。別のテーマからファイルをコピーして、それを元にテーマを作成するか、最初から始めることができます。

ステップ 2

APP::getThemes() メソッドによって返されるテーマのリストにテーマを追加します。このためには APP クラスの ZBase::getThemes() メソッドをオーバーライドします。これは include/classes/core/APP.php の閉じ中括弧の前に次のコードを追加することで実行できます。

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme')
    ]);
}
```

Attention:

引用符の最初のペアで指定する名前は、拡張子なしのテーマファイルの名前と一致する必要があることに注意してください。

複数のテーマを追加するには、最初のテーマの下にそれらをリストします。次に例を示します。

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme'),
        'anothertheme' => _('Another theme'),
        'onemoretheme' => _('One more theme')
    ]);
}
```

最後のテーマを除くすべてのテーマには、末尾にカンマが必要であることを注意してください。

Note:

グラフの色を変更するには、エントリを graph_theme データベーステーブルに追加する必要があります。

ステップ 3

新しいテーマをアクティブにします。

Zabbix フロントエンドで、新しいテーマをデフォルトのテーマに設定したり、ユーザープロファイルで各ユーザーのテーマを変更することができます。

新しいルックアンドフィールをお楽しみください！

9 デバッグモード

概要

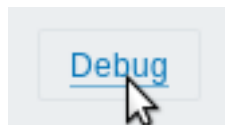
デバッグモードは、フロントエンドページの問題を診断するために使用できます。

設定

デバッグモードは、ユーザーグループに属する個々のユーザーに対してアクティブ化できます。

- **ユーザーグループ**で設定する。
- 設定された**ユーザーグループ**を表示する。

ユーザーグループに対してデバッグモードが有効になっている場合、そのユーザーにはブラウザウィンドウの右下隅にデバッグボタンが表示されます。



デバッグボタンをクリックすると、ページコンテンツの下に新しいウィンドウが開きます。このウィンドウには、API 呼び出しと個々の SQL ステートメントのリストとともにページの SQL 統計が含まれています。

```

***** Script profiler *****
Total time: 0.249825
Total SQL time: 0.139814
SQL count: 143 (selects: 117 | executes: 26)
Peak memory usage: 6M
Memory limit: 128M

1. hostgroup.get [latest.php:124]

Parameters:          Result:
Array                Array
(                    (
  [output] => Array   [4] => Array
    (                (
      [0] => groupid  [groupid] => 4
    )
  )
)

```

Hide debug

ページに問題がある場合は、このウィンドウを使用して問題の原因を調査できます。

Warning:
デバッグモードを有効にするとフロントエンドのパフォーマンスに悪影響を及ぼします。

10 Zabbix 用 Cookie

概要

このページには Zabbix 用 Cookie 一覧が表示されます。

名前	説明	値	範囲/最大-最小	Http のみ ^a	セキュア ^a
ZBX_SESSIONID	Base64 エンコードされた JSON として保存された Zabbix フロントエンドセッションデータ		セッション (ブラウジングセッションが終了すると期限切れ)	+	+
tab	アクティブなタブ番号。この Cookie は、複数のタブがあるページ (Host、Trigger、Action 設定ページなど) でのみ使用され、ユーザーがプライマリタブから別のタブ (Tags や Dependencies タブなど) に移動したときに作成されます。	例: 1	セッション (ブラウジングセッションが終了すると期限切れ)	-	-

^aHttpOnly が 'true' の場合、Cookie は HTTP プロトコルを介してのみアクセス可能になります。これは、JavaScript などのスクリプト言語から Cookie にアクセスできないことを意味します。この設定は、XSS 攻撃による個人情報の盗難を効果的に減らすのに役立ちます (ただし、すべてのブラウザでサポートされているわけではありません)

^aSecure は、Cookie がクライアントからの安全な HTTPS 接続を介してのみ送信される必要があることを示します。'true' に設定すると、Cookie は安全な接続が存在する場合にのみ設定されます。

0 はプライマリタブに使用されます。

Httpのみ^a

^aHttpOnly が'true' の場合、Cookie は HTTP プロトコルを介してのみアクセス可能になります。これは、JavaScript などのスクリプト言語から Cookie にアクセスできないことを意味します。この設定は、XSS 攻撃による個人情報の盗難を効果的に減らすのに役立ちます（ただし、すべてのブラウザでサポートされているわけではありません）

セキュア^a

^aSecure は、Cookie がクライアントからの安全な HTTPS 接続を介してのみ送信される必要があることを示します。'true' に設定すると、Cookie は安全な接続が存在する場合にのみ設定されます。

名前	説明	値	範囲/最大-最小		
browserwarningignore	ブラウザの使用に関する警告を無視するかどうか	yes	セッション（ブラウジングセッションが終了すると期限切れ）	-	-
system-message-ok	ページがリロードされるとすぐに表示されるメッセージ	プレーンテキストメッセージ	セッション（ブラウジングセッションが終了すると期限切れ）またはページがリロードされるとすぐに	+	-
system-message-error	ページがリロードされるとすぐに表示されるエラーメッセージ	プレーンテキストメッセージ	セッション（ブラウジングセッションが終了すると期限切れ）またはページがリロードされるとすぐに	+	-

Note:

Web サーバーディレクティブによる ZabbixCookie の「HttpOnly」フラグの強制はサポートされていません。

11 タイムゾーン

概要

フロントエンドのタイムゾーンは、フロントエンドでグローバルに設定し、個々のユーザーに合わせて調整できます。

The screenshot shows a configuration interface with several settings. The 'Default language' is set to 'English (en_US)'. The 'Default time zone' dropdown menu is open, showing a list of time zones. The selected option is 'System: (UTC+02:00) Europe/Riga'. Other visible options include 'System: (UTC+02:00) Europe/Riga', '(UTC-01:00) America/Scoresbysund', '(UTC-01:00) Atlantic/Azores', '(UTC-01:00) Atlantic/Cape_Verde', '(UTC-02:00) America/Noronha', '(UTC-02:00) Atlantic/South_Georgia', '(UTC-03:00) America/Araguaina', '(UTC-03:00) America/Argentina/Buenos_Aires', '(UTC-03:00) America/Argentina/Catamarca', '(UTC-03:00) America/Argentina/Cordoba', and '(UTC-03:00) America/Argentina/Jujuy'. Other settings visible include 'Default theme', '* Limit for search and filter results', '* Max number of columns and rows in overview tables', '* Max count of elements to show inside table cell', 'Show warning if Zabbix server is down', '* Working time', 'Show technical errors', and '* Max history display period'.

システムが選択されている場合、Web サーバーのタイムゾーンがフロントエンドに使用され（設定されている場合は php.ini の「date.timezone」の値）Zabbix サーバーは、実行しているマシンのタイムゾーンを使用します。

Note:

Zabbix サーバーは、通知でマクロを展開する（たとえば、{EVENT.TIME} はユーザーごとに異なるタイムゾーンに展開できません）とき、および通知が送信される際の制限時間にのみ、指定されたグローバル/ユーザータイムゾーンを使用します。（ユーザーメディア設定の「アクティブ時」設定を参照）

設定

グローバルタイムゾーン

- フロントエンドをインストールするときに手動で設定できます
- 管理 → 一般 → GUI で変更できます

ユーザーレベルタイムゾーン

- ユーザー設定/更新するときに設定できます
- ユーザープロファイルで各ユーザーが設定できます

13 パスワードのリセット

概要 このセクションでは、Zabbix のユーザーパスワードをリセットする手順について説明します。

ステップ Zabbix パスワードを忘れてログインできない場合は、Zabbix 管理者に問い合わせてください。

スーパー管理者ユーザーは、ユーザー設定フォーム 内のすべてのユーザーのパスワードを変更できます。

スーパー管理者がパスワードを忘れてログインできない場合は、次の SQL クエリを実行して、デフォルトパスワードをスーパー管理者ユーザーに適用する必要があります（「Admin」をスーパー管理者ユーザー名に置き換えます）

```
UPDATE users SET passwd = '$2a$10$ZXivHAEP2ZM.dLXTm6uPHOMVIARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username = 'Admin';
```

このクエリを実行すると、ユーザー名 'Admin' のパスワードは zabbix に設定されます。最初のログイン時に必ずパスワードをデフォルトから変更してください。

19. API

概要 Zabbix API を使用すると、Zabbix の設定をプログラムで取得して変更し、履歴データへのアクセスを提供できます。次の目的で広く使用されています。

- Zabbix と連携する新しいアプリケーションを作成する
- Zabbix をサードパーティソフトウェアと統合する
- ルーチンタスクを自動化する

Zabbix API は Web ベースの API であり、Web フロントエンドの一部として出荷され、JSON-RPC 2.0 プロトコルを使用します。これは、次の 2 つのことを意味します。

- API は一連の個別メソッドで構成されている
- クライアントと API 間の要求と応答は、JSON 形式を使用してエンコードされる

プロトコルと JSON の詳細については、[JSON-RPC 2.0 仕様](#) と [JSON 形式のホームページ](#) を参照してください。

構造 この API は、名目上、別々の API にグループ化された多数のメソッドから構成されています。それぞれのメソッドはある特定のタスクを
 実行します。例えば、host.create メソッドは host API に属し、新しいホストを作成するために使用されます。歴史的には、API は
 "classes" と呼ばれることもある。

Note:

ほとんどの API は、少なくとも get, create, update, delete という 4 つのメソッドを持ち、それぞれデータの取得、作成、更新、削除を
 行います。しかし、一部の API では、全く別のメソッドを提供することもあります。

リクエストの実行 フロントエンドを設定したら、リモート HTTP リクエストを使用して API を呼び出すことができます。これを行うには、フロントエンドディレクトリにある `api_jsonrpc.php` ファイルに HTTP POST リクエストを送信する必要があります。たとえば、Zabbix フロントエンドが `http://example.com/zabbix` にインストールされている場合、`apiinfo.version` メソッドを呼び出す HTTP リクエストは次のようになります。

Performing requests Once you've set up the frontend, you can use remote HTTP requests to call the API. To do that you need to send HTTP POST requests to the `api_jsonrpc.php` file located in the frontend directory. For example, if your Zabbix frontend is installed under `http://example.com/zabbix`, the HTTP request to call the `apiinfo.version` method may look like this:

```
POST http://example.com/zabbix/api_jsonrpc.php HTTP/1.1
Content-Type: application/json-rpc

{
  "jsonrpc": "2.0",
  "method": "apiinfo.version",
  "id": 1,
  "auth": null,
  "params": {}
}
```

リクエストの `Content-Type` ヘッダーは、`application/json-rpc`、`application/json`、または `application/jsonrequest` のいずれかの値に設定されている必要があります。

ワークフローの例 次のセクションでは、いくつかの使用例をより詳細に説明します。

認証 Zabbix のデータにアクセスするには次のいずれかのトークンで認証を行う必要があります。

- 既存の **API トークン** (Zabbix フロントエンドで作成、または **トークン API** を使用して作成) を使用。
- `user.login` メソッドで取得した認証トークンを使用。

たとえば、標準の Admin ユーザーとしてログインして新しい認証トークンを取得したい場合、JSON リクエストは次のようになります。

```
{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "username": "Admin",
    "password": "zabbix"
  },
  "id": 1,
  "auth": null
}
```

リクエストオブジェクトを詳しく見てみましょう。プロパティは次のとおりです：

`-jsonrpc` - API で使用される JSON-RPC プロトコルのバージョン。Zabbix API は JSON-RPC バージョン 2.0 を実装します。`-method` - 呼び出される API メソッド。`-params` - API メソッドに渡されるパラメータ。`-id` - リクエストの任意の識別子。`-auth` - ユーザー認証トークン。まだ持っていないので、「null」に設定されています。

資格情報を正しく提供した場合、応答が返されます。API にはユーザー認証トークンが含まれます。

```
{
  "jsonrpc": "2.0",
  "result": "0424bd59b807674191e7d77572075f33",
  "id": 1
}
```

レスポンスオブジェクトには次のプロパティが含まれます。

`-jsonrpc` - 繰り返しになりますが、JSON-RPC プロトコルのバージョン。`-result` - メソッドによって返されたデータ。`-id` - 対応するリクエストの識別子。

ホストの取得 ホストの取得

Zabbix のデータにアクセスするために使用できる有効なユーザー認証トークンを取得できました。次に例として `host.get` メソッドを使用し、構成されているすべての **ホスト** の ID、ホスト名、およびインターフェースを取得してみましょう。

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": [
      "hostid",
      "host"
    ],
    "selectInterfaces": [
      "interfaceid",
      "ip"
    ]
  },
  "id": 2,
  "auth": "0424bd59b807674191e7d77572075f33"
}
```

Attention:

auth プロパティに user.login を呼び出して取得した認証トークンを設定するよう注意してください。

レスポンスオブジェクトには、要求されたホストに関するデータが含まれます。

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "host": "Zabbix server",
      "interfaces": [
        {
          "interfaceid": "1",
          "ip": "127.0.0.1"
        }
      ]
    }
  ],
  "id": 2
}
```

Note:

パフォーマンス上の理由から、取得するオブジェクトプロパティを常にリストアップし、すべてのオブジェクトプロパティを取得しないようにすることをお勧めします。

新しいアイテムの作成 先ほどの host.get リクエストで取得したデータを使って、“Zabbix server” に新しいアイテムを作成してみましょう。これは `item.create` メソッドを使用して実行できます。

```
{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "name": "Free disk space on /home/joe/",
    "key_": "vfs.fs.size[/home/joe/,free]",
    "hostid": "10084",
    "type": 0,
    "value_type": 3,
    "interfaceid": "1",
    "delay": 30
  },
  "auth": "0424bd59b807674191e7d77572075f33",
  "id": 3
}
```

作成が成功した場合の応答には、新しく作成されたアイテムの ID が含まれます。これは、以降の要求でアイテムを参照するために使

用できます。

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "24759"
    ]
  },
  "id": 3
}
```

Note:

なお `item.create` メソッドや他の `create` メソッドも、オブジェクトの配列を受け入れて、1 回の API 呼び出しで複数のアイテムを作成することもできます。

複数のトリガーの作成 つまり `create` メソッドが配列を受け入れるのであれば、次のように複数のトリガーを追加することも可能です。

```
{
  "jsonrpc": "2.0",
  "method": "trigger.create",
  "params": [
    {
      "description": "Processor load is too high on {HOST.NAME}",
      "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
    },
    {
      "description": "Too many processes on {HOST.NAME}",
      "expression": "avg(/Linux server/proc.num[],5m)>300",
    }
  ],
  "auth": "0424bd59b807674191e7d77572075f33",
  "id": 4
}
```

作成が成功した場合の応答には、新しく作成したトリガーの ID が含まれます。

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "17369",
      "17370"
    ]
  },
  "id": 4
}
```

アイテムの更新 アイテムを有効にします。つまり `status` を "0" に設定します。

```
{
  "jsonrpc": "2.0",
  "method": "item.update",
  "params": {
    "itemid": "10092",
    "status": 0
  },
  "auth": "0424bd59b807674191e7d77572075f33",
  "id": 5
}
```

更新が成功した応答には、更新されたアイテムの ID が含まれます。


```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "10092"
    ]
  },
  "id": 5
}
```

Note:

なお `item.update` メソッドやその他の `update` メソッドも、オブジェクトの配列を受け入れ、1 回の API 呼び出しで複数のアイテムを更新することができます。

複数のトリガーの更新 複数のトリガーを有効にします。つまり `status` を 0 に設定します。

```
{
  "jsonrpc": "2.0",
  "method": "trigger.update",
  "params": [
    {
      "triggerid": "13938",
      "status": 0
    },
    {
      "triggerid": "13939",
      "status": 0
    }
  ],
  "auth": "0424bd59b807674191e7d77572075f33",
  "id": 6
}
```

更新が成功した応答には、更新されたトリガーの ID が含まれます。

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "13938",
      "13939"
    ]
  },
  "id": 6
}
```

Note:

これは、推奨される更新方法です。`host.massupdate` などの一部の API メソッドを使用すると、より単純なコードを記述できますが、これらのメソッドは将来のリリースで削除される予定のため、使用は推奨されません。

エラーハンドリング 今まで試したメソッドはすべて成功しました。しかし、API に対して間違った呼び出しを行おうとするとどうなるでしょうか？`host.create` を呼び出して別のホストを作成する際に、必須の `groups` パラメータを省略してみましょう。

```
{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "Linux server",
    "interfaces": [
      {
        "type": 1,
        "main": 1,

```

```

        "useip": 1,
        "ip": "192.168.3.1",
        "dns": "",
        "port": "10050"
    }
]
},
"id": 7,
"auth": "0424bd59b807674191e7d77572075f33"
}

```

レスポンスにはエラーメッセージが含まれます。

```

{
  "jsonrpc": "2.0",
  "error": {
    "code": -32602,
    "message": "Invalid params.",
    "data": "No groups for host \"Linux server\"."
  },
  "id": 7
}

```

エラーが発生した場合、レスポンスオブジェクトには `result` プロパティの代わりに、次のデータを持つ `error` プロパティが含まれます。

- `code` - エラーコード
- `message` - 簡単なエラーの要約
- `data` - より詳細なエラーメッセージ

エラーは、誤った入力値の使用、セッションタイムアウト、または存在しないオブジェクトへのアクセス試行など、さまざまなケースで発生する可能性があります。アプリケーションは、この種のエラーを適切に処理する必要があります。

API バージョン API のバージョン管理を簡素化するために、Zabbix 2.0.4 以降、API のバージョンは Zabbix 自体のバージョンと一致していますが、`apiinfo.version` メソッドを使用して、使用している API のバージョンを確認することもできます。これは、バージョン固有の機能を使用するようにアプリケーションを調整する場合に役立ちます。

メジャーバージョン内では、機能の下位互換性を保証します。メジャーリリース間で下位互換性のない変更を行う場合、通常は古い機能を次のリリースで非推奨として残し、その後のリリースで削除します。場合によっては、下位互換性を提供せずにメジャーリリース間で機能を削除することもあります。非推奨の機能には決して依存せず、できるだけ早く新しい代替機能に移行することが重要です。

Note:

API 変更ログ で、API に加えられたすべての変更を追うことができます。

参考文献 ここまでで Zabbix API を使い始めるのに十分な知識が得られましたが、これが全てではありません。更に詳しく知りたい場合は、[利用可能な API のリスト](#) を参照することをお勧めします。

メソッドリファレンス

このセクションでは、Zabbix API が提供する機能の概要を説明し、利用可能なクラスとメソッドの使い方を説明します。

監視 Zabbix API では、監視中に収集した履歴やその他のデータにアクセスすることができます。

高可用性 (HA) クラスタ

サーバノードとそのステータスの一覧を取得します。

High availability cluster API

History

Zabbix 監視プロセスによって収集された過去の値を取得し、プレゼンテーションやさらなる処理を行うことができます。

History API

Trends

Zabbix サーバで計算された Trends 値を取得し、プレゼンテーションやさらなる処理を行うことができます。

Trend API

イベント

trigger、ネットワークディスクバリアー、その他の Zabbix システムから生成されたイベントを取得し、より柔軟な状況管理や
サードパーティツールの統合を行うことができます。

Event API

問題

与えられたパラメータに従って、問題を取得します。

Problem API

サービスモニタリング

監視する IT インフラ/ビジネスサービスデータの階層表現を作成します。

Service API

サービスレベルアグリーメント (SLA)

サービスレベル目標 (SLO) を定義し、サービスのパフォーマンスに関する詳細なサービスレベル指標 (SLI) 情報を取得します。

SLA API

タスク

Zabbix server のタスクマネージャと連動し、タスクの作成とレスポンスの取得を行います。

Task API

設定 Zabbix API では、監視システムの設定を管理することができます。

Hosts と host groups

host groups、Host、およびホストインタフェース、ホストマクロ、メンテナンス期間など、ホストに関連するすべてのものを管理します。

[Host API](#) | [Host group API](#) | [Host interface API](#) | [User macro API](#) | [Value map API](#) | [Maintenance API](#)

Items

監視する item を定義します

Item API

Triggers

システムの問題を通知する trigger を設定する。trigger の依存関係を管理します。

Trigger API

グラフ

収集したデータを見やすくするために、グラフを編集したり、グラフの item を分けたりすることができます。

[Graph API](#) | [Graph item API](#)

テンプレート

テンプレートを管理し、ホストや他のテンプレートとリンクさせることができます。

[Template API](#) | [Value map API](#)

エクスポートとインポート

設定のバックアップ、マイグレーション、大規模な設定更新のために Zabbix の設定データをエクスポート、インポートする
ことができます。

Configuration API

ローレベルディスクバリアー

動的なエンティティを監視するために、ローレベルディスクバリアールール、item、trigger、グラフのプロトタイプを設定します。

[LLD rule API](#) | [Item prototype API](#) | [Trigger prototype API](#) | [Graph prototype API](#) | [Host prototype API](#)

イベント相関

カスタムイベントの相関ルールを作成します。

Correlation API

アクションとアラート

特定のイベントをユーザーに通知したり、リモートコマンドを自動的に実行するためのアクションと操作を定義します。
 生成されたアラートとその受信者に関する情報にアクセスできます。

Action API | Alert API

サービス

サービスレベルのモニタリングのためにサービスを管理し、あらゆるサービスの詳細な SLA 情報を取得することができます。

Service API

ダッシュボード

ダッシュボードを管理し、それに基づいてスケジュールされたレポートを作成します。

Dashboard API | Template dashboard API | Report API

マップ

マップを設定して、IT インフラストラクチャの詳細な動的表示を作成します。

Map API

Web モニタリング

Web アプリケーションやサービスを監視するための Web シナリオを設定します。

Web scenario API

ネットワーク検出

ネットワークレベルの検出ルールを管理し、新しいホストを自動的に検出して監視します。発見されたサービスやホストに
 関する情報への完全なアクセスを提供します。

Discovery rule API | Discovery check API | Discovered host API | Discovered service API

管理画面 Zabbix API を使用すると、監視システムの管理設定を変更することができます。

ユーザー

Zabbix にアクセスするユーザを追加し、ユーザグループに割り当て、パーミッションを付与します。
 ユーザーの権限を細かく管理するためのルールを作成します。各ユーザが行った設定変更の追跡メディアタイプや
 ユーザーがアラートを受信する複数の方法を設定することができます。

User API | User group API | User role API | Media type API | Audit log API

一般

特定のグローバル設定オプションを変更します。

Autoregistration API | Icon map API | Image API | User macro API | Settings API | Housekeeping API

正規表現

グローバルな正規表現を管理します。

Regular expression API

Proxy

分散モニタリングのセットアップで使用する proxy を管理します。

Proxy API

認証

認証の設定オプションを変更します。

Authentication API

API トークン

認可トークンを管理します。

Token API

スクリプト

日々の作業を支援するスクリプトを設定・実行します。

Script API

API 情報 Zabbix API のバージョンを取得し、アプリケーションでバージョン固有の機能を使用できるようにします。

API info API

Action

このクラスは、アクションと連動するように設計されています。

オブジェクトリファレンス:

- [Action](#)
- [Action condition](#)
- [Action operation](#)

利用可能なメソッド:

- [action.create](#) - 新しいアクションを作る
- [action.delete](#) - アクションを削除する
- [action.get](#) - アクションを取得する
- [action.update](#) - アクションを更新する

> Action object

次のオブジェクトは、actionAPI に直接関連しています。

Action

action オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
actionid	string	(読取専用) アクションの ID
esc_period (必須)	string	デフォルトの操作ステップ期間。少なくとも 60 秒でなければなりません。秒、サフィックスとユーザーマクロを備えたタイムユニットを受け入れます。

エスカレーションは、トリガー、内部、サービスアクションおよび通常の操作でのみサポートされていることに注意してください。

eventsource (必須)	integer	(定数) アクションが処理するイベントの種類。 サポートされているイベントタイプのリストについては、 イベント "Source" Property を参照してください。
name (必須)	string	アクション名
status	integer	アクションの有効無効 有効な値： 0 - (デフォルト) 有効 1 - 無効

プロパティ	タイプ	説明
pause_suppressed	integer	メンテナンス期間中にエスカレーションを一時停止するか 有効な値: 0 - しない 1 - (デフォルト) する
notify_if_canceled	integer	このパラメータはトリガーアクションに対してのみ有効です。 エスカレーションがキャンセルされたときに通知するか 有効な値: 0 - しない 1 - (デフォルト) する このパラメータはトリガーアクションに対してのみ有効です。

Action operation

action operation オブジェクトはアクション実行時の操作を設定します。次のプロパティがあります。

プロパティ	タイプ	説明
operationid	string	(読取専用) action operation の ID
operationtype (必須)	integer	操作のタイプ。 使用可能な値: 0 - メッセージを送信 1 - グローバルスクリプト 2 - ホストを追加 3 - ホストを削除 4 - ホストグループに追加 5 - ホストグループから削除 6 - テンプレートにリンク 7 - テンプレートからリンク解除 8 - ホストを有効化 9 - ホストを無効化 < br>10 - ホストインベントリモードを設定 トリガーおよびサービスアクションではタイプ'0' と '1' のみがサポートされ、内部アクションでは'0' のみがサポートされることに注意してください。ディスカバリおよび自動登録アクションでは、すべてのタイプがサポートされています。
actionid	string	(読取専用) 操作が属するアクションの ID
esc_period	string	エスカレーションステップの間隔 (秒単位)。60 秒より長くする必要がありません。秒、タイムサフィックス付きの単位、およびユーザーマクロを受け入れます。0 または 0 秒に設定すると、デフォルトのアクションエスカレーション間隔が使用されます。 エスカレーションは、トリガー、内部、およびサービスアクションに対してのみサポートされており、通常の操作でのみサポートされていることに注意してください。
esc_step_from	integer	エスカレーションを開始するステップ。 デフォルト: 1 エスカレーションは、トリガー、内部、およびサービスアクションに対してのみサポートされており、通常の操作でのみサポートされていることに注意してください。

プロパティ	タイプ	説明
esc_step_to	integer	<p>エスカレーションを終了するステップ。</p> <p>デフォルト: 1</p> <p>エスカレーションは、トリガー、内部およびサービスアクションでのみサポートされており、通常の操作でのみサポートされていることに注意してください。</p>
evaltype	integer	<p>動作条件の評価方法。</p> <p>使用可能な値: 0 - (デフォルト) AND / OR; 1 - AND; 2 - OR.</p>
opcommand	object	<p>オペレーションによって実行されるグローバルスクリプトに関するデータを含むオブジェクト。</p> <p>各オブジェクトには次のプロパティが1つあります。scriptid - (string) スクリプトの ID。</p>
opcommand_grp	array	<p>グローバルスクリプトオペレーションに必要です。 グローバルスクリプトを実行するホストグループ。</p> <p>各オブジェクトには次のプロパティがあります。 opcommand_grpid - (string, 読取専用) オブジェクトの ID; operationid - (string, 読取専用) 操作の ID; groupid - (string) ホストグループの ID。</p>
opcommand_hst	array	<p>opcommand_hst が設定されていない場合、グローバルスクリプト操作に必要です。 グローバルスクリプトを実行するホスト。</p> <p>各オブジェクトには次のプロパティがあります。 opcommand_hstid - (string, 読取専用) オブジェクトの ID; operationid - (string, 読取専用) 操作の ID; hostid - (string) ホストの ID。0 に設定すると、コマンドは現在のホストで実行されます。</p>
opconditions	array	<p>opcommand_grp が設定されていない場合、グローバルスクリプト操作に必要です。 トリガーアクションに使用される操作条件。</p>
opgroup	array	<p>操作条件オブジェクトの詳細は後述 ホストを追加するホストグループ。</p> <p>各オブジェクトには次のプロパティがあります。 operationid - (string, 読取専用) オペレーション ID groupid - (string) ホストグループ ID</p> <p>"ホストグループへ追加" および "ホストグループから削除" 操作に必要です。</p>
opmessage	object	<p>アクションオペレーションによって送信されたメッセージに関するデータを含むオブジェクト。</p> <p>アクションオペレーションメッセージオブジェクトは以下で詳しく説明しています。</p> <p>メッセージ操作に必須です。</p>

プロパティ	タイプ	説明
opmessage_grp	array	メッセージを送信するユーザーグループ 各オブジェクトには次のプロパティがあります。 operationid - (string, 読取専用) オペレーション ID usrgrpuid - (string) ユーザーグループ ID
opmessage_usr	array	opmessage_usr が設定されていない場合、メッセージ操作に必要です。 メッセージを送信するユーザー。 各オブジェクトには次のプロパティがあります。 operationid - (string, 読取専用) オペレーション ID userid - (string) ユーザー ID opmessage_grp が設定されていない場合、メッセージ操作に必要です。
optemplate	array	ホストをリンクするテンプレート。 各オブジェクトには次のプロパティがあります： operationid - (string, 読取専用) オペレーション ID templateid - (string) テンプレート ID
opinventry	object	"テンプレートへのリンク" および"テンプレートからのリンク解除" 操作に必要です。 インベントリモードにホストを設定します。 オブジェクトには次のプロパティがあります： operationid - (string, 読取専用) オペレーション ID; inventory_mode - (string) インベントリモード "ホストインベントリモードの設定" 操作に必要です。

Action operation message

operation message オブジェクトには、オペレーションによって送信されるメッセージに関するデータが含まれています。

プロパティ	タイプ	説明
default_msg	integer	デフォルトのアクションメッセージテキストと件名を使用するか 有効な値： 0 - オペレーションからのデータを使用 1 - (デフォルト) メディアタイプからのデータを使用
mediatypeid	string	メッセージの送信に使用されるメディアタイプの ID
message	string	オペレーションメッセージ本文
subject	string	オペレーションメッセージ件名

Action operation condition

action operation condition オブジェクトは、現在の操作を実行するために満たさなければならない条件を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
opconditionid	string	(読取専用) アクション動作条件の ID
conditiontype (必須)	integer	条件のタイプ 可能な値: 14 - イベントが確認された
value (必須)	string	比較する値
operationid	string	(読取専用) オペレーション ID

プロパティ	タイプ	説明
operator	integer	条件演算子 可能な値: 0 - (デフォルト) =。

次の演算子と値は、各操作条件タイプでサポートされています。

条件	条件名	対応演算子	期待値
14	イベント承認	=	イベントが承認されたか 可能な値: 0 - されていない 1 - されている

Action recovery operation

Action recovery operation オブジェクトは、障害が解決されたときに実行される操作を定義します。トリガー、内部、およびサービスアクションのリカバリオペレーションが可能です。次のプロパティがあります。

プロパティ	タイプ	説明
operationid	string	(読取専用) アクションオペレーション ID
operiontype (必須)	integer	オペレーションタイプ トリガーおよびサービスアクションに使用可能な値: 0 - メッセージを送信 1 - グローバルスクリプト 11 - 関連するすべてに通知 内部アクションに使用可能な値: 0 - メッセージを送信 11 - 関係者全員に通知
actionid	string	(読取専用) リカバリオペレーションが属するアクションの ID
opcommand	object	実行されるグローバルアクションタイプスクリプトに関するデータを含むオブジェクト。 各オブジェクトには次のプロパティが1つあります: scriptid - (string) アクションタイプスクリプトの ID
opcommand_grp	array	グローバルスクリプトオペレーションには必須です グローバルスクリプトを実行するホストグループ。 各オブジェクトには次のプロパティがあります。 opcommand_grpid - (string, 読取専用) オブジェクトの ID operationid - (string, 読取専用) オペレーション ID groupid - (string) ホストグループの ID
opcommand_hst	array	opcommand_hst が設定されていない場合、グローバルスクリプト操作に必須です。 グローバルスクリプトを実行するホスト 各オブジェクトには次のプロパティがあります opcommand_hstid - (string, 読取専用) オブジェクト ID operationid - (string, 読取専用) オペレーション ID; hostid - (string) ホスト ID。0 に設定するとコマンドは現在のホストで実行されます。 opcommand_grp が設定されていない場合、グローバルスクリプト操作に必須です。

プロパティ	タイプ	説明
opmessage	object	リカバリオペレーションによって送信されたメッセージに関するデータを含むオブジェクト。 操作メッセージオブジェクトは 上記で詳しく説明 しています。
opmessage_grp	array	 メッセージ操作に必須です。 メッセージを送信するユーザーグループ。 各オブジェクトには次のプロパティがあります。 operationid - (string, 読取専用) オペレーション ID usrgrpuid - (string) ユーザーグループの ID opmessage_usr が設定されていない場合、メッセージ操作に必須です。
opmessage_usr	array	メッセージを送信するユーザー。 各オブジェクトには次のプロパティがあります。 operationid - (string, 読取専用) オペレーション ID userid - (string) ユーザーの ID opmessage_grp が設定されていない場合、メッセージ操作に必須です。

Action update operation

action update operation オブジェクトは、障害が更新されたときに実行される操作を定義します (コメント、承認、深刻度の変更、または手動でクローズ)。トリガーとサービスのアクションは、更新操作が可能です。次のプロパティがあります。

プロパティ	タイプ	説明
operationid operatointype (必須)	string integer	(読取専用) アクションオペレーション ID オペレーションタイプ トリガーとサービスアクションの有効な値: 0 - メッセージを送信 1 - グローバルスクリプト 12 - 関係者全員に通知
opcommand	object	オペレーションによって実行されるグローバルアクションタイプスクリプトに関するデータを含むオブジェクト。 各オブジェクトには次のプロパティが1つあります。scriptid - (string) アクションタイプスクリプトの ID。
opcommand_grp	array	必須グローバルスクリプト操作。 グローバルスクリプトを実行するホストグループ 各オブジェクトには次のプロパティがあります groupid - (string) ホストグループの ID opcommand_hst が設定されていない場合、グローバルスクリプト操作に必要です
opcommand_hst	array	グローバルスクリプトを実行するホスト 各オブジェクトには次のプロパティがあります hostid - (文字列) ホストの ID。0 に設定すると、コマンドは現在のホストで実行されます opcommand_grp が設定されていない場合、グローバルスクリプト操作に必要です。
opmessage	object	更新操作によって送信されたメッセージに関するデータを含むオブジェクト。 操作メッセージオブジェクトは こちらで詳細に説明 します。

プロパティ	タイプ	説明
opmessage_grp	array	メッセージを送信するユーザーグループ 各オブジェクトには次のプロパティがあります usrgrpuid - (文字列) ユーザーグループの ID
opmessage_usr	array	opmessage_usr が設定されていない場合、send message オペレーションでのみ必要です send update message オペレーションでは無視されます メッセージを送信するユーザー 各オブジェクトには次のプロパティがあります userid - (文字列) ユーザーの ID opmessage_grp が設定されていない場合、send message オペレーションでのみ必要です。 send update message 操作では無視されます。

Action filter

action filter オブジェクトは、構成されたアクション操作を実行するために満たす必要がある一連の条件を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
conditions (必須)	array	結果のフィルタリングに使用するフィルター条件のセット
evaltype (必須)	integer	フィルタ条件の評価方法 使用可能な値: 0 - および/または; 1 - and 2 - or 3 - カスタム式
eval_formula	string	(読取専用) フィルター条件の評価に使用される式。式にはその formulaid によって特定のフィルター条件を参照する ID が含まれています。eval_formula の値は、カスタム式を持つフィルターの formula の値と同じです。
formula	string	カスタム式でフィルターの条件を評価するために使用されるユーザー定義の式。式には、その formulaid によって特定のフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、フィルター条件で定義された ID と正確に一致する必要があります。条件を未使用または省略したままにすることはできません。 カスタム式フィルタでは必須です。

Action filter condition

action filter condition オブジェクトは、アクション操作を実行する前にチェックする必要がある特定の条件を定義します。

プロパティ	タイプ	説明
conditionid	string	(読取専用) 動作条件 ID

プロパティ	タイプ	説明
conditiontype (必須)	integer	<p>条件のタイプ</p> <p>トリガーアクションに使用可能な値:</p> <ul style="list-style-type: none"> 0 - ホストグループ 1 - ホスト 2 - トリガー 3 - トリガー名 4 - トリガーの深刻度 6 - 期間 13 - ホストテンプレート 16 - 抑制された障害 25 - イベントタグ 26 - イベントタグ値 <p>< br> ディスカバリアクションに使用可能な値:</p> <ul style="list-style-type: none"> 7 - ホスト IP 8 - 検出されたサービスタイプ 9 - 検出されたサービスポート 10 - 検出ステータス 11 - アップタイムまたはダウンタイム期間 12 - 受信した値 18 - ディスカバリルール 19 - ディスカバリチェック 20 - プロキシ 21 - ディスカバリオブジェクト <p>自動登録アクションに使用可能な値:</p> <ul style="list-style-type: none"> 20 - プロキシ 22 - ホスト名 24 - ホストメタデータ <p>内部アクションに使用可能な値:</p> <ul style="list-style-type: none"> 0 - ホストグループ 1 - ホスト 13 - ホストテンプレート 23 - イベントタイプ 25 - イベントタグ 26 - イベントタグ値 <p>
 サービスアクションに使用できる値:</p> <ul style="list-style-type: none"> 25 - イベントタグ 26 - イベントタグ値 27 - サービス 28 - サービス名
value (必須)	string	比較する値
value2	string	比較する二次的な値。条件タイプが 26 の場合、トリガー、内部、およびサービスアクションに必要です。
actionid	string	(読取専用) 条件が属するアクション ID
formulaid	string	カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみを含めることができます。ID は、フィルタ条件を変更するときにユーザーが定義する必要がありますが、後で要求するときに新たに生成されます。

プロパティ	タイプ	説明
operator	integer	条件演算子 使用可能な値: 0 - (デフォルト) 等しい 1 - 等しくない 2 - 含む 3 - 含まない < br>4 - 入っている 5 - 以上 6 - 以下 7 - 入っていない 8 - 一致する 9 - 一致しない 10 - はい 11 - いいえ

::: notetip さまざまなタイプの式でフィルターを使用する方法をよりよく理解するには、[action.get](#) および[action.create](#) メソッドページの例を参照してください。:::

次の演算子と値は、条件の種類ごとにサポートされています。

条件	条件名	対応演算子	期待値
0	Host group	equals, does not equal	ホストグループ ID
1	Host	equals, does not equal	ホスト ID
2	Trigger	equals, does not equal	トリガー ID
3	Trigger name	含む、 含まない	トリガー名
4	Trigger severity	等しい 等しくない 以上 以下	トリガーの深刻度。サポートされているトリガーの深刻度のリストについては、 トリガーの"深刻度"プロパティ を参照してください。
5	Trigger value	equals	トリガー値。サポートされているトリガー値のリストについては、 トリガー"値"プロパティ を参照してください。
6	Time period	入っている, 入っていない	イベントがトリガーされた 時間帯
7	Host IP	equals, does not equal	確認する 1 つまたは複数の IP 範囲をコンマで区切って指定します。サポートされている IP 範囲の形式の詳細については、 ネットワーク検出構成 セクションを参照してください。
8	Discovered service type	equals, does not equal	検出されたサービスのタイプ。サービスのタイプは、サービスの検出に使用される検出チェックのタイプと一致します。サポートされているタイプのリストについては、 検出チェックの"タイプ"プロパティ を参照してください。
9	Discovered service port	equals, does not equal	コンマで区切られた 1 つまたは複数のポート範囲。
10	Discovery status	equals	検出されたオブジェクトのステータス 使用可能な値: 0 - ホストまたはサービスが稼働中 1 - ホストまたはサービスが停止中 2 - ホストまたはサービスが検出済み 3 - ホストまたはサービスが失われました。
11	Uptime or downtime duration	is greater than or equals, is less than or equals	検出されたオブジェクトが現在の状態になっている時間 (秒単位)

条件	条件名	対応演算子	期待値
12	Received values	equals, does not equal, is greater than or equals, is less than or equals, contains, does not contain	Zabbix エージェント、SNMPv1、SNMPv2、または SNMPv3 の 検出チェックを実行したときに返される値
13	Host template	equals, does not equal	リンクされたテンプレート ID
16	Problem is suppressed	Yes,No	値は不要です: "Yes" - 障害を抑制します。"No" - 障害を抑制し ません。
18	Discovery rule	equals does not equal	ディスカバリルール ID
19	Discovery check	equals does not equal	ディスカバリチェック ID
20	Proxy	equals does not equal	プロキシ ID
21	Discovery object	equals	ディスカバリイベントをトリガーしたオブジェクトのタイプ 使用可能な値: 1 - 検出されたホスト 2 - 検出されたサービス
22	Host name	contains does not contain matches does not match	ホスト名 自動登録条件では matches および does not match で正規表現 の使用がサポートされています。
23	Event type	equals	特定の内部イベント 使用可能な値: 0 - "サポートされていない" 状態のアイテム 1 - "通常" の状態のアイテム 2 - "サポートされていない" 状態の LLD ルール 3 - "通常" 状態の LLD ルール 4 - "不明" 状態のトリガー 5 - "通常" 状態のトリガー
24	Host metadata	contains does not contain matches does not match	自動登録されたホストのメタデータ。 正規表現の使用は matches および does not match でサポート されています。
25	Tag	equals does not equal contains does not contain	イベントタグ
26	Tag value	equals does not equal contains does not contain	イベントタグ値
27	Service	equals does not equal	サービス ID
28	Service name	equals does not equal	サービス名

action.create

説明

`object action.create(object/array actions)`

このメソッドでは、新しいアクションを作成することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーのロール設定で取り消すことができます。詳しくは、[User roles](#)を参照してください。

パラメータ

(object/array) Actions to create.

standard action propertiesに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	Action filter object for the action.
operations	array	Action operations to create for the action.
recovery_operations	array	Action recovery operations to create for the action.
update_operations	array	Action update operations to create for the action.

戻り値

(object) は `actionids` プロパティの下に作成されたアクションの ID を含むオブジェクトを返します。
返される ID の順序は、渡されたアクションの順序と一致します。

例

trigger アクションの作成

ホスト "10084" からのトリガーで、その名前に "memory" が含まれているものが問題状態になったときに実行される
アクションを作成しなさい。このアクションは、まず、ユーザーグループ "7" のすべてのユーザーにメッセージを送信
しなければなりません。もし、このイベントが 4 分以内に解決されない場合、グループ "2" のすべてのホストでスクリプト
"3" を実行する。トリガー回復時には、その問題に関して以前にメッセージを受け取ったすべてのユーザーに通知されます。
トリガーが更新されると、メッセージの件名と本文を含むメッセージが、すべてのメディアタイプで備考とコメントを
残したすべてのユーザーに送信されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "action.create",
  "params": {
    "name": "Trigger action",
    "eventsources": 0,
    "status": 0,
    "esc_period": "2m",
    "filter": {
      "evaltype": 0,
      "conditions": [
        {
          "conditiontype": 1,
          "operator": 0,
          "value": "10084"
        },
        {
          "conditiontype": 3,
          "operator": 2,
          "value": "memory"
        }
      ]
    },
    "operations": [
      {
        "operationtype": 0,
        "esc_period": "0s",
        "esc_step_from": 1,
        "esc_step_to": 2,
        "evaltype": 0,
        "opmessage_grp": [
```



```
    ]
  },
  "id": 1
}
```

検出アクションの作成

検出されたホストをテンプレート"10091"にリンクさせるアクションを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "action.create",
  "params": {
    "name": "Discovery action",
    "eventsources": 1,
    "status": 0,
    "filter": {
      "evaltype": 0,
      "conditions": [
        {
          "conditiontype": 21,
          "operator": 0,
          "value": "1"
        },
        {
          "conditiontype": 10,
          "operator": 0,
          "value": "2"
        }
      ]
    },
    "operations": [
      {
        "operationtype": 6,
        "optemplate": [
          {
            "templateid": "10091"
          }
        ]
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "actionids": [
      "18"
    ]
  },
  "id": 1
}
```

カスタム式フィルタの使用

カスタムフィルタ条件を使用するトリガーアクションを作成します。このアクションは、ホスト"10084"と"10106"に
 対して、"Warning"以上の重大度を持つ各トリガーのメッセージを送信する必要があります。式 ID "A"、"B"、"C" は
 任意に選択しました。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "action.create",
  "params": {
    "name": "Trigger action",
    "eventsources": 0,
    "status": 0,
    "esc_period": "2m",
    "filter": {
      "evaltype": 3,
      "formula": "A and (B or C)",
      "conditions": [
        {
          "conditiontype": 4,
          "operator": 5,
          "value": "2",
          "formulaid": "A"
        },
        {
          "conditiontype": 1,
          "operator": 0,
          "value": "10084",
          "formulaid": "B"
        },
        {
          "conditiontype": 1,
          "operator": 0,
          "value": "10106",
          "formulaid": "C"
        }
      ]
    },
    "operations": [
      {
        "operationtype": 0,
        "esc_period": "0s",
        "esc_step_from": 1,
        "esc_step_to": 2,
        "evaltype": 0,
        "opmessage_grp": [
          {
            "usrgrp": "7"
          }
        ],
        "opmessage": {
          "default_msg": 1,
          "mediatypeid": "1"
        }
      }
    ],
    "pause_suppressed": "0",
    "notify_if_canceled": "0"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "actionids": [

```

```
        "18"  
      ]  
    },  
    "id": 1  
  }  
}
```

エージェントの自動登録ルール作成

ホスト名に"SRV"が含まれているか、メタデータに"AlmaLinux"が含まれている場合、ホストをホストグループ"Linux servers"に追加します。

リクエスト:

```
{  
  "jsonrpc": "2.0",  
  "method": "action.create",  
  "params": {  
    "name": "Register Linux servers",  
    "eventsourcetype": "2",  
    "status": "0",  
    "filter": {  
      "evaltype": "2",  
      "conditions": [  
        {  
          "conditiontype": "22",  
          "operator": "2",  
          "value": "SRV"  
        },  
        {  
          "conditiontype": "24",  
          "operator": "2",  
          "value": "AlmaLinux"  
        }  
      ]  
    },  
    "operations": [  
      {  
        "operationtype": "4",  
        "opgroup": [  
          {  
            "groupid": "2"  
          }  
        ]  
      }  
    ]  
  },  
  "auth": "038e1d7b1735c6a5436ee9eae095879e",  
  "id": 1  
}
```

レスポンス:

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "actionids": [  
      19  
    ]  
  },  
  "id": 1  
}
```

参照

- [アクションフィルター](#)
- [アクション操作](#)

- スクリプト

ソース

CAction::create() in ui/include/classes/api/services/CAction.php.

action.delete

説明

object action.delete(array actionIds)

このメソッドは、アクションを削除することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(array) IDs of the actions to delete.

戻り値

(object) は actionids プロパティの下にある、削除されたアクションの ID を含むオブジェクトを返します。

例

複数のアクションを削除する

2つのアクションを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "action.delete",
  "params": [
    "17",
    "18"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "actionids": [
      "17",
      "18"
    ]
  },
  "id": 1
}
```

ソース

CAction::delete() in ui/include/classes/api/services/CAction.php.

action.get

説明

integer/array action.get(object parameters)

このメソッドで、指定したパラメーターに応じたアクションを取得できます。

Note:

このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細については、[ユーザーの役割](#)を参照してください。

パラメータ

(object) は希望する出力を定義するパラメータ

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
actionids	string/array	指定された ID のアクションのみを返します。
groupids	string/array	アクション条件で指定されたホストグループを使用するアクションのみを返します。
hostids	string/array	アクション条件で指定されたホストを使用するアクションのみを返します。
triggerids	string/array	アクション条件で指定されたトリガーを使用するアクションのみを返します。
mediatypeids	string/array	指定されたメディアタイプを使用してメッセージを送信するアクションのみを返します。
usrgrpsids	string/array	指定されたユーザーグループにメッセージを送信するように構成されたアクションのみを返します。
userid	string/array	指定されたユーザーにメッセージを送信するように構成されたアクションのみを返します。
scriptids	string/array	指定されたスクリプトを実行するように構成されたアクションのみを返します。
selectFilter	query	アクション条件フィルターを含む filter プロパティを返します。
selectOperations	query	Return an operations property with action operations.
selectRecoveryOperations	query	Return a recovery_operations property with action recovery operations.
selectUpdateOperations	query	Return an update_operations property with action update operations.
sortfield	string/array	Sort the result by the given properties.
countOutput	boolean	Possible values are: actionid , name and status . These parameters being common for all get methods are described in the reference commentary .
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列。
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数。

例

トリガーアクションを取得する

設定されたすべてのトリガーアクションを、アクションの実行条件および実行内容と共に取得します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "action.get",
  "params": {
```

```

    "output": "extend",
    "selectOperations": "extend",
    "selectRecoveryOperations": "extend",
    "selectUpdateOperations": "extend",
    "selectFilter": "extend",
    "filter": {
        "eventsources": 0
    }
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

レスポンス:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "actionid": "3",
      "name": "Report problems to Zabbix administrators",
      "eventsources": "0",
      "status": "1",
      "esc_period": "1h",
      "pause_suppressed": "1",
      "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [],
        "eval_formula": ""
      },
      "operations": [
        {
          "operationid": "3",
          "actionid": "3",
          "operationtype": "0",
          "esc_period": "0",
          "esc_step_from": "1",
          "esc_step_to": "1",
          "evaltype": "0",
          "opconditions": [],
          "opmessage": [
            {
              "default_msg": "1",
              "subject": "",
              "message": "",
              "mediatypeid" => "0"
            }
          ],
          "opmessage_grp": [
            {
              "usrgrp": "7"
            }
          ]
        }
      ],
      "recovery_operations": [
        {
          "operationid": "7",
          "actionid": "3",
          "operationtype": "11",
          "evaltype": "0",
          "opconditions": [],

```

```

        "opmessage": {
            "default_msg": "0",
            "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
            "message": "Trigger: {TRIGGER.NAME}\r\nTrigger status: {TRIGGER.STATUS}\r\nTrigger
            "mediatypeid": "0"
        }
    },
    ],
    "update_operations": [
        {
            "operationid": "31",
            "operationtype": "12",
            "evaltype": "0",
            "opmessage": {
                "default_msg": "1",
                "subject": "",
                "message": "",
                "mediatypeid": "0"
            }
        },
        {
            "operationid": "32",
            "operationtype": "0",
            "evaltype": "0",
            "opmessage": {
                "default_msg": "0",
                "subject": "Updated: {TRIGGER.NAME}",
                "message": "{USER.FULLNAME} updated problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.T
                "mediatypeid": "1"
            },
            "opmessage_grp": [
                {
                    "usrgrp": "7"
                }
            ],
            "opmessage_usr": []
        },
        {
            "operationid": "33",
            "operationtype": "1",
            "evaltype": "0",
            "opcommand": {
                "scriptid": "3"
            },
            "opcommand_hst": [
                {
                    "hostid": "10084"
                }
            ],
            "opcommand_grp": []
        }
    ]
}
],
"id": 1
}

```

ディスカバリーアクションの取得

設定されたすべてのディスカバリーアクションを、アクションの条件と操作とともに取得します。このフィルターは"and" 評価タイプを使用するため、formula プロパティは空で、eval_formula が自動的に生成されます。

リクエスト:

```

{
  "jsonrpc": "2.0",
  "method": "action.get",
  "params": {
    "output": "extend",
    "selectOperations": "extend"
    "selectFilter": "extend",
    "filter": {
      "eventsources": 1
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

レスポンス:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "actionid": "2",
      "name": "Auto discovery. Linux servers.",
      "eventsources": "1",
      "status": "1",
      "esc_period": "0s",
      "pause_suppressed": "1",
      "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [
          {
            "conditiontype": "10",
            "operator": "0",
            "value": "0",
            "value2": "",
            "formulaid": "B"
          },
          {
            "conditiontype": "8",
            "operator": "0",
            "value": "9",
            "value2": "",
            "formulaid": "C"
          },
          {
            "conditiontype": "12",
            "operator": "2",
            "value": "Linux",
            "value2": "",
            "formulaid": "A"
          }
        ]
      },
      "eval_formula": "A and B and C"
    },
    {
      "operations": [
        {
          "operationid": "1",
          "actionid": "2",
          "operationtype": "6",
          "esc_period": "0s",
          "esc_step_from": "1",
          "esc_step_to": "1",

```



```

        "evaltype": "0",
        "opconditions": [],
        "optemplate": [
            {
                "templateid": "10001"
            }
        ]
    },
    {
        "operationid": "2",
        "actionid": "2",
        "operationtype": "4",
        "esc_period": "0s",
        "esc_step_from": "1",
        "esc_step_to": "1",
        "evaltype": "0",
        "opconditions": [],
        "opgroup": [
            {
                "groupid": "2"
            }
        ]
    }
]
}
],
"id": 1
}
}

```

参照

- [Action filter](#)
- [Action operation](#)

ソース

CAction::get() in ui/include/classes/api/services/CAction.php.

action.update

説明

object action.update(object/array actions)

このメソッドにより、既存のアクションを更新することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) Action properties to be updated.

actionid プロパティは各アクションに対して定義する必要があり、その他のプロパティは任意です。渡されたプロパティのみが

更新され、他のプロパティは変更されません。

[standard action properties](#)に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	Action filter object to replace the current filter.
operations	array	Action operations to replace existing operations.
recovery_operations	array	Action recovery operations to replace existing recovery operations.
update_operations	array	Action update operations to replace existing update operations.

戻り値

(object) は `actionids` プロパティの下で更新されたアクションの ID を含むオブジェクトを返します。

例

アクションの無効化

ステータスを「1」に設定してアクションを無効にします。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "action.update",
  "params": {
    "actionid": "2",
    "status": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "actionids": [
      "2"
    ]
  },
  "id": 1
}
```

参照

- [Action filter](#)
- [Action operation](#)

ソース

`CAction::update()` in `ui/include/classes/api/services/CAction.php`.

Alert

このクラスは、アラートと連携することを想定しています。

オブジェクトリファレンス:

- [Alert](#)

利用可能なメソッド:

- `alert.get` - retrieve alerts

> Alert object

以下のオブジェクトは `alert` API に直接関連するものです。

Alert

Note:

Alert は Zabbix server によって作成され、API から変更することはできません。

Alert オブジェクトは、特定のアクション操作が正常に実行されたかどうかに関する情報を含んでいます。
 以下のプロパティを持ちます。

プロパティ	Type	説明
alertid	string	ID of the alert.
actionid	string	ID of the action that generated the alert.
alerttype	integer	Alert type. Possible values: 0 - message; 1 - remote command.
clock	timestamp	Time when the alert was generated.
error	string	Error text if there are problems sending a message or running a command.
esc_step	integer	Action escalation step during which the alert was generated.
eventid	string	ID of the event that triggered the action.
mediatypeid	string	ID of the media type that was used to send the message.
message	text	Message text. Used for message alerts.
retries	integer	Number of times Zabbix tried to send the message.
sendto	string	Address, user name or other identifier of the recipient. Used for message alerts.
status	integer	Status indicating whether the action operation has been executed successfully. Possible values for message alerts: 0 - message not sent. 1 - message sent. 2 - failed after a number of retries. 3 - new alert is not yet processed by alert manager. Possible values for command alerts: 0 - command not run. 1 - command run. 2 - tried to run the command on the Zabbix agent but it was unavailable.
subject	string	Message subject. Used for message alerts.
userid	string	ID of the user that the message was sent to.
p_eventid	string	ID of problem event, which generated the alert.
acknowledgeid	string	ID of acknowledgment, which generated the alert.

alert.get

説明

integer/array alert.get(object parameters)

このメソッドは、指定されたパラメータに従ってアラートを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
alertids	string/array	Return only alerts with the given IDs.
actionids	string/array	Return only alerts generated by the given actions.
eventids	string/array	Return only alerts generated by the given events.
groupids	string/array	Return only alerts generated by objects from the given host groups.
hostids	string/array	Return only alerts generated by objects from the given hosts.
mediatypeids	string/array	Return only message alerts that used the given media types.

パラメータ	Type	説明
objectids	string/array	Return only alerts generated by the given objects
userids	string/array	Return only message alerts that were sent to the given users.
eventobject	integer	Return only alerts generated by events related to objects of the given type. See event " object " for a list of supported object types.
eventsources	integer	Default: 0 - trigger. Return only alerts generated by events of the given type. See event " source " for a list of supported event types.
time_from	timestamp	Default: 0 - trigger events. Return only alerts that have been generated after the given time.
time_till	timestamp	Return only alerts that have been generated before the given time.
selectHosts	query	Return a hosts property with data of hosts that triggered the action operation.
selectMediatypes	query	Return a mediatypes property with an array of the media types that were used for the message alert.
selectUsers	query	Return a users property with an array of the users that the message was addressed to.
sortfield	string/array	Sort the result by the given properties. Possible values are: alertid , clock , eventid , mediatypeid , sendto and status .
countOutput	boolean	These parameters being common for all get methods are described in the reference commentary .
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

アクション ID でアラートを取得する

アクション"3"によって生成されたすべてのアラートを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "alert.get",
  "params": {
    "output": "extend",
    "actionids": "3"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "alertid": "1",
      "actionid": "3",
      "eventid": "21243",
      "userid": "1",
      "clock": "1362128008",
      "mediatypeid": "1",
      "sendto": "support@company.com",
      "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
      "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat",
      "status": "0",
      "retries": "3",
      "error": "",
      "esc_step": "1",
      "alerttype": "0",
      "p_eventid": "0",
      "acknowledgeid": "0"
    }
  ],
  "id": 1
}
```

参照

- [Host](#)
- [Media type](#)
- [User](#)

ソース

CAAlert::get() in ui/include/classes/api/services/CAAlert.php.

API info

このクラスは、API に関するメタ情報を取得するためのものです。

利用可能なメソッド:

- [apiinfo.version](#) - Zabbix API のバージョンの取得

apiinfo.version

説明

string apiinfo.version(array)

このメソッドでは、Zabbix API のバージョンを取得することができます。

Attention:

このメソッドは未認証のユーザーだけが利用できます。
JSON-RPC リクエストの auth パラメータを指定せずに呼び出す必要があります。

パラメータ

(array) このメソッドには、空の配列を指定します。

戻り値

(string) Zabbix API のバージョンを返します。

Note:

Zabbix 2.0.4 から、API のバージョンは Zabbix のバージョンと一致します。

例

API バージョンの取得

Zabbix API のバージョンを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "apiinfo.version",
  "params": [],
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": "4.0.0",
  "id": 1
}
```

ソース

CAPInfo::version() in ui/include/classes/api/services/CAPInfo.php.

Audit log

このクラスは、監査ログと連動するように設計されています。

オブジェクトリファレンス:

- **Audit log object**

利用可能なメソッド:

- **auditlog.get** - 監査ログを取得する

> Audit log object

以下のオブジェクトは auditlog API に直接関連するものである。

Audit log

Audit log オブジェクトは、ユーザーのアクションに関する情報を含んでいます。以下のプロパティを持ちます。

プロパティ	Type	説明
auditid	string	(readonly) ID of audit log entry. Generated using CUID algorithm.
userid	string	Audit log entry author userid.
username	string	Audit log entry author username.
clock	timestamp	Audit log entry creation timestamp.
ip	string	Audit log entry author IP address.

プロパティ	Type	説明
action	integer	Audit log entry action. Possible values are: 0 - Add; 1 - Update; 2 - Delete; 4 - Logout; 7 - Execute; 8 - Login; 9 - Failed login; 10 - History clear.
resourcetype	integer	Audit log entry resource type. Possible values are: 0 - User; 3 - Media type; 4 - Host; 5 - Action; 6 - Graph; 11 - User group; 13 - Trigger; 14 - Host group; 15 - Item; 16 - Image; 17 - Value map; 18 - Service; 19 - Map; 22 - Web scenario; 23 - Discovery rule; 25 - Script; 26 - Proxy; 27 - Maintenance; 28 - Regular expression; 29 - Macro; 30 - Template; 31 - Trigger prototype; 32 - Icon mapping; 33 - Dashboard; 34 - Event correlation; 35 - Graph prototype; 36 - Item prototype; 37 - Host prototype; 38 - Autoregistration; 39 - Module; 40 - Settings; 41 - Housekeeping; 42 - Authentication; 43 - Template dashboard; 44 - User role; 45 - Auth token; 46 - Scheduled report; 47 - High availability node; 48 - SLA.
resourceid	string	Audit log entry resource identifier.
resourceName	string	Audit log entry resource human readable name.
recordsetid	string	Audit log entry recordset ID. The audit log records created during the same operation will have the same recordset ID. Generated using CUID algorithm.

プロパティ	Type	説明
details	text	<p>Audit log entry details. The details are stored as JSON object where each property name is a path to property or nested object in which change occurred, and each value contain the data about the change of this property in array format.</p> <p>Possible value formats are: ["add"] - Nested object has been added; ["add", "<value>"] - The property of added object contain <value>; ["update"] - Nested object has been updated; ["update", "<new value>", "<old value>"] - The value of property of updated object was changed from <old value> to <new value>; ["delete"] - Nested object has been deleted.</p>

auditlog.get

説明

integer/array auditlog.get(object parameters)

このメソッドは、与えられたパラメータに従って監査ログレコードを取得することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
auditids	string/array	Return only audit log with the given IDs.
userids	string/array	Return only audit log that were created by the given users.
time_from	timestamp	Returns only audit log entries that have been created after or at the given time.
time_till	timestamp	Returns only audit log entries that have been created before or at the given time.
sortfield	string/array	Sort the result by the given properties.
filter	object	<p>Possible values are: auditid, userid, clock. Return only results that exactly match the given filter.</p> <p>Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against.</p> <p>Additionally supports filtering by details property fields: table_name, field_name.</p>
search	object	Case insensitive sub-string search in content of fields: username, ip, resourcename, details.
countOutput	boolean	These parameters being common for all get methods are described in the reference commentary .
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

監査ログの取得

最新の監査ログを 2 件取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "auditlog.get",
  "params": {
    "output": "extend",
    "sortfield": "clock",
    "sortorder": "DESC",
    "limit": 2
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "auditid": "cksstgfam0001yhdcc41y20q2",
      "userid": "1",
      "username": "Admin",
      "clock": "1629975715",
      "ip": "127.0.0.1",
      "action": "1",
      "resourcetype": "0",
      "resourceid": "0",
      "resourcename": "Jim",
      "recordsetid": "cksstgfal0000yhdcso67ondl",
      "details": "{\"user.name\": [\"update\", \"Jim\", \"\"], \"user.medias [37]\": [\"add\"], \"user.medias [37].id\": [\"1\"]}"
    },
    {
      "auditid": "ckssofl0p0001yhdcqxclsg8r",
      "userid": "1",
      "username": "Admin",
      "clock": "1629967278",
      "ip": "127.0.0.1",
      "action": "0",
      "resourcetype": "0",
      "resourceid": "20",
      "resourcename": "John",
      "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
      "details": "{\"user.username\": [\"add\", \"John\"], \"user.userid\": [\"add\", \"20\"], \"user.userid\": [\"20\"]}"
    }
  ],
  "id": 1
}
```

参照

- [Audit log object](#)

ソース

CAuditLog::get() in ui/include/classes/api/services/CAuditLog.php.

Authentication

このクラスは、認証設定と連動するように設計されています。

オブジェクトリファレンス:

- **Authentication**

利用可能なメソッド:

- **authentication.get** - 認証を取得する
- **authentication.update** - 認証を更新する

> Authentication object

以下のオブジェクトは authentication API に直接関連するものである。

Authentication

Authentication オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
authentication_type	integer	Default authentication. Possible values: 0 - (default) Internal; 1 - LDAP.
http_auth_enabled	integer	Enable HTTP authentication. Possible values: 0 - (default) Disable; 1 - Enable.
http_login_form	integer	Default login form. Possible values: 0 - (default) Zabbix login form; 1 - HTTP login form.
http_strip_domains	string	Remove domain name.
http_case_sensitive	integer	HTTP case sensitive login. Possible values: 0 - Off; 1 - (default) On.
ldap_configured	integer	Enable LDAP authentication. Possible values: 0 - Disable; 1 - (default) Enable.
ldap_host	string	LDAP host.
ldap_port	integer	LDAP port.
ldap_base_dn	string	LDAP base DN.
ldap_search_attribute	string	LDAP search attribute.
ldap_bind_dn	string	LDAP bind DN.
ldap_case_sensitive	integer	LDAP case sensitive login. Possible values: 0 - Off; 1 - (default) On.
ldap_bind_password	string	LDAP bind password.

プロパティ	Type	説明
saml_auth_enabled	integer	Enable SAML authentication. Possible values: 0 - (default) Disable; 1 - Enable.
saml_idp_entityid	string	SAML IdP entity ID.
saml_sso_url	string	SAML SSO service URL.
saml_slo_url	string	SAML SLO service URL.
saml_username_attribute	string	SAML username attribute.
saml_sp_entityid	string	SAML SP entity ID.
saml_nameid_format	string	SAML SP name ID format.
saml_sign_messages	integer	SAML sign messages. Possible values: 0 - (default) Do not sign messages; 1 - Sign messages.
saml_sign_assertions	integer	SAML sign assertions. Possible values: 0 - (default) Do not sign assertions; 1 - Sign assertions.
saml_sign_authn_requests	integer	SAML sign AuthN requests. Possible values: 0 - (default) Do not sign AuthN requests; 1 - Sign AuthN requests.
saml_sign_logout_requests	integer	SAML sign logout requests. Possible values: 0 - (default) Do not sign logout requests; 1 - Sign logout requests.
saml_sign_logout_responses	integer	SAML sign logout responses. Possible values: 0 - (default) Do not sign logout responses; 1 - Sign logout responses.
saml_encrypt_nameid	integer	SAML encrypt name ID. Possible values: 0 - (default) Do not encrypt name ID; 1 - Encrypt name ID.
saml_encrypt_assertions	integer	SAML encrypt assertions. Possible values: 0 - (default) Do not encrypt assertions; 1 - Encrypt assertions.
saml_case_sensitive	integer	SAML case sensitive login. Possible values: 0 - Off; 1 - (default) On.
passwd_min_length	integer	Password minimal length requirement. Possible range of values: 1-70 8 - default

プロパティ	Type	説明
passwd_check_rules	integer	Password checking rules. Possible bitmap values are: 0 - check password length; 1 - check if password uses uppercase and lowercase Latin letters; 2 - check if password uses digits; 4 - check if password uses special characters; 8 - (default) check if password is not in the list of commonly used passwords, does not contain derivations of word "Zabbix" or user's name, last name or username.

authentication.get

説明

object authentication.get(object parameters)

このメソッドは、与えられたパラメータに従って、認証オブジェクトを取得することができます。

Note:

このメソッドは、Super admin ユーザータイプでのみ利用可能です。このメソッドを呼び出すためのパーミッションは、
ユーザーロール設定で取り消すことができます。詳しくは、[User roles](#)をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	This parameter being common for all get methods described in the reference commentary .

戻り値

(object) 認証オブジェクトを戻します。

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "authentication.get",
  "params": {
    "output": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "authentication_type": "0",
    "http_auth_enabled": "0",
    "http_login_form": "0",
    "http_strip_domains": "",
    "http_case_sensitive": "1",
    "ldap_configured": "0",
    "ldap_host": ""
  }
}
```

```

        "ldap_port": "389",
        "ldap_base_dn": "",
        "ldap_search_attribute": "",
        "ldap_bind_dn": "",
        "ldap_case_sensitive": "1",
        "ldap_bind_password": "",
        "saml_auth_enabled": "0",
        "saml_idp_entityid": "",
        "saml_sso_url": "",
        "saml_slo_url": "",
        "saml_username_attribute": "",
        "saml_sp_entityid": "",
        "saml_nameid_format": "",
        "saml_sign_messages": "0",
        "saml_sign_assertions": "0",
        "saml_sign_authn_requests": "0",
        "saml_sign_logout_requests": "0",
        "saml_sign_logout_responses": "0",
        "saml_encrypt_nameid": "0",
        "saml_encrypt_assertions": "0",
        "saml_case_sensitive": "0",
        "passwd_min_length": "8",
        "passwd_check_rules": "8"
    },
    "id": 1
}

```

ソース

CAuthentication::get() in ui/include/classes/api/services/CAuthentication.php.

authentication.update

説明

object authentication.update(object authentication)

この方法では、既存の認証設定を更新することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
 ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object) 更新する認証のプロパティ

戻り値

(array) 更新されたパラメータの名前を配列で返します。

例

Request:

```

{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}

```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    "http_auth_enabled",
    "http_case_sensitive",
    "http_login_form"
  ],
  "id": 1
}
```

ソース

CAuthentication::update() in ui/include/classes/api/services/CAuthentication.php.

Autoregistration

このクラスは、自動登録と連動するように設計されています。

オブジェクトリファレンス:

- [Autoregistration](#)

利用可能なメソッド:

- [autoregistration.get](#) - 自動登録を取得する
- [autoregistration.update](#) - 自動登録を更新する

> Autoregistration object

以下のオブジェクトは autoregistration API に直接関連するものです。

Autoregistration

Autoregistration オブジェクトは、以下のプロパティを持つ。

プロパティ	Type	説明
tls_accept	integer	Type of allowed incoming connections for autoregistration. Possible values: 1 - allow insecure connections; 2 - allow TLS with PSK. 3 - allow both insecure and TLS with PSK connections.
tls_psk_identity	string	(write-only) PSK identity string. Do not put sensitive information in the PSK identity, it is transmitted unencrypted over the network to inform a receiver which PSK to use.
tls_psk	string	(write-only) PSK value string (an even number of hexadecimal characters).

autoregistration.get

説明

object autoregistration.get(object parameters)

このメソッドは、与えられたパラメータに従って、自動登録オブジェクトを取得することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは 1 つのパラメータのみをサポートしています。

パラメータ	Type	説明
output	query	This parameter being common for all get methods described in the reference commentary .

戻り値

(object) autoregistration オブジェクトを返します。

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "autoregistration.get",
  "params": {
    "output": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "tls_accept": "3"
  },
  "id": 1
}
```

ソース

CAutoregistration::get() in ui/include/classes/api/services/CAutoregistration.php.

autoregistration.update

説明

object autoregistration.update(object autoregistration)

このメソッドは、既存の自動登録の更新を行うことができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

Parameters

(object) 更新する自動登録のプロパティ

戻り値

(boolean) 更新に成功した場合、結果として True を返します。

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "autoregistration.update",
  "params": {
    "tls_accept": "3",
  }
}
```

```
    "tls_psk_identity": "PSK 001",
    "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 1
}
```

ソース

CAutoregistration::update() in ui/include/classes/api/services/CAutoregistration.php.

Configuration

このクラスは、Zabbix 設定データのエクスポートとインポートを行うためのものです。

利用可能なメソッド:

- `configuration.export` - 設定のエクスポート
- `configuration.import` - 設定のインポート
- `configuration.importcompare` - インポートファイルと現在のシステムの要素の比較

configuration.import

説明

`boolean configuration.import(object parameters)`

このメソッドは、シリアル化された文字列から設定データをインポートすることができます。

Note:

このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細は、[ユーザーの役割](#)を参照してください。

パラメータ

(object) インポートするデータと、そのデータをどのように処理するか of ルールを含むパラメータ

パラメータ	Type	説明
format (required)	string	Format of the serialized string. Possible values: yaml - YAML; xml - XML; json - JSON.
source (required)	string	Serialized string containing the configuration data.
rules (required)	object	Rules on how new and existing objects should be imported. The rules parameter is described in detail in the table below.

Note:

ルールが与えられない場合、設定は更新されません。

rules オブジェクトは以下のパラメータをサポートしています。

パラメータ	Type	説明
discoveryRules	object	<p>Rules on how to import LLD rules.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new LLD rules will be created; default: false; updateExisting - (boolean) if set to true, existing LLD rules will be updated; default: false; deleteMissing - (boolean) if set to true, LLD rules not present in the imported data will be deleted from the database; default: false.</p>
graphs	object	<p>Rules on how to import graphs.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new graphs will be created; default: false; updateExisting - (boolean) if set to true, existing graphs will be updated; default: false; deleteMissing - (boolean) if set to true, graphs not present in the imported data will be deleted from the database; default: false.</p>
groups	object	<p>Rules on how to import host groups.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new host groups will be created; default: false; updateExisting - (boolean) if set to true, existing host groups will be updated; default: false.</p>
hosts	object	<p>Rules on how to import hosts.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new hosts will be created; default: false; updateExisting - (boolean) if set to true, existing hosts will be updated; default: false.</p>
httptests	object	<p>Rules on how to import web scenarios.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new web scenarios will be created; default: false; updateExisting - (boolean) if set to true, existing web scenarios will be updated; default: false; deleteMissing - (boolean) if set to true, web scenarios not present in the imported data will be deleted from the database; default: false.</p>
images	object	<p>Rules on how to import images.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new images will be created; default: false; updateExisting - (boolean) if set to true, existing images will be updated; default: false.</p>
items	object	<p>Rules on how to import items.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new items will be created; default: false; updateExisting - (boolean) if set to true, existing items will be updated; default: false; deleteMissing - (boolean) if set to true, items not present in the imported data will be deleted from the database; default: false.</p>

パラメータ	Type	説明
maps	object	<p>Rules on how to import maps.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new maps will be created; default: false; updateExisting - (boolean) if set to true, existing maps will be updated; default: false.</p>
mediaTypes	object	<p>Rules on how to import media types.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new media types will be created; default: false; updateExisting - (boolean) if set to true, existing media types will be updated; default: false.</p>
templateLinkage	object	<p>Rules on how to import template links.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new links between templates and host will be created; default: false; deleteMissing - (boolean) if set to true, template links not present in the imported data will be deleted from the database; default: false.</p>
templates	object	<p>Rules on how to import templates.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new templates will be created; default: false; updateExisting - (boolean) if set to true, existing templates will be updated; default: false.</p>
templateDashboards	object	<p>Rules on how to import template dashboards.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new template dashboards will be created; default: false; updateExisting - (boolean) if set to true, existing template dashboards will be updated; default: false; deleteMissing - (boolean) if set to true, template dashboards not present in the imported data will be deleted from the database; default: false.</p>
triggers	object	<p>Rules on how to import triggers.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new triggers will be created; default: false; updateExisting - (boolean) if set to true, existing triggers will be updated; default: false; deleteMissing - (boolean) if set to true, triggers not present in the imported data will be deleted from the database; default: false.</p>
valueMaps	object	<p>Rules on how to import host or template value maps.</p> <p>Supported parameters: createMissing - (boolean) if set to true, new value maps will be created; default: false; updateExisting - (boolean) if set to true, existing value maps will be updated; default: false; deleteMissing - (boolean) if set to true, value maps not present in the imported data will be deleted from the database; default: false.</p>

戻り値

(boolean) インポートに成功した場合は true を返します。

例

ホストとアイテムのインポート

XML 文字列に含まれるホストと item をインポートします。XML に含まれる item が欠落している場合、その item は
 データベースから削除され、それ以外は変更されずに残ります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "configuration.import",
  "params": {
    "format": "xml",
    "rules": {
      "valueMaps": {
        "createMissing": true,
        "updateExisting": false
      },
      "hosts": {
        "createMissing": true,
        "updateExisting": true
      },
      "items": {
        "createMissing": true,
        "updateExisting": true,
        "deleteMissing": true
      }
    },
    "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>5.4</version><date>"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 1
}
```

ソース

CConfiguration::import() は ui/include/classes/api/services/CConfiguration.php にあります。

configuration.importcompare

説明

array configuration.importcompare(object parameters)

このメソッドは、インポートファイルを現在のシステム要素と比較することを可能にし、このインポートファイルがインポートされる場合に何が変更されるかを示します。

Note:

このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細は、[ユーザーの役割](#)を参照してください。

パラメータ

(object) インポート可能なデータと、そのデータをどのように処理するかのルールを含むパラメータ

パラメータ	Type	説明
format (required)	string	Format of the serialized string. Possible values: yaml - YAML; xml - XML; json - JSON.
source (required)	string	Serialized string containing the configuration data.
rules (required)	object	Rules on how new and existing objects should be imported. The rules parameter is described in detail in the table below.

Note:

ルールが与えられていない場合、更新するものはなく、結果は空になります。

Note:

比較は、ホストグループとテンプレートに対してのみ行われます。トリガーとグラフの比較は、インポートされたテンプレートに
 に対してのみ行われ、それ以外は「新規」とみなされます。

rules オブジェクトは以下のパラメータをサポートしています。

パラメータ	Type	説明
discoveryRules	object	Rules on how to import LLD rules. Supported parameters: createMissing - (boolean) if set to true, new LLD rules will be created; default: false; updateExisting - (boolean) if set to true, existing LLD rules will be updated; default: false; deleteMissing - (boolean) if set to true, LLD rules not present in the imported data will be deleted from the database; default: false.
graphs	object	Rules on how to import graphs. Supported parameters: createMissing - (boolean) if set to true, new graphs will be created; default: false; updateExisting - (boolean) if set to true, existing graphs will be updated; default: false; deleteMissing - (boolean) if set to true, graphs not present in the imported data will be deleted from the database; default: false.
groups	object	Rules on how to import host groups. Supported parameters: createMissing - (boolean) if set to true, new host groups will be created; default: false; updateExisting - (boolean) if set to true, existing host groups will be updated; default: false.
hosts	object	Rules on how to import hosts. Supported parameters: createMissing - (boolean) if set to true, new hosts will be created; default: false; updateExisting - (boolean) if set to true, existing hosts will be updated; default: false. This parameter will make no difference to the output. It is allowed only for consistency with configuration.import.

パラメータ	Type	説明
httptests	object	<p>Rules on how to import web scenarios.</p> <p>Supported parameters: <code>createMissing</code> - (boolean) if set to true, new web scenarios will be created; default: false; <code>updateExisting</code> - (boolean) if set to true, existing web scenarios will be updated; default: false; <code>deleteMissing</code> - (boolean) if set to true, web scenarios not present in the imported data will be deleted from the database; default: false.</p>
images	object	<p>Rules on how to import images.</p> <p>Supported parameters: <code>createMissing</code> - (boolean) if set to true, new images will be created; default: false; <code>updateExisting</code> - (boolean) if set to true, existing images will be updated; default: false.</p> <p>This parameter will make no difference to the output. It is allowed only for consistency with <code>configuration.import</code>.</p>
items	object	<p>Rules on how to import items.</p> <p>Supported parameters: <code>createMissing</code> - (boolean) if set to true, new items will be created; default: false; <code>updateExisting</code> - (boolean) if set to true, existing items will be updated; default: false; <code>deleteMissing</code> - (boolean) if set to true, items not present in the imported data will be deleted from the database; default: false.</p>
maps	object	<p>Rules on how to import maps.</p> <p>Supported parameters: <code>createMissing</code> - (boolean) if set to true, new maps will be created; default: false; <code>updateExisting</code> - (boolean) if set to true, existing maps will be updated; default: false.</p> <p>This parameter will make no difference to the output. It is allowed only for consistency with <code>configuration.import</code>.</p>
mediaTypes	object	<p>Rules on how to import media types.</p> <p>Supported parameters: <code>createMissing</code> - (boolean) if set to true, new media types will be created; default: false; <code>updateExisting</code> - (boolean) if set to true, existing media types will be updated; default: false.</p> <p>This parameter will make no difference to the output. It is allowed only for consistency with <code>configuration.import</code>.</p>
templateLinkage	object	<p>Rules on how to import template links.</p> <p>Supported parameters: <code>createMissing</code> - (boolean) if set to true, new links between templates and host will be created; default: false; <code>deleteMissing</code> - (boolean) if set to true, template links not present in the imported data will be deleted from the database; default: false.</p>

パラメータ	Type	説明
templates	object	Rules on how to import templates. Supported parameters: createMissing - (boolean) if set to true, new templates will be created; default: false; updateExisting - (boolean) if set to true, existing templates will be updated; default: false.
templateDashboards	object	Rules on how to import template dashboards. Supported parameters: createMissing - (boolean) if set to true, new template dashboards will be created; default: false; updateExisting - (boolean) if set to true, existing template dashboards will be updated; default: false; deleteMissing - (boolean) if set to true, template dashboards not present in the imported data will be deleted from the database; default: false.
triggers	object	Rules on how to import triggers. Supported parameters: createMissing - (boolean) if set to true, new triggers will be created; default: false; updateExisting - (boolean) if set to true, existing triggers will be updated; default: false; deleteMissing - (boolean) if set to true, triggers not present in the imported data will be deleted from the database; default: false.
valueMaps	object	Rules on how to import host or template value maps. Supported parameters: createMissing - (boolean) if set to true, new value maps will be created; default: false; updateExisting - (boolean) if set to true, existing value maps will be updated; default: false; deleteMissing - (boolean) if set to true, value maps not present in the imported data will be deleted from the database; default: false.

返却値

(array) 行われる構成の変更を含む配列を返します。

例

ホストと item のインポート

YAML 文字列に含まれるテンプレートと item をインポートします。YAML に含まれていない項目がある場合、それらは削除された
 ものとして表示され、それ以外は変更されずに残ります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "configuration.import",
  "params": {
    "format": "xml",
    "rules": {
      "groups": {
        "createMissing": true,
        "updateExisting": true
      },
      "templates": {
        "createMissing": true,
        "updateExisting": true
      },
      "items": {
```

```

        "createMissing": true,
        "updateExisting": true,
        "deleteMissing": true
    },
    "triggers": {
        "createMissing": true,
        "updateExisting": true,
        "deleteMissing": true
    },
    "discoveryRules": {
        "createMissing": true,
        "updateExisting": true,
        "deleteMissing": true
    },
    "valueMaps": {
        "createMissing": true,
        "updateExisting": false
    }
},
"source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><zabbix_export><version>5.4</version><date>20
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "templates": {
      "updated": [
        {
          "before": {
            "uuid": "e1bde9bf2f0544f5929f45b82502e744",
            "template": "Export template",
            "name": "Export template"
          },
          "after": {
            "uuid": "e1bde9bf2f0544f5929f45b82502e744",
            "template": "Export template",
            "name": "Export template"
          },
          "items": {
            "added": [
              {
                "after": {
                  "uuid": "3237bc89226e42ed8207574022470e83",
                  "name": "Item",
                  "key": "item.key",
                  "delay": "30s",
                  "valuemap": {
                    "name": "Host status"
                  }
                }
              },
              {
                "triggers": {
                  "added": [
                    {
                      "after": {
                        "uuid": "bd1ed0089e4b4f35b762c9d6c599c348",
                        "expression": "last(/Export template/item.key)=0",
                        "name": "Trigger"
                      }
                    }
                  ]
                }
              }
            ]
          }
        }
      ]
    }
  }
}

```

```

        }
    ]
}
},
],
"removed":[
    {
        "before":{
            "uuid":"bd3e7b28b3d544d6a83ed01ddaa65ab6",
            "name":"Old Item",
            "key":"ite_old.key",
            "delay":"30s",
            "valuemap":{
                "name":"Host status"
            }
        }
    }
]
},
"discovery_rules":{
    "updated":[
        {
            "before":{
                "uuid":"c91616bcf4a44f349539a1b40cb0979d",
                "name":"Discovery rule",
                "key":"rule.key"
            },
            "after":{
                "uuid":"c91616bcf4a44f349539a1b40cb0979d",
                "name":"Discovery rule",
                "key":"rule.key"
            },
            "item_prototypes":{
                "updated":[
                    {
                        "before":{
                            "uuid":"7e164881825744248b3039af3435cf4b",
                            "name":"Old item prototype",
                            "key":"prototype_old.key"
                        },
                        "after":{
                            "uuid":"7e164881825744248b3039af3435cf4b",
                            "name":"Item prototype",
                            "key":"prototype.key"
                        }
                    }
                ]
            }
        }
    ]
}
},
},
    "id":1
}
}

```

ソース

CConfiguration::importcompare() は ui/include/classes/api/services/CConfiguration.php にあります。

設定のエクスポート

説明

```
string configuration.export(object parameters)
```

このメソッドを使用すると、構成データをシリアル化された文字列としてエクスポートできます。

Note:

このメソッドは全てのタイプのユーザーが利用できます。メソッドを呼び出す権限はユーザーのユーザーロール設定で取り消すことができます。詳細については[ユーザーロール](#)を参照してください。

パラメーター

(object) エクスポートするオブジェクトと使用する形式を定義するパラメーター。

パラメータ	タイプ	説明
format (必須)	string	データのエクスポート形式 設定可能な値: yaml - YAML; xml - XML; json - JSON; raw - 未処理の PHP 配列。
prettyprint	boolean	インデントを追加して人間が読みやすい出力にします。 設定可能な値: true - インデントを追加します。 false - (デフォルト) インデントを追加しません。
options (必須)	object	エクスポートされるオブジェクト options オブジェクトには次のパラメータがあります: groups - (array) エクスポートするホストグループ ID hosts - (array) エクスポートするホスト ID images - (array) エクスポートするイメージ ID maps - (array) エクスポートするマップ ID mediaTypes - (array) エクスポートするメディアタイプ ID templates - (array) エクスポートするテンプレートの ID

戻り値

(string) 要求された構成データを含むシリアル化された文字列を返します。

例

テンプレートのエクスポート

Export the configuration of template "10571" as an XML string. テンプレート "10571" の設定を XML 文字列としてエクスポートします。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "configuration.export",
  "params": {
    "options": {
      "templates": [
        "10571"
      ]
    },
  },
  "format": "xml"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
```

```
"id": 1
}
```

レスポンス:

```
{
"jsonrpc": "2.0",
"result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>6.0</version><date>2023-01-
" id": 1
}
```

ソース

CConfiguration::export() は ui/include/classes/api/services/CConfiguration.php にあります。

Correlation

このクラスは、相関関係を扱うように設計されています。

オブジェクトリファレンス:

- [Correlation](#)

利用可能なメソッド:

- [correlation.create](#) - 新しい相関関係を作る
- [correlation.delete](#) - 相関関係の削除
- [correlation.get](#) - 相関関係の検索
- [correlation.update](#) - 相関関係の更新

> Correlation object

以下のオブジェクトは correlation API に直接関連するものです。

Correlation

Correlation オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
correlationid	string	(readonly) ID of the correlation.
name (required)	string	Name of the correlation.
description	string	Description of the correlation.
status	integer	Whether the correlation is enabled or disabled. Possible values are: 0 - (default) enabled; 1 - disabled.

相関オペレーション

相関操作オブジェクトは、相関が実行されたときに実行される操作を定義します。以下のプロパティを持ちます。

プロパティ	Type	説明
type (required)	integer	Type of operation. Possible values: 0 - close old events; 1 - close new event.

相関フィルタ

相関フィルタオブジェクトは、設定された相関処理を実行するために満たす必要のある一連の条件を定義します。
 このオブジェクトには、次のプロパティがあります。

プロパティ	Type	説明
evaltype (required)	integer	Filter condition evaluation method. Possible values: 0 - and/or; 1 - and; 2 - or; 3 - custom expression.
conditions (required)	array	Set of filter conditions to use for filtering results.
eval_formula	string	(readonly) Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its <code>formulaid</code> . The value of <code>eval_formula</code> is equal to the value of <code>formula</code> for filters with a custom expression.
formula	string	User-defined expression to be used for evaluating conditions of filters with a custom expression. The expression must contain IDs that reference specific filter conditions by its <code>formulaid</code> . The IDs used in the expression must exactly match the ones defined in the filter conditions: no condition can remain unused or omitted. Required for custom expression filters.

相関フィルター条件

相関フィルタ条件オブジェクトは、相関処理を実行する前に確認する必要のある特定の条件を定義します。

プロパティ	Type	説明
type (required)	integer	Type of condition. Possible values: 0 - old event tag; 1 - new event tag; 2 - new event host group; 3 - event tag pair; 4 - old event tag value; 5 - new event tag value.
tag	string	Event tag (old or new). Required when type of condition is: 0, 1, 4, 5.
groupid	string	Host group ID. Required when type of condition is: 2.
oldtag	string	Old event tag. Required when type of condition is: 3.
newtag	string	Old event tag. Required when type of condition is: 3.
value	string	Event tag (old or new) value. Required when type of condition is: 4, 5.
formulaid	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward.
operator	integer	Condition operator. Required when type of condition is: 2, 4, 5.

Note:

様々なタイプの式でフィルタを使用する方法を理解するために
 `correlation.get` と `correlation.create` メソッドの例を参照してください。

各条件タイプでサポートされる演算子および値は以下の通りです。

条件	条件名	サポートされる演算子	想定する値
2	Host group	=, <>	Host group ID.
4	Old event tag value	=, <>, like, not like	string
5	New event tag value	=, <>, like, not like	string

correlation.create

説明

object correlation.create(object/array correlations)

この方法によって、新しい相関関係を作り出すことができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
 ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 作成する相関関係

standard correlation propertiesに加えて、このメソッドは以下のパラメータを受け取ることができる。

	パラメータ	Type
operations (required)	array	Correlation operations to create for the correlation.
filter (required)	object	Correlation filter object for the correlation.

戻り値

(object) 作成された相関の ID を含むオブジェクトを返す。返される ID の順序は、渡された相関関係の順序と一致する。

例

イベントタグの相関関係を新規に作成する

AND/OR'の評価方法を用いて、1つの条件と1つの操作で相関を作成します。デフォルトでは、相関は有効になっています。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "correlation.create",
  "params": {
    "name": "new event tag correlation",
    "filter": {
      "evaltype": 0,
      "conditions": [
        {
          "type": 1,
          "tag": "ok"
        }
      ]
    },
    "operations": [
      {
        "type": 0
      }
    ]
  },
  "auth": "343baad4f88b4106b9b5961e77437688",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "correlationids": [
      "1"
    ]
  },
  "id": 1
}
```

カスタム式フィルタの使用

カスタム・フィルター条件を使用する相関を作成します。このとき式 ID "A" または "B" は任意に選択されています。
条件タイプは演算子 "<>" で "ホストグループ" とします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "correlation.create",
  "params": {
    "name": "new host group correlation",
    "description": "a custom description",
    "status": 0,
    "filter": {
      "evaltype": 3,
      "formula": "A or B",
      "conditions": [
        {
          "type": 2,
          "operator": 1,
          "formulaid": "A"
        },
        {
          "type": 2,
          "operator": 1,
          "formulaid": "B"
        }
      ]
    },
    "operations": [
      {
        "type": 1
      }
    ]
  },
  "auth": "343baad4f88b4106b9b5961e77437688",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "correlationids": [
      "2"
    ]
  },
  "id": 1
}
```

参照

- [Correlation filter](#)

- **Correlation operation**

ソース

CCorrelation::create() in ui/include/classes/api/services/CCorrelation.php.

correlation.delete

説明

object correlation.delete(array correlationids)

この方法では、相関関係を削除することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
 ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(array) 削除する相関関係の ID 配列

戻り値

(object) correlationids プロパティの下にある削除された相関の ID を含むオブジェクトを返します。

例

複数の関連付けを削除する

2つの相関を削除します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "correlation.delete",
  "params": [
    "1",
    "2"
  ],
  "auth": "343baad4f88b4106b9b5961e77437688",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "correlationids": [
      "1",
      "2"
    ]
  },
  "id": 1
}
```

ソース

CCorrelation::delete() in ui/include/classes/api/services/CCorrelation.php.

correlation.get

説明

integer/array correlation.get(object parameters)

この方法では、与えられたパラメータに応じた相関関係を検索することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
correlationids	string/array	Return only correlations with the given IDs.
selectFilter	query	Return a filter property with the correlation conditions.
selectOperations	query	Return an operations property with the correlation operations.
sortfield	string/array	Sort the result by the given properties.
		Possible values are: correlationid , name and status .
countOutput	boolean	These parameters being common for all get methods are described in the reference commentary .
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

相関関係を取得する

設定されたすべての相関を、相関条件とオペレーションとともに取得します。このフィルターは "and/or" 評価タイプを
 使用するため、`formula` プロパティは空で、`eval_formula` が自動的に生成されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "correlation.get",
  "params": {
    "output": "extend",
    "selectOperations": "extend",
    "selectFilter": "extend"
  },
  "auth": "343baad4f88b4106b9b5961e77437688",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "correlationid": "1",
```

```

    "name": "Correlation 1",
    "description": "",
    "status": "0",
    "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [
            {
                "type": "3",
                "oldtag": "error",
                "newtag": "ok",
                "formulaid": "A"
            }
        ],
        "eval_formula": "A"
    },
    "operations": [
        {
            "type": "0"
        }
    ]
  },
  "id": 1
}

```

参照

- [Correlation filter](#)
- [Correlation operation](#)

ソース

CCorrelation::get() in ui/include/classes/api/services/CCorrelation.php.

correlation.update

説明

object correlation.update(object/array correlations)

この方法によって、既存の相関関係を更新することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
 ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 更新される相関関係プロパティ

各相関には correlationid プロパティを定義する必要があり、その他のプロパティは任意です。渡されたプロパティのみが
 更新され、他のプロパティは変更されません。

[standard correlation properties](#)に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	Correlation filter object to replace the current filter.
operations	array	Correlation operations to replace existing operations.

戻り値

(object) correlationids' プロパティの下で更新された相関の ID を含むオブジェクトを返す。

例

相関を無効にする

Request:

```
{
  "jsonrpc": "2.0",
  "method": "correlation.update",
  "params": {
    "correlationid": "1",
    "status": "1"
  },
  "auth": "343baad4f88b4106b9b5961e77437688",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "correlationids": [
      "1"
    ]
  },
  "id": 1
}
```

条件を置き換えるが、評価方法は維持する

Request:

```
{
  "jsonrpc": "2.0",
  "method": "correlation.update",
  "params": {
    "correlationid": "1",
    "filter": {
      "conditions": [
        {
          "type": 3,
          "oldtag": "error",
          "newtag": "ok"
        }
      ]
    }
  },
  "auth": "343baad4f88b4106b9b5961e77437688",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "correlationids": [
      "1"
    ]
  },
  "id": 1
}
```

参照

- [Correlation filter](#)
- [Correlation operation](#)

ソース

Dashboard

このクラスは、ダッシュボードを扱うためのものです。

オブジェクトリファレンス:

- [Dashboard](#)
- [Dashboard page](#)
- [Dashboard widget](#)
- [Dashboard widget field](#)
- [Dashboard user](#)
- [Dashboard user group](#)

利用可能なメソッド:

- [dashboard.create](#) - 新しいダッシュボードの作成
- [dashboard.delete](#) - ダッシュボードの削除
- [dashboard.get](#) - ダッシュボードの取得
- [dashboard.update](#) - ダッシュボードの更新

> Dashboard object

以下のオブジェクトは dashboard API に直接関連するものです。

Dashboard

dashboard オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
dashboardid	string	(readonly) ID of the dashboard.
name (required)	string	Name of the dashboard.
userid	string	Dashboard owner user ID.
private	integer	Type of dashboard sharing. Possible values: 0 - public dashboard; 1 - (default) private dashboard.
display_period	integer	Default page display period (in seconds). Possible values: 10, 30, 60, 120, 600, 1800, 3600.
auto_start	integer	Default: 30. Auto start slideshow. Possible values: 0 - do not auto start slideshow; 1 - (default) auto start slideshow.

Dashboard page

Dashboard page オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
dashboard_pageid	string	(readonly) ID of the dashboard page.
name	string	Dashboard page name. Default: empty string.

プロパティ	Type	説明
display_period	integer	Dashboard page display period (in seconds). Possible values: 0, 10, 30, 60, 120, 600, 1800, 3600. Default: 0 (will use the default page display period).
widgets	array	Array of the dashboard widget objects.

Dashboard widget

Dashboard widget オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
widgetid	string	(readonly) ID of the dashboard widget.
type (required)	string	Type of the dashboard widget. Possible values: actionlog - Action log; clock - Clock; dataover - Data overview; discovery - Discovery status; favgraphs - Favorite graphs; favmaps - Favorite maps; graph - Graph (classic); graphprototype - Graph prototype; hostavail - Host availability; item - Item value; map - Map; navtree - Map Navigation Tree; plaintext - Plain text; problemhosts - Problem hosts; problems - Problems; problemsbysv - Problems by severity; slareport - SLA report; svggraph - Graph; systeminfo - System information; tophosts - Top hosts; trigover - Trigger overview; url - URL; web - Web monitoring.
name	string	Custom widget name.
x	integer	A horizontal position from the left side of the dashboard. Valid values range from 0 to 23.
y	integer	A vertical position from the top of the dashboard. Valid values range from 0 to 62.
width	integer	The widget width. Valid values range from 1 to 24.
height	integer	The widget height. Valid values range from 2 to 32.
view_mode	integer	The widget view mode. Possible values: 0 - (default) default widget view; 1 - with hidden header;
fields	array	Array of the dashboard widget field objects.

Dashboard widget field

Dashboard widget field オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
type (required)	integer	Type of the widget field. Possible values: 0 - Integer; 1 - String; 2 - Host group; 3 - Host; 4 - Item; 5 - Item prototype; 6 - Graph; 7 - Graph prototype; 8 - Map; 9 - Service; 10 - SLA.
name	string	Widget field name.
value (required)	mixed	Widget field value depending of type.

Dashboard user group

ユーザーグループに基づくダッシュボードの権限一覧です。以下のプロパティを持ちます。

Property	Type	Description
usrgrpId (required)	string	User group ID.
permission (required)	integer	Type of permission level. Possible values: 2 - read only; 3 - read-write;

Dashboard user

ユーザーに基づくダッシュボードの権限一覧です。以下のプロパティを持ちます。

Property	Type	Description
userid (required)	string	User ID.
permission (required)	integer	Type of permission level. Possible values: 2 - read only; 3 - read-write;

Dashboard widget fields

This page contains navigation links for dashboard widget parameters and possible property values for the respective **dashboard widget field** objects.

To see the parameters and property values for each widget, go to individual widget pages for:

- [Action log](#)
- [Clock](#)
- [Discovery status](#)
- [Favorite graphs](#)
- [Favorite maps](#)
- [Geomap](#)

- Graph
- Graph (classic)
- Graph prototype
- Host availability
- Item value
- Map
- Map navigation tree
- Plain text
- Problem hosts
- Problems
- SLA report
- System information
- Problems by severity
- Top hosts
- Trigger overview
- URL
- Web monitoring

Deprecated widgets:

- Data overview

Attention:

Deprecated widgets will be removed in the upcoming major release.

1 Action log

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Action log** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Action log widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Sort entries by	0	sort_triggers	3 - Time (ascending); 4 - (default) Time (descending); 5 - Type (ascending); 6 - Type (descending); 7 - Status (ascending); 8 - Status (descending); 11 - Recipient (ascending); 12 - Recipient (descending).
Show lines	0	show_lines	Valid values range from 1-100. Default: 25.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Action log widget. For more information on configuring a dashboard, see `dashboard.create`.

Configuring an Action log widget

Configure an Action log widget that displays 10 entries of action operation details, sorted by time (in ascending order).

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "actionlog",
            "name": "Action log",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 0,
                "name": "show_lines",
                "value": 10
              },
              {
                "type": 0,
                "name": "sort_triggers",
                "value": 3
              }
            ]
          }
        ]
      }
    ]
  },
  "userGroups": [
    {
      "usrgrpid": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}
```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

2 Clock

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Clock** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Clock widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.
Time type	0	time_type	0 - (default) Local time; 1 - Server time; 2 - Host time.

The following parameters are supported if Time type is set to "Host time".

Parameter	type	name	value
Item (required)	4	itemid	Item ID.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Clock widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Clock widget

Configure a Clock widget that displays Zabbix server time.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "clock",
            "name": "Clock",
            "x": 0,
            "y": 0,
            "width": 4,
            "height": 3,
```

```

        "view_mode": 0,
        "fields": [
            {
                "type": 0,
                "name": "time_type",
                "value": 1
            }
        ]
    },
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

3 データの概要

Attention:

This widget is deprecated and will be removed in the upcoming major release.

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Data overview** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Data overview widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.
Tags (the number in the the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)			
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value. Parameter Tag name required if configuring Tags.
Operator	0	tags.operator.0	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value. Parameter Tag value required if configuring Tags.

Parameter	type	name	value
Show suppressed problems	0	show_suppressed	0 - (default) Disabled; 1 - Enabled.
Hosts location	0	style	0 - (default) Left; 1 - Top.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Data overview widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Data overview widget

Configure a Data overview widget that displays data for host "10084" and only for items for which the tag with the name "component" contains value "cpu". In addition, display the data with hosts located on top.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "dataover",
            "name": "Data overview",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 3,
                "name": "hostids",
                "value": 10084
              },
              {
                "type": 1,
                "name": "tags.tag.0",
                "value": "component"
              },
              {
                "type": 0,
                "name": "tags.operator.0",
                "value": 0
              },
              {
                "type": 1,
                "name": "tags.value.0",
                "value": "cpu"
              },
              {
                "type": 0,
```

```

        "name": "style",
        "value": 1
      }
    ]
  },
  "userGroups": [
    {
      "usrgrpid": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

4 Discovery status

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Discovery status** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Discovery status widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Discovery status widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring Discovery status widget

Configure a Discovery status widget with the refresh interval set to 15 minutes.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "discovery",
            "name": "Discovery status",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "view_mode": 0,
            "fields": [
              {
                "type": 0,
                "name": "rf_rate",
                "value": 900
              }
            ]
          }
        ]
      }
    ]
  },
  "userGroups": [
    {
      "usrgrpid": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
}
```

```
"id": 1
}
```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

5 Favorite graphs

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Favorite graphs** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Favorite graphs widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Favorite graphs widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Favorite graphs widget

Configure a Favorite graphs widget with the refresh interval set to 10 minutes.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "favgraphs",
            "name": "Favorite graphs",
            "x": 0,
            "y": 0,
            "width": 4,
            "height": 3,
            "view_mode": 0,
            "fields": [
              {
                "type": 0,
                "name": "rf_rate",
                "value": 600
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "userGroups": [
      {
        "usrgrp": 7,
        "permission": 2
      }
    ],
    "users": [
      {
        "userid": 1,
        "permission": 3
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

6 Favorite maps

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Favorite maps** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Favorite maps widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Favorite maps widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Favorite maps widget

Configure a Favorite maps widget with the refresh interval set to 10 minutes.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "favmaps",
            "name": "Favorite maps",
            "x": 0,
            "y": 0,
            "width": 4,
            "height": 3,
            "view_mode": 0,
            "fields": [
              {
                "type": 0,
                "name": "rf_rate",
                "value": 600
              }
            ]
          }
        ]
      }
    ]
  },
  "userGroups": [
    {
      "usrgrpid": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}
```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

7 Geomap

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the [Geomap](#) widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Geomap widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.

Tags (the number in the the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)

Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value.

Parameter Tag name required if configuring Tags.

Parameter	type	name	value
Operator	0	tags.operator.0	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value.
Initial view	1	default_view	Parameter Tag value required if configuring Tags. Comma separated latitude, longitude, zoom level (optional, valid values range from 0-30). Example: 40.6892494,-74.0466891,10.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Geomap widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Geomap widget

Configure a Geomap widget that displays hosts from host groups "2" and "22" based on the following tag configuration: tag with the name "component" contains value "node", or tag with the name "location" equals value "New York". In addition, set the map initial view to coordinates "40.6892494" (latitude), "-74.0466891" (longitude) with the zoom level "10".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "geomap",
            "name": "Geomap",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 2,
                "name": "groupids",
                "value": 22
              },
              {
                "type": 2,
                "name": "groupids",
                "value": 2
              },
              {
                "type": 1,
                "name": "default_view",
                "value": "40.6892494,-74.0466891,10"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

        "type": 0,
        "name": "evaltype",
        "value": 2
      },
      {
        "type": 1,
        "name": "tags.tag.0",
        "value": "component"
      },
      {
        "type": 0,
        "name": "tags.operator.0",
        "value": 0
      },
      {
        "type": 1,
        "name": "tags.value.0",
        "value": "node"
      },
      {
        "type": 1,
        "name": "tags.tag.1",
        "value": "location"
      },
      {
        "type": 0,
        "name": "tags.operator.1",
        "value": 1
      },
      {
        "type": 1,
        "name": "tags.value.1",
        "value": "New York"
      }
    ]
  },
  ],
  "userGroups": [
    {
      "usrgrpId": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
],
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  }
}

```

```

    ],
    },
    "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

8 Graph

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Graph** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Graph widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.

Data set

The following parameters are supported for configuring a Data set.

Note:

The first number in the property name (e.g. `ds.hosts.0.0`, `ds.items.0.0`) represents the particular data set, while the second number, if present, represents the configured host or item.

Parameter	type	name	value
Host pattern (required)	1	ds.hosts.0.0	Host name or pattern (e.g. <code>Zabbix*</code>).
Item pattern (required)	1	ds.items.0.0	Item name or pattern (e.g. <code>*: Number of processed *values per second</code>).
Color	1	ds.color.0	Hexadecimal color code (e.g. <code>FF0000</code>).
Draw	0	ds.type.0	Default: <code>FF465C</code> . 0 - (default) Line; 1 - Points; 2 - Staircase; 3 - Bar.
Width	0	ds.width.0	Valid values range from 1-10. Default: 1. Parameter Width not available if Draw is set to "Points" or "Bar".

Parameter	type	name	value
Point size	0	ds.pointsize.0	Valid values range from 1-10. Default: 3. Parameter Point size not available if Draw is set to "Line", "Staircase" or "Bar".
Transparency	0	ds.transparency.0	Valid values range from 1-10. Default: 5.
Fill	0	ds.fill.0	Valid values range from 1-10. Default: 3. Parameter Fill not available if Draw is set to "Points" or "Bar".
Missing data	0	ds.missingdatafunc.0	0 - (default) None; 1 - Connected; 2 - Treat as 0.
Y-axis	0	ds.axisy.0	Parameter Missing data not available if Draw is set to "Points" or "Bar". 0 - (default) Left; 1 - Right.
Time shift	1	ds.timeshift.0	Valid time string (e.g. 3600, 1h, etc.). You may use time suffixes . Negative values are also allowed. Default: "" (empty).
Aggregation function	0	ds.aggregate_function.0	0 - (default) none; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Aggregation interval	1	ds.aggregate_interval.0	Valid time string (e.g. 3600, 1h, etc.). You may use time suffixes . Default: 1h.
Aggregate	0	ds.aggregate_grouping.0	0 (default) Each item; 1 - Data set. Parameter Aggregate not available if Aggregation function is set to "none".

Display options

The following parameters are supported for configuring Display options.

Parameter	type	name	value
History data selection	0	source	0 - (default) Auto; 1 - History; 2 - Trends.

Time period

The following parameters are supported for configuring Time period.

Parameter	type	name	value
Set custom time period	0	graph_time	0 - (default) Disabled; 1 - Enabled.

Parameter	type	name	value
From	1	time_from	Valid time string in format YYYY-MM-DD hh:mm:ss. Relative time period values (now, now/d, now/w-1w, etc.) are also supported.
To	1	time_to	Default: now-1h. Valid time string value in format YYYY-MM-DD hh:mm:ss. Relative time period values (now, now/d, now/w-1w, etc.) are also supported. Default: now.

Axes

The following parameters are supported for configuring Axes.

Parameter	type	name	value
Left Y	0	lefty	0 - Disabled; 1 - (default) Enabled.
Right Y	0	righty	Parameter available if Y-axis (in Data set configuration) is set to "Left". 0 - (default) Disabled; 1 - Enabled.
Min	1	lefty_min	Parameter available if Y-axis (in Data set configuration) is set to "Right". Any numeric value. Default: "" (empty).
Max	1	righty_min lefty_max	Any numeric value. Default: "" (empty).
Units (type)	0	righty_max lefty_units	0 - (default) Auto; 1 - Static.
Units (value)	1	righty_units lefty_static_units	Any string value. Default: "" (empty).
X-Axis	0	righty_static_units xaxis	0 - Disabled; 1 - (default) Enabled.

Legend

The following parameters are supported for configuring Legend.

Parameter	type	name	value
Show legend	0	legend	0 - Disabled; 1 - (default) Enabled.
Number of rows	0	legend_lines	Valid values range from 1-5. Default: 1.

Problems

The following parameters are supported for configuring Problems.

Parameter	type	name	value
Show problems	0	show_problems	0 - (default) Disabled; 1 - Enabled.
Selected items only	0	graph_item_problems	0 - Disabled; 1 - (default) Enabled.
Problem hosts	1	problemhosts.0	Host name. Note: The number in the property name references the configured host. To configure multiple hosts, create a dashboard widget field object for each host.
Severity	0	severities	0 - Not classified; 1 - Information; 2 - Warning; 3 - Average; 4 - High; 5 - Disaster. Default: 1, 2, 3, 4, 5 (all enabled). Note: To configure multiple values, create a dashboard widget field object for each value.
Problem Tags (the number in the the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)	1	problem_name	Problem event name (case insensitive, full name or part of it).
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value.
Operator	0	tags.operator.0	Parameter Tag name required if configuring Tags. 0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist. Parameter Operator required if configuring Tags.

Parameter	type	name	value
Tag value	1	tags.value.0	Any string value.

Parameter Tag value required if configuring Tags.

Overrides

The following parameters are supported for configuring Overrides.

Note:

The first number in the property name (e.g. or.hosts.0.0, or.items.0.0) represents the particular data set, while the second number, if present, represents the configured host or item.

Parameter	type	name	value
Host pattern (required)	1	or.hosts.0.0	Host name or pattern (e.g. Zabbix*).
Item pattern (required)	1	or.items.0.0	Item name or pattern (e.g. *: Number of processed *values per second).
Base color	1	or.color.0	Hexadecimal color code (e.g. FF0000).
Width	0	or.width.0	Valid values range from 1-10.
Draw	0	or.type.0	0 - Line; 1 - Points; 2 - Staircase; 3 - Bar.
Transparency	0	or.transparency.0	Valid values range from 1-10.
Fill	0	or.fill.0	Valid values range from 1-10.
Point size	0	or.pointsize.0	Valid values range from 1-10.
Missing data	0	or.missingdatafunc.0	0 - None; 1 - Connected; 2 - Treat as 0.
Y-axis	0	or.axisy.0	0 - Left; 1 - Right.
Time shift	1	or.timeshift.0	Valid time string (e.g. 3600, 1h, etc.). You may use time suffixes . Negative values are allowed.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Graph widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Graph widget

Configure a Graph widget in the following way:

- 4 data sets for a total of 9 items on 1 host.
- Each data set consists of a line that has a custom color, width, transparency, and fill.
- Data set 4 has a configured aggregation.
- Data in the graph are displayed for a time period of the last 3 hours.
- Problems in the graph are displayed only for the configured items.
- Graph has two Y axes of which the right Y axis displays values only for Data set 4.
- Graph legend displays configured items in 2 rows.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
```

```

"widgets": [
  {
    "type": "svggraph",
    "name": "Graph",
    "x": 0,
    "y": 0,
    "width": 12,
    "height": 5,
    "view_mode": 0,
    "fields": [
      {
        "type": 1,
        "name": "ds.hosts.0.0",
        "value": "Zabbix server"
      },
      {
        "type": 1,
        "name": "ds.items.0.0",
        "value": "Zabbix server: Utilization of poller data collector processes, i
      },
      {
        "type": 1,
        "name": "ds.color.0",
        "value": "FF0000"
      },
      {
        "type": 0,
        "name": "ds.width.0",
        "value": 3
      },
      {
        "type": 0,
        "name": "ds.transparency.0",
        "value": 3
      },
      {
        "type": 0,
        "name": "ds.fill.0",
        "value": 1
      },
      {
        "type": 1,
        "name": "ds.hosts.1.0",
        "value": "Zabbix server"
      },
      {
        "type": 1,
        "name": "ds.items.1.0",
        "value": "Zabbix server: Utilization of trapper data collector processes,
      },
      {
        "type": 1,
        "name": "ds.color.1",
        "value": "BF00FF"
      },
      {
        "type": 0,
        "name": "ds.width.1",
        "value": 3
      },
      {
        "type": 0,

```



```

        "name": "ds.transparency.1",
        "value": 3
    },
    {
        "type": 0,
        "name": "ds.fill.1",
        "value": 1
    },
    {
        "type": 1,
        "name": "ds.hosts.2.0",
        "value": "Zabbix server"
    },
    {
        "type": 1,
        "name": "ds.items.2.0",
        "value": "Zabbix server: Utilization of history syncer internal processes,"
    },
    {
        "type": 1,
        "name": "ds.color.2",
        "value": "0040FF"
    },
    {
        "type": 0,
        "name": "ds.width.2",
        "value": 3
    },
    {
        "type": 0,
        "name": "ds.transparency.2",
        "value": 3
    },
    {
        "type": 0,
        "name": "ds.fill.2",
        "value": 1
    },
    {
        "type": 1,
        "name": "ds.hosts.3.0",
        "value": "Zabbix server"
    },
    {
        "type": 1,
        "name": "ds.items.3.0",
        "value": "*: Number of processed *values per second"
    },
    {
        "type": 1,
        "name": "ds.color.3",
        "value": "000000"
    },
    {
        "type": 0,
        "name": "ds.transparency.3",
        "value": 0
    },
    {
        "type": 0,
        "name": "ds.fill.3",
        "value": 0
    }

```

```

    },
    {
      "type": 0,
      "name": "ds.axisy.3",
      "value": 1
    },
    {
      "type": 0,
      "name": "ds.aggregate_function.3",
      "value": 3
    },
    {
      "type": 1,
      "name": "ds.aggregate_interval.3",
      "value": "1m"
    },
    {
      "type": 0,
      "name": "ds.aggregate_grouping.3",
      "value": 1
    },
    {
      "type": 0,
      "name": "graph_time",
      "value": 1
    },
    {
      "type": 1,
      "name": "time_from",
      "value": "now-3h"
    },
    {
      "type": 0,
      "name": "legend_lines",
      "value": 2
    },
    {
      "type": 0,
      "name": "show_problems",
      "value": 1
    }
  }
]
}
],
"userGroups": [
  {
    "usrgrpId": 7,
    "permission": 2
  }
],
"users": [
  {
    "userid": 1,
    "permission": 3
  }
]
],
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}
```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

9 Graph (classic)

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Graph (classic)** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Graph (classic) widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Source	0	source_type	0 - (default) Graph; 1 - Simple graph.
Graph	6	graphid	Graph ID.
Item	4	itemid	Parameter Graph required if Source is set to "Graph". Item ID.
Show legend	0	show_legend	Parameter Item required if Source is set to "Simple graph". 0 - Disabled; 1 - (default) Enabled.
Dynamic item	0	dynamic	0 - (default) Disabled; 1 - Enabled.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Graph (classic) widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Graph (classic) widget

Configure a Graph (classic) widget that displays a simple graph for the item "42269".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
```

```

"name": "My dashboard",
"display_period": 30,
"auto_start": 1,
"pages": [
  {
    "widgets": [
      {
        "type": "graph",
        "name": "Graph (classic)",
        "x": 0,
        "y": 0,
        "width": 12,
        "height": 5,
        "view_mode": 0,
        "fields": [
          {
            "type": 0,
            "name": "source_type",
            "value": 1
          },
          {
            "type": 4,
            "name": "itemid",
            "value": 42269
          }
        ]
      }
    ]
  }
],
"userGroups": [
  {
    "usrgrpid": 7,
    "permission": 2
  }
],
"users": [
  {
    "userid": 1,
    "permission": 3
  }
],
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

10 Graph prototype

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Graph prototype** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Graph prototype widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Source	0	source_type	2 - (default) Graph prototype; 3 - Simple graph prototype.
Graph prototype	7	graphid	Graph prototype ID. Parameter Graph prototype required if Source is set to "Graph prototype".
Item prototype	5	itemid	Item prototype ID. Parameter Item prototype required if Source is set to "Simple graph prototype".
Show legend	0	show_legend	0 - Disabled; 1 - (default) Enabled.
Dynamic item	0	dynamic	0 - (default) Disabled; 1 - Enabled.
Columns	0	columns	Valid values range from 1-24. Default: 2.
Rows	0	rows	Valid values range from 1-16. Default: 1.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Graph prototype widget. For more information on configuring a dashboard, see `dashboard.create`.

Configuring a Graph prototype widget

Configure a Graph prototype widget that displays a grid of 3 graphs (3 columns, 1 row) created from an item prototype (ID: "42316") by low-level discovery.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "graphprototype",
```

```

        "name": "Graph prototype",
        "x": 0,
        "y": 0,
        "width": 16,
        "height": 5,
        "view_mode": 0,
        "fields": [
            {
                "type": 0,
                "name": "source_type",
                "value": 3
            },
            {
                "type": 5,
                "name": "itemid",
                "value": 42316
            },
            {
                "type": 0,
                "name": "columns",
                "value": 3
            }
        ]
    }
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
],
"users": [
    {
        "userid": 1,
        "permission": 3
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

11 Host availability

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Host availability** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Host availability widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.
Host groups	2	groupids	Host group ID . Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Interface type	0	interface_type	0 - None; 1 - Zabbix agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Default: 1, 2, 3, 4 (all enabled). Note: To configure multiple values, create a dashboard widget field object for each value.
Layout	0	layout	0 - (default) Horizontal; 1 - Vertical.
Show hosts in maintenance	0	maintenance	0 - (default) Disabled; 1 - Enabled.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Host availability widget. For more information on configuring a dashboard, see `dashboard.create`.

Configuring a Host availability widget

Configure a Host availability widget that displays availability information (in a vertical layout) for hosts in host group "4" with "Zabbix agent" and "SNMP" interfaces configured.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "hostavail",
            "name": "Host availability",
            "x": 0,
            "y": 0,
            "width": 6,
```

```

        "height": 3,
        "view_mode": 0,
        "fields": [
            {
                "type": 2,
                "name": "groupids",
                "value": 4
            },
            {
                "type": 0,
                "name": "interface_type",
                "value": 1
            },
            {
                "type": 0,
                "name": "interface_type",
                "value": 2
            },
            {
                "type": 0,
                "name": "layout",
                "value": 1
            }
        ]
    }
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
],
"users": [
    {
        "userid": 1,
        "permission": 3
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

12 Item value

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Item value** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Item value widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Item (required)	4	itemid	Item ID.
Show	0	show	1 - Description; 2 - Value; 3 - Time; 4 - Change indicator. Default: 1, 2, 3, 4 (all enabled). Note: To configure multiple values, create a dashboard widget field object for each value.
Dynamic item	0	dynamic	0 - (default) Disabled; 1 - Enabled.
Advanced configuration	0	adv_conf	0 - (default) Disabled; 1 - Enabled.

Advanced configuration

The following parameters are supported if Advanced configuration is set to "Enabled".

Parameter	type	name	value
Background color	1	bg_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).

Description

The following parameters are supported if Advanced configuration is set to "Enabled", and Show is set to "Description".

Parameter	type	name	value
Description	1	description	Any string value, including macros. Supported macros: {HOST.*}, {ITEM.*}, {INVENTORY.*}, User macros. Default: {ITEM.NAME}.
Horizontal position	0	desc_h_pos	0 - Left; 1 - (default) Center; 2 - Right. Two or more elements (Description, Value, Time) cannot share the same Horizontal position and Vertical position.

Parameter	type	name	value
Vertical position	0	desc_v_pos	0 - Top; 1 - Middle; 2 - (default) Bottom. Two or more elements (Description, Value, Time) cannot share the same Horizontal position and Vertical position.
Size	0	desc_size	Valid values range from 1-100. Default: 15.
Bold	0	desc_bold	0 - (default) Disabled; 1 - Enabled.
Color	1	desc_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).

Value

The following parameters are supported if Advanced configuration is set to "Enabled", and Show is set to "Value".

Parameter	type	name	value
Decimal places			
Decimal places	0	decimal_places	Valid values range from 1-10. Default: 2.
Size	0	decimal_size	Valid values range from 1-100. Default: 35.
Position			
Horizontal position	0	value_h_pos	0 - Left; 1 - (default) Center; 2 - Right. Two or more elements (Description, Value, Time) cannot share the same Horizontal position and Vertical position.
Vertical position	0	value_v_pos	0 - Top; 1 - (default) Middle; 2 - Bottom. Two or more elements (Description, Value, Time) cannot share the same Horizontal position and Vertical position.
Size	0	value_size	Valid values range from 1-100. Default: 45.
Bold	0	value_bold	0 - Disabled; 1 - (default) Enabled.
Color	1	value_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).
Units			
Units (checkbox)	0	units_show	0 - Disabled; 1 - (default) Enabled.
Units (value)	1	units	Any string value.
Position	0	units_pos	0 - Before value; 1 - Above value; 2 - (default) After value; 3 - Below value.
Size	0	units_size	Valid values range from 1-100. Default: 35.

Parameter	type	name	value
Bold	0	units_bold	0 - Disabled; 1 - (default) Enabled.
Color	1	units_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).

Time

The following parameters are supported if Advanced configuration is set to "Enabled", and Show is set to "Time".

Parameter	type	name	value
Horizontal position	0	time_h_pos	0 - Left; 1 - (default) Center; 2 - Right. Two or more elements (Description, Value, Time) cannot share the same Horizontal position and Vertical position.
Vertical position	0	time_v_pos	0 - (default) Top; 1 - Middle; 2 - Bottom. Two or more elements (Description, Value, Time) cannot share the same Horizontal position and Vertical position.
Size	0	time_size	Valid values range from 1-100. Default: 15.
Bold	0	time_bold	0 - (default) Disabled; 1 - Enabled.
Color	1	time_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).

Change indicator

The following parameters are supported if Advanced configuration is set to "Enabled", and Show is set to "Change indicator".

Parameter	type	name	value
Change indicator ↑ color	1	up_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).
Change indicator ↓ color	1	down_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).
Change indicator ⇅ color	1	updown_color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Item value widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring an Item value widget

Configure an Item value widget that displays the item value for the item "42266" (Zabbix agent availability). In addition, visually fine-tune the widget with multiple advanced options.

Request:

```

{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "item",
            "name": "Item value",
            "x": 0,
            "y": 0,
            "width": 4,
            "height": 3,
            "view_mode": 0,
            "fields": [
              {
                "type": 4,
                "name": "itemid",
                "value": 42266
              },
              {
                "type": 0,
                "name": "show",
                "value": 1
              },
              {
                "type": 0,
                "name": "show",
                "value": 2
              },
              {
                "type": 0,
                "name": "show",
                "value": 3
              },
              {
                "type": 0,
                "name": "adv_conf",
                "value": 1
              },
              {
                "type": 1,
                "name": "bg_color",
                "value": "D1C4E9"
              },
              {
                "type": 1,
                "name": "description",
                "value": "Agent status"
              },
              {
                "type": 0,
                "name": "desc_h_pos",
                "value": 0
              },
              {
                "type": 0,
                "name": "desc_v_pos",

```

```

        "value": 0
    },
    {
        "type": 0,
        "name": "desc_bold",
        "value": 1
    },
    {
        "type": 1,
        "name": "desc_color",
        "value": "F06291"
    },
    {
        "type": 0,
        "name": "value_h_pos",
        "value": 0
    },
    {
        "type": 0,
        "name": "value_size",
        "value": 25
    },
    {
        "type": 1,
        "name": "value_color",
        "value": "FFFF00"
    },
    {
        "type": 0,
        "name": "units_show",
        "value": 0
    },
    {
        "type": 0,
        "name": "time_h_pos",
        "value": 2
    },
    {
        "type": 0,
        "name": "time_v_pos",
        "value": 2
    },
    {
        "type": 0,
        "name": "time_size",
        "value": 10
    },
    {
        "type": 0,
        "name": "time_bold",
        "value": 1
    },
    {
        "type": 1,
        "name": "time_color",
        "value": "9FA8DA"
    }
}
]
}
],

```

```

    "userGroups": [
      {
        "usrgrpid": 7,
        "permission": 2
      }
    ],
    "users": [
      {
        "userid": 1,
        "permission": 3
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

13 Map

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Map** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Map widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.
Source type	0	source_type	1 - (default) Map; 2 - Map navigation tree.
Map	8	sysmapid	Map ID.
Linked widget reference	1	filter_widget_reference	Parameter Map required if Source type is set to "Map". Valid Map navigation tree widget parameter Reference value. Parameter Linked widget reference required if Source type is set to "Map navigation tree".

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Map widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Map widget

Configure a Map widget that displays the map "1".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "map",
            "name": "Map",
            "x": 0,
            "y": 0,
            "width": 18,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 8,
                "name": "sysmapid",
                "value": 1
              }
            ]
          }
        ]
      }
    ],
    "userGroups": [
      {
        "usrgrpid": 7,
        "permission": 2
      }
    ],
    "users": [
      {
        "userid": 1,
        "permission": 3
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
}
```

```
"id": 1
}
```

Configuring a linked Map widget

Configure a Map widget that is linked to a **Map navigation tree** widget.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "map",
            "name": "Map",
            "x": 0,
            "y": 5,
            "width": 18,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 0,
                "name": "source_type",
                "value": 2
              },
              {
                "type": 1,
                "name": "filter_widget_reference",
                "value": "ABCDE"
              }
            ]
          },
          {
            "type": "navtree",
            "name": "Map navigation tree",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 1,
                "name": "navtree.name.1",
                "value": "Element A"
              },
              {
                "type": 1,
                "name": "navtree.name.2",
                "value": "Element B"
              },
              {
                "type": 1,
                "name": "navtree.name.3",
                "value": "Element C"
              }
            ]
          }
        ]
      }
    ]
  }
}
```



```
{
  "type": 1,
  "name": "navtree.name.4",
  "value": "Element A1"
},
{
  "type": 1,
  "name": "navtree.name.5",
  "value": "Element A2"
},
{
  "type": 1,
  "name": "navtree.name.6",
  "value": "Element B1"
},
{
  "type": 1,
  "name": "navtree.name.7",
  "value": "Element B2"
},
{
  "type": 0,
  "name": "navtree.parent.4",
  "value": 1
},
{
  "type": 0,
  "name": "navtree.parent.5",
  "value": 1
},
{
  "type": 0,
  "name": "navtree.parent.6",
  "value": 2
},
{
  "type": 0,
  "name": "navtree.parent.7",
  "value": 2
},
{
  "type": 0,
  "name": "navtree.order.1",
  "value": 1
},
{
  "type": 0,
  "name": "navtree.order.2",
  "value": 2
},
{
  "type": 0,
  "name": "navtree.order.3",
  "value": 3
},
{
  "type": 0,
  "name": "navtree.order.4",
  "value": 1
},
{
  "type": 0,
```

```

        "name": "navtree.order.5",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.order.6",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.order.7",
        "value": 2
    },
    {
        "type": 8,
        "name": "navtree.sysmapid.6",
        "value": 1
    },
    {
        "type": 1,
        "name": "reference",
        "value": "ABCDE"
    }
}
]
}
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
],
"users": [
    {
        "userid": 1,
        "permission": 3
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)
- [Map navigation tree](#)

14 Map navigation tree

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Map navigation tree** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Map navigation tree widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.
Show unavailable maps	1	show_unavailable	0 - (default) Disabled; 1 - Enabled.
Reference	1	reference	Any string value consisting of 5 characters (e.g. ABCDE, JBPNL, etc.). Parameter Reference value is used in the Map widget (Linked widget reference) for linking with the Map navigation tree widget.

The following parameters are supported for configuring map navigation tree elements.

Parameter	type	name	value
Name	1	navtree.name.1	Any string value.
Linked map	8	navtree.sysmapid.1	Note: The number in the property name sets the element number. Map ID.
Parameters for creating element hierarchy	0	navtree.parent.1	Note: The number in the property name references the element to which the map is linked. Parent element number.
	0	navtree.order.1	Note: The number in the property name references the child element. The property value references the parent element. Element position in the map navigation tree. Note: The number in the property name references the element number. The property value references the element position in the map navigation tree. Parent element position is determined within the whole map navigation tree. Child element position is determined within the parent element.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Map navigation tree widget. For more information on configuring a dashboard, see `dashboard.create`.

Configuring a Map navigation tree widget

Configure a Map navigation tree widget that displays the following map navigation tree:

- Element A
 - Element A1
 - Element A2
- Element B
 - Element B1 (contains linked map "1" that can be displayed in a **linked Map widget**)
 - Element B2

- Element C

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "navtree",
            "name": "Map navigation tree",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 1,
                "name": "navtree.name.1",
                "value": "Element A"
              },
              {
                "type": 1,
                "name": "navtree.name.2",
                "value": "Element B"
              },
              {
                "type": 1,
                "name": "navtree.name.3",
                "value": "Element C"
              },
              {
                "type": 1,
                "name": "navtree.name.4",
                "value": "Element A1"
              },
              {
                "type": 1,
                "name": "navtree.name.5",
                "value": "Element A2"
              },
              {
                "type": 1,
                "name": "navtree.name.6",
                "value": "Element B1"
              },
              {
                "type": 1,
                "name": "navtree.name.7",
                "value": "Element B2"
              },
              {
                "type": 0,
                "name": "navtree.parent.4",
                "value": 1
              },
            ]
          },
        ]
      }
    ]
  }
}
```

```

        "type": 0,
        "name": "navtree.parent.5",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.parent.6",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.parent.7",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.order.1",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.order.2",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.order.3",
        "value": 3
    },
    {
        "type": 0,
        "name": "navtree.order.4",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.order.5",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.order.6",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.order.7",
        "value": 2
    },
    {
        "type": 8,
        "name": "navtree.sysmapid.6",
        "value": 1
    },
    {
        "type": 1,
        "name": "reference",
        "value": "ABCDE"
    }
}
]

```

```

    }
  ],
  "userGroups": [
    {
      "usrgrpid": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
],
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)
- [Map](#)

15 Plain text

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Plain text** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Plain text widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Items (required)	4	itemids	Item ID. Note: To configure multiple items, create a dashboard widget field object for each item.
Items location	0	style	0 - (default) Left; 1 - Top.

Parameter	type	name	value
Show lines	0	show_lines	Valid values range from 1-100.
Show text as HTML	0	show_as_html	Default: 25. 0 - (default) Disabled; 1 - Enabled.
Dynamic item	0	dynamic	0 - (default) Disabled; 1 - Enabled.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Plain text widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Plain text widget

Configure a Plain text widget that displays latest data for items "42269" and "42253". In addition, configure the item names to be located at the top of the data columns, and only 15 lines of data to be displayed.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "plaintext",
            "name": "Plain text",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "view_mode": 0,
            "fields": [
              {
                "type": 4,
                "name": "itemids",
                "value": 42269
              },
              {
                "type": 4,
                "name": "itemids",
                "value": 42253
              },
              {
                "type": 0,
                "name": "style",
                "value": 1
              },
              {
                "type": 0,
                "name": "show_lines",
                "value": 15
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "userGroups": [
      {
        "usrgrpid": 7,
        "permission": 2
      }
    ],
    "users": [
      {
        "userid": 1,
        "permission": 3
      }
    ]
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

16 Problem hosts

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Problem hosts** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Problem hosts widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Exclude host groups	2	exclude_groupids	Host group ID. Note: To exclude multiple host groups, create a dashboard widget field object for each host group.

Parameter	type	name	value
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.
Problem Severity	1 0	problem severities	Problem event name (case insensitive, full name or part of it). 0 - Not classified; 1 - Information; 2 - Warning; 3 - Average; 4 - High; 5 - Disaster. Default: 1, 2, 3, 4, 5 (all enabled). Note: To configure multiple values, create a dashboard widget field object for each value.
Tags (the number in the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)			
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value. Parameter Tag name required if configuring Tags.
Operator	0	tags.operator.0	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value. Parameter Tag value required if configuring Tags.

Parameter	type	name	value
Show suppressed problems	0	show_suppressed	0 - (default) Disabled; 1 - Enabled.
Hide groups without problems	0	hide_empty_groups	0 - (default) Disabled; 1 - Enabled.
Problem display	0	ext_ack	0 - (default) All; 1 - Unacknowledged only; 2 - Separated.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Problem hosts widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Problem hosts widget

Configure a Problem hosts widget that displays hosts from host groups "2" and "4" that have problems with a name that includes the string "CPU" and that have the following severities: "Warning", "Average", "High", "Disaster".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "problemhosts",
            "name": "Problem hosts",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 2,
                "name": "groupids",
                "value": 2
              },
              {
                "type": 2,
                "name": "groupids",
                "value": 4
              },
              {
                "type": 1,
                "name": "problem",
                "value": "cpu"
              },
              {
                "type": 0,
```

```

        "name": "severities",
        "value": 2
    },
    {
        "type": 0,
        "name": "severities",
        "value": 3
    },
    {
        "type": 0,
        "name": "severities",
        "value": 4
    },
    {
        "type": 0,
        "name": "severities",
        "value": 5
    }
    ]
}
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
],
"users": [
    {
        "userid": 1,
        "permission": 3
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

17 Problems

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Problems**

widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Problems widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Show	0	show	1 - (default) Recent problems; 2 - History; 3 - Problems.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Exclude host groups	2	exclude_groupids	Host group ID. Note: To exclude multiple host groups, create a dashboard widget field object for each host group.
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.
Problem Severity	1 0	problem severities	Problem event name (case insensitive, full name or part of it). 0 - Not classified; 1 - Information; 2 - Warning; 3 - Average; 4 - High; 5 - Disaster. Default: 1, 2, 3, 4, 5 (all enabled). Note: To configure multiple values, create a dashboard widget field object for each value.

Parameter	type	name	value
Tags (the number in the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)			
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value.
Operator	0	tags.operator.0	Parameter Tag name required if configuring Tags. 0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value.
Show tags	0	show_tags	Parameter Tag value required if configuring Tags. 0 - (default) None; 1 - 1; 2 - 2; 3 - 3.
Tag name (format)	0	tag_name_format	0 - (default) Full; 1 - Shortened; 2 - None.
Tag display priority	1	tag_priority	Parameter Tag name (format) not available if Show tags is set to "None". Comma-separated list of tags. Parameter Tag display priority not available if Show tags is set to "None".

Parameter	type	name	value
Show operational data	0	show_opdata	0 - (default) None; 1 - Separately; 2 - With problem name.
Show suppressed problems	0	show_suppressed	0 - (default) Disabled; 1 - Enabled.
Show unacknowledged only	0	unacknowledged	0 - (default) Disabled; 1 - Enabled.
Sort entries by	0	sort_triggers	1 - Severity (descending); 2 - Host (ascending); 3 - Time (ascending); 4 - (default) Time (descending); 13 - Severity (ascending); 14 - Host (descending); 15 - Problem (ascending); 16 - Problem (descending). For all values, except "Time (descending)" and "Time (ascending)", the parameter Show timeline must be set to "Disabled".
Show timeline	0	show_timeline	0 - Disabled; 1 - (default) Enabled. Parameter Show timeline available if Sort entries by is set to "Time (descending)" or "Time (ascending)".
Show lines	0	show_lines	Valid values range from 1-100. Default: 25.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Problems widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Problems widget

Configure a Problems widget that displays problems for host group "4" that satisfy the following conditions:

- Problems that have a tag with the name "scope" that contains values "performance" or "availability", or "capacity".
- Problems that have the following severities: "Warning", "Average", "High", "Disaster".

In addition, configure the widget to show tags and operational data.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
```

```

"type": "problems",
"name": "Problems",
"x": 0,
"y": 0,
"width": 12,
"height": 5,
"view_mode": 0,
"fields": [
  {
    "type": 2,
    "name": "groupids",
    "value": 4
  },
  {
    "type": 1,
    "name": "tags.tag.0",
    "value": "scope"
  },
  {
    "type": 0,
    "name": "tags.operator.0",
    "value": 0
  },
  {
    "type": 1,
    "name": "tags.value.0",
    "value": "performance"
  },
  {
    "type": 1,
    "name": "tags.tag.1",
    "value": "scope"
  },
  {
    "type": 0,
    "name": "tags.operator.1",
    "value": 0
  },
  {
    "type": 1,
    "name": "tags.value.1",
    "value": "availability"
  },
  {
    "type": 1,
    "name": "tags.tag.2",
    "value": "scope"
  },
  {
    "type": 0,
    "name": "tags.operator.2",
    "value": 0
  },
  {
    "type": 1,
    "name": "tags.value.2",
    "value": "capacity"
  },
  {
    "type": 0,
    "name": "severities",
    "value": 2
  }
]

```

```

    },
    {
      "type": 0,
      "name": "severities",
      "value": 3
    },
    {
      "type": 0,
      "name": "severities",
      "value": 4
    },
    {
      "type": 0,
      "name": "severities",
      "value": 5
    },
    {
      "type": 0,
      "name": "show_tags",
      "value": 1
    },
    {
      "type": 0,
      "name": "show_opdata",
      "value": 1
    }
  ]
}
],
"userGroups": [
  {
    "usrgrpid": 7,
    "permission": 2
  }
],
"users": [
  {
    "userid": 1,
    "permission": 3
  }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)

- `dashboard.update`

18 Problems by severity

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Problems by severity** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Problems by severity widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Exclude host groups	2	exclude_groupids	Host group ID. Note: To exclude multiple host groups, create a dashboard widget field object for each host group.
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.
Problem Severity	1 0	problem severities	Problem event name (case insensitive, full name or part of it). 0 - Not classified; 1 - Information; 2 - Warning; 3 - Average; 4 - High; 5 - Disaster. Default: 1, 2, 3, 4, 5 (all enabled). Note: To configure multiple values, create a dashboard widget field object for each value.

Parameter	type	name	value
Tags (the number in the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)			
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value.
Operator	0	tags.operator.0	Parameter Tag name required if configuring Tags. 0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value.
Show	0	show_type	Parameter Tag value required if configuring Tags. 0 - (default) Host groups; 1 - Totals.
Layout	0	layout	0 - (default) Horizontal; 1 - Vertical.
Show operational data	0	show_opdata	Parameter Layout not available if Show is set to "Host groups". 0 - (default) None; 1 - Separately; 2 - With problem name.
Show suppressed problems	0	show_suppressed	0 - (default) Disabled; 1 - Enabled.

Parameter	type	name	value
Hide groups without problems	0	hide_empty_groups	0 - (default) Disabled; 1 - Enabled. Parameter Hide groups without problems not available if Show is set to "Totals".
Problem display	0	ext_ack	0 - (default) All; 1 - Unacknowledged only; 2 - Separated.
Show timeline	0	show_timeline	0 - Disabled; 1 - (default) Enabled.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Problems by severity widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Problems by severity widget

Configure a Problems by severity widget that displays problem totals for all host groups.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "problemsbysv",
            "name": "Problems by severity",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 0,
                "name": "show_type",
                "value": 1
              }
            ]
          }
        ]
      }
    ]
  },
  "userGroups": [
    {
      "usrgrpId": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
}
```

```

    }
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

19 SLA report

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **SLA report** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the SLA report widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - (default) No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
SLA (required)	10	slaid	SLA ID.
Service	9	serviceid	Service ID.
Show periods	0	show_periods	Valid values range from 1-100.
From	1	date_from	Default: 20. Valid date string in format YYYY-MM-DD. Relative dates with modifiers d, w, M, y (e.g. now, now/d, now/w-1w, etc.) are supported.
To	1	date_to	Valid date string in format YYYY-MM-DD. Relative dates with modifiers d, w, M, y (e.g. now, now/d, now/w-1w, etc.) are supported.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the SLA report widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring an SLA report widget

Configure an SLA report widget that displays the SLA report for SLA "4" service "2" for the period of last 30 days.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "slareport",
            "name": "SLA report",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 10,
                "name": "slaid",
                "value": 4
              },
              {
                "type": 9,
                "name": "serviceid",
                "value": 2
              },
              {
                "type": 1,
                "name": "date_from",
                "value": "now-30d"
              },
              {
                "type": 1,
                "name": "date_to",
                "value": "now"
              }
            ]
          }
        ]
      }
    ]
  },
  "userGroups": [
    {
      "usrgrpId": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}
```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

20 System information

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **System Information** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the System Information widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - (default) 15 minutes.
Show	0	info_type	0 - (default) System stats; 1 - High availability nodes.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the System information widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a System information widget

Configure a System information widget that displays system stats with a refresh interval of 10 minutes.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "systeminfo",
            "name": "System information",
            "x": 0,

```

```

        "y": 0,
        "width": 12,
        "height": 5,
        "view_mode": 0,
        "fields": [
            {
                "type": 0,
                "name": "rf_rate",
                "value": 600
            }
        ]
    },
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

21 Top hosts

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Top Hosts** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Top Hosts widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.
Host Tags (the number in the the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)			
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value. Parameter Tag name required if configuring Tags.
Operator	0	tags.operator.0	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value. Parameter Tag value required if configuring Tags.

Parameter	type	name	value
Columns (see below)			
Order	0	order	2 - (default) Top N; 3 - Bottom N.
Order column	0	column	Column numeric value from the configured columns.
Host count	0	count	Valid values range from 1-100. Default: 10.

Columns

Columns have common parameters and additional parameters depending on the configuration of the parameter Data.

Note:

For all parameters related to columns the number in the property name (e.g. columns.name.0) references a column for which the parameter is configured.

The following parameters are supported for all columns.

Parameter	type	name	value
Name	1	columns.name.0	Any string value.
Data (required)	0	columns.data.0	1 - Item value; 2 - Host name; 3 - Text.
Base color (required)	1	columns.base_color.0	Hexadecimal color code (e.g. FF0000).

Item value

The following parameters are supported if Data is set to "Item value".

Note:

The first number in the Thresholds property name (e.g. columnsthresholds.color.0.0) references the column for which thresholds are configured, while the second number references threshold place in a list, sorted in ascending order. However, if thresholds are configured in a different order, the values will be sorted in ascending order after updating widget configuration in Zabbix frontend (e.g. "threshold.threshold.0": "5" → "threshold.threshold.0": "1"; "threshold.threshold.1": "1" → "threshold.threshold.1": "5").

Parameter	type	name	value
Item	1	columns.item.0	Valid item name.
Time shift (required)	1	columns.timeshift.0	Valid numeric or time string value (e.g. 3600 or 1h). You may use time suffixes . Negative values are allowed.
Aggregation function	0	columns.aggregate_function.0	(default) none; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.

Parameter	type	name	value
Aggregation interval	1	columns.aggregate_interval	Any string (e.g. 3600, 1h, etc.). You may use time suffixes . Parameter Aggregation interval not available if Aggregation function is set to none.
Display	0	columns.display	Default: 1h. 1 - (default) As is; 2 - Bar; 3 - Indicators.
Min	1	columns.min	Any numeric value.
Max	1	columns.max	Parameter Min not available if Display is set to "As is". Any numeric value.
History data	0	columns.history	Parameter Max not available if Display is set to "As is". 1 - (default) Auto; 2 - History; 3 - Trends.
Thresholds Color	1	columnsthresholds.color	Hexadecimal color code (e.g. FF0000). Default: "" (empty).
Threshold	1	columnsthresholds.threshold	Any numeric value.

Text

The following parameters are supported if Data is set to "Text".

Parameter	type	name	value
Text	1	columns.text	Any string value, including macros. Supported macros: {HOST.*}, {INVENTORY.*}. Parameter Text required if Data is set to "Text".

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Top hosts widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Top hosts widget

Configure a Top hosts widget that displays top hosts by CPU utilization in host group "4". In addition, configure the following custom columns: "Host name", "Utilization", "1m avg", "5m avg", "15m avg", "Processes".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "tophosts",
            "name": "Top hosts",
            "x": 0,
            "y": 0,

```

```

"width": 12,
"height": 5,
"view_mode": 0,
"fields": [
  {
    "type": 2,
    "name": "groupids",
    "value": 4
  },
  {
    "type": 1,
    "name": "columns.name.0",
    "value": ""
  },
  {
    "type": 0,
    "name": "columns.data.0",
    "value": 2
  },
  {
    "type": 1,
    "name": "columns.base_color.0",
    "value": "FFFFFF"
  },
  {
    "type": 1,
    "name": "columns.timeshift.0",
    "value": ""
  },
  {
    "type": 1,
    "name": "columns.item.0",
    "value": "System name"
  },
  {
    "type": 1,
    "name": "columns.name.1",
    "value": "Utilization"
  },
  {
    "type": 0,
    "name": "columns.data.1",
    "value": 1
  },
  {
    "type": 1,
    "name": "columns.base_color.1",
    "value": "4CAF50"
  },
  {
    "type": 1,
    "name": "columns.timeshift.1",
    "value": ""
  },
  {
    "type": 1,
    "name": "columns.item.1",
    "value": "CPU utilization"
  },
  {
    "type": 0,
    "name": "columns.display.1",

```

```

    "value": 3
  },
  {
    "type": 1,
    "name": "columns.min.1",
    "value": "0"
  },
  {
    "type": 1,
    "name": "columns.max.1",
    "value": "100"
  },
  {
    "type": 1,
    "name": "columnsthresholds.color.1.0",
    "value": "FFFF00"
  },
  {
    "type": 1,
    "name": "columnsthresholds.threshold.1.0",
    "value": "50"
  },
  {
    "type": 1,
    "name": "columnsthresholds.color.1.1",
    "value": "FF8000"
  },
  {
    "type": 1,
    "name": "columnsthresholds.threshold.1.1",
    "value": "80"
  },
  {
    "type": 1,
    "name": "columnsthresholds.color.1.2",
    "value": "FF4000"
  },
  {
    "type": 1,
    "name": "columnsthresholds.threshold.1.2",
    "value": "90"
  },
  {
    "type": 1,
    "name": "columns.name.2",
    "value": "1m avg"
  },
  {
    "type": 0,
    "name": "columns.data.2",
    "value": 1
  },
  {
    "type": 1,
    "name": "columns.base_color.2",
    "value": "FFFFFF"
  },
  {
    "type": 1,
    "name": "columns.timeshift.2",
    "value": ""
  },
},

```

```

{
  "type": 1,
  "name": "columns.item.2",
  "value": "Load average (1m avg)"
},
{
  "type": 1,
  "name": "columns.name.3",
  "value": "5m avg"
},
{
  "type": 0,
  "name": "columns.data.3",
  "value": 1
},
{
  "type": 1,
  "name": "columns.base_color.3",
  "value": "FFFFFF"
},
{
  "type": 1,
  "name": "columns.timeshift.3",
  "value": ""
},
{
  "type": 1,
  "name": "columns.item.3",
  "value": "Load average (5m avg)"
},
{
  "type": 1,
  "name": "columns.name.4",
  "value": "15m avg"
},
{
  "type": 0,
  "name": "columns.data.4",
  "value": 1
},
{
  "type": 1,
  "name": "columns.base_color.4",
  "value": "FFFFFF"
},
{
  "type": 1,
  "name": "columns.timeshift.4",
  "value": ""
},
{
  "type": 1,
  "name": "columns.item.4",
  "value": "Load average (15m avg)"
},
{
  "type": 1,
  "name": "columns.name.5",
  "value": "Processes"
},
{
  "type": 0,

```

```

        "name": "columns.data.5",
        "value": 1
    },
    {
        "type": 1,
        "name": "columns.base_color.5",
        "value": "FFFFFF"
    },
    {
        "type": 1,
        "name": "columns.timeshift.5",
        "value": ""
    },
    {
        "type": 1,
        "name": "columns.item.5",
        "value": "Number of processes"
    },
    {
        "type": 0,
        "name": "column",
        "value": 1
    }
    ]
}
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
],
"users": [
    {
        "userid": 1,
        "permission": 3
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

22 Trigger overview

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Trigger Overview** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Trigger Overview widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Show	0	show	1 - (default) Recent problems; 2 - Any; 3 - Problems.
Host groups	2	groupids	Host group ID.
Hosts	3	hostids	Host ID.

Note: To configure multiple host groups, create a dashboard widget field object for each host group.

Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.

Tags
(the number in the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)

Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value.

Parameter Tag name required if configuring Tags.

Parameter	type	name	value
Operator	0	tags.operator.0	0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value.
Show suppressed problems	0	show_suppressed	Parameter Tag value required if configuring Tags. 0 - (default) Disabled; 1 - Enabled.
Hosts location	0	style	0 - (default) Left; 1 - Top.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Trigger overview widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Trigger overview widget

Configure a Trigger overview widget that displays trigger states for all host groups that have triggers with a tag that has the name "scope" and contains value "availability".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "trigover",
            "name": "Trigger overview",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 1,
                "name": "tags.tag.0",
                "value": "scope"
              },
              {
                "type": 0,
                "name": "tags.operator.0",
                "value": 0
              }
            ]
          }
        ]
      }
    ]
  }
}
```



```

        "type": 1,
        "name": "tags.value.0",
        "value": "availability"
    }
    ]
}
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
],
"users": [
    {
        "userid": 1,
        "permission": 3
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

23 URL

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the [URL](#) widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the URL widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - (default) No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.

Parameter	type	name	value
URL (required)	1	url	Valid URL string.
Dynamic item	0	dynamic	0 - (default) Disabled; 1 - Enabled.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the URL widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a URL widget

Configure a URL widget that displays the home page of Zabbix manual.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "url",
            "name": "URL",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 1,
                "name": "url",
                "value": "https://www.zabbix.com/documentation/6.0/en"
              }
            ]
          }
        ]
      }
    ]
  },
  "userGroups": [
    {
      "usrgrp_id": 7,
      "permission": 2
    }
  ],
  "users": [
    {
      "userid": 1,
      "permission": 3
    }
  ]
},
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

24 Web monitoring

Description

These parameters and the possible property values for the respective dashboard widget field objects allow to configure the **Web monitoring** widget in `dashboard.create` and `dashboard.update` methods.

Parameters

The following parameters are supported for the Web monitoring widget.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - No refresh; 10 - 10 seconds; 30 - 30 seconds; 60 - (default) 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Host groups	2	groupids	Host group ID. Note: To configure multiple host groups, create a dashboard widget field object for each host group.
Exclude host groups	2	exclude_groupids	Host group ID. Note: To exclude multiple host groups, create a dashboard widget field object for each host group.
Hosts	3	hostids	Host ID. Note: To configure multiple hosts, create a dashboard widget field object for each host. For multiple hosts, the parameter Host groups must either be not configured at all or configured with at least one host group that the configured hosts belong to.

Parameter	type	name	value
Tags (the number in the property name (e.g. tags.tag.0) references tag order in the tag evaluation list)			
Evaluation type	0	evaltype	0 - (default) And/Or; 2 - Or.
Tag name	1	tags.tag.0	Any string value.
Operator	0	tags.operator.0	Parameter Tag name required if configuring Tags. 0 - Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
Tag value	1	tags.value.0	Parameter Operator required if configuring Tags. Any string value.
Show hosts in maintenance	0	maintenance	Parameter Tag value required if configuring Tags. 0 - Disabled; 1 - (default) Enabled.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the Web monitoring widget. For more information on configuring a dashboard, see [dashboard.create](#).

Configuring a Web monitoring widget

Configure a Web monitoring widget that displays a status summary of the active web monitoring scenarios for host group "4".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
```

```

"name": "My dashboard",
"display_period": 30,
"auto_start": 1,
"pages": [
  {
    "widgets": [
      {
        "type": "web",
        "name": "Web monitoring",
        "x": 0,
        "y": 0,
        "width": 6,
        "height": 3,
        "view_mode": 0,
        "fields": [
          {
            "type": 2,
            "name": "groupids",
            "value": 4
          }
        ]
      }
    ]
  }
],
"userGroups": [
  {
    "usrgrpid": 7,
    "permission": 2
  }
],
"users": [
  {
    "userid": 1,
    "permission": 3
  }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "3"
    ]
  },
  "id": 1
}

```

See also

- [Dashboard widget field](#)
- [dashboard.create](#)
- [dashboard.update](#)

dashboard.create

説明

object dashboard.create(object/array dashboards)

この方法では、新しいダッシュボードを作成することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object/array) 作成するダッシュボード

[standard dashboard properties](#) に加えて、このメソッドは次のパラメータを受け付けます。

パラメータ	Type	説明
pages (required)	array	Dashboard pages to be created for the dashboard. Dashboard pages will be ordered in the same order as specified. At least one dashboard page object is required for pages property.
users	array	Dashboard user shares to be created on the dashboard.
userGroups	array	Dashboard user group shares to be created on the dashboard.

戻り値

(object) dashboardids プロパティの下に作成されたダッシュボードの ID を含むオブジェクトを返します。
 返される ID の順序は、渡されたダッシュボードの順序と一致します。

例

ダッシュボードを作成する

1つのダッシュボードページに、タグ付きの Problems ウィジェットを1つ、2種類の共有（ユーザーグループとユーザー）を
 使用して、“My dashboard” という名前のダッシュボードを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.create",
  "params": {
    "name": "My dashboard",
    "display_period": 30,
    "auto_start": 1,
    "pages": [
      {
        "widgets": [
          {
            "type": "problems",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 1,
                "name": "tags.tag.0",
                "value": "service"
              },
              {
                "type": 0,
                "name": "tags.operator.0",
                "value": 1
              }
            ]
          },
          {
            "type": 1,
            "name": "tags.value.0",
```

```

        "value": "zabbix_server"
    }
    ]
}
],
"userGroups": [
    {
        "usrgrpid": "7",
        "permission": 2
    }
],
"users": [
    {
        "userid": "4",
        "permission": 3
    }
]
],
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}

```

参照

- [Dashboard page](#)
- [Dashboard widget](#)
- [Dashboard widget field](#)
- [Dashboard user](#)
- [Dashboard user group](#)

ソース

CDashboard::create() in ui/include/classes/api/services/CDashboard.php.

dashboard.delete

説明

object dashboard.delete(array dashboardids)

ダッシュボードを削除する方法です。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で `
` 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(array) 削除するダッシュボードの ID 配列

戻り値

(object) ``dashboardids プロパティの下にある、削除されたダッシュボードの ID を含むオブジェクトを返します。

例

複数のダッシュボードを削除する

2つのダッシュボードを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.delete",
  "params": [
    "2",
    "3"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "2",
      "3"
    ]
  },
  "id": 1
}
```

ソース

CDashboard::delete() in ui/include/classes/api/services/CDashboard.php.

dashboard.get

説明

integer/array dashboard.get(object parameters)

このメソッドは、指定されたパラメータに従ってダッシュボードを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
dashboardids	string/array	Return only dashboards with the given IDs.
selectPages	query	Return a pages property with dashboard pages, correctly ordered.
selectUsers	query	Return a users property with users that the dashboard is shared with.
selectUserGroups	query	Return a userGroups property with user groups that the dashboard is shared with.
sortfield	string/array	Sort the result by the given properties.
countOutput	boolean	Possible value is: dashboardid. These parameters being common for all get methods are described in detail in the reference commentary page.
editable	boolean	
excludeSearch	boolean	

パラメータ	Type	説明
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ID でダッシュボードを取得する

ダッシュボード"1" "2" についての全データを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.get",
  "params": {
    "output": "extend",
    "selectPages": "extend",
    "selectUsers": "extend",
    "selectUserGroups": "extend",
    "dashboardids": [
      "1",
      "2"
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "dashboardid": "1",
      "name": "Dashboard",
      "userid": "1",
      "private": "0",
      "display_period": "30",
      "auto_start": "1",
      "users": [],
      "userGroups": [],
      "pages": [
        {
          "dashboard_pageid": "1",
          "name": "",
          "display_period": "0",
          "widgets": [
            {
              "widgetid": "9",
              "type": "systeminfo",
            }
          ]
        }
      ]
    }
  ]
}
```

```

        "name": "",
        "x": "12",
        "y": "8",
        "width": "12",
        "height": "5",
        "view_mode": "0",
        "fields": []
    },
    {
        "widgetid": "8",
        "type": "problemsbysv",
        "name": "",
        "x": "12",
        "y": "4",
        "width": "12",
        "height": "4",
        "view_mode": "0",
        "fields": []
    },
    {
        "widgetid": "7",
        "type": "problemhosts",
        "name": "",
        "x": "12",
        "y": "0",
        "width": "12",
        "height": "4",
        "view_mode": "0",
        "fields": []
    },
    {
        "widgetid": "6",
        "type": "discovery",
        "name": "",
        "x": "6",
        "y": "9",
        "width": "6",
        "height": "4",
        "view_mode": "0",
        "fields": []
    },
    {
        "widgetid": "5",
        "type": "web",
        "name": "",
        "x": "0",
        "y": "9",
        "width": "6",
        "height": "4",
        "view_mode": "0",
        "fields": []
    },
    {
        "widgetid": "4",
        "type": "problems",
        "name": "",
        "x": "0",
        "y": "3",
        "width": "12",
        "height": "6",
        "view_mode": "0",
        "fields": []
    }

```

```

        },
        {
            "widgetid": "3",
            "type": "favmaps",
            "name": "",
            "x": "8",
            "y": "0",
            "width": "4",
            "height": "3",
            "view_mode": "0",
            "fields": []
        },
        {
            "widgetid": "1",
            "type": "favgraphs",
            "name": "",
            "x": "0",
            "y": "0",
            "width": "4",
            "height": "3",
            "view_mode": "0",
            "fields": []
        }
    ]
},
{
    "dashboard_pageid": "2",
    "name": "",
    "display_period": "0",
    "widgets": []
},
{
    "dashboard_pageid": "3",
    "name": "Custom page name",
    "display_period": "60",
    "widgets": []
}
],
{
    "dashboardid": "2",
    "name": "My dashboard",
    "userid": "1",
    "private": "1",
    "display_period": "60",
    "auto_start": "1",
    "users": [
        {
            "userid": "4",
            "permission": "3"
        }
    ],
    "userGroups": [
        {
            "usrgrpid": "7",
            "permission": "2"
        }
    ],
    "pages": [
        {
            "dashboard_pageid": "4",
            "name": "",

```

```

        "display_period": "0",
        "widgets": [
            {
                "widgetid": "10",
                "type": "problems",
                "name": "",
                "x": "0",
                "y": "0",
                "width": "12",
                "height": "5",
                "view_mode": "0",
                "fields": [
                    {
                        "type": "2",
                        "name": "groupids",
                        "value": "4"
                    }
                ]
            }
        ]
    },
    "id": 1
}

```

参照

- [Dashboard page](#)
- [Dashboard widget](#)
- [Dashboard widget field](#)
- [Dashboard user](#)
- [Dashboard user group](#)

ソース

CDashboard::get() in ui/include/classes/api/services/CDashboard.php.

dashboard.update

説明

object dashboard.update(object/array dashboards)

この方法では、既存のダッシュボードを更新することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object/array) 更新されるダッシュボードのプロパティ

dashboardid プロパティは各ダッシュボードに対して指定する必要があり、その他のプロパティは任意です。
 指定されたプロパティのみが更新されます。

[standard dashboard properties](#) に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
pages	array	Dashboard pages to replace the existing dashboard pages. Dashboard pages are updated by the <code>dashboard_pageid</code> property. New dashboard pages will be created for objects without <code>dashboard_pageid</code> property and the existing dashboard pages will be deleted if not reused. Dashboard pages will be ordered in the same order as specified. Only the specified properties of the dashboard pages will be updated. At least one dashboard page object is required for <code>pages</code> property.
users	array	Dashboard user shares to replace the existing elements.
userGroups	array	Dashboard user group shares to replace the existing elements.

戻り値

(object) `dashboardids` プロパティの下にある、更新されたダッシュボードの ID を含むオブジェクトを返します。

例

ダッシュボードの名前変更

ダッシュボードの名前を "SQL サーバーの状態" に変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.update",
  "params": {
    "dashboardid": "2",
    "name": "SQL server status"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "2"
    ]
  },
  "id": 1
}
```

ダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換え、3 番目のページとして

新しいページを追加します。他のダッシュボードページをすべて削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.update",
  "params": {
    "dashboardid": "2",
    "pages": [
      {
        "dashboard_pageid": 1,
        "name": 'Renamed Page'
      },
      {
        "dashboard_pageid": 2,
```

```
        "widgets": [
          {
            "type": "clock",
            "x": 0,
            "y": 0,
            "width": 4,
            "height": 3
          }
        ]
      },
      {
        "display_period": 60
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "2"
    ]
  },
  "id": 2
}
```

ダッシュボードの所有者を変更する

admins と super admins のみ利用可能です

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dashboard.update",
  "params": {
    "dashboardid": "2",
    "userid": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 2
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "2"
    ]
  },
  "id": 2
}
```

参照

- [Dashboard page](#)
- [Dashboard widget](#)
- [Dashboard widget field](#)
- [Dashboard user](#)
- [Dashboard user group](#)

ソース

CDashboard::update() in ui/include/classes/api/services/CDashboard.php.

Discovered host

このクラスは、検出されたホストと連携して動作するように設計されています。

オブジェクトリファレンス:

- [Discovered host](#)

利用可能なメソッド:

- [dhost.get](#) - 検出されたホストを検索する

> Discovered host object

以下のオブジェクトは dhost API に直接関連するものです。

Discovered host

Note:

検出されたホストは Zabbix server によって作成され、API から変更することはできません。

検出されたホストオブジェクトは、ネットワークディスカバリールールによって検出されたホストに関する情報を含んでいます。

 このオブジェクトは以下のプロパティを持ちます。

プロパティ	Type	説明
dhostid	string	ID of the discovered host.
druleid	string	ID of the discovery rule that detected the host.
lastdown	timestamp	Time when the discovered host last went down.
lastup	timestamp	Time when the discovered host last went up.
status	integer	Whether the discovered host is up or down. A host is up if it has at least one active discovered service.

Possible values:
0 - host up;
1 - host down.

dhost.get

説明

integer/array dhost.get(object parameters)

このメソッドは、与えられたパラメータに従って、検出されたホストを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
dhostids	string/array	Return only discovered hosts with the given IDs.
druleids	string/array	Return only discovered hosts that have been created by the given discovery rules.

パラメータ	Type	説明
dserviceids	string/array	Return only discovered hosts that are running the given services.
selectDRules	query	Return a drules property with an array of the discovery rules that detected the host.
selectDServices	query	Return a dservices property with the discovered services running on the host.
limitSelects	integer	Supports count. Limits the number of records returned by subselects.
sortfield	string/array	Applies to the following subselects: selectDServices - results will be sorted by dserviceid. Sort the result by the given properties.
countOutput	boolean	Possible values are: dhostid and druleid. These parameters being common for all get methods are described in detail in the reference commentary .
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

検出されたホストを検出ルールで取得する

ディスクバリールール"4"によって検出されたすべてのホストと、それらが実行している検出されたサービスを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dhost.get",
  "params": {
    "output": "extend",
    "selectDServices": "extend",
    "druleids": "4"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "dservices": [
        {
          "dserviceid": "1",
          "dhostid": "1",

```



```

        "type": "4",
        "key_": "",
        "value": "",
        "port": "80",
        "status": "0",
        "lastup": "1337697227",
        "lastdown": "0",
        "dcheckid": "5",
        "ip": "192.168.1.1",
        "dns": "station.company.lan"
    }
],
"dhostid": "1",
"druleid": "4",
"status": "0",
"lastup": "1337697227",
"lastdown": "0"
},
{
    "dservices": [
        {
            "dserviceid": "2",
            "dhostid": "2",
            "type": "4",
            "key_": "",
            "value": "",
            "port": "80",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0",
            "dcheckid": "5",
            "ip": "192.168.1.4",
            "dns": "john.company.lan"
        }
    ],
    "dhostid": "2",
    "druleid": "4",
    "status": "0",
    "lastup": "1337697234",
    "lastdown": "0"
},
{
    "dservices": [
        {
            "dserviceid": "3",
            "dhostid": "3",
            "type": "4",
            "key_": "",
            "value": "",
            "port": "80",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0",
            "dcheckid": "5",
            "ip": "192.168.1.26",
            "dns": "printer.company.lan"
        }
    ],
    "dhostid": "3",
    "druleid": "4",
    "status": "0",
    "lastup": "1337697234",

```

```

        "lastdown": "0"
    },
    {
        "dservices": [
            {
                "dserviceid": "4",
                "dhostid": "4",
                "type": "4",
                "key_": "",
                "value": "",
                "port": "80",
                "status": "0",
                "lastup": "1337697234",
                "lastdown": "0",
                "dcheckid": "5",
                "ip": "192.168.1.7",
                "dns": "mail.company.lan"
            }
        ],
        "dhostid": "4",
        "druleid": "4",
        "status": "0",
        "lastup": "1337697234",
        "lastdown": "0"
    }
],
"id": 1
}

```

参照

- [Discovered service](#)
- [Discovery rule](#)

ソース

CDHost::get() in ui/include/classes/api/services/CDHost.php.

Discovered service

このクラスは、検出されたサービスと連携するように設計されています。

オブジェクトリファレンス:

- [Discovered service](#)

利用可能なメソッド:

- [dservice.get](#) - 検出されたサービスを検索する

> Discovered service object

以下のオブジェクトは dservice API に直接関連するものです。

Discovered service

Note:

発見されたサービスは Zabbix server によって作成され、API から変更することはできません。

発見されたサービスオブジェクトは、ホスト上のネットワーク発見ルールによって発見されたサービスに関する情報を
 含んでいます。このオブジェクトは以下のプロパティを持ちます。

プロパティ	Type	説明
dserviceid	string	ID of the discovered service.
dcheckid	string	ID of the discovery check used to detect the service.
dhostid	string	ID of the discovered host running the service.
dns	string	DNS of the host running the service.
ip	string	IP address of the host running the service.
lastdown	timestamp	Time when the discovered service last went down.
lastup	timestamp	Time when the discovered service last went up.
port	integer	Service port number.
status	integer	Status of the service.
		Possible values: 0 - service up; 1 - service down.
value	string	Value returned by the service when performing a Zabbix agent, SNMPv1, SNMPv2 or SNMPv3 discovery check.

dservice.get

説明

integer/array dservice.get(object parameters)

このメソッドは、指定されたパラメータに従って、検出されたサービスを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
dserviceids	string/array	Return only discovered services with the given IDs.
dhostids	string/array	Return only discovered services that belong to the given discovered hosts.
dcheckids	string/array	Return only discovered services that have been detected by the given discovery checks.
druleids	string/array	Return only discovered services that have been detected by the given discovery rules.
selectDRules	query	Return a drules property with an array of the discovery rules that detected the service.
selectDHosts	query	Return a dhosts property with an array the discovered hosts that the service belongs to.
selectHosts	query	Return a hosts property with the hosts with the same IP address and proxy as the service.
		Supports count.
limitSelects	integer	Limits the number of records returned by subselects.
		Applies to the following subselects: selectHosts - result will be sorted by hostid.
sortfield	string/array	Sort the result by the given properties.
		Possible values are: dserviceid, dhostid and ip.
countOutput	boolean	These parameters being common for all get methods are described in detail in the reference commentary .
editable	boolean	
excludeSearch	boolean	

パラメータ	Type	説明
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ホスト上で検出されたサービスを取得する

発見されたホスト "11" 上で検出されたすべてのサービスを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dservice.get",
  "params": {
    "output": "extend",
    "dhostids": "11"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "dserviceid": "12",
      "dhostid": "11",
      "value": "",
      "port": "80",
      "status": "1",
      "lastup": "0",
      "lastdown": "1348650607",
      "dcheckid": "5",
      "ip": "192.168.1.134",
      "dns": "john.local"
    },
    {
      "dserviceid": "13",
      "dhostid": "11",
      "value": "",
      "port": "21",
      "status": "1",
      "lastup": "0",
      "lastdown": "1348650610",
      "dcheckid": "6",
      "ip": "192.168.1.134",
      "dns": "john.local"
    }
  ]
}
```

```

    ],
    "id": 1
}

```

参照

- [Discovered host](#)
- [Discovery check](#)
- [Host](#)

ソース

CDServic::get() in ui/include/classes/api/services/CDServic.php.

Discovery check

ディスカバリーチェックと連動したクラスです。

オブジェクトリファレンス:

- [Discovery check](#)

利用可能なメソッド:

- [dcheck.get](#) - ディスカバリーチェックの取得

> Discovery check object

以下のオブジェクトは dcheck API に直接関連するものです。

Discovery check

ディスカバリーチェックオブジェクトは、ネットワークディスカバリールールによって実行される特定のチェックを定義します。

 このオブジェクトは以下のプロパティを持ちます。

プロパティ	Type	説明
dcheckid	string	(readonly) ID of the discovery check.
druleid	string	(readonly) ID of the discovery rule that the check belongs to.
key_	string	The value of this property differs depending on the type of the check: - key to query for Zabbix agent checks, required; - SNMP OID for SNMPv1, SNMPv2 and SNMPv3 checks, required.
ports	string	One or several port ranges to check separated by commas. Used for all checks except for ICMP.
snmp_community	string	Default: 0. SNMP community.
snmpv3_authpassphrase	string	Required for SNMPv1 and SNMPv2 agent checks. Authentication passphrase used for SNMPv3 agent checks with security level set to authNoPriv or authPriv.
snmpv3_authprotocol	integer	Authentication protocol used for SNMPv3 agent checks with security level set to authNoPriv or authPriv. Possible values: 0 - (default) MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
snmpv3_contextname	string	SNMPv3 context name. Used only by SNMPv3 checks.
snmpv3_privpassphrase	string	Privacy passphrase used for SNMPv3 agent checks with security level set to authPriv.

プロパティ	Type	説明
snmpv3_privprotocol	integer	Privacy protocol used for SNMPv3 agent checks with security level set to authPriv. Possible values: 0 - (default) DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
snmpv3_securitylevel	string	Security level used for SNMPv3 agent checks. Possible values: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
snmpv3_securityname type (required)	string integer	Security name used for SNMPv3 agent checks. Type of check. Possible values: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet.
uniq	integer	Whether to use this check as a device uniqueness criteria. Only a single unique check can be configured for a discovery rule. Used for Zabbix agent, SNMPv1, SNMPv2 and SNMPv3 agent checks. Possible values: 0 - (default) do not use this check as a uniqueness criteria; 1 - use this check as a uniqueness criteria.
host_source	integer	Source for host name. Possible values: 1 - (default) DNS; 2 - IP; 3 - discovery value of this check.
name_source	integer	Source for visible name. Possible values: 0 - (default) not specified; 1 - DNS; 2 - IP; 3 - discovery value of this check.

dcheck.get

説明

integer/array dcheck.get(object parameters)

このメソッドは、指定されたパラメータに従って、ディスカバリーチェックを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
dcheckids	string/array	Return only discovery checks with the given IDs.
druleids	string/array	Return only discovery checks that belong to the given discovery rules.
dserviceids	string/array	Return only discovery checks that have detected the given discovered services.
sortfield	string/array	Sort the result by the given properties. Possible values are: dcheckid and druleid.
countOutput	boolean	These parameters being common for all get methods are described in detail in the reference commentary .
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します。:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ディスカバリールール⁶のディスカバリーチェックを取得する

ディスカバリールール⁶で 사용되는すべてのディスカバリーチェックを取得する。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "dcheck.get",
  "params": {
    "output": "extend",
    "dcheckids": "6"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
```

```

    {
        "dcheckid": "6",
        "druleid": "4",
        "type": "3",
        "key_": "",
        "snmp_community": "",
        "ports": "21",
        "snmpv3_securityname": "",
        "snmpv3_securitylevel": "0",
        "snmpv3_authpassphrase": "",
        "snmpv3_privpassphrase": "",
        "uniq": "0",
        "snmpv3_authprotocol": "0",
        "snmpv3_privprotocol": "0",
        "host_source": "1",
        "name_source": "0"
    }
],
    "id": 1
}

```

ソース

CDCheck::get() in ui/include/classes/api/services/CDCheck.php.

Discovery rule

このクラスは、ネットワークディスカバリールールと連動するように設計されています。

Note:

この API は、ネットワークディスカバリールールを扱うためのものです。低レベルのディスカバリールールについては、
[LLD rule API](#) を参照してください。

オブジェクトリファレンス:

- [Discovery rule](#)

利用可能なメソッド:

- [drule.create](#) - 新しいディスカバリールールを作成する
- [drule.delete](#) - ディスカバリールールの削除
- [drule.get](#) - ディスカバリールールの取得
- [drule.update](#) - ディスカバリールールの更新

> Discovery rule object

次のオブジェクトは、druleAPI に直接関連しています。

Discovery rule

ディスカバリールールオブジェクトは、ネットワークディスカバリールールを定義します。以下の特性があります。

プロパティ	タイプ	説明
druleid	string	(読取専用) ディスカバリールールの ID
iprange (必須)	string	確認する 1 つまたは複数の IP 範囲。複数の場合はカンマ区切り。 サポートされている IP 範囲の形式詳細については network discovery configuration セクションを参照してください
name (必須)	string	ディスカバリールール名

プロパティ	タイプ	説明
delay	string	ディスカバリルールの実行間隔。秒、接尾辞付きの時間単位、およびユーザーマクロを受け入れます。
nextcheck	timestamp	デフォルト:1h (読取専用) 次にディスカバリルールが実行される時刻
proxy_hostid	string	ディスカバリに使用されるプロキシの ID
status	integer	検出ルールが有効か無効か
		可能な値: 0 - (デフォルト) 有効 1 - 無効

drule.create

説明

object drule.create(object/array discoveryRules)

この方法によって、新しいディスカバリルールを作成することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 作成するディスカバリルール

standard discovery rule properties に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
dchecks (required)	array	Discovery checks to create for the discovery rule.

戻り値

(object) druleids プロパティの下に作成されたディスカバリルールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたディスカバリルールの順序と一致します。

例

ディスカバリルールを作成する

ローカルネットワークで Zabbix エージェントを実行しているマシンを見つけるためのディスカバリルールを作成します。ルールはポート 10050 で単一の Zabbix エージェントチェックを使用する必要があります。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "drule.create",
  "params": {
    "name": "Zabbix agent discovery",
    "iprange": "192.168.1.1-255",
    "dchecks": [
      {
        "type": "9",
        "key_": "system.uname",
        "ports": "10050",
        "uniq": "0"
      }
    ]
  },
}
```

```
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "druleids": [
      "6"
    ]
  },
  "id": 1
}
```

参照

- [Discovery check](#)

ソース

CDRule::create() in ui/include/classes/api/services/CDRule.php.

drule.delete

説明

object drule.delete(array discoveryRuleIds)

この方法では、ディスカバリールールを削除することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(array) 削除するディスカバリールールの ID 配列

戻り値

(object) druleids プロパティの下にある、削除されたディスカバリールールの ID を含むオブジェクトを返します。

例

複数のディスカバリールールを削除する

2つのディスカバリールールを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "drule.delete",
  "params": [
    "4",
    "6"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "druleids": [
      "4",

```

```

        "6"
    ]
},
"id": 1
}

```

ソース

CDRule::delete() in ui/include/classes/api/services/CDRule.php.

drule.get

説明

integer/array drule.get(object parameters)

このメソッドは、与えられたパラメータに従ってディスカバリールールを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
dhostids	string/array	Return only discovery rules that created the given discovered hosts.
druleids	string/array	Return only discovery rules with the given IDs.
dserviceids	string/array	Return only discovery rules that created the given discovered services.
selectDChecks	query	Return a dchecks property with the discovery checks used by the discovery rule.
selectDHosts	query	Supports count. Return a dhosts property with the discovered hosts created by the discovery rule.
limitSelects	integer	Supports count. Limits the number of records returned by subselects.
sortfield	string/array	Applies to the following subselects: selectDChecks - results will be sorted by dcheckid; selectDHosts - results will be sorted by dhostsid. Sort the result by the given properties.
countOutput	boolean	Possible values are: druleid and name. These parameters being common for all get methods are described in detail in the reference commentary .
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

すべてのディスカバリールールを取得する

設定されたすべてのディスカバリールールと、それらが使用するディスカバリーチェックを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "drule.get",
  "params": {
    "output": "extend",
    "selectDChecks": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "druleid": "2",
      "proxy_hostid": "0",
      "name": "Local network",
      "iprange": "192.168.3.1-255",
      "delay": "5s",
      "nextcheck": "1348754327",
      "status": "0",
      "dchecks": [
        {
          "dcheckid": "7",
          "druleid": "2",
          "type": "3",
          "key_": "",
          "snmp_community": "",
          "ports": "21",
          "snmpv3_securityname": "",
          "snmpv3_securitylevel": "0",
          "snmpv3_authpassphrase": "",
          "snmpv3_privpassphrase": "",
          "uniq": "0",
          "snmpv3_authprotocol": "0",
          "snmpv3_privprotocol": "0",
          "host_source": "1",
          "name_source": "0"
        },
        {
          "dcheckid": "8",
          "druleid": "2",
          "type": "4",
          "key_": "",
          "snmp_community": "",
          "ports": "80",
          "snmpv3_securityname": "",
          "snmpv3_securitylevel": "0",
          "snmpv3_authpassphrase": "",
          "snmpv3_privpassphrase": "",
          "uniq": "0",

```

```

        "snmpv3_authprotocol": "0",
        "snmpv3_privprotocol": "0",
        "host_source": "1",
        "name_source": "0"
    }
]
},
{
    "druleid": "6",
    "proxy_hostid": "0",
    "name": "Zabbix agent discovery",
    "iprange": "192.168.1.1-255",
    "delay": "1h",
    "nextcheck": "0",
    "status": "0",
    "dchecks": [
        {
            "dcheckid": "10",
            "druleid": "6",
            "type": "9",
            "key_": "system.uname",
            "snmp_community": "",
            "ports": "10050",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "host_source": "2",
            "name_source": "3"
        }
    ]
}
],
"id": 1
}

```

参照

- [Discovered host](#)
- [Discovery check](#)

ソース

CDRule::get() in ui/include/classes/api/services/CDRule.php.

drule.update

説明

object drule.update(object/array discoveryRules)

この方法によって、既存のディスカバリールールを更新することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 更新するディスカバリールールのプロパティ

druleid プロパティは各ディスカバリールールに対して定義する必要があり、他のプロパティはすべて任意です。
渡されたプロパティのみが更新され、他のプロパティは変更されません。

standard discovery rule propertiesに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
dchecks	array	Discovery checks to replace existing checks.

戻り値

(object) druleids プロパティの下にある、更新された探索ルールの ID を含むオブジェクトを返します。

例

ディスカバリールールの IP レンジを変更する

ディスカバリールールの IP 範囲を"192.168.2.1-255" に変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "drule.update",
  "params": {
    "druleid": "6",
    "iprange": "192.168.2.1-255"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "druleids": [
      "6"
    ]
  },
  "id": 1
}
```

参照

- [Discovery check](#)

ソース

CDRule::update() in ui/include/classes/api/services/CDRule.php.

Event

このクラスは、イベントと連動するように設計されています。

オブジェクトリファレンス:

- [Event](#)

利用可能なメソッド:

- [event.get](#) - イベントの取得
- [event.acknowledge](#) - イベントの確認

> Event object

以下のオブジェクトは event API に直接関連するものです。

Event

Note:

Event は Zabbix server によって作成され、API から変更することはできません。

Event オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
eventid	string	ID of the event.
source	integer	Type of the event. Possible values: 0 - event created by a trigger; 1 - event created by a discovery rule; 2 - event created by active agent autoregistration; 3 - internal event; 4 - event created on service status update.
object	integer	Type of object that is related to the event. Possible values for trigger events: 0 - trigger. Possible values for discovery events: 1 - discovered host; 2 - discovered service. Possible values for autoregistration events: 3 - auto-registered host. Possible values for internal events: 0 - trigger; 4 - item; 5 - LLD rule. Possible values for service events: 6 - service.
objectid	string	ID of the related object.
acknowledged	integer	Whether the event has been acknowledged.
clock	timestamp	Time when the event was created.
ns	integer	Nanoseconds when the event was created.
name	string	Resolved event name.
value	integer	State of the related object. Possible values for trigger and service events: 0 - OK; 1 - problem. Possible values for discovery events: 0 - host or service up; 1 - host or service down; 2 - host or service discovered; 3 - host or service lost. Possible values for internal events: 0 - "normal" state; 1 - "unknown" or "not supported" state.

This parameter is not used for active agent autoregistration events.

プロパティ	Type	説明
severity	integer	Event current severity. Possible values: 0 - not classified; 1 - information; 2 - warning; 3 - average; 4 - high; 5 - disaster.
r_eventid	string	Recovery event ID
c_eventid	string	ID of the event that was used to override (close) current event under global correlation rule. See <code>correlationid</code> to identify exact correlation rule. This parameter is only defined when the event is closed by global correlation rule.
correlationid	string	ID of the correlation rule that generated closing of the problem. This parameter is only defined when the event is closed by global correlation rule.
userid	string	User ID if the event was manually closed.
suppressed	integer	Whether the event is suppressed. Possible values: 0 - event is in normal state; 1 - event is suppressed.
opdata	string	Operational data with expanded macros.
urls	array of Media type URLs	Active media types URLs.

Event tag

Event tag オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
tag	string	Event tag name.
value	string	Event tag value.

Media type URLs

media type URL を持つオブジェクトは、以下のプロパティを持つ。

プロパティ	Type	説明
name	string	Media type defined URL name.
url	string	Media type defined URL value.

結果は、イベントメニューエントリーが有効になっているアクティブなメディアタイプについてのみ、エントリーを `
` 含んでいます。プロパティで使用されているマクロは展開されますが、一方のプロパティが展開されていないマクロを `
` 含む場合、両方のプロパティが結果から除外されます。サポートされているマクロは、[page](#)で説明されています。

event.acknowledge

説明

`object event.acknowledge(object/array parameters)`

このメソッドにより、イベントを更新することができます。以下の更新操作を行うことができます:

- イベントを閉じる。イベントがすでに解決されている場合、このアクションはスキップされます。
- イベントを確認します。イベントがすでに確認されている場合、このアクションはスキップされます。
- メッセージを追加します。
- イベントの重要度を変更します。イベントの重要度がすでに同じ場合は、このアクションはスキップされます。

Attention:

トリガーイベントのみ更新可能です。問題イベントのみ更新可能です。イベントを閉じる、またはイベントの重要度を
 変更するには、トリガに対する読み取り/書き込み権限が必要です。イベントを閉じるには、トリガで手動によるクローズを
 許可する必要があります。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object/array) 実行すべきイベントと更新操作の ID を含むパラメータ

パラメータ	Type	説明
eventids (required)	string/object	IDs of the events to acknowledge.
action (required)	integer	Event update action(s). This is bitmask field, any combination of values is acceptable. Possible values: 1 - close problem; 2 - acknowledge event; 4 - add message; 8 - change severity; 16 - unacknowledge event.
message	string	Text of the message. Required , if action contains 'add message' flag.
severity	integer	New severity for events. Required , if action contains 'change severity' flag. Possible values: 0 - not classified; 1 - information; 2 - warning; 3 - average; 4 - high; 5 - disaster.

戻り値

(object) eventids プロパティの下で更新されたイベントの ID を含むオブジェクトを返します。

例

イベントの確認

1つのイベントを確認し、メッセージを残します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "event.acknowledge",
  "params": {
    "eventids": "20427",
    "action": 6,
    "message": "Problem resolved."
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "eventids": [
      "20427"
    ]
  },
  "id": 1
}
```

イベントの重要度を変更する

複数のイベントの重要度を変更し、メッセージを残すことができます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "event.acknowledge",
  "params": {
    "eventids": ["20427", "20428"],
    "action": 12,
    "message": "Maintenance required to fix it.",
    "severity": 4
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "eventids": [
      "20427",
      "20428"
    ]
  },
  "id": 1
}
```

ソース

CEvent::acknowledge() in ui/include/classes/api/services/CEvent.php.

event.get

説明

integer/array event.get(object parameters)

このメソッドは、与えられたパラメータに従ってイベントを取得することができます。

Attention:

このメソッドは、削除されたエンティティのイベントを返すことがあります。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

Parameters

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
eventids	string/array	Return only events with the given IDs.
groupids	string/array	Return only events created by objects that belong to the given host groups.
hostids	string/array	Return only events created by objects that belong to the given hosts.
objectids	string/array	Return only events created by the given objects.
source	integer	Return only events with the given type. Refer to the event object page for a list of supported event types.
object	integer	Default: 0 - trigger events. Return only events created by objects of the given type. Refer to the event object page for a list of supported object types.
acknowledged	boolean	Default: 0 - trigger.
suppressed	boolean	If set to true return only acknowledged events. true - return only suppressed events; false - return events in the normal state.
severities	integer/array	Return only events with given event severities. Applies only if object is trigger.
evaltype	integer	Rules for tag searching. Possible values: 0 - (default) And/Or; 2 - Or.
tags	array of objects	Return only events with given tags. Exact match by tag and case-insensitive search by value and operator. Format: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]. An empty array returns all events. Possible operator types: 0 - (default) Like; 1 - Equal; 2 - Not like; 3 - Not equal 4 - Exists; 5 - Not exists.
eventid_from	string	Return only events with IDs greater or equal to the given ID.
eventid_till	string	Return only events with IDs less or equal to the given ID.
time_from	timestamp	Return only events that have been created after or at the given time.
time_till	timestamp	Return only events that have been created before or at the given time.
problem_time_from	timestamp	Returns only events that were in the problem state starting with <code>problem_time_from</code> . Applies only if the source is trigger event and object is trigger. Mandatory if <code>problem_time_till</code> is specified.
problem_time_till	timestamp	Returns only events that were in the problem state until <code>problem_time_till</code> . Applies only if the source is trigger event and object is trigger. Mandatory if <code>problem_time_from</code> is specified.
value	integer/array	Return only events with the given values.
selectHosts	query	Return a <code>hosts</code> property with hosts containing the object that created the event. Supported only for events generated by triggers, items or LLD rules.
selectRelatedObject	query	Return a <code>relatedObject</code> property with the object that created the event. The type of object returned depends on the event type.
select_alerts	query	Return an <code>alerts</code> property with alerts generated by the event. Alerts are sorted in reverse chronological order.

パラメータ	Type	説明
select_acknowledges	query	Return an <code>acknowledges</code> property with event updates. Event updates are sorted in reverse chronological order. The event update object has the following properties: <code>acknowledgeid</code> - (string) acknowledgment's ID; <code>userid</code> - (string) ID of the user that updated the event; <code>eventid</code> - (string) ID of the updated event; <code>clock</code> - (timestamp) time when the event was updated; <code>message</code> - (string) text of the message; <code>action</code> - (integer) update action that was performed see event.acknowledge ; <code>old_severity</code> - (integer) event severity before this update action; <code>new_severity</code> - (integer) event severity after this update action; <code>username</code> - (string) username of the user that updated the event; <code>name</code> - (string) name of the user that updated the event; <code>surname</code> - (string) surname of the user that updated the event. Supports count.
selectTags	query	Return a <code>tags</code> property with event tags.
selectSuppressionData	query	Return a <code>suppression_data</code> property with the list of maintenances: <code>maintenanceid</code> - (string) ID of the maintenance; <code>suppress_until</code> - (integer) time until the event is suppressed.
sortfield	string/array	Sort the result by the given properties. Possible values are: <code>eventid</code> , <code>objectid</code> and <code>clock</code> .
countOutput	boolean	These parameters being common for all get methods are described in detail in the reference commentary page.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

トリガーイベントの取得

トリガー"13926" から最新のイベントを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "event.get",
  "params": {
    "output": "extend",
    "select_acknowledges": "extend",
    "selectTags": "extend",
    "selectSuppressionData": "extend",
    "objectids": "13926",
    "sortfield": ["clock", "eventid"],
```

```

    "sortorder": "DESC"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "eventid": "9695",
      "source": "0",
      "object": "0",
      "objectid": "13926",
      "clock": "1347970410",
      "value": "1",
      "acknowledged": "1",
      "ns": "413316245",
      "name": "MySQL is down",
      "severity": "5",
      "r_eventid": "0",
      "c_eventid": "0",
      "correlationid": "0",
      "userid": "0",
      "opdata": "",
      "acknowledges": [
        {
          "acknowledgeid": "1",
          "userid": "1",
          "eventid": "9695",
          "clock": "1350640590",
          "message": "Problem resolved.\n\r----[BULK ACKNOWLEDGE]----",
          "action": "6",
          "old_severity": "0",
          "new_severity": "0",
          "username": "Admin",
          "name": "Zabbix",
          "surname": "Administrator"
        }
      ],
      "suppression_data": [
        {
          "maintenanceid": "15",
          "suppress_until": "1472511600"
        }
      ],
      "suppressed": "1",
      "tags": [
        {
          "tag": "service",
          "value": "mysqld"
        },
        {
          "tag": "error",
          "value": ""
        }
      ]
    },
    {
      "eventid": "9671",
      "source": "0",

```

```

    "object": "0",
    "objectid": "13926",
    "clock": "1347970347",
    "value": "0",
    "acknowledged": "0",
    "ns": "0",
    "name": "Unavailable by ICMP ping",
    "severity": "4",
    "r_eventid": "0",
    "c_eventid": "0",
    "correlationid": "0",
    "userid": "0",
    "opdata": "",
    "acknowledges": [],
    "suppression_data": [],
    "suppressed": "0",
    "tags": []
  }
],
  "id": 1
}

```

期間別イベントの取得

2012年10月9日から10日の間に作成されたすべてのイベントを逆順に取得します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "event.get",
  "params": {
    "output": "extend",
    "time_from": "1349797228",
    "time_till": "1350661228",
    "sortfield": ["clock", "eventid"],
    "sortorder": "desc"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "eventid": "20616",
      "source": "0",
      "object": "0",
      "objectid": "14282",
      "clock": "1350477814",
      "value": "1",
      "acknowledged": "0",
      "ns": "0",
      "name": "Less than 25% free in the history cache",
      "severity": "3",
      "r_eventid": "0",
      "c_eventid": "0",
      "correlationid": "0",
      "userid": "0",
      "opdata": "",
      "suppressed": "0"
    },
  ],
}

```

```

    {
      "eventid": "20617",
      "source": "0",
      "object": "0",
      "objectid": "14283",
      "clock": "1350477814",
      "value": "0",
      "acknowledged": "0",
      "ns": "0",
      "name": "Zabbix trapper processes more than 75% busy",
      "severity": "3",
      "r_eventid": "0",
      "c_eventid": "0",
      "correlationid": "0",
      "userid": "0",
      "opdata": "",
      "suppressed": "0"
    },
    {
      "eventid": "20618",
      "source": "0",
      "object": "0",
      "objectid": "14284",
      "clock": "1350477815",
      "value": "1",
      "acknowledged": "0",
      "ns": "0",
      "name": "High ICMP ping loss",
      "severity": "3",
      "r_eventid": "0",
      "c_eventid": "0",
      "correlationid": "0",
      "userid": "0",
      "opdata": "",
      "suppressed": "0"
    }
  ],
  "id": 1
}

```

参照

- [Alert](#)
- [Item](#)
- [Host](#)
- [LLD rule](#)
- [Trigger](#)

ソース

CEvent::get() in ui/include/classes/api/services/CEvent.php.

Graph

このクラスは、グラフを操作するように設計されています。

オブジェクトのリファレンス:

- [Graph](#)

利用可能なメソッド:

- [graph.create](#) - 新しいグラフの作成
- [graph.delete](#) - グラフの削除

- `graph.get` - グラフの取得
- `graph.update` - グラフの更新

> Graph object

以下のオブジェクトは graph API に直接関連するものです。

Graph

Graph グラフオブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
<code>graphid</code>	string	(読取専用) グラフ ID
<code>height</code> (必須)	integer	グラフの高さ (ピクセル単位)
<code>name</code> (必須)	string	グラフ名
<code>width</code> (必須)	integer	グラフの幅 (ピクセル単位)
<code>flags</code>	integer	(読取専用) グラフの元 使用可能な値: 0 - (デフォルト) プレーングラフ 4 - ディスカバリされたグラフ グラフのレイアウトタイプ
<code>graphtype</code>	integer	使用可能な値: 0 - (デフォルト) 通常 1 - 積み上げ 2 - 円 3 - 分解円
<code>percent_left</code>	float	左パーセンタイル
<code>percent_right</code>	float	デフォルト: 0 右パーセンタイル
<code>show_3d</code>	integer	デフォルト: 0 円グラフと分解円グラフを 3D で表示するか
<code>show_legend</code>	integer	可能な値: 0 - (デフォルト) 2D で表示 1 - 3D で表示 グラフに凡例を表示するか
<code>show_work_period</code>	integer	使用可能な値: 0 - 非表示 1 - (デフォルト) 表示 グラフに作業時間を表示するか
<code>show_triggers</code>	integer	使用可能な値: 0 - 非表示 1 - (デフォルト) 表示 グラフにトリガーラインを表示するか
<code>templateid</code>	string	(読取専用) 親テンプレートグラフの ID
<code>yaxismax</code>	float	Y 軸の固定最大値 デフォルト: 100

プロパティ	Type	説明
yaxismin	float	Y 軸の固定最小値
ymin_itemid	string	デフォルト: 0 Y 軸の最大値として使用されるアイテムの ID
ymin_type	integer	Y 軸の最大値計算方法 使用可能な値: 0 - (デフォルト) 計算 1 - 固定 2 - アイテム
ymax_itemid	string	Y 軸の最小値として使用されるアイテムの ID
ymax_type	integer	Y 軸の最小値計算方法 使用可能な値: 0 - (デフォルト) 計算 1 - 固定 2 - アイテム
uuid	string	インポートされたグラフを既存のグラフにリンクするために使用されるユニバーサル一意識別子。テンプレートのグラフにのみ使用されます。指定されていない場合は自動生成されます。

更新操作の場合、このフィールドは読み取り専用です。

graph.create

説明

`object graph.create(object/array graphs)`

この方法で、新しいグラフを作成することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 作成するグラフ。

[standard graph properties](#)に加えて、このメソッドは以下のパラメータを受け付ける。

パラメータ	Type	説明
gitems (required)	array	Graph items to be created for the graph.

戻り値

(object) `graphids` プロパティの下に作成されたグラフの ID を含むオブジェクトを返します。返される ID の順番は、
渡されたグラフの順番と一致する。

例

グラフの作成

2つの項目からなるグラフを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graph.create",
  "params": {
    "name": "MySQL bandwidth",
```

```
    "width": 900,
    "height": 200,
    "gitems": [
      {
        "itemid": "22828",
        "color": "00AA00"
      },
      {
        "itemid": "22829",
        "color": "3333FF"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "graphids": [
      "652"
    ]
  },
  "id": 1
}
```

参照

- [Graph item](#)

ソース

CGraph::create() in ui/include/classes/api/services/CGraph.php.

graph.delete

説明

object graph.delete(array graphIds)

グラフを削除することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(array) 削除するグラフの ID 配列

戻り値

(object) graphids プロパティの下にある、削除されたグラフの ID を含むオブジェクトを返す。

例

複数のグラフを削除する

2つのグラフを削除する。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graph.delete",
  "params": [
```

```

    "652",
    "653"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "graphids": [
      "652",
      "653"
    ]
  },
  "id": 1
}

```

ソース

CGraph::delete() in ui/include/classes/api/services/CGraph.php.

graph.get

説明

integer/array graph.get(object parameters)

このメソッドでは、与えられたパラメータに応じたグラフを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
graphids	string/array	Return only graphs with the given IDs.
groupids	string/array	Return only graphs that belong to hosts in the given host groups.
templateids	string/array	Return only graph that belong to the given templates.
hostids	string/array	Return only graphs that belong to the given hosts.
itemids	string/array	Return only graphs that contain the given items.
templated	boolean	If set to true return only graphs that belong to templates.
inherited	boolean	If set to true return only graphs inherited from a template.
expandName	flag	Expand macros in the graph name.
selectGroups	query	Return a groups property with the host groups that the graph belongs to.
selectTemplates	query	Return a templates property with the templates that the graph belongs to.
selectHosts	query	Return a hosts property with the hosts that the graph belongs to.
selectItems	query	Return an items property with the items used in the graph.
selectGraphDiscovery	query	Return a graphDiscovery property with the graph discovery object. The graph discovery objects links the graph to a graph prototype from which it was created.

It has the following properties:

graphid - (string) ID of the graph;
parent_graphid - (string) ID of the graph prototype from which the graph has been created.

パラメータ	Type	説明
selectGraphItems	query	Return a <code>gitems</code> property with the items used in the graph.
selectDiscoveryRule	query	Return a <code>discoveryRule</code> property with the low-level discovery rule that created the graph.
filter	object	Return only those results that exactly match the given filter. Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against. Supports additional filters: <code>host</code> - technical name of the host that the graph belongs to; <code>hostid</code> - ID of the host that the graph belongs to. Sort the result by the given properties.
sortfield	string/array	Possible values are: <code>graphid</code> , <code>name</code> and <code>graphtype</code> . These parameters being common for all get methods are described in detail in the reference commentary page.
countOutput	boolean	
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

ホストからグラフを取得する

ホスト "10107" から全てのグラフを取得し、名前順でソートします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graph.get",
  "params": {
    "output": "extend",
    "hostids": 10107,
    "sortfield": "name"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "graphid": "612",
      "name": "CPU jumps",
      "width": "900",
      "height": "200",
      "yaxismin": "0",

```

```

    "yaxismax": "100",
    "templateid": "439",
    "show_work_period": "1",
    "show_triggers": "1",
    "graphtype": "0",
    "show_legend": "1",
    "show_3d": "0",
    "percent_left": "0",
    "percent_right": "0",
    "ymin_type": "0",
    "ymax_type": "0",
    "ymin_itemid": "0",
    "ymax_itemid": "0",
    "flags": "0"
  },
  {
    "graphid": "613",
    "name": "CPU load",
    "width": "900",
    "height": "200",
    "yaxismin": "0",
    "yaxismax": "100",
    "templateid": "433",
    "show_work_period": "1",
    "show_triggers": "1",
    "graphtype": "0",
    "show_legend": "1",
    "show_3d": "0",
    "percent_left": "0",
    "percent_right": "0",
    "ymin_type": "1",
    "ymax_type": "0",
    "ymin_itemid": "0",
    "ymax_itemid": "0",
    "flags": "0"
  },
  {
    "graphid": "614",
    "name": "CPU utilization",
    "width": "900",
    "height": "200",
    "yaxismin": "0",
    "yaxismax": "100",
    "templateid": "387",
    "show_work_period": "1",
    "show_triggers": "0",
    "graphtype": "1",
    "show_legend": "1",
    "show_3d": "0",
    "percent_left": "0",
    "percent_right": "0",
    "ymin_type": "1",
    "ymax_type": "1",
    "ymin_itemid": "0",
    "ymax_itemid": "0",
    "flags": "0"
  },
  {
    "graphid": "645",
    "name": "Disk space usage /",
    "width": "600",
    "height": "340",

```

```

        "yaxismin": "0",
        "yaxismax": "0",
        "templateid": "0",
        "show_work_period": "0",
        "show_triggers": "0",
        "graphtype": "2",
        "show_legend": "1",
        "show_3d": "1",
        "percent_left": "0",
        "percent_right": "0",
        "ymin_type": "0",
        "ymax_type": "0",
        "ymin_itemid": "0",
        "ymax_itemid": "0",
        "flags": "4"
    }
],
    "id": 1
}

```

参照

- [Discovery rule](#)
- [Graph item](#)
- [Item](#)
- [Host](#)
- [Host group](#)
- [Template](#)

ソース

CGraph::get() in ui/include/classes/api/services/CGraph.php.

graph.update

説明

object graph.update(object/array graphs)

この方法によって、既存のグラフを更新することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 更新されるグラフのプロパティ

各グラフには graphid プロパティを定義する必要があり、その他のプロパティは任意である。渡されたプロパティのみが
更新され、その他のプロパティは変更されません。

[standard graph properties](#) に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	Graph items to replace existing graph items. If a graph item has the gitemid property defined it will be updated, otherwise a new graph item will be created.

戻り値

(object) graphids プロパティの下で更新されたグラフの ID を含むオブジェクトを返す。

例

Y スケールの最大値を設定する

Y スケールの最大値を固定値 100 に設定します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graph.update",
  "params": {
    "graphid": "439",
    "ymax_type": 1,
    "yaxismax": 100
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "graphids": [
      "439"
    ]
  },
  "id": 1
}
```

ソース

CGraph::update() in ui/include/classes/api/services/CGraph.php.

Graph item

このクラスは、グラフアイテムを操作するように設計されています。

オブジェクトのリファレンス:

- [Graph item](#)

利用可能なメソッド:

- [graphitem.get](#) - グラフアイテムの取得

> Graph item object

以下のオブジェクトは graphitem API に直接関連するものです。

Graph item

Note:

グラフアイテムは graph API を介してのみ変更することができる。

グラフアイテムオブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
gitemid	string	(readonly) ID of the graph item.
color (required)	string	Graph item's draw color as a hexadecimal color code.
itemid (required)	string	ID of the item.

プロパティ	Type	説明
calc_fnc	integer	Value of the item that will be displayed. Possible values: 1 - minimum value; 2 - (default) average value; 4 - maximum value; 7 - all values; 9 - last value, used only by pie and exploded graphs.
drawtype	integer	Draw style of the graph item. Possible values: 0 - (default) line; 1 - filled region; 2 - bold line; 3 - dot; 4 - dashed line; 5 - gradient line.
graphid	string	ID of the graph that the graph item belongs to.
sortorder	integer	Position of the item in the graph.
type	integer	Default: starts with 0 and increases by one with each entry. Type of graph item. Possible values: 0 - (default) simple; 2 - graph sum, used only by pie and exploded graphs.
yaxiside	integer	Side of the graph where the graph item's Y scale will be drawn. Possible values: 0 - (default) left side; 1 - right side.

graphitem.get

説明

integer/array graphitem.get(object parameters)

このメソッドは、与えられたパラメータに従ってグラフアイテムを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
graphids	string/array	Return only graph items that belong to the given graphs.
itemids	string/array	Return only graph items with the given item IDs.
type	integer	Return only graph items with the given type. Refer to the graph item object page for a list of supported graph item types.
selectGraphs	query	Return a graphs property with an array of graphs that the item belongs to.

パラメータ	Type	説明
sortfield	string/array	Sort the result by the given properties.
countOutput	boolean	Possible values are: gitemid. These parameters being common for all get methods are described in detail in the reference commentary page.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

グラフからグラフアイテムを取得する

グラフで使用されているすべてのグラフアイテムを、item やホストに関する追加情報とともに取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graphitem.get",
  "params": {
    "output": "extend",
    "graphids": "387"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "gitemid": "1242",
      "graphid": "387",
      "itemid": "22665",
      "drawtype": "1",
      "sortorder": "1",
      "color": "FF5555",
      "yaxisside": "0",
      "calc_fnc": "2",
      "type": "0",
      "key_": "system.cpu.util[,steal]",
      "hostid": "10001",
      "flags": "0",
      "host": "Linux"
    },
    {
      "gitemid": "1243",
      "graphid": "387",
      "itemid": "22668",
      "drawtype": "1",
      "sortorder": "2",
      "color": "55FF55",
      "yaxisside": "0",

```

```

        "calc_fnc": "2",
        "type": "0",
        "key_": "system.cpu.util[,softirq]",
        "hostid": "10001",
        "flags": "0",
        "host": "Linux"
    },
    {
        "gitemid": "1244",
        "graphid": "387",
        "itemid": "22671",
        "drawtype": "1",
        "sortorder": "3",
        "color": "009999",
        "yaxisside": "0",
        "calc_fnc": "2",
        "type": "0",
        "key_": "system.cpu.util[,interrupt]",
        "hostid": "10001",
        "flags": "0",
        "host": "Linux"
    }
],
    "id": 1
}

```

参照

- [Graph](#)

ソース

CGraphItem::get() in ui/include/classes/api/services/CGraphItem.php.

Graph prototype

このクラスは、graph prototype を扱うためのものです。

オブジェクトリファレンス:

- [Graph prototype](#)

利用可能なメソッド:

- [graphprototype.create](#) - 新しい graph prototype の作成
- [graphprototype.delete](#) - graph prototype の削除
- [graphprototype.get](#) - graph prototype を取得する
- [graphprototype.update](#) - graph prototype を更新する

> Graph prototype object

以下のオブジェクトは graphprototype API に直接関連するものである。

Graph prototype

graph prototype オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
graphid	string	(読取専用) グラフプロトタイプ ID
height (必須)	integer	グラフプロトタイプの高さ (ピクセル単位)
name (必須)	string	グラフプロトタイプ名

プロパティ	タイプ	説明
width (必須)	integer	グラフプロトタイプの幅 (ピクセル単位)
graphtype	integer	グラフプロトタイプのレイアウトタイプ 使用可能な値: 0 - (デフォルト) ノーマル 1 - 積算グラフ 2 - 円グラフ 3 - 分解円グラフ
percent_left	float	左パーセンタイルライン
percent_right	float	デフォルト: 0 右パーセンタイルライン
show_3d	integer	デフォルト: 0. ディスカバリされた円グラフと分解円グラフを 3D で表示するか 使用可能な値: 0 - (デフォルト) 2D 表示 1 - 3D 表示
show_legend	integer	ディスカバリされたグラフに凡例を表示するか 使用可能な値: 0 - 非表示 1 - (デフォルト) 表示
show_work_period	integer	ディスカバリされたグラフにワーキングタイムを表示するか 使用可能な値: 0 - 非表示 1 - (デフォルト) 表示
templateid	string	(読取専用) 親テンプレートグラフプロトタイプの ID
yaxismax	float	Y 軸の固定最大値
yaxismin	float	Y 軸の固定最小値
ymax_itemid	string	Y 軸の最大値として使用されるアイテムの ID
ymax_type	integer	Zabbix 6.0.7 以降、ユーザーが特定のアイテムにアクセスできない場合、グラフは ymax_type が '0' (計算済み) に設定されるようにレンダリングされます。 Y 軸の最大値計算方法 使用可能な値: 0 - (デフォルト) 計算 1 - 固定 2 - アイテム
ymin_itemid	string	Y 軸の最小値として使用されるアイテムの ID
ymin_type	integer	Zabbix 6.0.7 以降、ユーザーが特定のアイテムにアクセスできない場合、グラフは ymin_type が '0' (計算済み) に設定されるようにレンダリングされます。 Y 軸の最小値計算方法 使用可能な値: 0 - (デフォルト) 計算 1 - 固定 2 - アイテム
discover	integer	グラフプロトタイプのディスカバリステータス。 使用可能な値: 0 - (デフォルト) 新しいグラフが検出されます。 1 - 新しいグラフは検出されず、既存のグラフは失われたものとしてマークされます。

プロパティ	タイプ	説明
uuid	string	インポートされたグラフプロトタイプを既存のものにリンクするために使用されるユニバーサル一意識別子。テンプレートのグラフプロトタイプにのみ使用されます。指定されていない場合は自動生成されます。 更新操作の場合、このフィールドは読み取り専用です。

graphprototype.create

説明

object graphprototype.create(object/array graphPrototypes)

このメソッドに graph prototypes を作成することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) Graph prototypes を作成する。

standard graph prototype propertiesに加えて、このメソッドは以下のパラメータを受け付ける。

パラメータ	Type	説明
gitems (required)	array	Graph items to be created for the graph prototypes. Graph items can reference both items and item prototypes, but at least one item prototype must be present.

戻り値

(object) graphids プロパティの下に作成されたグラフプロトタイプの ID を含むオブジェクトを返します。
返された ID の順番は、渡されたグラフプロトタイプの順番と一致する。

例

graph prototype の作成

2つの項目からなる graph prototype を作成する。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graphprototype.create",
  "params": {
    "name": "Disk space usage {#FSNAME}",
    "width": 900,
    "height": 200,
    "gitems": [
      {
        "itemid": "22828",
        "color": "00AA00"
      },
      {
        "itemid": "22829",
        "color": "3333FF"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "graphids": [
      "652"
    ]
  },
  "id": 1
}
```

参照

- [Graph item](#)

ソース

CGraphPrototype::create() in ui/include/classes/api/services/CGraphPrototype.php.

graphprototype.delete

説明

object graphprototype.delete(array graphPrototypeIds)

graph prototypes を削除することができるメソッドです。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。詳細については、[User roles](#)を参照してください。

パラメータ

(array) 削除する graph prototypes の ID

戻り値

(object) graphids プロパティの下で削除された graph prototypes の ID を含むオブジェクトを返します。

例

複数の graph prototype を削除する

2つの graph prototype を削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graphprototype.delete",
  "params": [
    "652",
    "653"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "graphids": [
      "652",
      "653"
    ]
  },
}
```

```
"id": 1  
}
```

ソース

CGraphPrototype::delete() in ui/include/classes/api/services/CGraphPrototype.php.

graphprototype.get

説明

integer/array graphprototype.get(object parameters)

このメソッドは、与えられたパラメータに従って、グラフの graph prototypes を取得することができます

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
 詳しくは、 [User roles](#) を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
discoveryids	string/array	Return only graph prototypes that belong to the given discovery rules.
graphids	string/array	Return only graph prototypes with the given IDs.
groupids	string/array	Return only graph prototypes that belong to hosts in the given host groups.
hostids	string/array	Return only graph prototypes that belong to the given hosts.
inherited	boolean	If set to true return only graph prototypes inherited from a template.
itemids	string/array	Return only graph prototypes that contain the given item prototypes.
templated	boolean	If set to true return only graph prototypes that belong to templates.
templateids	string/array	Return only graph prototypes that belong to the given templates.
selectDiscoveryRule	query	Return a discoveryRule property with the LLD rule that the graph prototype belongs to.
selectGraphItems	query	Return a gitems property with the graph items used in the graph prototype.
selectGroups	query	Return a groups property with the host groups that the graph prototype belongs to.
selectHosts	query	Return a hosts property with the hosts that the graph prototype belongs to.
selectItems	query	Return an items property with the items and item prototypes used in the graph prototype.
selectTemplates	query	Return a templates property with the templates that the graph prototype belongs to.
filter	object	Return only those results that exactly match the given filter. Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against. Supports additional filters: host - technical name of the host that the graph prototype belongs to; hostid - ID of the host that the graph prototype belongs to. Sort the result by the given properties.
sortfield	string/array	Possible values are: graphid , name and graphtype . These parameters being common for all get methods are described in detail in the reference commentary .
countOutput	boolean	
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	

パラメータ	Type	説明
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のいずれかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

LLD ルールから graph prototypes を取得する

LLD ルールからすべての graph prototypes を取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graphprototype.get",
  "params": {
    "output": "extend",
    "discoveryids": "27426"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "graphid": "1017",
      "parent_itemid": "27426",
      "name": "Disk space usage {#FSNAME}",
      "width": "600",
      "height": "340",
      "yaxismin": "0.0000",
      "yaxismax": "0.0000",
      "templateid": "442",
      "show_work_period": "0",
      "show_triggers": "0",
      "graphtype": "2",
      "show_legend": "1",
      "show_3d": "1",
      "percent_left": "0.0000",
      "percent_right": "0.0000",
      "ymin_type": "0",
      "ymax_type": "0",
      "ymin_itemid": "0",
      "ymax_itemid": "0",
      "discover": "0"
    }
  ],
  "id": 1
}
```

LLD ルールから graph prototypes を取得する

LLD ルールからすべての graph prototypes を取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graphprototype.get",
  "params": {
    "output": "extend",
    "discoveryids": "27426"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "graphid": "1017",
      "parent_itemid": "27426",
      "name": "Disk space usage {#FSNAME}",
      "width": "600",
      "height": "340",
      "yaxismin": "0.0000",
      "yaxismax": "0.0000",
      "templateid": "442",
      "show_work_period": "0",
      "show_triggers": "0",
      "graphtype": "2",
      "show_legend": "1",
      "show_3d": "1",
      "percent_left": "0.0000",
      "percent_right": "0.0000",
      "ymin_type": "0",
      "ymax_type": "0",
      "ymin_itemid": "0",
      "ymax_itemid": "0",
      "discover": "0"
    }
  ],
  "id": 1
}
```

参照

- [Discovery rule](#)
- [Graph item](#)
- [Item](#)
- [Host](#)
- [Host group](#)
- [Template](#)

ソース

CGraphPrototype::get() in ui/include/classes/api/services/CGraphPrototype.php.

graphprototype.update

説明

object graphprototype.update(object/array graphPrototypes)

この方法により、既存の graph prototypes を更新することができる。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、ユーザーロール設定で
 取り消すことができます。
 詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 更新する Graph prototype プロパティ

graph prototype ごとに graphid プロパティを定義する必要がある。他のプロパティはすべてオプションである。
 渡されたプロパティのみが更新され、他のプロパティは変更されません。

standard graph prototype properties に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	Graph items to replace existing graph items. If a graph item has the gitemid property defined it will be updated, otherwise a new graph item will be created.

戻り値

(object) graphids プロパティの下で更新された graph prototypes の ID を含むオブジェクトを返します。

例

graph prototype のサイズを変更する

graph prototype のサイズを 1100 から 400 ピクセルに変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "graphprototype.update",
  "params": {
    "graphid": "439",
    "width": 1100,
    "height": 400
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "graphids": [
      "439"
    ]
  },
  "id": 1
}
```

ソース

CGraphPrototype::update() in ui/include/classes/api/services/CGraphPrototype.php.

High availability node

このクラスは、高可用性クラスターの一部であるサーバーノード、またはスタンドアロンのサーバーインスタンスで動作するように
 設計されています。

オブジェクトリファレンス:

- [High availability node](#)

利用可能なメソッド:

- `hanode.get` - ノード検索

> High availability node object

以下のオブジェクトは、Zabbix server の HA (高可用性) クラスターの運用に関連するものです。

High availability node

Note:

ノードは Zabbix server によって作成され、API から変更することはできません。

High availability node object は、以下のプロパティを持ちます。

プロパティ	Type	説明
ha_nodeid	string	ID of the node.
name	string	Name assigned to the node, using the HANodeName configuration entry of zabbix_server.conf. Empty for a server running in standalone mode.
address	string	IP or DNS name where the node connects from.
port	integer	Port on which the node is running.
lastaccess	integer	Heartbeat time, t.i. time of last update from the node. UTC timestamp.
status	integer	State of the node. Possible values: 0 - standby; 1 - stopped manually; 2 - unavailable; 3 - active.

hanode.get

説明

`integer/array hanode.get(object parameters)`

このメソッドは、指定されたパラメーターに従って High availability cluster node のリストを取得することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
詳細については、[User roles](#)を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
ha_nodeids	string/array	Return only nodes with the given node IDs.
filter	object	Return only those results that exactly match the given filter. Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against.
sortfield	string/array	Allows filtering by the node properties: name, address, status. Sort the result by the given properties. Possible values are: name, lastaccess, status.

パラメータ	Type	説明
countOutput	flag	These parameters being common for all get methods are described in detail in the reference commentary .
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(integer/array) のいずれかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ノードのリストを状態順に取得

リクエスト :

```
{
  "jsonrpc": "2.0",
  "method": "hanode.get",
  "params": {
    "preservekeys": true,
    "sortfield": "status",
    "sortorder": "DESC"
  },
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

レスポンス :

```
{
  "jsonrpc": "2.0",
  "result": {
    "ckuo7i1nw000h0sajj3l3hh8u": {
      "ha_nodeid": "ckuo7i1nw000h0sajj3l3hh8u",
      "name": "node-active",
      "address": "192.168.1.13",
      "port": "10051",
      "lastaccess": "1635335704",
      "status": "3"
    },
    "ckuo7i1nw000e0sajwfttc1mp": {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    },
    "ckuo7i1nv000c0sajz85xcrtt": {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    "ckuo7i1nv000a0saj1fcdkeu4": {
      "ha_nodeid": "ckuo7i1nv000a0saj1fcdkeu4",
      "name": "node2",

```

```
    "address": "192.168.1.6",
    "port": "10051",
    "lastaccess": "1635335705",
    "status": "0"
  }
},
"id": 1
}
```

特定のノードの ID によるリストの取得

リクエスト :

```
{
  "jsonrpc": "2.0",
  "method": "hanode.get",
  "params": {
    "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
  },
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

レスポンス :

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    }
  ],
  "id": 1
}
```

停止しているノードのリストを取得

リクエスト :

```
{
  "jsonrpc": "2.0",
  "method": "hanode.get",
  "params": {
    "output": ["ha_nodeid", "address", "port"],
    "filter": {
      "status": 1
    }
  },
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

レスポンス :

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "ha_nodeid": "ckuo7i1nw000g0sajjsjre7e3",
      "address": "192.168.1.12",
      "port": "10051"
    },
    {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "address": "192.168.1.8",
      "port": "10052"
    },
    {
      "ha_nodeid": "ckuo7i1nv000d0sajd95y1b6x",
      "address": "192.168.1.9",
      "port": "10053"
    }
  ],
  "id": 1
}
```

待機ノード数の取得

リクエスト :

```
{
  "jsonrpc": "2.0",
  "method": "hanode.get",
  "params": {
    "countOutput": true,
    "filter": {
      "status": 0
    }
  },
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

レスポンス :

```
{
  "jsonrpc": "2.0",
  "result": "3",
  "id": 1
}
```

特定の IP アドレスのノードの状態を確認

リクエスト :

```
{
  "jsonrpc": "2.0",
  "method": "hanode.get",
  "params": {
    "output": ["name", "status"],
    "filter": {
      "address": ["192.168.1.7", "192.168.1.13"]
    }
  },
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

レスポンス :

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "name": "node3",
      "status": "0"
    },
    {
      "name": "node-active",
      "status": "3"
    }
  ],
  "id": 1
}

```

ソース

CHaNode::get() in ui/include/classes/api/services/CHaNode.php.

History

このクラスは、history データを扱うためのものです。

オブジェクトリファレンス:

- [History](#)

利用可能なメソッド:

- [history.get](#) - history データの検索

> History object

以下のオブジェクトは history API に直接関連するものです。

Note:

history オブジェクトは item の情報の種類によって異なります。これらは Zabbix server によって作成され、API から変更することはできません。

Float history

float history オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
clock	timestamp	Time when that value was received.
itemid	string	ID of the related item.
ns	integer	Nanoseconds when the value was received.
value	float	Received value.

Integer history

Integer history オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
clock	timestamp	Time when that value was received.
itemid	string	ID of the related item.
ns	integer	Nanoseconds when the value was received.
value	integer	Received value.

String history

String history オブジェクトは、以下のプロパティを持つ。

プロパティ	Type	説明
clock	timestamp	Time when that value was received.
itemid	string	ID of the related item.
ns	integer	Nanoseconds when the value was received.
value	string	Received value.

Text history

Text history オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
id	string	ID of the history entry.
clock	timestamp	Time when that value was received.
itemid	string	ID of the related item.
ns	integer	Nanoseconds when the value was received.
value	text	Received value.

Log history

Log history オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
id	string	ID of the history entry.
clock	timestamp	Time when that value was received.
itemid	string	ID of the related item.
logeventid	integer	Windows event log entry ID.
ns	integer	Nanoseconds when the value was received.
severity	integer	Windows event log entry level.
source	string	Windows event log entry source.
timestamp	timestamp	Windows event log entry time.
value	text	Received value.

history.clear

説明

```
object history.clear(array itemids)
```

item history を消去するメソッドです。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
 詳細については、[User roles](#)を参照してください。

パラメータ

(array) クリアするアイテムの ID 配列

戻り値

(object) itemids プロパティの下にある、クリアされた item の ID を含むオブジェクトを返す。

例

history の消去

Request:

```
{
  "jsonrpc": "2.0",
  "method": "history.clear",
  "params": [
    "10325",
    "13205"
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "10325",
      "13205"
    ]
  },
  "id": 1
}
```

ソース

CHistory::clear() in ui/include/classes/api/services/CHistory.php.

history.get

説明

integer/array history.get(object parameters)

このメソッドは、与えられたパラメータに従って history データを取得することができます。

参照: [known issues](#)

Attention:

このメソッドは、削除されたエンティティの history データがまだ housekeeper によって削除されていない場合、そのデータを
 返すことがあります。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
 詳しくは、[User roles](#)を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、以下のパラメータをサポートしています。

パラメータ	Type	説明
history	integer	History object types to return. Possible values: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text. Default: 3.
hostids	string/array	Return only history from the given hosts.
itemids	string/array	Return only history from the given items.

パラメータ	Type	説明
time_from	timestamp	Return only values that have been received after or at the given time.
time_till	timestamp	Return only values that have been received before or at the given time.
sortfield	string/array	Sort the result by the given properties.
countOutput	boolean	Possible values are: itemid and clock. These parameters being common for all get methods are described in detail in the reference commentary page.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のいずれかを返します:

- オブジェクトの配列
- もし countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

item の history データを取得する

numeric(float) item から受け取った最新の値を 10 件返します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "history.get",
  "params": {
    "output": "extend",
    "history": 0,
    "itemids": "23296",
    "sortfield": "clock",
    "sortorder": "DESC",
    "limit": 10
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "23296",
      "clock": "1351090996",
      "value": "0.085",
      "ns": "563157632"
    },
    {
      "itemid": "23296",
      "clock": "1351090936",
      "value": "0.16",
      "ns": "549216402"
    }
  ],
}
```

```

    {
      "itemid": "23296",
      "clock": "1351090876",
      "value": "0.18",
      "ns": "537418114"
    },
    {
      "itemid": "23296",
      "clock": "1351090816",
      "value": "0.21",
      "ns": "522659528"
    },
    {
      "itemid": "23296",
      "clock": "1351090756",
      "value": "0.215",
      "ns": "507809457"
    },
    {
      "itemid": "23296",
      "clock": "1351090696",
      "value": "0.255",
      "ns": "495509699"
    },
    {
      "itemid": "23296",
      "clock": "1351090636",
      "value": "0.36",
      "ns": "477708209"
    },
    {
      "itemid": "23296",
      "clock": "1351090576",
      "value": "0.375",
      "ns": "463251343"
    },
    {
      "itemid": "23296",
      "clock": "1351090516",
      "value": "0.315",
      "ns": "447947017"
    },
    {
      "itemid": "23296",
      "clock": "1351090456",
      "value": "0.275",
      "ns": "435307141"
    }
  ],
  "id": 1
}

```

ソース

CHistory::get() in ui/include/classes/api/services/CHistory.php.

Host

このクラスは、ホストとの連携を想定しています。

オブジェクトリファレンス:

- [Host](#)

- **Host inventory**

利用可能なメソッド:

- **host.create** - 新しいホストの作成
- **host.delete** - ホストの削除
- **host.get** - ホストの検索
- **host.massadd** - ホストに関連オブジェクトを追加する
- **host.massremove** - ホストから関連オブジェクトを削除する
- **host.massupdate** - ホストから関連するオブジェクトを交換または削除する
- **host.update** - ホストの更新

> Host object

以下のオブジェクトは、ホスト API に直接関連するものです。

Host

host オブジェクトは以下のプロパティを持ちます。

プロパティ	Type	説明
hostid	string	(readonly) ID of the host.
host (required)	string	Technical name of the host.
description	text	Description of the host.
flags	integer	(readonly) Origin of the host. Possible values: 0 - a plain host; 4 - a discovered host.
inventory_mode	integer	Host inventory population mode. Possible values are: -1 - (default) disabled; 0 - manual; 1 - automatic.
ipmi_authtype	integer	IPMI authentication algorithm. Possible values are: -1 - (default) default; 0 - none; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+.
ipmi_password	string	IPMI password.
ipmi_privilege	integer	IPMI privilege level. Possible values are: 1 - callback; 2 - (default) user; 3 - operator; 4 - admin; 5 - OEM.
ipmi_username	string	IPMI username.
maintenance_from	timestamp	(readonly) Starting time of the effective maintenance.
maintenance_status	integer	(readonly) Effective maintenance status. Possible values are: 0 - (default) no maintenance; 1 - maintenance in effect.

プロパティ	Type	説明
maintenance_type	integer	(readonly) Effective maintenance type. Possible values are: 0 - (default) maintenance with data collection; 1 - maintenance without data collection.
maintenanceid	string	(readonly) ID of the maintenance that is currently in effect on the host.
name	string	Visible name of the host.
proxy_hostid	string	Default: host property value. ID of the proxy that is used to monitor the host.
status	integer	Status and function of the host. Possible values are: 0 - (default) monitored host; 1 - unmonitored host.
tls_connect	integer	Connections to host. Possible values are: 1 - (default) No encryption; 2 - PSK; 4 - certificate.
tls_accept	integer	Connections from host. Possible bitmap values are: 1 - (default) No encryption; 2 - PSK; 4 - certificate.
tls_issuer	string	Certificate issuer.
tls_subject	string	Certificate subject.
tls_psk_identity	string	(write-only) PSK identity. Required if either <code>tls_connect</code> or <code>tls_accept</code> has PSK enabled. Do not put sensitive information in the PSK identity, it is transmitted unencrypted over the network to inform a receiver which PSK to use.
tls_psk	string	(write-only) The preshared key, at least 32 hex digits. Required if either <code>tls_connect</code> or <code>tls_accept</code> has PSK enabled.

Host インベントリ

Host インベントリオブジェクトは、以下のプロパティを持ちます。

Note:

各プロパティには固有の ID 番号があり、ホストインベントリフィールドと item の関連付けに使用されます。

ID	プロパティ	Type	説明
4	alias	string	Alias.
11	asset_tag	string	Asset tag.
28	chassis	string	Chassis.
23	contact	string	Contact person.
32	contract_number	string	Contract number.
47	date_hw_decomm	string	HW decommissioning date.
46	date_hw_expiry	string	HW maintenance expiry date.
45	date_hw_install	string	HW installation date.
44	date_hw_purchase	string	HW purchase date.
34	deployment_status	string	Deployment status.
14	hardware	string	Hardware.
15	hardware_full	string	Detailed hardware.
39	host_netmask	string	Host subnet mask.
38	host_networks	string	Host networks.
40	host_router	string	Host router.
30	hw_arch	string	HW architecture.

ID	プロパティ	Type	説明
33	installer_name	string	Installer name.
24	location	string	Location.
25	location_lat	string	Location latitude.
26	location_lon	string	Location longitude.
12	macaddress_a	string	MAC address A.
13	macaddress_b	string	MAC address B.
29	model	string	Model.
3	name	string	Name.
27	notes	string	Notes.
41	oob_ip	string	OOB IP address.
42	oob_netmask	string	OOB host subnet mask.
43	oob_router	string	OOB router.
5	os	string	OS name.
6	os_full	string	Detailed OS name.
7	os_short	string	Short OS name.
61	poc_1_cell	string	Primary POC mobile number.
58	poc_1_email	string	Primary email.
57	poc_1_name	string	Primary POC name.
63	poc_1_notes	string	Primary POC notes.
59	poc_1_phone_a	string	Primary POC phone A.
60	poc_1_phone_b	string	Primary POC phone B.
62	poc_1_screen	string	Primary POC screen name.
68	poc_2_cell	string	Secondary POC mobile number.
65	poc_2_email	string	Secondary POC email.
64	poc_2_name	string	Secondary POC name.
70	poc_2_notes	string	Secondary POC notes.
66	poc_2_phone_a	string	Secondary POC phone A.
67	poc_2_phone_b	string	Secondary POC phone B.
69	poc_2_screen	string	Secondary POC screen name.
8	serialno_a	string	Serial number A.
9	serialno_b	string	Serial number B.
48	site_address_a	string	Site address A.
49	site_address_b	string	Site address B.
50	site_address_c	string	Site address C.
51	site_city	string	Site city.
53	site_country	string	Site country.
56	site_notes	string	Site notes.
55	site_rack	string	Site rack location.
52	site_state	string	Site state.
54	site_zip	string	Site ZIP/postal code.
16	software	string	Software.
18	software_app_a	string	Software application A.
19	software_app_b	string	Software application B.
20	software_app_c	string	Software application C.
21	software_app_d	string	Software application D.
22	software_app_e	string	Software application E.
17	software_full	string	Software details.
10	tag	string	Tag.
1	type	string	Type.
2	type_full	string	Type details.
35	url_a	string	URL A.
36	url_b	string	URL B.
37	url_c	string	URL C.
31	vendor	string	Vendor.

Host tag

Host tag オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
tag (required)	string	Host tag name.
value	string	Host tag value.

host.create

説明

object host.create(object/array hosts)

このメソッドでは、新しいホストを作成することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
 詳細については、[User roles](#)を参照してください。

パラメータ

(object/array) 作成するホスト

standard host propertiesに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
groups (required)	object/array	Host groups to add the host to. The host groups must have the groupid property defined.
interfaces	object/array	Interfaces to be created for the host.
tags	object/array	Host tags .
templates	object/array	Templates to be linked to the host. The templates must have the templateid property defined.
macros	object/array	User macros to be created for the host.
inventory	object	Host inventory properties.

戻り値

(object) **hostids** プロパティの下に生成されたホストの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストの
 順序と一致します。

例

ホストの作成

IP インターフェースとタグを持つ "Linux server" というホストを作成し、グループに追加し、テンプレートをリンクし、
 ホストインベントリに MAC アドレスを設定します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "Linux server",
    "interfaces": [
      {
        "type": 1,
        "main": 1,
        "useip": 1,
        "ip": "192.168.3.1",
        "dns": "",
        "port": "10050"
      }
    ]
  },
}
```

```

    "groups": [
      {
        "groupid": "50"
      }
    ],
    "tags": [
      {
        "tag": "Host name",
        "value": "Linux server"
      }
    ],
    "templates": [
      {
        "templateid": "20045"
      }
    ],
    "macros": [
      {
        "macro": "${USER_ID}",
        "value": "123321"
      },
      {
        "macro": "${USER_LOCATION}",
        "value": "0:0:0",
        "description": "latitude, longitude and altitude coordinates"
      }
    ],
    "inventory_mode": 0,
    "inventory": {
      "macaddress_a": "01234",
      "macaddress_b": "56768"
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "107819"
    ]
  },
  "id": 1
}

```

SNMP インターフェイスを持つホストの作成

SNMPv3 インタフェースを持つホスト“SNMP host”を詳細とともに作成します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "SNMP host",
    "interfaces": [
      {
        "type": 2,
        "main": 1,
        "useip": 1,

```

```

        "ip": "127.0.0.1",
        "dns": "",
        "port": "161",
        "details": {
            "version": 3,
            "bulk": 0,
            "securityname": "mysecurityname",
            "contextname": "",
            "securitylevel": 1
        }
    }
],
"groups": [
    {
        "groupid": "4"
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}

```

Creating a host with PSK encryption

Create a host called "PSK host" with PSK encryption configured. Note that the host has to be **pre-configured to use PSK**.

Request:

```

{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "PSK host",
        "interfaces": [
            {
                "type": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050",
                "useip": 1,
                "main": 1
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "tls_accept": 2,
        "tls_connect": 2,
        "tls_psk_identity": "PSK 001",
        "tls_psk": "1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952"
    }
},

```



```
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10590"
    ]
  },
  "id": 1
}
```

参照

- [Host group](#)
- [Template](#)
- [User macro](#)
- [Host interface](#)
- [Host inventory](#)
- [Host tag](#)

ソース

CHost::create() in ui/include/classes/api/services/CHost.php.

host.delete

説明

object host.delete(array hosts)

このメソッドでは、ホストを削除することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
詳細については、[User roles](#)を参照してください。

パラメータ

(array) 削除するホストの ID 配列

戻り値

(object) hostids プロパティの下にある、削除されたホストの ID を含むオブジェクトを返します。

例

複数のホストを削除する

2つのホストを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.delete",
  "params": [
    "13",
    "32"
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "13",
      "32"
    ]
  },
  "id": 1
}
```

ソース

CHost::delete() in ui/include/classes/api/services/CHost.php.

host.get

説明

integer/array host.get(object parameters)

このメソッドは、与えられたパラメータにしたがってホストを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーロール設定で取り消すことができます。
 詳しくは、 [User roles](#) を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
groupids	string/array	Return only hosts that belong to the given groups.
dserviceids	string/array	Return only hosts that are related to the given discovered services.
graphids	string/array	Return only hosts that have the given graphs.
hostids	string/array	Return only hosts with the given host IDs.
httptestids	string/array	Return only hosts that have the given web checks.
interfaceids	string/array	Return only hosts that use the given interfaces.
itemids	string/array	Return only hosts that have the given items.
maintenanceids	string/array	Return only hosts that are affected by the given maintenances.
monitored_hosts	flag	Return only monitored hosts.
proxy_hosts	flag	Return only proxies.
proxyids	string/array	Return only hosts that are monitored by the given proxies.
templated_hosts	flag	Return both hosts and templates.
templateids	string/array	Return only hosts that are linked to the given templates.
triggerids	string/array	Return only hosts that have the given triggers.
with_items	flag	Return only hosts that have items.
with_item_prototypes	flag	Overrides the with_monitored_items and with_simple_graph_items parameters. Return only hosts that have item prototypes.
with_simple_graph_item_prototypes	flag	Overrides the with_simple_graph_item_prototypes parameter. Return only hosts that have item prototypes, which are enabled for creation and have numeric type of information.
with_graphs	flag	Return only hosts that have graphs.
with_graph_prototypes	flag	Return only hosts that have graph prototypes.
with_httptests	flag	Return only hosts that have web checks.
with_monitored_httptests	flag	Overrides the with_monitored_httptests parameter. Return only hosts that have enabled web checks.

パラメータ	Type	説明
with_monitored_items	flag	Return only hosts that have enabled items.
with_monitored_triggers	flag	Overrides the <code>with_simple_graph_items</code> parameter. Return only hosts that have enabled triggers. All of the items used in the trigger must also be enabled.
with_simple_graph_items	flag	Return only hosts that have items with numeric type of information.
with_triggers	flag	Return only hosts that have triggers.
withProblemsSuppressed	boolean	Overrides the <code>with_monitored_triggers</code> parameter. Return hosts that have suppressed problems. Possible values: <code>null</code> - (default) all hosts; <code>true</code> - only hosts with suppressed problems; <code>false</code> - only hosts with unsuppressed problems.
evaltype	integer	Rules for tag searching. Possible values: 0 - (default) And/Or; 2 - Or.
severities	integer/array	Return hosts that have only problems with given severities. Applies only if problem object is trigger.
tags	array/object	Return only hosts with given tags. Exact match by tag and case-sensitive or case-insensitive search by tag value depending on operator value. Format: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]. An empty array returns all hosts. Possible operator values: 0 - (default) Contains; 1 - Equals; 2 - Not like; 3 - Not equal 4 - Exists; 5 - Not exists.
inheritedTags	boolean	Return hosts that have given tags also in all of their linked templates. Default: Possible values: <code>true</code> - linked templates must also have given tags; <code>false</code> - (default) linked template tags are ignored.
selectDiscoveries	query	Return a <code>discoveries</code> property with host low-level discovery rules.
selectDiscoveryRule	query	Supports count. Return a <code>discoveryRule</code> property with the low-level discovery rule that created the host (from host prototype in VMware monitoring).
selectGraphs	query	Return a <code>graphs</code> property with host graphs.
selectGroups	query	Supports count. Return a <code>groups</code> property with host groups data that the host belongs to.

パラメータ	Type	説明
selectHostDiscovery	query	Return a <code>hostDiscovery</code> property with host discovery object data. The host discovery object links a discovered host to a host prototype or a host prototypes to an LLD rule and has the following properties: <code>host</code> - (string) host of the host prototype; <code>hostid</code> - (string) ID of the discovered host or host prototype; <code>parent_hostid</code> - (string) ID of the host prototype from which the host has been created; <code>parent_itemid</code> - (string) ID of the LLD rule that created the discovered host; <code>lastcheck</code> - (timestamp) time when the host was last discovered; <code>ts_delete</code> - (timestamp) time when a host that is no longer discovered will be deleted.
selectHttpTests	query	Return an <code>httpTests</code> property with host web scenarios.
selectInterfaces	query	Supports count. Return an <code>interfaces</code> property with host interfaces.
selectInventory	query	Supports count. Return an <code>inventory</code> property with host inventory data.
selectItems	query	Return an <code>items</code> property with host items.
selectMacros	query	Supports count. Return a <code>macros</code> property with host macros.
selectParentTemplates	query	Return a <code>parentTemplates</code> property with templates that the host is linked to.
selectDashboards	query	Supports count. Return a <code>dashboards</code> property.
selectTags	query	Supports count. Return a <code>tags</code> property with host tags.
selectInheritedTags	query	Return an <code>inheritedTags</code> property with tags that are on all templates which are linked to host.
selectTriggers	query	Return a <code>triggers</code> property with host triggers.
selectValueMaps	query	Supports count. Return a <code>valuemaps</code> property with host value maps.
filter	object	Return only those results that exactly match the given filter. Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against.
limitSelects	integer	Allows filtering by interface properties. Limits the number of records returned by subselects. Applies to the following subselects: <code>selectParentTemplates</code> - results will be sorted by <code>host</code> ; <code>selectInterfaces</code> ; <code>selectItems</code> - sorted by <code>name</code> ; <code>selectDiscoveries</code> - sorted by <code>name</code> ; <code>selectTriggers</code> - sorted by <code>description</code> ; <code>selectGraphs</code> - sorted by <code>name</code> ; <code>selectDashboards</code> - sorted by <code>name</code> .
search	object	Return results that match the given wildcard search. Accepts an array, where the keys are property names, and the values are strings to search for. If no additional options are given, this will perform a LIKE "%...%" search. Allows searching by interface properties. Works only with text fields.

パラメータ	Type	説明
searchInventory	object	Return only hosts that have inventory data matching the given wildcard search.
sortfield	string/array	This parameter is affected by the same additional parameters as <code>search</code> . Sort the result by the given properties.
countOutput	boolean	Possible values are: <code>hostid</code> , <code>host</code> , <code>name</code> , <code>status</code> . These parameters being common for all get methods are described in detail in the reference commentary .
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のいずれかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

名前によるデータの取得

"Zabbix server" と "Linux server" という名前の 2 つのホストに関するすべてのデータを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "filter": {
      "host": [
        "Zabbix server",
        "Linux server"
      ]
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10160",
      "proxy_hostid": "0",
      "host": "Zabbix server",
      "status": "0",
      "lastaccess": "0",
      "ipmi_authtype": "-1",
      "ipmi_privilege": "2",
      "ipmi_username": "",
      "ipmi_password": "",

```

```

        "maintenanceid": "0",
        "maintenance_status": "0",
        "maintenance_type": "0",
        "maintenance_from": "0",
        "name": "Zabbix server",
        "flags": "0",
        "description": "The Zabbix monitoring server.",
        "tls_connect": "1",
        "tls_accept": "1",
        "tls_issuer": "",
        "tls_subject": "",
        "inventory_mode": "1"
    },
    {
        "hostid": "10167",
        "proxy_hostid": "0",
        "host": "Linux server",
        "status": "0",
        "lastaccess": "0",
        "ipmi_authtype": "-1",
        "ipmi_privilege": "2",
        "ipmi_username": "",
        "ipmi_password": "",
        "maintenanceid": "0",
        "maintenance_status": "0",
        "maintenance_type": "0",
        "maintenance_from": "0",
        "name": "Linux server",
        "flags": "0",
        "description": "",
        "tls_connect": "1",
        "tls_accept": "1",
        "tls_issuer": "",
        "tls_subject": "",
        "inventory_mode": "1"
    }
],
    "id": 1
}

```

ホストグループの取得

ホスト“Zabbix server”が所属するグループ名を取得しますが、ホスト自体の詳細は取得しません。

Request:

```

{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 2
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10085",
      "groups": [
        {
          "groupid": "2",
          "name": "Linux servers",
          "internal": "0",
          "flags": "0"
        },
        {
          "groupid": "4",
          "name": "Zabbix servers",
          "internal": "0",
          "flags": "0"
        }
      ]
    }
  ],
  "id": 2
}

```

リンク先テンプレートの取得

ホスト"10084"にリンクしているテンプレートの ID および名前を取得します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["hostid"],
    "selectParentTemplates": [
      "templateid",
      "name"
    ],
    "hostids": "10084"
  },
  "id": 1,
  "auth": "70785d2b494a7302309b48afcdb3a401"
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "parentTemplates": [
        {
          "name": "Linux",
          "templateid": "10001"
        },
        {
          "name": "Zabbix Server",
          "templateid": "10047"
        }
      ]
    }
  ],
  "id": 1
}

```

```
}
```

Retrieving hosts by template

Retrieve hosts that have the "10001" (Linux by Zabbix agent) template linked to them.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["hostid", "name"],
    "templateids": "10001"
  },
  "auth": "70785d2b494a7302309b48afcdb3a401",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "templateid": "10001",
      "hosts": [
        {
          "hostid": "10084",
          "name": "Zabbix server"
        },
        {
          "hostid": "10603",
          "name": "Host 1"
        },
        {
          "hostid": "10604",
          "name": "Host 2"
        }
      ]
    }
  ],
  "id": 1
}
```

ホストインベントリのデータで検索する

ホストインベントリの"OS" フィールドに"Linux" が含まれるホストを検索します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": [
      "host"
    ],
    "selectInventory": [
      "os"
    ],
    "searchInventory": {
      "os": "Linux"
    }
  },
  "id": 2,
  "auth": "7f9e00124c75e8f25facd5c093f3e9a0"
}
```



```
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "host": "Zabbix server",
      "inventory": {
        "os": "Linux Ubuntu"
      }
    },
    {
      "hostid": "10107",
      "host": "Linux server",
      "inventory": {
        "os": "Linux Mint"
      }
    }
  ],
  "id": 1
}
```

ホストタグで検索

タグ"Host name" が"Linux server" と等しいホストを検索します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["hostid"],
    "selectTags": "extend",
    "evaltype": 0,
    "tags": [
      {
        "tag": "Host name",
        "value": "Linux server",
        "operator": 1
      }
    ]
  },
  "auth": "7f9e00124c75e8f25facd5c093f3e9a0",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10085",
      "tags": [
        {
          "tag": "Host name",
          "value": "Linux server"
        },
        {
          "tag": "OS",
          "value": "RHEL 7"
        }
      ]
    }
  ]
}
```

```
    ]
  }
],
"id": 1
}
```

Retrieve hosts that have these tags not only on host level but also in their linked parent templates.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["name"],
    "tags": [{"tag": "A", "value": "1", "operator": "0"}],
    "inheritedTags": true
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10623",
      "name": "PC room 1"
    },
    {
      "hostid": "10601",
      "name": "Office"
    }
  ],
  "id": 1
}
```

問題の深刻度によるホストの検索

問題の深刻度が"Disaster"のホストを検索します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["name"],
    "severities": 5
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10160",
      "name": "Zabbix server"
    }
  ],
  "id": 1
}
```

```
}
```

“Average” と “High” の問題を持つホストを取得する。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["name"],
    "severities": [3, 4]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "20170",
      "name": "Database"
    },
    {
      "hostid": "20183",
      "name": "workstation"
    }
  ],
  "id": 1
}
```

問題の深刻度によるホストの検索

問題の深刻度が “Disaster” のホストを検索します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["name"],
    "severities": 5
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10160",
      "name": "Zabbix server"
    }
  ],
  "id": 1
}
```

“Average” と “High” の問題を持つホストを取得する。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["name"],
    "severities": [3, 4]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "20170",
      "name": "Database"
    },
    {
      "hostid": "20183",
      "name": "workstation"
    }
  ],
  "id": 1
}
```

参照

- [Host group](#)
- [Template](#)
- [User macro](#)
- [Host interface](#)

ソース

`CHost::get()` in `ui/include/classes/api/services/CHost.php`.

host.massadd

説明

`object host.massadd(object parameters)`

このメソッドでは、与えられたすべてのホストに、複数の関連オブジェクトを同時に追加することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、`
` ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object) 更新するホストの ID、および全てのホストに追加するオブジェクトを含むパラメータ

このメソッドは、以下のパラメータを受け付けます。

パラメータ	Type	説明
hosts (required)	object/array	Hosts to be updated. The hosts must have the <code>hostid</code> property defined.
groups	object/array	Host groups to add to the given hosts. The host groups must have the <code>groupid</code> property defined.
interfaces	object/array	Host interfaces to be created for the given hosts.
macros	object/array	User macros to be created for the given hosts.

パラメータ	Type	説明
templates	object/array	Templates to link to the given hosts.

The templates must have the `templateid` property defined.

戻り値

(object) `hostids` プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

マクロの追加

2つのホストに2つの新しいマクロを追加します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.massadd",
  "params": {
    "hosts": [
      {
        "hostid": "10160"
      },
      {
        "hostid": "10167"
      }
    ],
    "macros": [
      {
        "macro": "${TEST1}",
        "value": "MACROTEST1"
      },
      {
        "macro": "${TEST2}",
        "value": "MACROTEST2",
        "description": "Test description"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10160",
      "10167"
    ]
  },
  "id": 1
}
```

参照

- [host.update](#)
- [Host group](#)
- [Template](#)
- [User macro](#)
- [Host interface](#)

ソース

CHost::massAdd() in ui/include/classes/api/services/CHost.php.

host.massremove

説明

object host.massremove(object parameters)

このメソッドは、複数のホストから関連するオブジェクトを削除することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object) 更新するホストと削除するオブジェクトの ID を含むパラメータ

パラメータ	Type	説明
hostids (required)	string/array	IDs of the hosts to be updated.
groupids	string/array	Host groups to remove the given hosts from.
interfaces	object/array	Host interfaces to remove from the given hosts. The host interface object must have the ip, dns and port properties defined.
macros	string/array	User macros to delete from the given hosts.
templateids	string/array	Templates to unlink from the given hosts.
templateids_clear	string/array	Templates to unlink and clear from the given hosts.

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

テンプレートのリンク解除

2つのホストからテンプレートのリンクを解除し、テンプレート化されたエンティティをすべて削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.massremove",
  "params": {
    "hostids": ["69665", "69666"],
    "templateids_clear": "325"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "69665",
      "69666"
    ]
  },
  "id": 1
}
```

参照

- [host.update](#)
- [User macro](#)
- [Host interface](#)

ソース

CHost::massRemove() in ui/include/classes/api/services/CHost.php.

host.massupdate

説明

object host.massupdate(object parameters)

このメソッドでは、複数のホストで同時に関連するオブジェクトの交換や削除、プロパティの更新を行うことができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object) 更新するホストの ID と更新すべきプロパティを含むパラメータ

[standard host properties](#)に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
hosts (required)	object/array	Hosts to be updated.
groups	object/array	The hosts must have the <code>hostid</code> property defined. Host groups to replace the current host groups the hosts belong to.
interfaces	object/array	The host groups must have the <code>groupid</code> property defined. Host interfaces to replace the current host interfaces on the given hosts.
inventory	object	Host inventory properties. Host inventory mode cannot be updated using the <code>inventory</code> parameter, use <code>inventory_mode</code> instead.
macros	object/array	User macros to replace the current user macros on the given hosts.
templates	object/array	Templates to replace the currently linked templates on the given hosts.
templates_clear	object/array	The templates must have the <code>templateid</code> property defined. Templates to unlink and clear from the given hosts.
		The templates must have the <code>templateid</code> property defined.

戻り値

(object) `hostids` プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

複数ホストの監視を有効にする

2 台のホストの監視を有効にする。(ステータスを 0 にする)

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.massupdate",
  "params": {
    "hosts": [
      {
        "hostid": "69665"
      }
    ],
  }
}
```

```

        {
            "hostid": "69666"
        }
    ],
    "status": 0
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}

```

参照

- [host.update](#)
- [host.massadd](#)
- [host.massremove](#)
- [Host group](#)
- [Template](#)
- [User macro](#)
- [Host interface](#)

ソース

`CHost::massUpdate()` in `ui/include/classes/api/services/CHost.php`.

host.update

説明

`object host.update(object/array hosts)`

このメソッドでは、既存のホストを更新することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、`
` ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object/array) 更新されるホストプロパティ

`hostid` プロパティは各ホストに対して定義する必要があり、他のプロパティはすべてオプションです。 `
` 与えられたプロパティのみが更新され、その他のプロパティは変更されません。

しかし、ホストの `technical name`(実ホスト名) を更新すると、ホストの `visible name` (表示名指定されていないか空の場合) も `
` ホストの `technical name` の値で更新されることに注意してください。

[standard host properties](#)に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
<code>groups</code>	<code>object/array</code>	Host <code>groups</code> to replace the current host groups the host belongs to.

The host groups must have the `groupid` property defined. All host groups that are not listed in the request will be unlinked.

パラメータ	Type	説明
interfaces	object/array	Host interfaces to replace the current host interfaces.
tags	object/array	All interfaces that are not listed in the request will be removed. Host tags to replace the current host tags.
inventory	object	All tags that are not listed in the request will be removed. Host inventory properties.
macros	object/array	User macros to replace the current user macros.
templates	object/array	All macros that are not listed in the request will be removed. Templates to replace the currently linked templates. All templates that are not listed in the request will be only unlinked.
templates_clear	object/array	The templates must have the <code>templateid</code> property defined. Templates to unlink and clear from the host.
		The templates must have the <code>templateid</code> property defined.

Note:

Zabbix フロントエンドとは異なり、`name`(表示名) と `host`(ホスト名) が同じ場合、API 経由で `host` を更新しても、`
` 自動的に `name` は更新されません。両方のプロパティを明示的に更新する必要があります。

戻り値

(object) `hostids` プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

ホストの有効化

ホストの監視を有効にする、つまりステータスを 0 にします。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.update",
  "params": {
    "hostid": "10126",
    "status": 0
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10126"
    ]
  },
  "id": 1
}
```

テンプレートのリンク削除

ホストから 2 つのテンプレートのリンクを削除してクリアします。

リクエスト:

```
{
  "jsonrpc": "2.0",
```

```
"method": "host.update",
"params": {
  "hostid": "10126",
  "templates_clear": [
    {
      "templateid": "10124"
    },
    {
      "templateid": "10125"
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10126"
    ]
  },
  "id": 1
}
```

ホストマクロの更新

すべてのホストマクロを新しい2つのマクロに置き換えます。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.update",
  "params": {
    "hostid": "10126",
    "macros": [
      {
        "macro": "{$PASS}",
        "value": "password"
      },
      {
        "macro": "{$DISC}",
        "value": "sda",
        "description": "Updated description"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10126"
    ]
  },
  "id": 1
}
```

ホストインベントリの更新

インベントリモードの変更とロケーションの追加

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.update",
  "params": {
    "hostid": "10387",
    "inventory_mode": 0,
    "inventory": {
      "location": "Latvia, Riga"
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10387"
    ]
  },
  "id": 1
}
```

ホストタグの更新

すべてのホストタグを新しいものに置き換えます。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.update",
  "params": {
    "hostid": "10387",
    "tags": {
      "tag": "OS",
      "value": "RHEL 7"
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10387"
    ]
  },
  "id": 1
}
```

Updating host encryption

Update the host "10590" to use PSK encryption only for connections from host to Zabbix server, and change the PSK identity and PSK key. Note that the host has to be **pre-configured to use PSK**.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "host.update",
  "params": {
    "hostid": "10590",
    "tls_connect": 1,
    "tls_accept": 2,
    "tls_psk_identity": "PSK 002",
    "tls_psk": "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9"
  },
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10590"
    ]
  },
  "id": 1
}
```

参照

- [host.massadd](#)
- [host.massupdate](#)
- [host.massremove](#)
- [Host group](#)
- [Template](#)
- [User macro](#)
- [Host interface](#)
- [Host inventory](#)
- [Host tag](#)

ソース

`CHost::update()` in `ui/include/classes/api/services/CHost.php`.

Host group

このクラスは、`host groups` との連携を想定しています。

オブジェクトリファレンス:

- [Host group](#)

利用可能なメソッド:

- [hostgroup.create](#) - 新しい `host group` の作成
- [hostgroup.delete](#) - `host group` の削除
- [hostgroup.get](#) - `host group` の取得
- [hostgroup.massadd](#) - `host group` への関連オブジェクトの追加
- [hostgroup.massremove](#) - `host group` から関連オブジェクトを削除
- [hostgroup.massupdate](#) - `host group` から関連するオブジェクトを交換または削除する
- [hostgroup.update](#) - `host group` の更新

> Host group object

以下のオブジェクトは `hostgroup API` に直接関連するものです。

Host group

host group object オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
groupid	string	(readonly) ID of the host group.
name (required)	string	Name of the host group.
flags	integer	(readonly) Origin of the host group. Possible values: 0 - a plain host group; 4 - a discovered host group.
internal	integer	(readonly) Whether the group is used internally by the system. An internal group cannot be deleted. Possible values: 0 - (default) not internal; 1 - internal.
uuid	string	Universal unique identifier, used for linking imported host groups to already existing ones. Auto-generated, if not given.

For update operations this field is readonly.

hostgroup.create

説明

object hostgroup.create(object/array hostGroups)

このメソッドでは、新しいホストグループを作成することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object/array) 作成するホストグループ
 このメソッドは、[standard host group properties](#) を持つホストグループを受け付けます。

戻り値

(object) groupids プロパティの下に作成されたホストグループの ID を含むオブジェクトを返します。
 返される ID の順序は、渡されたホストグループの順序と一致します。

例

host group の作成

"Linux servers" という host group を作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.create",
  "params": {
    "name": "Linux servers"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "groupids": [
      "107819"
    ]
  },
  "id": 1
}
```

ソース

CHostGroup::create() in ui/include/classes/api/services/CHostGroup.php.

hostgroup.delete

説明

object hostgroup.delete(array hostGroupIds)

このメソッドでは、host group を削除することができます。

以下の場合、ホストグループを削除することはできません。

- このグループに属するホストのみが含まれている。
- 内部グループとしてマークされている。
- ホストプロトタイプで使用されている。
- グローバルスクリプトで使用されている。
- 相関関係で使用されている。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(array) 削除する host group の ID 配列

戻り値

(object) groupids プロパティの下にある、削除された host group の ID を含むオブジェクトを返す。

例

複数の host group を削除する

2つの host group を削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.delete",
  "params": [
    "107824",
    "107825"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "groupids": [
      "107824",
      "107825"
    ]
  }
}
```

```

    ]
  },
  "id": 1
}

```

ソース

CHostGroup::delete() in ui/include/classes/api/services/CHostGroup.php.

hostgroup.get

説明

integer/array hostgroup.get(object parameters)

このメソッドは、与えられたパラメータに従って、host group を取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーのロール設定で
取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートしています。

パラメータ	タイプ	説明
graphids	string/array	指定されたグラフを持つホストまたはテンプレートを含まないホストグループのみを返します。
groupids	string/array	指定されたホストグループ ID を持つホストグループのみを返します。
hostids	string/array	指定されたホストを含むホストグループのみを返します。
maintenanceids	string/array	指定されたメンテナンスの影響を受けるホストグループのみを返します。
monitored_hosts	flag	監視対象ホストを含むホストグループのみを返します。
real_hosts	flag	ホストを含むホストグループのみを返します。
templated_hosts	flag	テンプレートを含まないホストグループのみを返します。
templateids	string/array	指定されたテンプレートを含まないホストグループのみを返します。
triggerids	string/array	指定されたトリガーを持つホストまたはテンプレートを含まないホストグループのみを返します。
with_graphs	flag	グラフのあるホストを含むホストグループのみを返します。
with_graph_prototypes	flag	グラフプロトタイプを持つホストを含むホストグループのみを返します。
with_hosts_and_templates	flag	ホスト または テンプレートを含まないホストグループのみを返します。
with_httpstests	flag	Web チェックのあるホストを含むホストグループのみを返します。
with_items	flag	<code>with_monitored_httpstests</code> パラメータをオーバーライドします。アイテムを含むホストまたはテンプレートを含まないホストグループのみを返します。
with_item_prototypes	flag	<code>with_monitored_items</code> および <code>with_simple_graph_items</code> パラメータをオーバーライドします。アイテムプロトタイプを持つホストを含むホストグループのみを返しません。
with_simple_graph_item_prototypes	flag	<code>with_simple_graph_item_prototypes</code> パラメータをオーバーライドします。アイテムプロトタイプを持つホストを含むホストグループのみを返しません。これは、作成が有効で、数値型の情報を持ちます。
with_monitored_httpstests	flag	Web チェックが有効になっているホストを含むホストグループのみを返します。

パラメータ	タイプ	説明
with_monitored_items	flag	アイテムが有効になっているホストまたはテンプレートを含むホストグループのみを返します。
with_monitored_triggers	flag	with_simple_graph_items パラメータをオーバーライドします。トリガーが有効になっているホストを含むホストグループのみを返します。トリガーで使用されるすべての項目も有効にする必要があります。
with_simple_graph_items	flag	数値項目を持つホストを含むホストグループのみを返します。
with_triggers	flag	トリガーを持つホストを含むホストグループのみを返します。
selectDiscoveryRule	query	with_monitored_triggers パラメータをオーバーライドします。ホストグループを作成した LLD ルールを含む <code>discoveryRule</code> プロパティを返します。
selectGroupDiscovery	query	ホストグループディスカバリオブジェクトを含む <code>groupDiscovery</code> プロパティを返します。 ホストグループディスカバリオブジェクトは、検出されたホストグループをホストグループプロトタイプにリンクし、次のプロパティを持ちます。 <code>groupid</code> - (string) 検出されたホストグループの ID。 <code>lastcheck</code> - (timestamp) ホストグループが最後に検出された時刻。 <code>name</code> - (string) ホストグループプロトタイプの名前。 <code>parent_group_prototypeid</code> - (string) ホストグループが作成されたホストグループプロトタイプの ID。 <code>ts_delete</code> - (timestamp) 検出されなくなったホストグループが削除される時刻。
selectHosts	query	ホストグループに属するホストを含む <code>hosts</code> プロパティを返します。
selectTemplates	query	<code>count</code> をサポートします。ホストグループに属するテンプレートを含む <code>templates</code> プロパティを返します。
limitSelects	integer	<code>count</code> をサポートします。副選択によって返されるレコードの数を制限します。
sortfield	string/array	次のサブセレクトに適用されます。 <code>selectHosts</code> - 結果は <code>host</code> でソートされます。 <code>selectTemplates</code> - 結果は <code>host</code> でソートされます。 指定されたプロパティで結果を並べ替えます。
countOutput	boolean	使用可能な値は次のとおりです: <code>groupid</code> 、 <code>name</code> すべての <code>get</code> メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) のどちらかを返します:

- オブジェクトの配列
- もし `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

名前によるデータの取得

"Zabbix servers" と "Linux servers" という名前の 2 つのホストグループに関するすべてのデータを取得します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.get",
  "params": {
    "output": "extend",
    "filter": {
      "name": [
        "Zabbix servers",
        "Linux servers"
      ]
    }
  },
  "auth": "6f38cddc44cfbb6c1bd186f9a220b5a0",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "groupid": "2",
      "name": "Linux servers",
      "internal": "0"
    },
    {
      "groupid": "4",
      "name": "Zabbix servers",
      "internal": "0"
    }
  ],
  "id": 1
}
```

参照

- [Host](#)
- [Template](#)

ソース

CHostGroup::get() in ui/include/classes/api/services/CHostGroup.php.

hostgroup.massadd

説明

object hostgroup.massadd(object parameters)

この方法では、与えられたすべての host group に、複数の関連オブジェクトを同時に追加することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object) 更新する host groups の ID と、全ての host groups に追加するオブジェクトを含むパラメータ。

このメソッドは、以下のパラメータを受け付ける。

パラメータ	Type	説明
groups (required)	object/array	Host groups to be updated.
hosts	object/array	The host groups must have the <code>groupid</code> property defined. Hosts to add to all host groups.
templates	object/array	The hosts must have the <code>hostid</code> property defined. Templates to add to all host groups.
		The templates must have the <code>templateid</code> property defined.

戻り値

(object) `groupids` プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

host group への host の追加

ID が 5 と 6 の host group に 2 台の host を追加します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.massadd",
  "params": {
    "groups": [
      {
        "groupid": "5"
      },
      {
        "groupid": "6"
      }
    ],
    "hosts": [
      {
        "hostid": "30050"
      },
      {
        "hostid": "30001"
      }
    ]
  },
  "auth": "f223adf833b2bf2ff38574a67bba6372",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "groupids": [
      "5",
      "6"
    ]
  },
  "id": 1
}
```

参照

- [Host](#)
- [Template](#)

ソース

CHostGroup::massAdd() in ui/include/classes/api/services/CHostGroup.php.

hostgroup.massremove

説明

object hostgroup.massremove(object parameters)

このメソッドは、複数の host group から関連するオブジェクトを削除することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object) 更新する host group の ID と削除するオブジェクトを含むパラメータ

パラメータ	Type	説明
groupids (required)	string/array	IDs of the host groups to be updated.
hostids	string/array	Hosts to remove from all host groups.
templateids	string/array	Templates to remove from all host groups.

戻り値

(object) groupids プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

host group から host を削除する

与えられた host group から 2 つの host を削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.massremove",
  "params": {
    "groupids": [
      "5",
      "6"
    ],
    "hostids": [
      "30050",
      "30001"
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "groupids": [
      "5",
      "6"
    ]
  },
  "id": 1
}
```

ソース

CHostGroup::massRemove() in ui/include/classes/api/services/CHostGroup.php.

hostgroup.massupdate

説明

object hostgroup.massupdate(object parameters)

この方法は、複数の host group において、ホストとテンプレートを指定したものに置き換えることができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(object) 更新する host group の ID と更新すべきオブジェクトを含むパラメータ。

パラメータ	Type	説明
groups (required)	object/array	Host groups to be updated.
hosts (required)	object/array	The host groups must have the <code>groupid</code> property defined. Hosts to replace the current hosts on the given host groups. All other hosts, except the ones mentioned, will be excluded from host groups. Discovered hosts will not be affected.
templates (required)	object/array	The hosts must have the <code>hostid</code> property defined. Templates to replace the current templates on the given host groups. All other templates, except the ones mentioned, will be excluded from host groups. The templates must have the <code>templateid</code> property defined.

戻り値

(object) `groupids` プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

host group 内のホストとテンプレートの入れ替え

host group 内のすべてのホストを指定されたホストに置き換え、ホストグループ内のすべてのテンプレートのリンクを解除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.massupdate",
  "params": {
    "groups": [
      {
        "groupid": "6"
      }
    ],
    "hosts": [
      {
        "hostid": "30050"
      }
    ],
    "templates": []
  },
  "auth": "f223adf833b2bf2ff38574a67bba6372",
}
```

```
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "groupids": [
      "6",
    ]
  },
  "id": 1
}
```

参照

- [hostgroup.update](#)
- [hostgroup.massadd](#)
- [Host](#)
- [Template](#)

ソース

`CHostGroup::massUpdate()` in `ui/include/classes/api/services/CHostGroup.php`.

hostgroup.update

説明

`object hostgroup.update(object/array hostGroups)`

このメソッドでは、既存の `hosts group` を更新することができます。

Note:

このメソッドは、Admin および Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、`
` ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

パラメータ

(`object/array`) `Host group properties` が更新されます。

`groupid` プロパティは各ホストグループに対して定義する必要があり、他のプロパティはすべてオプションです。`
` 与えられたプロパティのみが更新され、その他のプロパティは変更されません。

戻り値

(`object`) `groupids` プロパティの下で更新された `host group` の ID を含むオブジェクトを返します。

例

host group の名前の変更

host group の名前を "Linux hosts" に変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostgroup.update",
  "params": {
    "groupid": "7",
    "name": "Linux hosts"
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "groupids": [
      "7"
    ]
  },
  "id": 1
}
```

ソース

CHostGroup::update() in ui/include/classes/api/services/CHostGroup.php.

Host interface

このクラスは、host interface との連携を想定しています。

オブジェクトリファレンス:

- [Host interface](#)

利用可能なメソッド:

- [hostinterface.create](#) - 新しい host interface の作成
- [hostinterface.delete](#) - host interface の削除
- [hostinterface.get](#) - host interface の取得
- [hostinterface.massadd](#) - host interface へ関連オブジェクトの追加
- [hostinterface.massremove](#) - host interface から関連オブジェクトの削除
- [hostinterface.replacehostinterfaces](#) - ホスト上の host interface 置換
- [hostinterface.update](#) - host interface の更新

> Host interface object

以下のオブジェクトは `hostinterfaceAPI` に直接関連するものです。

Host interface

host interface オブジェクトには、次のプロパティがあります。

Attention:

IP と DNS の両方が必要ではありますが、DNS を使用しない場合は空の文字列に設定してください。

プロパティ	タイプ	説明
available	integer	(読み取り専用) ホストインターフェイスの可用性。 値は次のとおりです。 0 - (デフォルト) 不明 1 - 利用可能 2 - 利用不可
details	array	インターフェイスの追加オブジェクト。インターフェイス'タイプ'が SNMP の場合は必須
disable_until	timestamp	(読み取り専用) 使用できないホストインターフェイスの次回ポーリング時間
dns (必須)	string	インターフェイスで使用される DNS 名。
error	string	IP 経由で接続する場合は、空の文字列にすることができます。 (読み取り専用) ホストインターフェイスが使用できない場合のエラーテキスト。
errors_from	timestamp	(読み取り専用) ホストインターフェイスが使用できなくなった時刻。
hostid (必須)	string	インターフェイスが属するホストの ID
interfaceid	string	(読み取り専用) インターフェイスの ID

プロパティ	タイプ	説明
ip (必須)	string	インターフェースで使用される IP アドレス。
main (必須)	integer	DNS 経由で接続する場合は、空の文字列にすることができます。 インターフェースがホストでデフォルトとして使用されているかどうか。 ホストのデフォルトとして設定できるインターフェイスは 1 つだけです。
port (必須)	string	値は次のとおりです。 0 - デフォルトではありません 1 - デフォルトです インターフェースで使用されるポート番号。ユーザーマクロを含めることができます。
type (必須)	integer	インターフェースタイプ 値は次のとおりです。 1 - エージェント 2 - SNMP 3 - IPMI 4 - JMX
useip (必須)	integer	IP 経由で接続するかどうか。 値は次のとおりです。 0 - ホスト DNS 名を使用して接続します。 1 - ホスト IP アドレスを使用して接続します。

Details tag

Details tag オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
version (必須)	integer	SNMP インターフェースのバージョン。 値は次のとおりです。 1 - SNMPv1 2 - SNMPv2c 3 - SNMPv3
bulk	integer	バルク SNMP リクエストを使用するかどうか。 値は次のとおりです。 0 - バルクリクエストを使用しない 1 - (デフォルト) - バルクリクエストを使用する
community	string	SNMP コミュニティ (必須) SNMPv1 および SNMPv2 インターフェースでのみ使用されます。
securityname	string	SNMPv3 セキュリティ名。SNMPv3 インターフェースでのみ使用されま
securitylevel	integer	SNMPv3 セキュリティレベル。SNMPv3 インターフェースでのみ使用されま
authpassphrase	string	値は次のとおりです。 0 - (デフォルト) - noAuthNoPriv 1 - authNoPriv 2 - authPriv SNMPv3 認証パスフレーズ。SNMPv3 インターフェースでのみ使用されま
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。SNMPv3 インターフェースでのみ使用されま

プロパティ	タイプ	説明
authprotocol	integer	SNMPv3 認証プロトコル。SNMPv3 インターフェースでのみ使用されま す。 値は次のとおりです。 0 - (デフォルト) - MD5 1 - SHA1 2 - SHA224 3 - SHA256 4 - SHA384 5 - SHA512
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使 用されます。 値は次のとおりです。 0 - (デフォルト) - DES 1 - AES128 2 - AES192 3 - AES256 4 - AES192C 5 - AES256C
contextname	string	SNMPv3 コンテキスト名。SNMPv3 インターフェースでのみ使用されま す。

hostinterface.create

説明

`object hostinterface.create(object/array hostInterfaces)`

このメソッドでは、新しいホストインターフェイスを作成できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するホストインターフェイス。このメソッドは、[標準のホストインターフェイスプロパティ](#)を持つホストインターフェイスを受け入れます。

戻り値

(object) `interfaceids` プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストインターフェイスの順序と一致します。

例

新しいインターフェイスの作成

ホスト "30052" にセカンダリ IP エージェントインターフェイスを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.create",
  "params": {
    "hostid": "30052",
    "main": "0",
    "type": "1",
    "useip": "1"
    "ip": "127.0.0.1",
    "dns": "",
    "port": "10050",
  },
}
```



```
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
      "30062"
    ]
  },
  "id": 1
}
```

SNMPの詳細を指定してインターフェースを作成する

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.create",
  "params": {
    "hostid": "10456",
    "main": "0",
    "type": "2",
    "useip": "1",
    "ip": "127.0.0.1",
    "dns": "",
    "port": "1601",
    "details": {
      "version": "2",
      "bulk": "1",
      "community": "{$SNMP_COMMUNITY}"
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
      "30063"
    ]
  },
  "id": 1
}
```

参照

- [hostinterface.massadd](#)
- [host.massadd](#)

ソース

CHostInterface::create() in ui/include/classes/api/services/CHostInterface.php.

hostinterface.delete

説明

object hostinterface.delete(array hostInterfaceIds)

このメソッドでは、ホストインターフェースを削除できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するホストインターフェースの ID

戻り値

(object) interfaceids プロパティの下で削除されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェースの削除

ID 30062 のホストインターフェイスを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.delete",
  "params": [
    "30062"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
      "30062"
    ]
  },
  "id": 1
}
```

参照

- [hostinterface.massremove](#)
- [host.massremove](#)

ソース

CHostInterface::delete() in ui/include/classes/api/services/CHostInterface.php.

hostinterface.get

説明

integer/array hostinterface.get(object parameters)

このメソッドを使用では、指定されたパラメーターに従ってホストインターフェイスを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは、次のパラメータをサポートしています。

パラメータ	タイプ	説明
hostids	string/array	指定されたホストによって使用されるホストインターフェースのみを返します。
interfaceids	string/array	指定された ID を持つホストインターフェースのみを返します。
itemids	string/array	指定されたアイテムで使用されるホストインターフェースのみを返します。
triggerids	string/array	指定されたトリガーのアイテムによって使用されるホストインターフェースのみを返します。
selectItems	query	インターフェースを使用するアイテムを含む <code>items</code> プロパティを返します。
selectHosts	query	<code>count</code> をサポートします。 インターフェースを使用するホストの配列を含む <code>hosts</code> プロパティを返します。
limitSelects	integer	副選択によって返されるレコードの数を制限します。
sortfield	string/array	次の副選択に適用されます： <code>selectItems</code> 指定されたプロパティで結果を並べ替えます。
countOutput	boolean	指定可能な値は <code>interfaceid</code> 、 <code>dns</code> 、 <code>ip</code> です。 すべての <code>get</code> メソッドに共通するこれらのパラメータについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
nodeids	string/array	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列。
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数。

例

ホストインターフェースの取得

ホスト "30057" が使用するインターフェースに関するすべてのデータを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.get",
  "params": {
    "output": "extend",
    "hostids": "30057"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
```

```

"result": [
  {
    "interfaceid": "50039",
    "hostid": "30057",
    "main": "1",
    "type": "1",
    "useip": "1",
    "ip": ":::1",
    "dns": "",
    "port": "10050",
    "available": "0",
    "error": "",
    "errors_from": "0",
    "disable_until": "0",
    "details": []
  },
  {
    "interfaceid": "55082",
    "hostid": "30057",
    "main": "0",
    "type": "1",
    "useip": "1",
    "ip": "127.0.0.1",
    "dns": "",
    "port": "10051",
    "available": "0",
    "error": "",
    "errors_from": "0",
    "disable_until": "0",
    "details": {
      "version": "2",
      "bulk": "0",
      "community": "{$SNMP_COMMUNITY}"
    }
  }
],
"id": 1
}

```

参照

- [Host](#)
- [Item](#)

ソース

CHostInterface::get() in ui/include/classes/api/services/CHostInterface.php.

hostinterface.massadd

説明

object hostinterface.massadd(object parameters)

このメソッドでは、ホストインターフェイスを複数のホストに同時に追加できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 指定されたホストで作成されるホストインターフェイスを含むパラメータ。

このメソッドは、次のパラメーターを受け付けます。

パラメータ	タイプ	説明
hosts (必須)	object/array	更新するホスト。 ホストには <code>hostid</code> プロパティを定義する必要があります。
interfaces (必須)	object/array	指定されたホスト上に作成する ホストインターフェース

戻り値

(object) `interfaceids` プロパティの下で作成されたホストインターフェースの ID を含むオブジェクトを返します。

例

インターフェースの作成

2つのホストにインターフェースを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.massadd",
  "params": {
    "hosts": [
      {
        "hostid": "30050"
      },
      {
        "hostid": "30052"
      }
    ],
    "interfaces": {
      "dns": "",
      "ip": "127.0.0.1",
      "main": 0,
      "port": "10050",
      "type": 1,
      "useip": 1
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
      "30069",
      "30070"
    ]
  },
  "id": 1
}
```

参照

- [hostinterface.create](#)
- [host.massadd](#)
- [Host](#)

ソース

`CHostInterface::massAdd()` in `ui/include/classes/api/services/CHostInterface.php`.

hostinterface.massremove

説明

object hostinterface.massremove(object parameters)

このメソッドでは、指定されたホストからホストインターフェイスを削除できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 更新するホストと削除するインターフェイスの ID を含むパラメーター

パラメータ	タイプ	説明
hostids (必須)	string/array	更新するホストの ID
interfaces (必須)	object/array	指定されたホストから削除するホストインターフェイス。 ホストインターフェイスオブジェクトには、ip、dns、および port プロパティが定義されている必要があります。

戻り値

(object) interfaceids プロパティの下で削除されたホストインターフェイスの ID を含むオブジェクトを返します。

例

インターフェイスの削除

2つのホストから"127.0.0.1"SNMP インターフェイスを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.massremove",
  "params": {
    "hostids": [
      "30050",
      "30052"
    ],
    "interfaces": {
      "dns": "",
      "ip": "127.0.0.1",
      "port": "161"
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
      "30069",
      "30070"
    ]
  },
  "id": 1
}
```

参照

- [hostinterface.delete](#)
- [host.massremove](#)

ソース

CHostInterface::massRemove() in ui/include/classes/api/services/CHostInterface.php.

hostinterface.replacehostinterfaces

説明

object hostinterface.replacehostinterfaces(object parameters)

このメソッドを使用では、特定のホスト上のすべてのホストインターフェースを置き換えることができます。

Note:

このメソッドは、Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 更新するホストの ID と新しいホストインターフェースを含むパラメーター。

パラメータ	タイプ	説明
hostid (必須)	string	更新するホストの ID
interfaces (必須)	object/array	置換後のホストインターフェース

戻り値

(object) interfaceids プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェイスの置換

すべてのホストインターフェースを単一のエージェントインターフェースに置換します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.replacehostinterfaces",
  "params": {
    "hostid": "30052",
    "interfaces": {
      "dns": "",
      "ip": "127.0.0.1",
      "main": 1,
      "port": "10050",
      "type": 1,
      "useip": 1
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
```

```
        "30081"
    ]
},
"id": 1
}
```

参照

- [host.update](#)
- [host.massupdate](#)

ソース

CHostInterface::replaceHostInterfaces() in ui/include/classes/api/services/CHostInterface.php.

hostinterface.update

説明

object hostinterface.update(object/array hostInterfaces)

このメソッドでは、既存のホストインターフェイスを更新できます。

Note:

このメソッドは、Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) **ホストインターフェイスのプロパティ**が更新されます。

interfaceid プロパティは、ホストインターフェイスごとに定義する必要があり、他のすべてのプロパティはオプションです。指定されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) interfaceids プロパティの下で更新されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェイスポートの変更

ホストインターフェイスのポートを変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostinterface.update",
  "params": {
    "interfaceid": "30048",
    "port": "30050"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "interfaceids": [
      "30048"
    ]
  },
  "id": 1
}
```


ソース

CHostInterface::update() in ui/include/classes/api/services/CHostInterface.php.

Host prototype

このクラスは host prototypes との連携を想定しています。

オブジェクトリファレンス:

- [Host prototype](#)
- [Host prototype inventory](#)
- [Group link](#)
- [Group prototype](#)

利用可能なメソッド:

- [hostprototype.create](#) - 新しいホストプロトタイプを作成
- [hostprototype.delete](#) - ホストプロトタイプの削除
- [hostprototype.get](#) - ホストプロトタイプの取得
- [hostprototype.update](#) - ホストプロトタイプの更新

> Host prototype object

以下のオブジェクトは hostprototypeAPI に直接関連しています。

Host prototype

host prototype オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
hostid	string	(読み取り専用) ホストプロトタイプの ID。
host (必須)	string	ホストプロトタイプの名前
name	string	ホストプロトタイプの表示名
status	integer	デフォルト: host プロパティ値 ホストプロトタイプのステータス
inventory_mode	integer	値は次のとおり 0 - (デフォルト) 監視対象ホスト 1 - 監視対象外ホスト ホストインベントリ作成モード
templateid	string	値は次のとおり -1 - (デフォルト) 無効 0 - 手動 1 - 自動
discover	integer	(読み取り専用) 親テンプレートホストプロトタイプの ID ホストプロトタイプの検出ステータス
custom_interfaces	integer	値: 0 - (デフォルト) 新しいホストが検出されます 1 - 新しいホストは検出されず、既存のホストは失われたものとしてマークされます ホストプロトタイプによって作成されたホストのインターフェースのソース
		値: 0 - (デフォルト) 親ホストからインターフェースを継承 1 - ホストプロトタイプのカスタムインターフェースを使用

プロパティ	タイプ	説明
uuid	string	インポートされたホストプロトタイプを既存のプロトタイプにリンクするために使用されるユニバーサル一意識別子。テンプレートのホストプロトタイプにのみ使用されます。指定されていない場合は自動生成されます。

update の場合、このフィールドは読み取り専用です。

Group link

group link オブジェクトは、ホストプロトタイプをホストグループにリンクし、次のプロパティを持ちます。

プロパティ	タイプ	説明
group_prototypeid	string	(読み取り専用) グループリンクの ID
groupid (必須)	string	ホストグループの ID
hostid	string	(読み取り専用) ホストプロトタイプの ID
templateid	string	(読み取り専用) 親テンプレートグループリンクの ID

Group prototype

グループプロトタイプオブジェクトは、検出されたホスト用に作成されるグループを設定し、次のプロパティを持ちます。

プロパティ	タイプ	説明
group_prototypeid	string	(読み取り専用) グループプロトタイプの ID
name (必須)	string	グループプロトタイプの名前
hostid	string	(読み取り専用) ホストプロトタイプの ID
templateid	string	(読み取り専用) 親テンプレートグループのプロトタイプの ID。

Host prototype tag

Host prototype tag オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	ホストプロトタイプタグ名
value	string	ホストプロトタイプタグ値

Custom interface

custom interface オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
dns	string	インターフェースで使用される DNS 名
ip	string	接続が DNS 経由で行われる場合は必須。マクロを含めることができません。 インターフェースで使用される IP アドレス
main (必須)	integer	接続が IP 経由で行われる場合は必須。マクロを含めることができます。 インターフェイスがホストでデフォルトとして使用されているかどうか。 ホストでデフォルトとして設定できるインターフェースは 1 つだけです

値は次のとおり
0 - デフォルトではありません
1 - デフォルトです

プロパティ	タイプ	説明
port (必須)	string	インターフェイスで使用されるポート番号。ユーザーマクロと LLD マクロを含めることができます。
type (必須)	integer	インターフェイスタイプ 値は次のとおり 1 - エージェント 2 - SNMP 3 - IPMI 4 - JMX
useip (必須)	integer	IP 経由で接続するかどうか 値は次のとおり 0 - ホスト DNS 名を使用して接続 1 - ホスト IP アドレスを使用して接続
details	array	インターフェイスの追加オブジェクト。インターフェイス'タイプ'が SNMP の場合は必須。

Custom interface details

Custom interface details オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
version (必須)	integer	SNMP インターフェースのバージョン。 値は次のとおりです。 1 - SNMPv1 2 - SNMPv2c 3 - SNMPv3
bulk	integer	バルク SNMP リクエストを使用するかどうか。 値は次のとおりです。 0 - バルクリクエストを使用しない 1 - (デフォルト) - バルクリクエストを使用する
community	string	SNMP コミュニティ。SNMPv1 および SNMPv2 インターフェイスでのみ使用されます。
securityname	string	SNMPv3 セキュリティ名。SNMPv3 インターフェイスでのみ使用されま
securitylevel	integer	SNMPv3 セキュリティレベル。SNMPv3 インターフェイスでのみ使用されま
authpassphrase	string	SNMPv3 認証パスフレーズ。SNMPv3 インターフェイスでのみ使用されま
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。SNMPv3 インターフェイスでのみ使用されま
authprotocol	integer	SNMPv3 認証プロトコル。SNMPv3 インターフェイスでのみ使用されま
		値は次のとおりです。 0 - (デフォルト) - MD5 1 - SHA1 2 - SHA224 3 - SHA256 4 - SHA384 5 - SHA512

プロパティ	タイプ	説明
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 値は次のとおりです。 0 - (デフォルト) - DES 1 - AES128 2 - AES192 3 - AES256 4 - AES192C 5 - AES256C
contextname	string	SNMPv3 コンテキスト名。SNMPv3 インターフェースでのみ使用されません。

hostprototype.create

説明

object hostprototype.create(object/array hostPrototypes)

このメソッドでは、新しいホストプロトタイプを作成できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するホストプロトタイプ

このメソッドは標準のホストプロトタイププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
groupLinks (必須)	array	ホストプロトタイプ用に作成するグループリンク
ruleid (必須)	string	ホストプロトタイプが属する LLD ルールの ID
groupPrototypes	array	ホストプロトタイプ用に作成されるグループプロトタイプ
macros	object/array	ホストプロトタイプ用に作成するユーザーマクロ
tags	object/array	ホストプロトタイプタグ。
interfaces	object/array	ホストプロトタイプカスタムインターフェース。
templates	object/array	ホストプロトタイプにリンクするテンプレート

テンプレートには `templateid` プロパティを定義する必要があります。

戻り値

(object) `hostids` プロパティの下で作成されたホストプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストプロトタイプの順序と一致します。

例

ホストプロトタイプの作成

グループプロトタイプ "`{#HV.NAME}`"、タグペア "Datacenter": "`{#DATACENTER.NAME}`" およびコミュニティ `{$SNMP_COMMUNITY}` を使用したカスタム SNMPv2 インターフェイス 127.0.0.1:161 を使用して、LLD ルール "23542" にホストプロトタイプ "`{#VM.NAME}`" を作成し、ホストグループ 「2」 にリンクします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostprototype.create",
```

```

"params": {
  "host": "{#VM.NAME}",
  "ruleid": "23542",
  "custom_interfaces": "1",
  "groupLinks": [
    {
      "groupid": "2"
    }
  ],
  "groupPrototypes": [
    {
      "name": "{#HV.NAME}"
    }
  ],
  "tags": [
    {
      "tag": "Datacenter",
      "value": "{#DATACENTER.NAME}"
    }
  ],
  "interfaces": [
    {
      "main": "1",
      "type": "2",
      "useip": "1",
      "ip": "127.0.0.1",
      "dns": "",
      "port": "161",
      "details": {
        "version": "2",
        "bulk": "1",
        "community": "{$SNMP_COMMUNITY}"
      }
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10103"
    ]
  },
  "id": 1
}

```

参照

- [Group link](#)
- [Group prototype](#)
- [Host prototype tag](#)
- [Custom interface](#)
- [User macro](#)

ソース

CHostPrototype::create() in ui/include/classes/api/services/CHostPrototype.php.

hostprototype.delete

説明

object hostprototype.delete(array hostPrototypeIds)

このメソッドでは、ホストのプロトタイプを削除できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するホストプロトタイプの ID

戻り値

(object) hostids プロパティの下で削除されたホストプロトタイプの ID を含むオブジェクトを返します。

例

複数のホストプロトタイプ削除

2つのホストプロトタイプを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostprototype.delete",
  "params": [
    "10103",
    "10105"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10103",
      "10105"
    ]
  },
  "id": 1
}
```

ソース

CHostPrototype::delete() in ui/include/classes/api/services/CHostPrototype.php.

hostprototype.get

説明

integer/array hostprototype.get(object parameters)

このメソッドでは、指定されたパラメーターを基にホストプロトタイプを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
hostids	string/array	指定された ID を持つホストプロトタイプのみを返します
discoveryids	string/array	指定された LLD ルールに属するホストプロトタイプのみを返します
inherited	boolean	true に設定すると、テンプレートから継承されたアイテムのみが返されます
selectDiscoveryRule	query	ホストプロトタイプが属する LLD ルールを使用して <code>discoveryRule</code> プロパティを返します
selectInterfaces	query	<code>interfaces</code> プロパティをホストプロトタイプカスタムインターフェイスとともに返します
selectGroupLinks	query	ホストプロトタイプのグループリンクを含む <code>groupLinks</code> プロパティを返します
selectGroupPrototypes	query	<code>groupPrototypes</code> プロパティを、ホストプロトタイプのグループプロトタイプとともに返します
selectMacros	query	ホストプロトタイプマクロを使用して <code>macros</code> プロパティを返します
selectParentHost	query	ホストプロトタイプが属するホストの <code>parent Host</code> プロパティを返します
selectTags	query	<code>tags</code> プロパティをホストプロトタイプタグとともに返します
selectTemplates	query	ホストプロトタイプにリンクされたテンプレートを含む <code>templates</code> プロパティを返します
sortfield	string/array	<code>count</code> をサポートします 指定されたプロパティで結果を並べ替えます。
countOutput	boolean	値は <code>hostid,host,name</code> および <code>status</code> です すべての <code>get</code> メソッドに共通するこれらのパラメーターについては Generic Zabbix API information ページで詳しく説明されています
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得されたオブジェクトの数

例

LLD ルールからホストプロトタイプの取得

LLD ルールから、すべてのホストプロトタイプ、グループリンク、グループプロトタイプ、およびタグを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostprototype.get",
  "params": {
    "output": "extend",
    "selectInterfaces": "extend",
    "selectGroupLinks": "extend",
    "selectGroupPrototypes": "extend",
    "selectTags": "extend",
    "discoveryids": "23554"
  }
}
```

```
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10092",
      "host": "#{HV.UUID}",
      "name": "#{HV.UUID}",
      "status": "0",
      "templateid": "0",
      "discover": "0",
      "custom_interfaces": "1",
      "inventory_mode": "-1",
      "groupLinks": [
        {
          "group_prototypeid": "4",
          "hostid": "10092",
          "groupid": "7",
          "templateid": "0"
        }
      ],
      "groupPrototypes": [
        {
          "group_prototypeid": "7",
          "hostid": "10092",
          "name": "#{CLUSTER.NAME}",
          "templateid": "0"
        }
      ],
      "tags": [
        {
          "tag": "Datacenter",
          "value": "#{DATACENTER.NAME}"
        },
        {
          "tag": "Instance type",
          "value": "#{INSTANCE_TYPE}"
        }
      ],
      "interfaces": [
        {
          "main": "1",
          "type": "2",
          "useip": "1",
          "ip": "127.0.0.1",
          "dns": "",
          "port": "161",
          "details": {
            "version": "2",
            "bulk": "1",
            "community": "${SNMP_COMMUNITY}"
          }
        }
      ]
    }
  ],
  "id": 1
}
```



```
}
```

参照

- [Group link](#)
- [Group prototype](#)
- [User macro](#)

ソース

CHostPrototype::get() in ui/include/classes/api/services/CHostPrototype.php.

hostprototype.update

説明

object hostprototype.update(object/array hostPrototypes)

このメソッドでは、既存のホストプロトタイプを更新できます。

Note:

このメソッドは Admin および Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるホストプロトタイププロパティ。

hostid プロパティは、ホストプロトタイプごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のホストプロトタイププロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
groupLinks	array	ホストプロトタイプの現在のグループリンクを置き換えます。
groupPrototypes	array	ホストプロトタイプ上の既存のグループプロトタイプを置き換えます。
macros	object/array	現在のユーザーマクロを置き換えます。
tags	object/array	リクエストにリストされていないマクロはすべて削除されます。 ホストプロトタイプタグを使用して現在のタグを置き換えます。
interfaces	object/array	リクエストにリストされていないタグはすべて削除されます。 現在のインターフェースを置き換えるためのプロトタイプカスタムインターフェースをホストします。
templates	object/array	カスタムインターフェースオブジェクトには、そのすべてのパラメーターが含まれている必要があります。 リクエストにリストされていないインターフェースはすべて削除されます。 現在リンクされているテンプレートを置き換えます。
		テンプレートには templateid プロパティが定義されている必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストプロトタイプの ID を含むオブジェクトを返します。

例

ホストプロトタイプの無効化

ホストプロトタイプを無効にするため、ステータスを 1 に設定します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostprototype.update",
  "params": {
    "hostid": "10092",
    "status": 1
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10092"
    ]
  },
  "id": 1
}
```

ホストプロトタイプタグの更新

ホストプロトタイプタグを新しいタグに置き換えます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "hostprototype.update",
  "params": {
    "hostid": "10092",
    "tags": [
      {
        "tag": "Datacenter",
        "value": "#{DATACENTER.NAME}"
      },
      {
        "tag": "Instance type",
        "value": "#{INSTANCE_TYPE}"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10092"
    ]
  },
  "id": 1
}
```

ホストプロトタイプカスタムインターフェイスの更新

継承されたインターフェイスをホストプロトタイプのカスタムインターフェイスに置き換えます。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "hostprototype.update",
  "params": {
    "hostid": "10092",
    "custom_interfaces": "1",
    "interfaces": [
      {
        "main": "1",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "161",
        "details": {
          "version": "2",
          "bulk": "1",
          "community": "${SNMP_COMMUNITY}"
        }
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10092"
    ]
  },
  "id": 1
}

```

参照

- [Group link](#)
- [Group prototype](#)
- [Host prototype tag](#)
- [Custom interface](#)
- [User macro](#)

ソース

`CHostPrototype::update()` in `ui/include/classes/api/services/CHostPrototype.php`.

Housekeeping

このクラスは、housekeeping 機能と合わせて動くクラスです。

オブジェクトリファレンス:

- [Housekeeping](#)

利用可能なメソッド:

- [housekeeping.get](#) - housekeeping の検索
- [housekeeping.update](#) - housekeeping の更新

> Housekeeping object

以下のオブジェクトは housekeeping API に直接関連するものです。

Housekeeping

settings オブジェクトは、以下のプロパティを持ちます。

プロパティ	Type	説明
hk_events_mode	integer	Enable internal housekeeping for events and alerts. Possible values: 0 - Disable; 1 - (default) Enable.
hk_events_trigger	string	Trigger data storage period. Accepts seconds and time unit with suffix. Default: 365d.
hk_events_service	string	Service data storage period. Accepts seconds and time unit with suffix. Default: 1d.
hk_events_internal	string	Internal data storage period. Accepts seconds and time unit with suffix. Default: 1d.
hk_events_discovery	string	Network discovery data storage period. Accepts seconds and time unit with suffix. Default: 1d.
hk_events_autoreg	string	Autoregistration data storage period. Accepts seconds and time unit with suffix. Default: 1d.
hk_services_mode	integer	Enable internal housekeeping for services. Possible values: 0 - Disable; 1 - (default) Enable.
hk_services	string	Services data storage period. Accepts seconds and time unit with suffix. Default: 365d.
hk_audit_mode	integer	Enable internal housekeeping for audit. Possible values: 0 - Disable; 1 - (default) Enable.
hk_audit	string	Audit data storage period. Accepts seconds and time unit with suffix. Default: 365d.
hk_sessions_mode	integer	Enable internal housekeeping for sessions. Possible values: 0 - Disable; 1 - (default) Enable.
hk_sessions	string	Sessions data storage period. Accepts seconds and time unit with suffix. Default: 365d.
hk_history_mode	integer	Enable internal housekeeping for history. Possible values: 0 - Disable; 1 - (default) Enable.

プロパティ	Type	説明
hk_history_global	integer	Override item history period. Possible values: 0 - Do not override; 1 - (default) Override.
hk_history	string	History data storage period. Accepts seconds and time unit with suffix. Default: 90d.
hk_trends_mode	integer	Enable internal housekeeping for trends. Possible values: 0 - Disable; 1 - (default) Enable.
hk_trends_global	integer	Override item trend period. Possible values: 0 - Do not override; 1 - (default) Override.
hk_trends	string	Trends data storage period. Accepts seconds and time unit with suffix. Default: 365d.
db_extension	string	(readonly) Configuration flag DB extension. If this flag is set to "timescaledb" then the server changes its behavior for housekeeping and item deletion.
compression_status	integer	Enable TimescaleDB compression for history and trends. Possible values: 0 - (default) Off; 1 - On.
compress_older	string	Compress history and trends records older than specified period. Accepts seconds and time unit with suffix. Default: 7d.
compression_availability	integer	(readonly) Compression availability. Possible values: 0 - Unavailable; 1 - Available.

housekeeping.get

説明

`object housekeeping.get(object parameters)`

このメソッドは、与えられたパラメータに従って、housekeeping オブジェクトを取得することができます。

Note:

このメソッドは、どのタイプのユーザーでも利用可能です。このメソッドを呼び出す許可は、ユーザーのロール設定で
取り消すことができます。詳しくは [User roles](#) をご覧ください。

パラメータ

(object) 希望する出力を定義するパラメータ

このメソッドは 1 つのパラメータのみをサポートしています。

パラメータ	Type	説明
output	query	This parameter being common for all get methods described in the reference commentary .

戻り値

(object) housekeeping オブジェクトを返します

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "housekeeping.get",
  "params": {
    "output": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hk_events_mode": "1",
    "hk_events_trigger": "365d",
    "hk_events_service": "1d",
    "hk_events_internal": "1d",
    "hk_events_discovery": "1d",
    "hk_events_autoreg": "1d",
    "hk_services_mode": "1",
    "hk_services": "365d",
    "hk_audit_mode": "1",
    "hk_audit": "365d",
    "hk_sessions_mode": "1",
    "hk_sessions": "365d",
    "hk_history_mode": "1",
    "hk_history_global": "0",
    "hk_history": "90d",
    "hk_trends_mode": "1",
    "hk_trends_global": "0",
    "hk_trends": "365d",
    "db_extension": "",
    "compression_status": "0",
    "compress_older": "7d"
  },
  "id": 1
}
```

ソース

CHousekeeping ::get() in ui/include/classes/api/services/CHousekeeping.php.

housekeeping.update

説明

object housekeeping.update(object housekeeping)

このメソッドでは、既存の housekeeping 設定を更新することができます。

Note:

このメソッドは、Super admin ユーザータイプにのみ使用できます。このメソッドを呼び出す許可は、
 ユーザーロール設定で取り消すことができます。詳細については、[User roles](#) を参照してください。

Parameters

(object) 更新される Housekeeping プロパティ

戻り値

(array) 更新されたパラメータの名前を配列で返します。

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "housekeeping.update",
  "params": {
    "hk_events_mode": "1",
    "hk_events_trigger": "200d",
    "hk_events_internal": "2d",
    "hk_events_discovery": "2d"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    "hk_events_mode",
    "hk_events_trigger",
    "hk_events_internal",
    "hk_events_discovery"
  ],
  "id": 1
}
```

ソース

CHousekeeping::update() in ui/include/classes/api/services/CHousekeeping.php.

Icon map

このクラスは、アイコンマップを操作するように設計されています。

オブジェクトリファレンス:

- [Icon map](#) - [Icon mapping](#)

利用可能なメソッド:

- [iconmap.create](#) - 新しいアイコンマップの作成
- [iconmap.delete](#) - アイコンマップの削除
- [iconmap.get](#) - アイコンマップの取得
- [iconmap.update](#) - アイコンマップの更新

> Icon map object

次のオブジェクトは `iconmapAPI` に直接関連しています。

Icon map

icon map オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
<code>iconmapid</code>	string	(読み取り専用) アイコンマップの ID
<code>default_iconid</code> (必須)	string	デフォルトアイコンの ID
<code>name</code> (必須)	string	アイコンマップの名前

Icon mapping

アイコンマッピングオブジェクトは、特定のインベントリフィールド値を持つホストに使用される特定のアイコンを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
iconmappingid	string	(読み取り専用) アイコンマップの ID
iconid (必須)	string	アイコンマッピングで使用されるアイコンの ID
expression (必須)	string	インベントリフィールド照合式
inventory_link (必須)	integer	ホストインベントリフィールドの ID
iconmapid	string	サポートされているインベントリフィールドのリストについては、 ホストインベントリオブジェクト を参照してください。 (必須) アイコンマッピングが属するアイコンマップの ID
sortorder	integer	(必須) アイコンマップ内のアイコンマッピングの位置

iconmap.create

説明

```
object iconmap.create(object/array iconMaps)
```

このメソッドでは新しいアイコンマップを作成できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) アイコンマップの作成

標準のアイコンマッププロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
mappings (必須)	array	アイコンマップ用に作成する アイコンマッピング

戻り値

(object) iconmapids プロパティの下で作成されたアイコンマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイコンマップの順序と一致します。

例

アイコンマップを作成する

さまざまなタイプのホストを表示するアイコンマップを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "iconmap.create",
  "params": {
    "name": "Type icons",
    "default_iconid": "2",
    "mappings": [
      {
        "inventory_link": 1,
        "expression": "server",
        "iconid": "3"
      }
    ]
  }
}
```



```
        {
            "inventory_link": 1,
            "expression": "switch",
            "iconid": "4"
        }
    ],
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

参照

- [Icon mapping](#)

ソース

ClconMap::create() in ui/include/classes/api/services/ClconMap.php.

iconmap.delete

説明

object iconmap.delete(array iconMapIds)

このメソッドではアイコンマップを削除できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するアイコンマップの ID

戻り値

(object) iconmapids プロパティの下にある削除されたアイコンマップの ID を含むオブジェクトを返します。

例

複数のアイコンマップの削除

アイコンマップを二つ削除します

Request:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "auth": "3a57200802b24cda67c4e4010b50c065",
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "iconmapids": [
      "2",
      "5"
    ]
  },
  "id": 1
}
```

ソース

ClconMap::delete() in ui/include/classes/api/services/ClconMap.php.

iconmap.get

説明

integer/array iconmap.get(object parameters)

このメソッドでは、指定されたパラメーターに従ってアイコンマップを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
iconmapids	string/array	指定された ID のアイコンマップのみを返します
sysmapids	string/array	指定されたマップで使用されているアイコンマップのみを返します
selectMappings	query	アイコンマッピングを使用して mappings プロパティを返します
sortfield	string/array	指定されたプロパティで結果を並べ替えます。 可能な値は iconmapid と name です。
countOutput	boolean	すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 で詳しく説明
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- **countOutput** パラメータが使用されている場合、取得されたオブジェクトの数

例

アイコンマップを取得する

アイコンマップ"3" に関するすべてのデータを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "iconmap.get",
  "params": {
    "iconmapids": "3",
    "output": "extend",
    "selectMappings": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "mappings": [
        {
          "iconmappingid": "3",
          "iconmapid": "3",
          "iconid": "6",
          "inventory_link": "1",
          "expression": "server",
          "sortorder": "0"
        },
        {
          "iconmappingid": "4",
          "iconmapid": "3",
          "iconid": "10",
          "inventory_link": "1",
          "expression": "switch",
          "sortorder": "1"
        }
      ],
      "iconmapid": "3",
      "name": "Host type icons",
      "default_iconid": "2"
    }
  ],
  "id": 1
}
```

参照

- [Icon mapping](#)

ソース

ClconMap::get() in ui/include/classes/api/services/ClconMap.php.

iconmap.update

説明

object iconmap.update(object/array iconMaps)

このメソッドをでは既存のアイコンマップを更新できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるアイコンマップのプロパティ

iconmapid プロパティはアイコンマップごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準のアイコンマッププロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
mappings	array	既存のアイコンマッピングを置き換えます

戻り値

(object) iconmapids プロパティの下にある更新されたアイコンマップの ID を含むオブジェクトを返します

例

アイコンマップの名前変更

アイコンマップの名前を"OS アイコン"に変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "iconmap.update",
  "params": {
    "iconmapid": "1",
    "name": "OS icons"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "iconmapids": [
      "1"
    ]
  },
  "id": 1
}
```

参照

- [Icon mapping](#)

ソース

ClconMap::update() in ui/include/classes/api/services/ClconMap.php.

Image

このクラスでは、画像を処理できます。

オブジェクトリファレンス

- [Image](#)

利用可能なメソッド:

- [image.create](#) - 新しい画像の作成
- [image.delete](#) - 画像の削除
- [image.get](#) - 画像の取得
- [image.update](#) - 画像の更新

> Image object

次のオブジェクトは、imageAPI に直接関連しています。

Image

image オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
imageid	string	(読み専用) 画像の ID
name (必須)	string	画像の名前
imagetype	integer	画像タイプ 値: 1 - (デフォルト) アイコン 2 - 背景画像

image.create

Description

object image.create(object/array images)

このメソッドでは、新しい画像を作成できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 画像の作成

標準の画像プロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
name (必須)	string	画像の名前
imagetype (必須)	integer	画像のタイプ 値: 1 - (デフォルト) アイコン 2 - 背景画像
image (必須)	string	Base64 でエンコードされた画像。エンコードされた画像の最大サイズは 1MB です。最大サイズは ZBX_MAX_IMAGE_SIZE 定数値で調整できます。サポートされている画像形式は、PNG、JPEG、GIF です。

戻り値

(object) imageids プロパティの下で作成された画像の ID を含むオブジェクトを返します。返される ID の順序は、渡された画像の順序と一致します。

例

画像の作成

クラウドアイコンを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "image.create",
  "params": {
    "imagetype": 1,
  }
}
```

```
    "name": "Cloud_(24)",
    "image": "iVBORw0KGgoAAAANSUHEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgE
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "imageids": [
      "188"
    ]
  },
  "id": 1
}
```

ソース

CImage::create() in ui/include/classes/api/services/CImage.php.

image.delete

説明

object image.delete(array imageIds)

このメソッドでは画像を削除できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除する画像の ID

戻り値

(object) imageids プロパティの下で削除された画像の ID を含むオブジェクトを返します。

例

複数の画像を削除

二つの画像を削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "image.delete",
  "params": [
    "188",
    "192"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "imageids": [
      "188",

```

```
        "192"  
    ],  
    },  
    "id": 1  
}
```

ソース

CImage::delete() in ui/include/classes/api/services/CImage.php.

image.get

説明

integer/array image.get(object parameters)

このメソッドでは、指定されたパラメーターに従って画像を取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします

パラメータ	タイプ	説明
imageids	string/array	指定された ID の画像のみを返します
sysmapids	string/array	指定されたマップで使用されている画像を返します
select_image	flag	Base64 でエンコードされた画像を含む image プロパティを返します
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	利用可能な値は <code>imageid</code> と <code>name</code> です。 すべての <code>get</code> メソッドに共通するこれらのパラメーターについては フレックス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得されたオブジェクトの数

例

画像を取得する

ID "2" の画像のすべてのデータを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "image.get",
  "params": {
    "output": "extend",
    "select_image": true,
    "imageids": "2"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "imageid": "2",
      "imagetype": "1",
      "name": "Cloud_(24)",
      "image": "iVBORwOKGgoAAAANSUHEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAA"
    }
  ],
  "id": 1
}
```

ソース

CImage::get() in ui/include/classes/api/services/CImage.php.

image.update

説明

object image.update(object/array images)

このメソッドでは、既存の画像を更新できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新する画像のプロパティ

imageid プロパティは画像ごとに定義する必要があり、プロパティはすべてオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

標準の画像プロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
image	string	Base64 でエンコードされた画像。エンコードされた画像の最大サイズは 1MB です。最大サイズは ZBX_MAX_IMAGE_SIZE 定数値で調整できます。サポートされている画像形式は、PNG、JPEG、GIF です。

戻り値

(object) imageids プロパティの下で更新された画像の ID を含むオブジェクトを返します。

例

画像の名前変更

画像の名前を "Cloud icon" に変更

Request:

```
{
  "jsonrpc": "2.0",
  "method": "image.update",
  "params": {
    "imageid": "2",
    "name": "Cloud icon"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "imageids": [
      "2"
    ]
  },
  "id": 1
}
```

ソース

CImage::update() in ui/include/classes/api/services/CImage.php.

Item

このクラスは、アイテムを操作するように設計されています。

オブジェクトリファレンス:

- [Item](#)

利用可能なメソッド:

- [item.create](#) - 新しいアイテムの作成
- [item.delete](#) - アイテムの削除
- [item.get](#) - アイテムの取得
- [item.update](#) - アイテムの更新

> Item object

次のオブジェクトは、item API に直接関連しています。

item

Note:

Zabbix API を使用して Web アイテムを直接作成、更新、または削除することはできません。

item オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
itemid	string	(読取専用) アイテム ID

プロパティ	タイプ	説明
delay (必須)	string	<p>アイテムの更新間隔。秒またはサフィックス付きの時間単位 (30s、1m、2h、1d) を受け入れます。</p> <p>オプションで、1 つまたは複数のカスタム間隔 を柔軟な間隔またはスケジューリングとして指定できます。</p> <p>複数の間隔はセミコロンで区切ります。</p> <p>ユーザーマクロを使用できます。1 つのマクロでフィールド全体を埋める必要があります。フィールド内の複数のマクロまたはテキストと混合されたマクロはサポートされていません。</p> <p>柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます (例: <code>{FLEX_INTERVAL}/{FLEX_PERIOD}</code>)。</p> <p>Zabbix トラッパー、依存アイテム、および <code>mqtt.get</code> キーを使用した Zabbix エージェント (アクティブ) のオプション。</p> <p>アイテムが属するホストまたはテンプレートの ID</p>
hostid (required)	string	<p>更新操作の場合、このフィールドは 読取専用です。</p> <p>アイテムのホストインターフェースの ID</p>
interfaceid (必須)	string	<p>ホストアイテムにのみ使用されます。Zabbix エージェント (アクティブ)、Zabbix インターナル、Zabbix トラッパー、計算、依存、データベースモニター、およびスクリプトアイテムには必要ありません。HTTP エージェントアイテムのオプションです。</p>
key_ (required)	string	アイテムキー
name (required)	string	アイテム名
type (required)	integer	<p>アイテムのタイプ</p> <p>使用可能な値:</p> <ul style="list-style-type: none"> 0 - Zabbix エージェント 2 - Zabbix トラッパー 3 - シンプルチェック 5 - Zabbix インターナル 7 - Zabbix エージェント (アクティブ) 9 - Web アイテム 10 - 外部チェック 11 - データベース監視 12 - IPMI エージェント 13 - SSH エージェント 14 - Telnet エージェント 15 - 計算済み 16 - JMX エージェント 17 - SNMP トラップ 18 - 依存アイテム 19 - HTTP エージェント 20 - SNMP エージェント 21 - スクリプト
url (required)	string	<p>URL 文字列。HTTP エージェントアイテムタイプにのみ必要です。ユーザーマクロ <code>{HOST.IP}</code>、<code>{HOST.CONN}</code>、<code>{HOST.DNS}</code>、<code>{HOST.HOST}</code>、<code>{HOST.NAME}</code>、<code>{ITEM.ID}</code>、<code>{ITEM.KEY}</code> をサポートします。</p>
value_type (必須)	integer	<p>アイテムの情報タイプ</p> <p>使用可能な値:</p> <ul style="list-style-type: none"> 0 - 浮動小数点数 1 - キャラクター 2 - ログ 3 - 符号なしの数値 4 - テキスト

プロパティ	タイプ	説明
allow_traps	integer	HTTP エージェントアイテムフィールド。トラッパーアイテムタイプと同様に、値の入力を許可します。 0 - (デフォルト) 着信データの受け入れを許可しません。 1 - 着信データの受け入れを許可します。
authtype	integer	SSH エージェントアイテムまたは HTTP エージェントアイテムでのみ使用されます。 SSH エージェント認証方法の使用可能な値: 0 - (デフォルト) パスワード 1 - 公開鍵 HTTP エージェント認証方式の使用可能な値: 0 - (デフォルト) なし 1 - Basic 2 - NTLM 3 - Kerberos
description	string	アイテムの説明
error	string	(読取専用) アイテムの更新に問題がある場合のエラーテキスト
flags	integer	(読取専用) アイテムの起源 使用可能な値: 0 - プレーンなアイテム 4 - 発見されたアイテム
follow_redirects	integer	HTTP エージェントアイテムフィールド。データのプール中に応答リダイレクトに従うか 0 - リダイレクトに従いません 1 - (デフォルト) リダイレクトに従います
headers	object	HTTP エージェントアイテムフィールド。ヘッダー名がキーとして使用され、ヘッダー値が値として使用される HTTP(S) リクエストヘッダーを持つオブジェクト。 例: { "User-Agent": "Zabbix" }
history	string	履歴データを保存する時間の単位。ユーザーマクロも受け入れます。 デフォルト: 90d
http_proxy	string	HTTP エージェントアイテムフィールド。HTTP(S) プロキシ接続文字列。
inventory_link	integer	アイテムによって入力されるホストインベントリフィールド ID サポートされているホストインベントリフィールドのリストについては、 ホストインベントリページ を参照してください。
ipmi_sensor	string	デフォルト: 0。 IPMI センサー。IPMI アイテムでのみ使用されます
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト値: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi
lastclock	timestamp	(読取専用) アイテムが最後に更新された時刻。 デフォルトでは、過去 24 時間以内の値のみが表示されます。管理 → 一般設定 メニューセクションのデータの保存期間パラメータ値を変更することで、この期間を延長できます。
lastns	integer	(読取専用) アイテムが最後に更新されたときのナノ秒 デフォルトでは過去 24 時間以内の値のみが表示されます。管理 → 一般設定 メニューセクションのデータの保存期間パラメータ値を変更することで、この期間を延長できます。

プロパティ	タイプ	説明
lastvalue	string	(読取専用) アイテムの最後の値 デフォルトでは過去 24 時間以内の値のみが表示されます。管理 → 一般設定 メニューセクションのデータの保存期間パラメータ値を変更することで、この期間を延長できます。
logtimefmt	string	ログエントリの時刻形式。ログアイテムでのみ使用されます。
master_itemid	integer	マスターアイテム ID 最大 3 つの従属アイテムの再帰と最大 29999 の従属アイテムが許可されます
output_format	integer	従属アイテムに必要です HTTP エージェントアイテムフィールド。レスポンスを JSON に変換する必要があります。 0 - (デフォルト) そのまま保存します 1 - JSON に変換します
params	string	アイテムのタイプに応じた追加パラメータ: - SSH および Telnet アイテムの実行スクリプト - データベース監視アイテムの SQL クエリ - 計算アイテムの式 - スクリプトのスクリプトアイテム
parameters	array	スクリプトアイテムの追加パラメーター。'name' および'value' プロパティを持つオブジェクトの配列。name は一意である必要があります
password	string	認証用のパスワード。簡易チェック、SSH、Telnet、データベース監視、JMX、HTTP エージェントのアイテムで使用
post_type	integer	HTTP エージェントアイテムフィールド。posts プロパティに格納される投稿データ本体のタイプ。 0 - (デフォルト) 生データ。 2 - JSON データ。 3 - XML データ。
posts	string	HTTP エージェントアイテムフィールド。HTTP(S) リクエストの本文データ。post_type で使用します。
prevvalue	string	(読取専用) アイテムの以前の値
privatekey	string	秘密鍵ファイル名
publickey	string	公開鍵ファイル名
query_fields	array	HTTP エージェントアイテムフィールド。クエリパラメータ。'キー': '値' のペアを持つオブジェクトの配列。値は空の文字列にすることができます。
request_method	integer	HTTP エージェントアイテムフィールド。リクエストメソッドのタイプ。 0 - (デフォルト) GET 1 - POST 2 - PUT 3 - HEAD
retrieve_mode	integer	HTTP エージェントアイテムフィールド。レスポンスのどの部分を保存するか。 0 - (デフォルト) 本文 1 - ヘッダー 2 - 本文とヘッダーの両方
snmp_oid	string	request_method HEAD の場合、許可される値は 1 のみ SNMP OID。
ssl_cert_file	string	HTTP エージェントアイテムフィールド。公開 SSL キーファイルパス
ssl_key_file	string	HTTP エージェントアイテムフィールド。プライベート SSL キーファイルパス

プロパティ	タイプ	説明
ssl_key_password	string	HTTP エージェントアイテムフィールド。SSL キーファイルのパスワード
state	integer	(読取専用) アイテムの状態 使用可能な値: 0 - (デフォルト) 通常 1 - サポートされていません
status	integer	アイテムのステータス 使用可能な値: 0 - (デフォルト) 有効なアイテム 1 - 無効なアイテム
status_codes	string	HTTP エージェントアイテムフィールド。コンマで区切られた必須の HTTP ステータスコードの範囲。コンマ区切りリストの一部としてユーザーマクロもサポートします。
templateid	string	例: 200,200-{\$M},{M},200-400 (読取専用) 親テンプレートアイテムの ID。
timeout	string	ヒント: hostid プロパティを使用して、アイテムが属するテンプレートを指定します。 アイテムデータポーリングリクエストのタイムアウト。HTTP エージェントおよびスクリプトアイテムに使用されます。ユーザーマクロをサポートします。
trapper_hosts	string	デフォルト: 3 秒 最大値: 60 秒 許可されたホスト。トラッパーアイテムまたは HTTP エージェントアイテムで使用されます。
trends	string	トレンドデータを保存する時間の単位。ユーザーマクロも受け入れます。
units	string	デフォルト: 365d 値の単位
username	string	認証用のユーザー名。簡易チェック、SSH、Telnet、データベース監視、JMX、HTTP エージェントのアイテムで使用
uuid	string	SSH、Telnet のアイテムで必須 JMX で使用する場合、パスワードもユーザー名と一緒に指定するか、両方のプロパティを空白のままにする必要があります。 インポートされたアイテムを既存のアイテムにリンクするために使用されるユニバーサル意識別子。テンプレートのアイテムにのみ使用されます。指定されていない場合は自動生成されます。
valuemapid	string	更新操作の場合、このフィールドは 読取専用です。 関連付けられた値マップ ID
verify_host	integer	HTTP エージェントアイテムフィールド。URL のホスト名がホスト証明書の Common Name フィールドまたは Subject Alternate Name フィールドにあることを検証します。
verify_peer	integer	0 - (デフォルト) 検証しません。 1 - 検証します。 HTTP エージェントアイテムフィールド。検証はホスト証明書の認証です。 0 - (デフォルト) 検証しません。 1 - 検証します。

Item tag

item tag オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	アイテムタグ名
value	string	アイテムタグ値

アイテムの前処理

アイテム前処理オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
type (必須)	integer	前処理オプションのタイプ。 使用可能な値: 1 - カスタム乗数 2 - 右トリム 3 - 左トリム 4 - トリム 5 - 正規表現マッチング 6 - ブール値から 10 進数へ 7 - 8 進数から 10 進数へ 8 - 16 進数から 10 進数へ 9 - 単純な変更 10 - 1 秒あたりの変更 11 - XML XPath 12 - JSONPath 13 - 範囲内 14 - 正規表現に一致 15 - 正規表現に不一致 16 - JSON のエラーチェック 17 - XML のエラーチェック; 18 - 正規表現を使用したエラーチェック 19 - 変更なしで破棄 20 - ハートビートで変更されていないものを破棄 21 - JavaScript 22 - Prometheus パターン 23 - Prometheus から JSON へ 24 - CSV から JSON へ 25 - 置換 26 - サポートされていないチェック 27 - XML から JSON へ
params (必須)	string	前処理オプションで使用される追加パラメーター。複数のパラメータは LF (\n) 文字で区切られます。
error_handler (必須)	integer	前処理ステップが失敗した場合に使用されるアクションタイプ 使用可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定されます。 1 - 値を破棄します。 2 - カスタム値を設定します。 3 - カスタムエラーメッセージを設定します。
error_handler_params (必須)	string	エラーハンドラーのパラメーター。error_handler とともに使用されます。 error_handler が 0 または 1 の場合、空にする必要があります。 error_handler が 2 の場合、空にすることができます。 error_handler が 3 の場合、空にすることはできません。

前処理の種類ごとに、次のパラメーターとエラーハンドラーがサポートされています。

前処理の種類	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートする
1	カスタム乗数	number ^{1,6}			0, 1, 2, 3
2	右トリム	list of characters ²			

前処理の種類	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートする
3	左トリム	list of characters ²			
4	トリム	list of characters ²			
5	正規表現マッチング	pattern ³	output ²		0, 1, 2, 3
6	ブール値から 10 進数へ				0, 1, 2, 3
7	8 進数から 10 進数へ				0, 1, 2, 3
8	16 進数から 10 進数へ				0, 1, 2, 3
9	単純な変更				0, 1, 2, 3
10	一秒あたりの変更				0, 1, 2, 3
11	XML XPath	path ⁴			0, 1, 2, 3
12	JSONPath	path ⁴			0, 1, 2, 3
13	範囲内	min ^{1, 6}	max ^{1, 6}		0, 1, 2, 3
14	正規表現に一致	pattern ³			0, 1, 2, 3
15	正規表現に不一致	pattern ³			0, 1, 2, 3
16	JSON のエラーチェック	path ⁴			0, 1, 2, 3
17	XML のエラーチェック	path ⁴			0, 1, 2, 3
18	正規表現を使用したエラーチェック	pattern ³	output ²		0, 1, 2, 3
19	変更なしで破棄				
20	ハートビートで変更されていないものを破棄	seconds ^{5, 6}			
21	JavaScript	script ²			
22	Prometheus パターン	pattern ^{6, 7}	value, label, function	output ^{8, 9}	0, 1, 2, 3
23	Prometheus から JSON へ	pattern ^{6, 7}			0, 1, 2, 3
24	CSV から JSON へ	character ²	character ²	0,1	0, 1, 2, 3
25	置換	search string ²	replacement ²		
26	サポートされていないチェック				1, 2, 3
27	XML から JSON へ				0, 1, 2, 3

¹ 整数または浮動小数点数

² 文字列

³ 正規表現

⁴ JSONPath または XML XPath

⁵ 正の整数 (タイムサフィックスをサポート, 例: 30s, 1m, 2h, 1d)

⁶ ユーザーマクロ

⁷ 構文に従う Prometheus パターン: <metric name>{<label name>=<label value>, ...} == <value> 各 Prometheus パターンコンポーネント (メトリック、ラベル名、ラベル値、およびメトリック値) は、ユーザーマクロにすることができます。

⁸ Prometheus は次の構文で出力します: <label name> (ユーザーマクロを使用できます) label が 2 番目のパラメーターとして選択されている場合。

⁹ 集計関数の 1 つ: sum,min,max,avg,count (function が 2 番目のパラメーターとして選択されている場合)

item.create

説明

object item.create(object/array items)

このメソッドを使用すると、新しいアイテムを作成できます。

Note:

Zabbix API を介して Web アイテムを作成することはできません。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するアイテム

このメソッドは標準アイテムプロパティに加えて、次のパラメータを受け入れます。

パラメータ	タイプ	説明
preprocessing	array	アイテムの事前処理 オプション
tags	array	アイテムタグ

戻り値

(object) itemids プロパティの下で作成されたアイテムの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムの順序と一致します。

例

アイテムの作成

ID "30074" のホストの空きディスク容量を監視するために、2 つのアイテムタグを持つ数値の Zabbix エージェントアイテムを作成します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "name": "Free disk space on /home/joe/",
    "key_": "vfs.fs.size[/home/joe/,free]",
    "hostid": "30074",
    "type": 0,
    "value_type": 3,
    "interfaceid": "30084",
    "tags": [
      {
        "tag": "Disk usage"
      },
      {
        "tag": "Equipment",
        "value": "Workstation"
      }
    ],
    "delay": "30s"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "24758"
    ]
  },
  "id": 1
}
```

ホストインベントリアイテムの作成

ホストの"OS" インベントリフィールドに入力する Zabbix エージェントアイテムを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "name": "uname",
    "key_": "system.uname",

```



```
    "hostid": "30021",
    "type": 0,
    "interfaceid": "30007",
    "value_type": 1,
    "delay": "10s",
    "inventory_link": 5
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "24759"
    ]
  },
  "id": 1
}
```

事前処理を伴うアイテムの作成

乗数を使用してアイテムを作成します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "name": "Device uptime",
    "key_": "sysUpTime",
    "hostid": "11312",
    "type": 4,
    "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
    "value_type": 1,
    "delay": "60s",
    "units": "uptime",
    "interfaceid": "1156",
    "preprocessing": [
      {
        "type": 1,
        "params": "0.01",
        "error_handler": 1,
        "error_handler_params": ""
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44210"
    ]
  },
  "id": 1
}
```

依存アイテムの作成

ID 24759 のマスターアイテムの依存アイテムを作成します。同じホストへの依存のみが許可されるため、マスターと依存アイテムは同じホスト ID を持つ必要があります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "hostid": "30074",
    "name": "Dependent test item",
    "key_": "dependent.item",
    "type": 18,
    "master_itemid": "24759",
    "value_type": 2
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44211"
    ]
  },
  "id": 1
}
```

HTTP エージェントアイテムの作成

JSON レスポンスの事前処理で POST リクエストメソッドアイテムを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "url": "http://127.0.0.1/http.php",
    "query_fields": [
      {
        "mode": "json"
      },
      {
        "min": "10"
      },
      {
        "max": "100"
      }
    ],
    "interfaceid": "1",
    "type": 19,
    "hostid": "10254",
    "delay": "5s",
    "key_": "json",
    "name": "HTTP agent example JSON",
    "value_type": 0,
    "output_format": 1,
    "preprocessing": [
      {
        "type": 12,

```

```

        "params": "$.random",
        "error_handler": 0,
        "error_handler_params": ""
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 2
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23865"
    ]
  },
  "id": 3
}

```

スクリプトアイテムの作成

スクリプトアイテムを使用して簡単なデータコレクションを作成します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "name": "Script example",
    "key_": "custom.script.item",
    "hostid": "12345",
    "type": 21,
    "value_type": 4,
    "params": "var request = new CurlHttpRequest();\nreturn request.Post(\"https://postman-echo.com/post\");",
    "parameters": [
      {
        "name": "host",
        "value": "{HOST.CONN}"
      }
    ]
  },
  "timeout": "6s",
  "delay": "30s"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 2
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23865"
    ]
  },
  "id": 3
}

```

ソース

CItem::create() in ui/include/classes/api/services/CItem.php.

item.delete

説明

object item.delete(array itemIds)

このメソッドを使用すると、アイテムを削除できます。

Note:

Zabbix API を使用して Web アイテムを削除することはできません。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するアイテムの ID

戻り値

(object) itemids プロパティの下で削除されたアイテムの ID を含むオブジェクトを返します。

例

複数のアイテムを削除する

2つのアイテムを削除します。マスターアイテムが削除されると、依存アイテムとアイテムプロトタイプは自動的に削除されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.delete",
  "params": [
    "22982",
    "22986"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "22982",
      "22986"
    ]
  },
  "id": 1
}
```

ソース

CItem::delete() in ui/include/classes/api/services/CItem.php.

item.get

説明

integer/array item.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従ってアイテムを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートしています。

パラメータ	タイプ	説明
itemids	string/array	指定された ID を持つアイテムのみを返します
groupids	string/array	指定されたグループのホストに属するアイテムのみを返します
templateids	string/array	指定されたテンプレートに属するアイテムのみを返します
hostids	string/array	指定されたホストに属するアイテムのみを返します
proxyids	string/array	指定されたプロキシによって監視されているアイテムのみを返します
interfaceids	string/array	指定されたホストインターフェイスを使用するアイテムのみを返します
graphids	string/array	指定されたグラフで使用されているアイテムのみを返します
triggerids	string/array	指定されたトリガーで使用されるアイテムのみを返します
webitems	flag	結果に Web アイテムを含めます
inherited	boolean	"true" に設定すると、テンプレートから継承されたアイテムのみが返されます。
templated	boolean	"true" に設定すると、テンプレートに属するアイテムのみが返されます。
monitored	boolean	"true" に設定すると、監視対象ホストに属する有効なアイテムのみが返されます。
group	string	指定された名前のグループに属するアイテムのみを返します
host	string	指定された名前のホストに属するアイテムのみを返します
evaltype	integer	タグ検索のルール
tags	array of objects	<p>使用可能な値: 0 - (デフォルト) And/Or 2 - Or</p> <p>指定されたタグを持つアイテムのみを返します。タグによる完全一致と、演算子の値に応じたタグ値による大文字と小文字の区別または区別なしの検索。</p> <p>フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]</p> <p>空の配列はすべての項目を返します。</p> <p>使用可能な演算子: 0 - (デフォルト) Like 1 - 等しい 2 - Not Like 3 - 等しくない 4 - 存在する 5 - 存在しない</p>
with_triggers	boolean	"true" に設定すると、トリガーで使用されるアイテムのみが返されます。
selectHosts	query	項目が属するホストの配列を持つ hosts プロパティを返します。
selectInterfaces	query	アイテムが使用するホストインターフェイスの配列を含む interfaces プロパティを返します
selectTriggers	query	アイテムが使用されているトリガーを含む triggers プロパティを返します。
selectGraphs	query	count をサポートします。 アイテムを含むグラフの graphs プロパティを返します。
selectDiscoveryRule	query	count をサポートします。 アイテムを作成した LLD ルールを含む discoveryRule プロパティを返します。
selectItemDiscovery	query	アイテムディスカバリオブジェクトを含む itemDiscovery プロパティを返します。

パラメータ	タイプ	説明
アイテムディスカバリ オブジェクトは、アイ テムをそのアイテムの 作成元のアイテムプロ トタイプにリンクしま す。		
以下のプロパティがあ ります。		
itemdiscoveryid - (string) アイテムデ ィスカバリの ID		
itemid - (string) 発 見されたアイテムの ID		
parent_itemid - (string) アイテムが 作成されたアイテムプロ トタイプの ID		
key_ - (string) アイ テムプロトタイプのキ ー		
lastcheck - (timestamp) アイテ ムが最後に発見された 時間		
ts_delete - (timestamp) 検出さ れなくなったアイテム が削除される時刻		

パラメータ	タイプ	説明
selectPreprocessing	query	<p>アイテムの自演処理オプションを含む preprocessing プロパティを返しません。</p> <p>次のプロパティがあります。 type - (string) 事前処理オプションタイプ: 1 - カスタム乗数。 2 - 右トリム。 3 - 左トリム。 4 - トリム; 5 - 正規表現マッチング。 6 - ブール値から 10 進数。 7 - 8 進数から 10 進数。 8 - 16 進数から 10 進数。 9 - 簡単な変更。 10 - 1 秒ごとに変化します。 11 - XML XPath 12 - JSONPath 13 - 範囲内。 14 - 正規表現に一致。 15 - 正規表現と一致しません。 16 - JSON のエラーをチェックします。 17 - XML のエラーをチェックします。 18 - 正規表現を使用してエラーをチェックします。 19 - 変更せずに破棄します。 20 - ハートビートで変更されていないものを破棄します。 21 - JavaScript; 22 - プロメテウスパターン。 23 - プロメテウスから JSON へ。 24 - CSV から JSON; 25 - 交換; 26 - サポートされていない値をチェックします。 27 - XML から JSON へ。</p> <p>params - (string) 前処理オプションで使用される追加パラメータ。複数のパラメータは、LF (\n) 文字で区切られます。 error_handler - (文字列) 前処理ステップが失敗した場合に使用されるアクションタイプ: 0 - エラーメッセージは Zabbix サーバーによって設定されます。 1 - 値を破棄します。 2 - カスタム値を設定します。 3 - カスタムエラーメッセージを設定します。</p> <p>error_handler_params - (string) エラーハンドラのパラメータ。 tags プロパティでアイテムタグを返します。 valuemap プロパティをアイテム値マップと共に返します。 指定されたフィルタに正確に一致する結果のみを返します。</p> <p>配列を受け入れます。キーはプロパティ名で、値は単一の値または照合する値の配列です。
 追加のフィルタをサポート: host - アイテムが属するホストの技術名。 サブセレクトによって返されるレコードの数を制限します。</p> <p>次のサブセレクトに適用されます: selectGraphs - 結果は name でソートされます; selectTriggers - 結果は description でソートされます。 指定されたプロパティで結果を並べ替えます。</p> <p>使用可能な値は itemid,name,key_,delay,history,trends,type,status です。 すべての get メソッドに共通するこれらのパラメータについては リファレンス解説 ページで詳しく説明されています。</p>
selectTags	query	
selectValueMap	query	
filter	object	
limitSelects	integer	
sortfield	string/array	
countOutput	boolean	

パラメータ	タイプ	説明
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

キーによる項目の検索

トリガー固有のホスト ID で使用され、アイテムキーに"system.cpu" という単語が含まれるすべてのアイテムを取得し、結果を名前で並べ替えます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "output": "extend",
    "hostids": "10084",
    "with_triggers": true,
    "search": {
      "key_": "system.cpu"
    },
    "sortfield": "name"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "42269",
      "type": "18",
      "snmp_oid": "",
      "hostid": "10084",
      "name": "CPU utilization",
      "key_": "system.cpu.util",
      "delay": "0",
      "history": "7d",
      "trends": "365d",
      "status": "0",
      "value_type": "0",
      "trapper_hosts": "",
      "units": "%",
      "logtimefmt": "",
      "templateid": "42267",
      "valuemapid": "0",

```



```

"params": "",
"ipmi_sensor": "",
"authtype": "0",
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"flags": "0",
"interfaceid": "0",
"description": "CPU utilization in %.",
"inventory_link": "0",
"evaltype": "0",
"jmx_endpoint": "",
"master_itemid": "42264",
"timeout": "3s",
"url": "",
"query_fields": [],
"posts": "",
"status_codes": "200",
"follow_redirects": "1",
"post_type": "0",
"http_proxy": "",
"headers": [],
"retrieve_mode": "0",
"request_method": "0",
"output_format": "0",
"ssl_cert_file": "",
"ssl_key_file": "",
"ssl_key_password": "",
"verify_peer": "0",
"verify_host": "0",
"allow_traps": "0",
"uuid": "",
"state": "0",
"error": "",
"parameters": [],
"lastclock": "0",
"lastns": "0",
"lastvalue": "0",
"prevvalue": "0"
},
{
"itemid": "42259",
"type": "0",
"snmp_oid": "",
"hostid": "10084",
"name": "Load average (15m avg)",
"key_": "system.cpu.load[all,avg15]",
"delay": "1m",
"history": "7d",
"trends": "365d",
"status": "0",
"value_type": "0",
"trapper_hosts": "",
"units": "",
"logtimefmt": "",
"templateid": "42219",
"valuemapid": "0",
"params": "",
"ipmi_sensor": "",
"authtype": "0",
"username": "",

```

```

    "password": "",
    "publickey": "",
    "privatekey": "",
    "flags": "0",
    "interfaceid": "1",
    "description": "",
    "inventory_link": "0",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "0",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": [],
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0"
  },
  {
    "itemid": "42249",
    "type": "0",
    "snmp_oid": "",
    "hostid": "10084",
    "name": "Load average (1m avg)",
    "key_": "system.cpu.load[all,avg1]",
    "delay": "1m",
    "history": "7d",
    "trends": "365d",
    "status": "0",
    "value_type": "0",
    "trapper_hosts": "",
    "units": "",
    "logtimefmt": "",
    "templateid": "42209",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "flags": "0",

```

```

"interfaceid": "1",
"description": "",
"inventory_link": "0",
"evaltype": "0",
"jmx_endpoint": "",
"master_itemid": "0",
"timeout": "3s",
"url": "",
"query_fields": [],
"posts": "",
"status_codes": "200",
"follow_redirects": "1",
"post_type": "0",
"http_proxy": "",
"headers": [],
"retrieve_mode": "0",
"request_method": "0",
"output_format": "0",
"ssl_cert_file": "",
"ssl_key_file": "",
"ssl_key_password": "",
"verify_peer": "0",
"verify_host": "0",
"allow_traps": "0",
"uuid": "",
"state": "0",
"error": "",
"parameters": [],
"lastclock": "0",
"lastns": "0",
"lastvalue": "0",
"prevvalue": "0"
},
{
"itemid": "42257",
"type": "0",
"snmp_oid": "",
"hostid": "10084",
"name": "Load average (5m avg)",
"key_": "system.cpu.load[all,avg5]",
"delay": "1m",
"history": "7d",
"trends": "365d",
"status": "0",
"value_type": "0",
"trapper_hosts": "",
"units": "",
"logtimefmt": "",
"templateid": "42217",
"valuemapid": "0",
"params": "",
"ipmi_sensor": "",
"authtype": "0",
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"flags": "0",
"interfaceid": "1",
"description": "",
"inventory_link": "0",
"evaltype": "0",

```

```

    "jmx_endpoint": "",
    "master_itemid": "0",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": [],
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0"
  },
  {
    "itemid": "42260",
    "type": "0",
    "snmp_oid": "",
    "hostid": "10084",
    "name": "Number of CPUs",
    "key_": "system.cpu.num",
    "delay": "1m",
    "history": "7d",
    "trends": "365d",
    "status": "0",
    "value_type": "3",
    "trapper_hosts": "",
    "units": "",
    "logtimefmt": "",
    "templateid": "42220",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "flags": "0",
    "interfaceid": "1",
    "description": "",
    "inventory_link": "0",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "0",
    "timeout": "3s",
    "url": "",

```

```

    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": [],
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0"
  }
],
  "id": 1
}

```

キーによる依存アイテムの検索

ID が"10116" で、キーに"apache" という単語が含まれるすべての依存アイテムをホストから取得します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "output": "extend",
    "hostids": "10116",
    "search": {
      "key_": "apache"
    },
    "filter": {
      "type": 18
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "25550",
      "type": "18",
      "snmp_oid": "",
      "hostid": "10116",
      "name": "Days",
      "key_": "apache.status.uptime.days",
      "delay": "0",

```

```

    "history": "90d",
    "trends": "365d",
    "status": "0",
    "value_type": "3",
    "trapper_hosts": "",
    "units": "",
    "logtimefmt": "",
    "templateid": "0",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "flags": "0",
    "interfaceid": "0",
    "description": "",
    "inventory_link": "0",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "25545",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": [],
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0"
  },
  {
    "itemid": "25555",
    "type": "18",
    "snmp_oid": "",
    "hostid": "10116",
    "name": "Hours",
    "key_": "apache.status.uptime.hours",
    "delay": "0",
    "history": "90d",
    "trends": "365d",
    "status": "0",
    "value_type": "3",

```

```

    "trapper_hosts": "",
    "units": "",
    "logtimefmt": "",
    "templateid": "0",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "flags": "0",
    "interfaceid": "0",
    "description": "",
    "inventory_link": "0",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "25545",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": [],
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0"
  }
],
  "id": 1
}

```

HTTP エージェントアイテムを探す

特定のホスト ID の投稿本文タイプが XML の HTTP エージェントアイテムを検索します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "hostids": "10255",
    "filter": {
      "type": 19,
      "post_type": 3
    }
  }
}

```

```
    }  
  },  
  "id": 3,  
  "auth": "d678e0b85688ce578ff061bd29a20d3b"  
}
```

Response:

```
{  
  "jsonrpc": "2.0",  
  "result": [  
    {  
      "itemid": "28252",  
      "type": "19",  
      "snmp_oid": "",  
      "hostid": "10255",  
      "name": "template item",  
      "key_": "ti",  
      "delay": "30s",  
      "history": "90d",  
      "trends": "365d",  
      "status": "0",  
      "value_type": "3",  
      "trapper_hosts": "",  
      "units": "",  
      "logtimefmt": "",  
      "templateid": "0",  
      "valuemapid": "0",  
      "params": "",  
      "ipmi_sensor": "",  
      "authtype": "0",  
      "username": "",  
      "password": "",  
      "publickey": "",  
      "privatekey": "",  
      "flags": "0",  
      "interfaceid": "0",  
      "description": "",  
      "inventory_link": "0",  
      "evaltype": "0",  
      "jmx_endpoint": "",  
      "master_itemid": "0",  
      "timeout": "3s",  
      "url": "localhost",  
      "query_fields": [  
        {  
          "mode": "xml"  
        }  
      ],  
      "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",  
      "status_codes": "200",  
      "follow_redirects": "0",  
      "post_type": "3",  
      "http_proxy": "",  
      "headers": [],  
      "retrieve_mode": "1",  
      "request_method": "3",  
      "output_format": "0",  
      "ssl_cert_file": "",  
      "ssl_key_file": "",  
      "ssl_key_password": "",  
      "verify_peer": "0",  
      "verify_host": "0",  
    }  
  ]  
}
```



```

        "allow_traps": "0",
        "uuid": "",
        "state": "0",
        "error": "",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "",
        "prevvalue": ""
    }
],
    "id": 3
}

```

事前処理ルールを使用してアイテムを取得する

特定のホスト ID のすべてのアイテムとその事前処理ルールを取得します。

Request:

```

{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}

```

参照

- [Discovery rule](#)
- [Graph](#)
- [Host](#)
- [Host interface](#)
- [Trigger](#)

ソース

CItem::get() in ui/include/classes/api/services/CItem.php.

item.update

説明

object item.update(object/array items)

このメソッドを使用すると、既存のアイテムを更新できます。

Note:

Zabbix API を介して Web アイテムを更新することはできません。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するアイテムプロパティ

itemid プロパティはアイテムごとに定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準アイテムプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
preprocessing	array	現在の事前処理オプションを置き換える アイテム事前処理オプション
tags	array	アイテムタグ

戻り値

(object) itemids プロパティの下で更新されたアイテムの ID を含むオブジェクトを返します。

例

アイテムの有効化

ステータスを"0" に設定してアイテムを有効化します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.update",
  "params": {
    "itemid": "10092",
    "status": 0
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "10092"
    ]
  },
  "id": 1
}
```

依存アイテムの更新

依存アイテム名とマスターアイテム ID を更新します。同じホストへの依存関係のみが許可されるため、マスターと依存アイテムは同じホスト ID を持つ必要があります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.update",
  "params": {
    "name": "Dependent item updated name",
    "master_itemid": "25562",
    "itemid": "189019"
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "189019"
    ]
  },
  "id": 1
}
```

HTTP エージェントアイテムの更新

アイテム値トラップを有効にします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.update",
  "params": {
    "itemid": "23856",
    "allow_traps": 1
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23856"
    ]
  },
  "id": 1
}
```

事前処理によるアイテムの更新

アイテム事前処理ルール“In range” でアイテムを更新します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.update",
  "params": {
    "itemid": "23856",
    "preprocessing": [
      {
        "type": 13,
        "params": "\n100",
      }
    ]
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

```
        "error_handler": 1,
        "error_handler_params": ""
    }
]
},
"auth": "700ca65537074ec963db7efabda78259",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23856"
    ]
  },
  "id": 1
}
```

スクリプトアイテムの更新

別のスクリプトでスクリプトアイテムを更新し、以前のスクリプトで使用された不要なパラメーターを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "item.update",
  "params": {
    "itemid": "23865",
    "parameters": [],
    "script": "Zabbix.Log(3, 'Log test');\nreturn 1;"
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23865"
    ]
  },
  "id": 1
}
```

ソース

CItem::update() in ui/include/classes/api/services/CItem.php.

Item prototype

このクラスは、アイテムのプロトタイプで動作するように設計されています。

オブジェクトリファレンス:

- [Item prototype](#)

利用可能なメソッド:

- [itemprototype.create](#) - 新しいアイテムプロトタイプを作成する

- `itemprototype.delete` - アイテムプロトタイプ削除
- `itemprototype.get` - アイテムのプロトタイプの取得
- `itemprototype.update` - アイテムのプロトタイプの更新

> Item prototype object

次のオブジェクトは、`itemprototype` API に直接関連しています。

Item prototype

item prototype オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
<code>itemid</code>	string	(読取専用) アイテムプロトタイプ ID
<code>delay</code> (必須)	string	アイテムプロトタイプの更新間隔。秒またはサフィックス付きの時間単位 (30s、1m、2h、1d) を受け入れます。 オプションで、1 つまたは複数の カスタム間隔 を柔軟な間隔またはスケジューリングとして指定できます。 複数の間隔はセミコロンで区切ります。 ユーザーマクロを使用できます。1 つのマクロでフィールド全体を埋める必要があります。フィールド内の複数のマクロまたはテキストと混合されたマクロはサポートされていません。 柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます (例: <code>{ \$FLEX_INTERVAL } / { \$FLEX_PERIOD }</code>)。 Zabbix トラッパー、依存アイテム、および <code>mqtt.get</code> キーを使用した Zabbix エージェント (アクティブ) のオプション。 アイテムプロトタイプが属するホストまたはテンプレートの ID
<code>hostid</code> (required)	string	更新操作の場合、このフィールドは 読取専用です。 アイテムが属する LLD ルールの ID。
<code>ruleid</code> (required)	string	更新操作の場合、このフィールドは 読取専用です。 アイテムプロトタイプのホストインターフェースの ID
<code>interfaceid</code> (必須)	string	ホストアイテムプロトタイプにのみ使用されます。Zabbix エージェント (アクティブ)、Zabbix インターナル、Zabbix トラッパー、計算、依存、データベースモニター、およびスクリプトアイテムには必要ありません。 HTTP エージェントアイテムのオプションです。
<code>key_</code> (required)	string	アイテムプロトタイプキー
<code>name</code> (required)	string	アイテムプロトタイプ名
<code>type</code> (required)	integer	
<code>url</code> (required)	string	URL 文字列。HTTP エージェントアイテムプロトタイプタイプにのみ必要です。LLD マクロとユーザーマクロ <code>{HOST.IP}</code> 、 <code>{HOST.CONN}</code> 、 <code>{HOST.DNS}</code> 、 <code>{HOST.HOST}</code> 、 <code>{HOST.NAME}</code> 、 <code>{ITEM.ID}</code> 、 <code>{ITEM.KEY}</code> をサポートします。
<code>value_type</code> (必須)	integer	アイテムプロトタイプの情報タイプ 使用可能な値: 0 - 浮動小数点数 1 - キャラクター 2 - ログ 3 - 符号なしの数値 4 - テキスト
<code>allow_traps</code>	integer	HTTP エージェント項目フィールド。トラッパーアイテムタイプと同様に、値の入力を許可します。 0 - (デフォルト) 着信データの受け入れを許可しません。 1 - 着信データの受け入れを許可します。

プロパティ	タイプ	説明
authtype	integer	SSH エージェントアイテムまたは HTTP エージェントアイテムでのみ使用されます。 SSH エージェント認証方法の使用可能な値: 0 - (デフォルト) パスワード 1 - 公開鍵 HTTP エージェント認証方式の使用可能な値: 0 - (デフォルト) なし 1 - Basic 2 - NTLM 3 - Kerberos
description	string	アイテムプロトタイプの説明
follow_redirects	integer	HTTP エージェントアイテムプロトタイプフィールド。データのプール中に応答リダイレクトに従うか 0 - リダイレクトに従いません 1 - (デフォルト) リダイレクトに従います
headers	object	HTTP エージェントアイテムプロトタイプフィールド。ヘッダー名がキーとして使用され、ヘッダー値が値として使用される、HTTP(S) 要求ヘッダーを持つオブジェクト。 例: { "User-Agent": "Zabbix" }
history	string	履歴データを保存する時間の単位。ユーザーマクロと LLD マクロを受け入れます。 デフォルト: 90d
http_proxy	string	HTTP エージェントアイテムプロトタイプフィールド。HTTP(S) プロキシ接続文字列。
ipmi_sensor	string	IPMI センサー。IPMI アイテムプロトタイプでのみ使用されます
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト値: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi
logtimefmt	string	ログエントリの時刻形式。ログアイテムプロトタイプでのみ使用されません。
master_itemid	integer	マスターアイテム ID 最大 3 つの従属アイテムプロトタイプの再帰と最大 29999 の従属アイテムプロトタイプが許可されます
output_format	integer	従属アイテムに必要です HTTP エージェントアイテムプロトタイプフィールド。レスポンスを JSON に変換する必要があります。 0 - (デフォルト) そのまま保存します 1 - JSON に変換します
params	string	アイテムプロトタイプのタイプに応じた追加パラメータ: - SSH および Telnet アイテムプロトタイプの実行スクリプト - データベース監視アイテムプロトタイプの SQL クエリ - 計算アイテムプロトタイプの式 - スクリプトのスクリプトアイテムプロトタイプ
parameters	array	スクリプトアイテムプロトタイプの追加パラメーター。'name' および 'value' プロパティを持つオブジェクトの配列。name は一意である必要があります
password	string	認証用のパスワード。簡易チェック、SSH、Telnet、データベース監視、JMX、HTTP エージェントのアイテムプロトタイプで使用

プロパティ	タイプ	説明
post_type	integer	HTTP エージェントアイテムプロトタイプフィールド。posts プロパティに格納される投稿データ本体のタイプ。 0 - (デフォルト) 生データ。 2 - JSON データ。 3 - XML データ。
posts	string	HTTP エージェントアイテムプロトタイプフィールド。HTTP(S) リクエストの本文データ。post_type で使用します。
privatekey	string	秘密鍵ファイル名
publickey	string	公開鍵ファイル名
query_fields	array	HTTP エージェントアイテムプロトタイプフィールド。クエリパラメータ。'キー': '値' のペアを持つオブジェクトの配列。値は空の文字列にすることができます。
request_method	integer	HTTP エージェントアイテムプロトタイプフィールド。リクエストメソッドのタイプ。 0 - (デフォルト) GET 1 - POST 2 - PUT 3 - HEAD
retrieve_mode	integer	HTTP エージェントアイテムプロトタイプフィールド。レスポンスのどの部分を保存するか。 0 - (デフォルト) 本文 1 - ヘッダー 2 - 本文とヘッダーの両方
snmp_oid	string	request_method HEAD の場合、許可される値は 1 のみ SNMP OID。
ssl_cert_file	string	HTTP エージェントアイテムプロトタイプフィールド。公開 SSL キーファイルパス
ssl_key_file	string	HTTP エージェントアイテムプロトタイプフィールド。プライベート SSL キーファイルパス
ssl_key_password	string	HTTP エージェントアイテムプロトタイプフィールド。SSL キーファイルのパスワード
status	integer	アイテムプロトタイプのステータス 使用可能な値: 0 - (デフォルト) 有効なアイテムプロトタイプ 1 - 無効なアイテムプロトタイプ 3 - サポートされていないアイテムプロトタイプ
status_codes	string	HTTP エージェントアイテムプロトタイプフィールド。コンマで区切られた必須の HTTP ステータスコードの範囲。コンマ区切りリストの一部としてユーザーマクロと LLD マクロもサポートします。 例: 200,200-{\$M},{\$M},200-400
templateid	string	(読取専用) 親テンプレートアイテムプロトタイプの ID
timeout	string	アイテムデータポーリングリクエストのタイムアウト。HTTP エージェントおよびスクリプトアイテムプロトタイプに使用されます。ユーザーマクロと LLD マクロをサポートします。 デフォルト: 3 秒 最大値: 60 秒
trapper_hosts	string	許可されたホスト。トラッパーアイテムプロトタイプまたは HTTP エージェントアイテムプロトタイプで使用されます。
trends	string	トレンドデータを保存する時間の単位。ユーザーマクロと LLD マクロも受け入れます。
units	string	デフォルト: 365d 値の単位

プロパティ	タイプ	説明
username	string	認証用のユーザー名。簡易チェック、SSH、Telnet、データベース監視、JMX、HTTP エージェントのアイテムプロトタイプで使用
uuid	string	SSH、Telnet のアイテムプロトタイプで必須 JMX で使用する場合、パスワードもユーザー名と一緒に指定するか、両方のプロパティを空白のままにする必要があります。 インポートされたアイテムプロトタイプを既存のアイテムプロトタイプにリンクするために使用されるユニバーサル意識別子。テンプレートのアイテムプロトタイプにのみ使用されます。指定されていない場合は自動生成されます。
valuemapid	string	更新操作の場合、このフィールドは 読取専用です。 関連付けられた値マップ ID
verify_host	integer	HTTP エージェントアイテムプロトタイプフィールド。URL のホスト名がホスト証明書の Common Name フィールドまたは Subject Alternate Name フィールドにあることを検証します。
verify_peer	integer	0 - (デフォルト) 検証しません。 1 - 検証します。 HTTP エージェントアイテムプロトタイプフィールド。検証はホスト証明書の認証です。
discover	integer	0 - (デフォルト) 検証しません。 1 - 検証します。 アイテムプロトタイプのディスカバリステータス。
		使用可能な値: 0 - (デフォルト) 新しいアイテムが発見されます。 1 - 新しいアイテムは発見されず、既存のアイテムは紛失としてマークされます。

Item prototype tag

item prototype tag オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tag	string	アイテムプロトタイプタグ名 (必須)
value	string	アイテムプロトタイプタグ値

アイテムプロトタイプの前処理

アイテムプロトタイプの前処理オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
type (必須)	integer	前処理オプションのタイプ。 使用可能な値: 1 - カスタム乗数 2 - 右トリム 3 - 左トリム 4 - トリム 5 - 正規表現マッチング 6 - ブール値から 10 進数へ 7 - 8 進数から 10 進数へ 8 - 16 進数から 10 進数へ 9 - 単純な変更 10 - 1 秒あたりの変更 11 - XML XPath 12 - JSONPath 13 - 範囲内 14 - 正規表現に一致 15 - 正規表現に不一致 16 - JSON のエラーチェック 17 - XML のエラーチェック; 18 - 正規表現を使用したエラーチェック 19 - 変更なしで破棄 20 - ハートビートで変更されていないものを破棄 21 - JavaScript 22 - Prometheus パターン 23 - Prometheus から JSON へ 24 - CSV から JSON へ 25 - 置換 26 - サポートされていないチェック 27 - XML から JSON へ
params (必須)	string	前処理オプションで使用される追加パラメーター。複数のパラメータは LF (\n) 文字で区切られます。
error_handler (必須)	integer	前処理ステップが失敗した場合に使用されるアクションタイプ 使用可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定されます。 1 - 値を破棄します。 2 - カスタム値を設定します。 3 - カスタムエラーメッセージを設定します。
error_handler_params (必須)	string	エラーハンドラーのパラメーター。error_handler とともに使用されます。 error_handler が 0 または 1 の場合、空にする必要があります。 error_handler が 2 の場合、空にすることができます。 error_handler が 3 の場合、空にすることはできません。

前処理の種類ごとに、次のパラメーターとエラーハンドラーがサポートされています。

前処理の種類	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートする
1	カスタム乗数	number ^{1, 6}			0, 1, 2, 3
2	右トリム	list of characters ²			
3	左トリム	list of characters ²			
4	トリム	list of characters ²			
5	正規表現マッチング	pattern ³	output ²		0, 1, 2, 3
6	ブール値から 10 進数へ				0, 1, 2, 3
7	8 進数から 10 進数へ				0, 1, 2, 3
8	16 進数から 10 進数へ				0, 1, 2, 3
9	単純な変更				0, 1, 2, 3
10	一秒あたりの変更				0, 1, 2, 3
11	XML XPath	path ⁴			0, 1, 2, 3
12	JSONPath	path ⁴			0, 1, 2, 3

前処理の種類	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートするバージョン
13	範囲内	min ^{1,6}	max ^{1,6}		0, 1, 2, 3
14	正規表現に一致	pattern ³			0, 1, 2, 3
15	正規表現に不一致	pattern ³			0, 1, 2, 3
16	JSON のエラーチェック	path ⁴			0, 1, 2, 3
17	XML のエラーチェック	path ⁴			0, 1, 2, 3
18	正規表現を使用したエラーチェック	pattern ³	output ²		0, 1, 2, 3
19	変更なしで破棄				
20	ハートビートで変更されていないものを破棄	seconds ^{5,6}			
21	JavaScript	script ²			
22	Prometheus パターン	pattern ^{6,7}	value, label, function	output ^{8,9}	0, 1, 2, 3
23	Prometheus から JSON へ	pattern ^{6,7}			0, 1, 2, 3
24	CSV から JSON へ	character ²	character ²	0,1	0, 1, 2, 3
25	置換	search string ²	replacement ²		
26	サポートされていないチェック				1, 2, 3
27	XML から JSON へ				0, 1, 2, 3

¹ 整数または浮動小数点数

² 文字列

³ 正規表現

⁴ JSONPath または XML XPath

⁵ 正の整数 (タイムサフィックスをサポート, 例: 30s, 1m, 2h, 1d)

⁶ ユーザーマクロ LLD マクロ

⁷ 構文に従う Prometheus パターン: <metric name>{<label name>="<label value>", ...} == <value> 各 Prometheus パターンコンポーネント (メトリック、ラベル名、ラベル値、およびメトリック値) は、ユーザーマクロか LLD マクロにすることができます。

⁸ Prometheus は次の構文で出力します: <label name> (ユーザーマクロまたは LLD マクロを使用できます) label が 2 番目のパラメーターとして選択されている場合。

⁹ 集計関数の 1 つ: sum,min,max,avg,count (function が 2 番目のパラメーターとして選択されている場合)

itemprototype.create

説明

object itemprototype.create(object/array itemPrototypes)

このメソッドを使用すると、新しいアイテムのプロトタイプを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するアイテムプロトタイプ。

このメソッドは標準アイテムプロトタイププロパティに加えて、次のパラメーターを受け入れます

パラメータ	タイプ	説明
ruleid	string	アイテムが属する LLD ルールの ID (必須)
preprocessing	array	アイテムプロトタイプ事前処理 オプション
tags	array	アイテムプロトタイプタグ。

戻り値

(object) itemids プロパティの下で作成されたアイテムプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムプロトタイプの順序と一致します。

例

アイテムプロトタイプの作成

アイテムのプロトタイプを作成して、検出されたファイルシステムのディスクの空き容量を監視します。検出されたアイテムは、数値の Zabbix エージェントアイテムである必要があり、30 秒ごとに更新されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.create",
  "params": {
    "name": "Free disk space on {#FSNAME}",
    "key_": "vfs.fs.size[{#FSNAME},free]",
    "hostid": "10197",
    "ruleid": "27665",
    "type": 0,
    "value_type": 3,
    "interfaceid": "112",
    "delay": "30s"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27666"
    ]
  },
  "id": 1
}
```

事前処理でアイテムプロトタイプを作成する

2 番目のステップとして、1 秒あたりの変化とカスタム倍率を使用してアイテムを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.create",
  "params": {
    "name": "Incoming network traffic on {#IFNAME}",
    "key_": "net.if.in[{#IFNAME}]",
    "hostid": "10001",
    "ruleid": "27665",
    "type": 0,
    "value_type": 3,
    "delay": "60s",
    "units": "bps",
    "interfaceid": "1155",
    "preprocessing": [
      {
        "type": 10,
        "params": "",
        "error_handler": 0,
        "error_handler_params": ""
      },
      {
        "type": 1,
        "params": "8",
        "error_handler": 2,
        "error_handler_params": "10"
      }
    ]
  }
}
```

```
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44211"
    ]
  },
  "id": 1
}
```

依存アイテムのプロトタイプ作成

ID 44211 のマスターアイテムプロトタイプの依存アイテムプロトタイプを作成します。同じホスト (テンプレート/ディスクバリエーション) への依存のみが許可されるため、マスターアイテムと依存アイテムは同じホスト ID とルール ID を持つ必要があります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.create",
  "params": {
    "hostid": "10001",
    "ruleid": "27665",
    "name": "Dependent test item prototype",
    "key_": "dependent.prototype",
    "type": 18,
    "master_itemid": "44211",
    "value_type": 3
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44212"
    ]
  },
  "id": 1
}
```

HTTP エージェントアイテムのプロトタイプを作成する

ユーザーマクロ、クエリフィールド、およびカスタムヘッダーを使用して URL でアイテムプロトタイプを作成します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.create",
  "params": {
    "type": "19",
    "hostid": "10254",
    "ruleid": "28256",
    "interfaceid": "2",
    "name": "api item prototype example",
    "key_": "api_http_item",
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

```

    "value_type": 3,
    "url": "{$URL_PROTOTYPE}",
    "query_fields": [
      {
        "min": "10"
      },
      {
        "max": "100"
      }
    ],
    "headers": {
      "X-Source": "api"
    },
    "delay": "35"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "28305"
    ]
  },
  "id": 1
}

```

スクリプトアイテムのプロトタイプを作成する

スクリプトアイテムのプロトタイプを使用して、単純なデータコレクションを作成します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "itemprototype.create",
  "params": {
    "name": "Script example",
    "key_": "custom.script.itemprototype",
    "hostid": "12345",
    "type": 21,
    "value_type": 4,
    "params": "var request = new CurlHttpRequest();\nreturn request.Post(\"https://postman-echo.com/post\");",
    "parameters": [
      {
        "name": "host",
        "value": "{HOST.CONN}"
      }
    ]
  },
  "timeout": "6s",
  "delay": "30s"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 2
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [

```

```
        "23865"  
    ]  
  },  
  "id": 3  
}
```

ソース

CItemPrototype::create() in ui/include/classes/api/services/CItemPrototype.php.

itemprototype.delete

説明

object itemprototype.delete(array itemPrototypeIds)

このメソッドを使用すると、アイテムのプロトタイプを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するアイテムプロトタイプの ID

戻り値

(object) prototypeids プロパティの下で削除されたアイテムプロトタイプの ID を含むオブジェクトを返します。

例

複数のアイテムプロトタイプ削除

2つのアイテムプロトタイプを削除します。マスターアイテムまたはアイテムプロトタイプが削除されると、依存アイテムプロトタイプは自動的に削除されます。

Request:

```
{  
  "jsonrpc": "2.0",  
  "method": "itemprototype.delete",  
  "params": [  
    "27352",  
    "27356"  
  ],  
  "auth": "3a57200802b24cda67c4e4010b50c065",  
  "id": 1  
}
```

Response:

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "prototypeids": [  
      "27352",  
      "27356"  
    ]  
  },  
  "id": 1  
}
```

ソース

CItemPrototype::delete() in ui/include/classes/api/services/CItemPrototype.php.

itemprototype.get

説明

integer/array itemprototype.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従ってアイテムのプロトタイプを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートしています

パラメータ	タイプ	説明
discoveryids	string/array	指定された LLD ルールに属するアイテムプロトタイプのみを返します
graphids	string/array	指定されたグラフプロトタイプで使用されているアイテムプロトタイプのみを返します
hostids	string/array	指定されたホストに属するアイテムプロトタイプのみを返します
inherited	boolean	true に設定すると、テンプレートから継承されたアイテムプロトタイプのみが返されます
itemids	string/array	指定された ID を持つアイテムプロトタイプのみを返します
monitored	boolean	true に設定すると監視対象ホストに属する有効なアイテムプロトタイプのみが返されます
templated	boolean	true に設定するとテンプレートに属するアイテムプロトタイプのみが返されます
templateids	string/array	指定されたテンプレートに属するアイテムプロトタイプのみを返します
triggerids	string/array	指定されたトリガープロトタイプで使用されるアイテムプロトタイプのみを返します
selectDiscoveryRule	query	アイテムプロトタイプが属するローレベルディスカバリルールを含む <code>discoveryRule</code> プロパティを返します
selectGraphs	query	アイテムプロトタイプが使用されているグラフプロトタイプを含む <code>graph_prototype</code> プロパティを返します。
selectHosts	query	<code>count</code> をサポートします アイテムのプロトタイプが属するホストの配列を持つ <code>hosts</code> プロパティを返します
selectTags	query	<code>tags</code> プロパティでアイテムプロトタイプタグを返します
selectTriggers	query	アイテムプロトタイプが使用されているトリガープロトタイプを含む <code>triggers</code> プロパティを返します。
		<code>count</code> をサポートします。

パラメータ	タイプ	説明
selectPreprocessing	query	<p>アイテムの事前処理オプションを含む事前処理プロパティを返します。</p> <p>次のプロパティがあります: type - (string) 事前処理オプションタイプ: 1 - カスタム乗数 2 - 右トリム 3 - 左トリム 4 - トリム 5 - 正規表現一致 6 - ブール値から 10 進数へ 7 - 8 進数から 10 進数へ 8 - 16 進数から 10 進数へ 9 - 単純な変更 10 - 1 秒ごとの変更 11 - XML XPath 12 - JSONPath 13 - 範囲内 14 - 正規表現に一致 15 - 正規表現に一致しない 16 - JSON のエラーをチェック 17 - XML エラーをチェック 18 - 正規表現を使用してエラーをチェック 19 - 変更なしで破棄 20 - ハートビートで変更なしで破棄 21 - JavaScript 22 - Prometheus パターン 23 - Prometheus から JSON 24 - CSV から JSON 25 - 置換 26 - サポートされていない値のチェック 27 - XML から JSON</p> <p>params - (string) 追加パラメータ前処理オプションで使用されます。複数のパラメータは LF(\n) 文字で区切られます。 error_handler - (string) 前処理ステップが失敗した場合に使用されるアクションタイプ: 0 - エラーメッセージは Zabbix サーバーによって設定されます。 1 - 値を破棄します。 2 - カスタム値を設定します。 3 - カスタムエラーメッセージを設定します。</p> <p>error_handler_params - (string) エラーハンドラパラメータ</p>
selectValueMap filter	query object	<p>アイテムプロトタイプ値マップを含むvaluemapプロパティを返します 指定されたフィルタに正確に一致する結果のみを返します。</p> <p>配列を受け入れます。キーはプロパティ名で、値は単一の値または照合する値の配列です。
追加のフィルタをサポート:</p>
limitSelects	integer	<p>host - アイテムプロトタイプが属するホストの技術名 副選択によって返されるレコードの数を制限します</p> <p>次のサブセレクトに適用されます selectGraphs - 結果は name でソートされます selectTriggers - 結果は description でソートされます 指定されたプロパティで結果を並べ替えます。</p>
sortfield	string/array	
countOutput	boolean	<p>可能な値は次のとおりです: itemid,name,key_,delay,type,status すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。</p>
editable	boolean	
excludeSearch	boolean	

パラメータ	タイプ	説明
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

LLD ルールからアイテムプロトタイプを取得する

特定の LLD ルール ID のすべてのアイテムプロトタイプを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.get",
  "params": {
    "output": "extend",
    "discoveryids": "27426"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "23077",
      "type": "0",
      "snmp_oid": "",
      "hostid": "10079",
      "name": "Incoming network traffic on en0",
      "key_": "net.if.in[en0]",
      "delay": "1m",
      "history": "1w",
      "trends": "365d",
      "status": "0",
      "value_type": "3",
      "trapper_hosts": "",
      "units": "bps",
      "logtimefmt": "",
      "templateid": "0",
      "valuemapid": "0",
      "params": "",
      "ipmi_sensor": "",
      "authtype": "0",
      "username": "",
      "password": "",
      "publickey": "",
      "privatekey": "",
      "interfaceid": "0",
    }
  ]
}
```

```

    "description": "",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "0",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "discover": "0",
    "uuid": "",
    "parameters": []
  },
  {
    "itemid": "10010",
    "type": "0",
    "snmp_oid": "",
    "hostid": "10001",
    "name": "Processor load (1 min average per core)",
    "key_": "system.cpu.load[percpu,avg1]",
    "delay": "1m",
    "history": "1w",
    "trends": "365d",
    "status": "0",
    "value_type": "0",
    "trapper_hosts": "",
    "units": "",
    "logtimefmt": "",
    "templateid": "0",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "interfaceid": "0",
    "description": "The processor load is calculated as system CPU load divided by number of CPU c",
    "evaltype": "0",
    "jmx_endpoint": "",
    "master_itemid": "0",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",

```

```

    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "lastclock": "0",
    "lastns": "0",
    "lastvalue": "0",
    "prevvalue": "0",
    "discover": "0",
    "uuid": "",
    "parameters": []
  }
],
  "id": 1
}

```

依存アイテムの検索

アイテム ID"25545" の依存アイテムを 1 つ検索します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "output": "extend",
    "filter": {
      "type": "18",
      "master_itemid": "25545"
    },
    "limit": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "25547",
      "type": "18",
      "snmp_oid": "",
      "hostid": "10116",
      "name": "Seconds",
      "key_": "apache.status.uptime.seconds",
      "delay": "0",
      "history": "90d",
      "trends": "365d",
      "status": "0",
      "value_type": "3",
      "trapper_hosts": "",
      "units": "",
      "logtimefmt": "",
      "templateid": "0",

```

```

    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "interfaceid": "0",
    "description": "",
    "evaltype": "0",
    "master_itemid": "25545",
    "jmx_endpoint": "",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "output_format": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "discover": "0",
    "uuid": "",
    "parameters": []
  }
],
  "id": 1
}

```

HTTP エージェントアイテムのプロトタイプを見つける

特定のホスト ID のリクエストメソッド HEAD を持つ HTTP エージェントアイテムプロトタイプを検索します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "itemprototype.get",
  "params": {
    "hostids": "10254",
    "filter": {
      "type": "19",
      "request_method": "3"
    }
  },
  "id": 17,
  "auth": "d678e0b85688ce578ff061bd29a20d3b"
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {

```

```

        "itemid": "28257",
        "type": "19",
        "snmp_oid": "",
        "hostid": "10254",
        "name": "discovered",
        "key_": "item[#{INAME}]",
        "delay": "#{IUPDATE}",
        "history": "90d",
        "trends": "30d",
        "status": "0",
        "value_type": "3",
        "trapper_hosts": "",
        "units": "",
        "logtimefmt": "",
        "templateid": "28255",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "interfaceid": "2",
        "description": "",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "3s",
        "url": "#{IURL}",
        "query_fields": [],
        "posts": "",
        "status_codes": "",
        "follow_redirects": "0",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "3",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "discover": "0",
        "uuid": "",
        "parameters": []
    }
],
    "id": 17
}

```

参照

- [Host](#)
- [Graph prototype](#)
- [Trigger prototype](#)

ソース

CItemPrototype::get() in ui/include/classes/api/services/CItemPrototype.php.

itemprototype.update

説明

object itemprototype.update(object/array itemPrototypes)

このメソッドを使用すると、既存のアイテムプロトタイプを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するアイテムプロトタイプのプロパティ

アイテムプロトタイプごとに itemid プロパティを定義する必要があります。他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準アイテムプロトタイププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
preprocessing	array	現在の事前処理オプションを置き換えるアイテムプロトタイプの事前処理オプション
tags	array	アイテムプロトタイプタグ

戻り値

(object) itemids プロパティの下で更新されたアイテムプロトタイプの ID を含むオブジェクトを返します。

例

アイテムプロトタイプのインターフェイス変更

検出されたアイテムが使用するホストインターフェイスを変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.update",
  "params": {
    "itemid": "27428",
    "interfaceid": "132"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27428"
    ]
  },
  "id": 1
}
```

依存アイテムプロトタイプを更新

依存アイテムプロトタイプを新しいマスターアイテムプロトタイプ ID で更新します。同じホスト (テンプレート/ディスカバリルール) への依存関係のみが許可されるため、マスターアイテムと依存アイテムは同じホスト ID とルール ID を持つ必要があります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.update",
  "params": {
    "master_itemid": "25570",
    "itemid": "189030"
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "189030"
    ]
  },
  "id": 1
}
```

HTTP エージェントアイテムプロトタイプを更新

クエリフィールドを変更し、すべてのカスタムヘッダーを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.update",
  "params": {
    "itemid": "28305",
    "query_fields": [
      {
        "random": "qwertyuiopasdfghjklzxcvbnm"
      }
    ],
    "headers": []
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "28305"
    ]
  },
  "id": 1
}
```

アイテムの事前処理オプションの更新

アイテムの事前処理ルール “乗数” でアイテムのプロトタイプを更新します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "itemprototype.update",
  "params": {
    "itemid": "44211",

```

```

    "preprocessing": [
      {
        "type": 1,
        "params": "4",
        "error_handler": 2,
        "error_handler_params": "5"
      }
    ]
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44211"
    ]
  },
  "id": 1
}

```

スクリプトアイテムプロトタイプを更新

スクリプトアイテムのプロトタイプを別のスクリプトで更新し、以前のスクリプトで使用された不要なパラメーターを削除します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "itemprototype.update",
  "params": {
    "itemid": "23865",
    "parameters": [],
    "script": "Zabbix.Log(3, 'Log test');\nreturn 1;"
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23865"
    ]
  },
  "id": 1
}

```

ソース

CItemPrototype::update() in ui/include/classes/api/services/CItemPrototype.php.

LLD rule

このクラスはローレベルディスカバリルールで機能するように設計されています。

オブジェクトリファレンス:

- [LLD rule](#)

利用可能なメソッド:

- `discoveryrule.copy` - ローレベルディスカバリルールのコピー
- `discoveryrule.create` - 新しいローレベルディスカバリルールの作成
- `discoveryrule.delete` - ローレベルディスカバリルールの削除
- `discoveryrule.get` - ローレベルディスカバリルールの取得
- `discoveryrule.update` - ローレベルディスカバリルールの更新

> LLD rule object

次のオブジェクトは、`discoveryruleAPI` に直接関連しています。

LLD rule

LLD ルールオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
<code>itemid</code>	string	(読取専用) LLD ルールの ID
<code>delay</code> (必須)	string	LLD ルールの更新間隔。接尾辞があり、フレキシブルインターバルかスケジューリング間隔のいずれかで構成される 1 つ以上のカスタム間隔がある場合とない場合の秒または時間単位をシリアル化された文字列として受け入れます。ユーザーマクロも受け入れます。フレキシブルインターバルはスラッシュで区切られた 2 つのマクロとして記述できます。間隔はセミコロンで区切られます。
<code>hostid</code> (必須)	string	Zabbix トラッパー、依存アイテム、および <code>mqtt.get</code> キーを使用する Zabbix エージェント (アクティブ) の場合は必須ではありません LLD ルールが属するホストの ID
<code>interfaceid</code> (必須)	string	LLD ルールのホストインターフェース ID。ホスト LLD ルールにのみ使用されます。
<code>key_</code> (必須)	string	Zabbix エージェント (アクティブ)、Zabbix 内部、Zabbix トラッパー、依存、データベースモニター、およびスクリプト LLD ルールには不要です。HTTP エージェント LLD ルールの場合はオプションになります。 LLD ルールキー
<code>name</code> (必須)	string	LLD ルール名
<code>type</code> (必須)	integer	LLD ルールのタイプ。 利用可能な値: 0 - Zabbix エージェント 2 - Zabbix トラッパー 3 - シンプルチェック 5 - Zabbix インターナル 7 - Zabbix エージェント (アクティブ) 10 - 外部チェック 11 - データベースモニター 12 - IPMI エージェント 13 - SSH エージェント 14 - TELNET エージェント 16 - JMX エージェント 18 - 依存アイテム 19 - HTTP エージェント 20 - SNMP エージェント 21 - スクリプト
<code>url</code> (必須)	string	HTTP エージェント LLD ルールに必要な URL 文字列。ユーザーマクロ <code>{HOST.IP}</code> 、 <code>{HOST.CONN}</code> 、 <code>{HOST.DNS}</code> 、 <code>{HOST.HOST}</code> 、 <code>{HOST.NAME}</code> 、 <code>{ITEM.ID}</code> 、 <code>{ITEM.KEY}</code> をサポートします。

プロパティ	タイプ	説明
allow_traps	integer	HTTP エージェント LLD ルールフィールド。トラップアイテムタイプと同様に値の入力を許可します。 0 - (デフォルト) 受信データの受け入れを許可しません。 1 - 受信データの受け入れを許可します。
authtype	integer	SSH エージェントまたは HTTP エージェントの LLD ルールで使用されません。 SSH エージェント認証方法の可能な値: 0 - (デフォルト) パスワード 1 - 公開鍵 HTTP エージェント認証方式の可能な値: 0 - (デフォルト) なし 1 - basic 2 - NTLM
error	string	(読取専用)LLD ルールの更新に問題がある場合のエラーテキスト
follow_redirects	integer	HTTP エージェント LLD ルールフィールド。データのプール中に応答リダイレクトを追跡します。 0 - リダイレクトを追跡しません 1 - (デフォルト) リダイレクトを追跡します。
headers	object	HTTP エージェント LLD ルールフィールド。HTTP(S) リクエストヘッダーを持つオブジェクト。ヘッダー名はキーとして使用され、ヘッダー値は値として使用されます。 例: { "User-Agent": "Zabbix" }
http_proxy	string	HTTP エージェント LLD ルールフィールド。HTTP(s) プロキシ接続文字列。
ipmi_sensor	string	IPMI センサー。IPMILLD ルールでのみ使用されます。
jmx_endpoint	string	JMX エージェントのカスタム接続文字列 デフォルト値: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi
lifetime	string	検出されなくなったアイテムが削除されるまでの期間。秒、接尾辞付きの時間単位、およびユーザーマクロを受け入れます。 デフォルト:30d
master_itemid	integer	マスターアイテム ID 最大 3 つの従属アイテムの再帰と 999 の従属アイテム最大数が許可されます。 ディスカバリルールを別のディスカバリルールのマスターアイテムにすることはできません。
output_format	integer	従属アイテムに必須です。 HTTP エージェント LLD ルールフィールド。応答を JSON に変換する必要があります。 0 - (デフォルト) そのまま保存します。 1 - JSON に変換します。
params	string	LLD ルールのタイプに応じた追加パラメーター: - SSH および TelnetLLD ルール用に実行されたスクリプト - データベースモニター LLD ルール用の SQL クエリ - 計算された LLD ルール用の式
parameters	array	スクリプトタイプ LLD ルールの追加パラメーター。'name' および'value' プロパティを持つオブジェクトの配列。name は一意である必要があります。
password	string	認証用のパスワード。シンプルチェック、SSH、Telnet、データベースモニター、JMX および HTTP エージェント LLD ルールで使用されます。

プロパティ	タイプ	説明
post_type	integer	HTTP エージェント LLD ルールフィールド。posts プロパティに保存されている投稿データ本文のタイプ。 0 - (デフォルト) そのままのデータ 2 - JSON データ 3 - XML データ
posts	string	HTTP エージェント LLD ルールフィールド。HTTP(s) リクエストの本文データ。post_type とともに使用されます。
privatekey	string	秘密鍵ファイルの名前
publickey	string	公開鍵ファイルの名前。
query_fields	array	HTTP エージェント LLD ルールフィールド。クエリパラメータ。'key' : 'value' のペアを持つオブジェクトの配列。value は空の文字列にすることができます。
request_method	integer	HTTP エージェント LLD ルールフィールド。リクエストメソッドのタイプ。 0 - (デフォルト) GET 1 - POST 2 - PUT 3 - HEAD
retrieve_mode	integer	HTTP エージェント LLD ルールフィールド。応答のどの部分を保存するか。 0 - (デフォルト) 本文 1 - ヘッダー 2 - 本文とヘッダーの両方が保存されます request_method HEAD の場合、許可される値は 1 つだけです。
snmp_oid	string	SNMP OID.
ssl_cert_file	string	HTTP エージェント LLD ルールフィールド。SSL 公開鍵ファイルパス
ssl_key_file	string	HTTP エージェント LLD ルールフィールド。SSL 秘密鍵ファイルパス
ssl_key_password	string	HTTP エージェント LLD ルールフィールド。SSL キーファイルのパスワード。
state	integer	(読取専用)LLD ルールの状態 利用可能な値: 0 - (デフォルト) 通常 1 - サポートされていません
status	integer	LLD ルールのステータス 可能な値: 0 - (デフォルト) 有効な LLD ルール 1 - 無効な LLD ルール
status_codes	string	HTTP エージェント LLD ルールフィールド。必要な HTTP ステータスコードの範囲をコンマで区切ります。カンマ区切りリストの一部としてユーザーマクロもサポートします。 例:200,200-{\$M},{M},200-400
templateid	string	(読取専用) 親テンプレート LLD ルールの ID
timeout	string	アイテムデータのポーリング要求のタイムアウト。HTTP エージェントおよびスクリプト LLD ルールに使用されます。ユーザーマクロをサポートします。 デフォルト:3s 最大値:60s
trapper_hosts	string	許可されたホスト。トラッパー LLD ルールまたは HTTP エージェント LLD ルールによって使用されます。
username	string	認証用のユーザー名。シンプルチェック、SSH、Telnet、データベースモニター、JMX および HTTP エージェント LLD ルールで使用されます。 SSH および TelnetLLD ルールで必要です。

プロパティ	タイプ	説明
uuid	string	インポートされた LLD ルールを既存のルールにリンクするために使用されるユニバーサル意識別子。テンプレートの LLD ルールにのみ使用されます。指定されていない場合は自動生成されます。
verify_host	integer	更新操作の場合、このフィールドは読取専用です。 HTTP エージェント LLD ルールフィールド。URL の検証ホスト名は、ホスト証明書の共通名フィールドまたはサブジェクト代替名フィールドにあります。
verify_peer	integer	HTTP エージェント LLD ルールフィールド。検証は本物のホスト証明書です。
		0 - (デフォルト) 検証しない 1 - 検証する

LLD ルールフィルター

LLD ルールフィルターオブジェクトは、検出されたオブジェクトをフィルター処理するために使用できる一連の条件を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
conditions (必須)	array	結果のフィルタリングに使用する一連のフィルター条件
evaltype (必須)	integer	フィルタ条件の評価方法
		可能な値: 0 - and/or 1 - and 2 - or 3 - カスタム式
eval_formula	string	(読取専用) フィルタ条件の評価に使用される生成された式。式には <code>formulaid</code> によって特定のフィルター条件を参照する ID が含まれています。eval_formula の値は、カスタム式を持つフィルターの formula の値と同じです。
formula	string	カスタム式を使用してフィルターの条件を評価するために使用されるユーザー定義の式。式には、その <code>formulaid</code> によって特定のフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、フィルター条件で定義された ID と完全に一致する必要があります。条件を未使用のままにしたり省略したりすることはできません。
		カスタム式フィルターで使用します。

LLD ルールフィルター条件

LLD ルールフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
macro (必須)	string	チェックを実行する LLD マクロ
value (必須)	string	比較する値
formulaid	string	カスタム式から条件を参照するために使用される任意の ID。大文字のみを含めることができます。ID は、フィルター条件を変更するときにユーザーが定義する必要がありますが、後で要求すると新たに生成されます。

プロパティ	タイプ	説明
operator	integer	条件演算子。 可能な値: 8 - (デフォルト) 正規表現に一致 9 - 正規表現に不一致 12 - 存在する 13 - 存在しない

Note:

さまざまなタイプの式でフィルターを使用する方法をよりよく理解するには `discoveryrule.get` と `discoveryrule.create` メソッドページの例を参照してください。

LLD マクロパス

LLD マクロパスには、次のプロパティがあります。

プロパティ	タイプ	説明
lld_macro (必須)	string	LLD マクロ
path (必須)	string	対応するマクロに割り当てられる値のセレクター

LLD ルールの事前処理

LLD ルールの事前処理オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
type (必須)	integer	事前処理オプションタイプ 利用可能な値: 5 - 正規表現の一致 11 - XML XPath 12 - JSONPath 15 - 正規表現の不一致 16 - JSON のエラーチェック 17 - XML のエラーチェック 20 - ハートビートで変更せずに破棄 23 - プロメテウスから JSON 24 - CSV から JSON 25 - 置換 27 - XML から JSON
params (必須)	string	事前処理オプションで使用される追加のパラメーター。複数のパラメーターは LF (\n) 文字で区切られます。
error_handler (必須)	integer	事前処理ステップが失敗した場合に使用されるアクションタイプ。 利用可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定されます 1 - 値を破棄 2 - カスタム値を設定 3 - カスタムエラーメッセージを設定 エラーハンドラパラメーター。 <code>error_handler</code> とともに使用されます。
error_handler_params (必須)	string	<code>error_handler</code> が 0 または 1 の場合は空白にする必要があります。 <code>error_handler</code> が 2 の場合は空白にすることもできます。 <code>error_handler</code> が 3 の場合は必須です。

次のパラメータとエラーハンドラは、各事前処理タイプでサポートされています。

事前処理タイプ	名前	パラメーター 1	パラメーター 2	パラメーター 3	サポートされているエラーハンドラ
5	正規表現	pattern ¹	output ²		0, 1, 2, 3
11	XML XPath	path ³			0, 1, 2, 3
12	JSON Path	path ³			0, 1, 2, 3
15	正規表現と不一致	pattern ¹			0, 1, 2, 3
16	JSON エラーチェック	path ³			0, 1, 2, 3
17	XML エラーチェック	path ³			0, 1, 2, 3
20	ハートビートで変更せずに破棄	seconds ^{4, 5, 6}			
23	プロメテウスから JSON へ	pattern ^{5, 7}			0, 1, 2, 3
24	CSV から JSON へ	character ²	character ²	0,1	0, 1, 2, 3
25	置換	search string ²	replacement ²		
27	XML から JSON へ				0, 1, 2, 3

¹ 正規表現

² string

³ JSONPath または XML XPath

⁴ 正の整数 (時間接尾辞をサポート例: 30s, 1m, 2h, 1d)

⁵ user macro

⁶ LLD macro

⁷ 構文に従ったプロメテウスの出力:<metric name>{<label name>=<label value>", ...} == <value>. Each Prometheus pattern component (metric, label name, label value and metric value) can be user macro.

⁸ 構文に従ったプロメテウスの出力: <label name>.

LLD ルールのオーバーライド

LLD ルールオーバーライドオブジェクトは、さまざまなプロトタイプオブジェクトのプロパティをオーバーライドするために使用される一連のルール (フィルター、条件、および操作) を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
name (必須)	string	一意なオーバーライド名
step (必須)	integer	オーバーライドの一意なオーダーナンバー
stop	integer	一致する場合は次のオーバーライドの処理を停止します。 可能な値: 0 - (デフォルト) オーバーライドの処理を停止しません 1 - フィルターが一致する場合はオーバーライドの処理を停止します
filter	object	オーバーライドフィルター
operations	array	オーバーライドの操作

LLD ルールオーバーライドフィルター

LLD ルールオーバーライドフィルターオブジェクトは、検出されたオブジェクトと一致する場合にオーバーライドが適用される一連の条件を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
evaltype (必須)	integer	フィルタ条件の評価方法を上書きします。 利用可能な値: 0 - and/or 1 - and 2 - or 3 - カスタム式
conditions (必須)	array	検出されたオブジェクトの照合に使用するオーバーライドフィルター条件のセット。
eval_formula	string	(読取専用) オーバーライドフィルター条件の評価に使用される生成された式。式には、その formulaid によって特定のオーバーライドフィルター条件を参照する ID が含まれています。eval_formula の値は、カスタム式を持つフィルターの formula の値と同じです。

プロパティ	タイプ	説明
formula	string	カスタム式を使用してオーバーライドフィルターの条件を評価するために使用されるユーザー定義式。式には、その formulaid によって特定のオーバーライドフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、オーバーライドフィルター条件で定義された ID と完全に一致する必要があります。条件を未使用のままにしたり省略したりすることはできません。 カスタム式オーバーライドフィルターに必要です。

LLD ルールオーバーライドフィルター条件

LLD ルールオーバーライドフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
macro (必須)	string	チェックを実行する LLD マクロ
value (必須)	string	比較する値
formulaid	string	カスタム式から条件を参照するために使用される任意の一意の ID。大文字のみを含めることができます。ID は、フィルター条件を変更するときにユーザーが定義する必要がありますが、後で要求すると新たに生成されます。
operator	integer	条件演算子 可能な値: 8 - (デフォルト) 正規表現に一致 9 - 正規表現と不一致 12 - 存在する 13 - 存在しない

LLD ルールオーバーライド操作

LLD ルールのオーバーライド操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。次のプロパティがあります。

プロパティ	タイプ	説明
operationobject (必須)	integer	アクションを実行するために検出されたオブジェクトのタイプ。 可能な値: 0 - アイテムのプロトタイプ 1 - トリガーのプロトタイプ 2 - グラフのプロトタイプ 3 - ホストのプロトタイプ
operator	integer	条件演算子をオーバーライドします。 可能な値: 0 - (デフォルト) 等しい 1 - 等しくない 2 - 含む 3 - 含まない 8 - 一致する 9 - 一致しない
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフ、またはホストのプロトタイプ名に一致するパターン。
opstatus	object	アイテム、トリガー、およびホストのプロトタイプオブジェクトの操作ステータスオブジェクトをオーバーライドします。
opdiscover	object	オーバーライド操作の検出ステータスオブジェクト (すべてのオブジェクトタイプ)

プロパティ	タイプ	説明
opperiod	object	アイテムプロトタイプオブジェクトの操作期間（更新間隔）オブジェクトを上書きします。
ophistory	object	アイテムプロトタイプオブジェクトの操作履歴オブジェクトを上書きします。
optrends	object	アイテムプロトタイプオブジェクトの操作トレンドオブジェクトをオーバーライドします。
opseverity	object	トリガープロトタイプオブジェクトの操作深刻度オブジェクトをオーバーライドします。
optag	array	トリガーおよびホストのプロトタイプオブジェクトの操作タグオブジェクトをオーバーライドします。
optemplate	array	ホストプロトタイプオブジェクトの操作テンプレートオブジェクトをオーバーライドします。
opinVENTORY	object	ホストプロトタイプオブジェクトの操作インベントリオブジェクトを上書きします。

LLD ルールオーバーライド操作ステータス

LLD ルールは、検出されたオブジェクトに設定されている操作ステータスを上書きします。次のプロパティがあります。

プロパティ	タイプ	説明
status (必須)	integer	選択したオブジェクトのステータスを上書きします。 利用可能な値: 0 - 作成を有効に 1 - 作成を無効に

LLD ルールオーバーライド操作ディスカバリ

LLD ルールは、検出されたオブジェクトに設定されている操作のディスカバリステータスを上書きします。次のプロパティがあります。

プロパティ	タイプ	説明
discover (必須)	integer	選択したオブジェクトのディスカバリステータスを上書きします。 利用可能な値: 0 - オブジェクトの検出を続行します 1 - 新しいオブジェクトは検出されず、既存のオブジェクトは失われたものとしてマークされます。

LLD ルールオーバーライド動作期間

LLD ルールオーバーライド操作期間は、検出されたアイテムに設定される更新間隔値（カスタム間隔をサポート）です。次のプロパティがあります。

プロパティ	タイプ	説明
delay (必須)	string	アイテムのプロトタイプの更新間隔を上書きします。接尾辞（30s,1m,2h,1d）が付いた秒または時間単位およびフレキシブルなスケジューリング間隔とユーザーマクロまたは LLD マクロを受け入れます。複数の間隔はセミコロンで区切られます。

LLD ルール操作履歴上書き

LLD ルールは、検出されたアイテムに設定されている操作履歴値を上書きします。次のプロパティがあります。

プロパティ	タイプ	説明
history (必須)	string	履歴データを保存する期間の時間単位であるアイテムプロトタイプの履歴を上書きします。ユーザーマクロと LLD マクロも受け入れます。

LLD ルールによるオペレーショントレンドの上書き

LLD ルールは、検出されたアイテムに設定されているオペレーショントレンド値をオーバーライドします。次のプロパティがあります。

プロパティ	タイプ	説明
trends (必須)	string	トレンドデータを保存する期間の時間単位であるアイテムプロトタイプのトレンドをオーバーライドします。ユーザーマクロと LLD マクロも受け入れます。

LLD ルールはオペレーションの深刻度をオーバーライドします

LLD ルールは、検出されたトリガーに設定されているオペレーションの深刻度の値をオーバーライドします。次のプロパティがあります。

プロパティ	タイプ	説明
severity (必須)	integer	トリガープロトタイプの重大度を上書きします。 可能な値は次のとおりです:0 - (デフォルト) 未分類 1 - 情報 2 - 警告 3 - 軽度の障害 4 - 重度の障害 5 - 致命的な障害

LLD ルールオーバーライドオペレーションタグ

LLD ルールオーバーライド操作タグオブジェクトには、検出されたオブジェクトに設定されているタグ名と値が含まれています。次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	新しいタグ名
value	string	新しいタグ値

LLD ルールオーバーライドオペレーションテンプレート

LLD ルールは、検出されたホストにリンクされている操作テンプレートオブジェクトをオーバーライドします。次のプロパティがあります。

プロパティ	タイプ	説明
templateid (必須)	string	ホストプロトタイプのリンクされたテンプレートのテンプレートをオーバーライドします。

LLD ルールによるオペレーションインベントリ上書き

LLD ルールは検出されたホストに設定されている操作インベントリモード値を上書きします。次のプロパティがあります。

プロパティ	タイプ	説明
inventory_mode (必須)	integer	ホストプロトタイプインベントリモードをオーバーライドします。 利用可能な値: -1 - 無効 0 - (デフォルト) 手動 1 - 自動

discoveryrule.copy

説明

object discoveryrule.copy(object parameters)

このメソッドで、すべてのプロトタイプを含む LLD ルールを指定されたホストにコピーできます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) コピーする LLD ルールとターゲットホストを定義するパラメーター。

パラメータ	タイプ	説明
discoveryids	array	コピーする LLD ルールの ID
hostids	array	LLD ルールをコピーするホストの ID

戻り値

(boolean) コピーが成功した場合は true を返します。

例

LLD ルールを複数のホストにコピーします

LLD ルールを 2 つのホストにコピー

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.copy",
  "params": {
    "discoveryids": [
      "27426"
    ],
    "hostids": [
      "10196",
      "10197"
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 1
}
```

ソース

CDiscoveryrule::copy() in ui/include/classes/api/services/CDiscoveryRule.php.

discoveryrule.create

説明

object discoveryrule.create(object/array lldRules)

このメソッドでは、新しい LLD ルールを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成する LLD ルール

このメソッドは標準の LLD ルールプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
filter	object	LLD ルールの LLD ルールフィルターオブジェクト
preprocessing	array	LLD ルール事前処理オプション
lld_macro_paths	array	LLD ルール lld_macro_path オプション
overrides	array	LLD ルールオーバーライドオプション

戻り値

(object) itemids プロパティの下で作成された LLD ルールの ID を含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールの順序と一致します。

例

LLD ルールの作成

Zabbix エージェント LLD ルールを作成して、マウントされたファイルシステムを検出します。発見されたアイテムは 30 秒ごとに更新されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Mounted filesystem discovery",
    "key_": "vfs.fs.discovery",
    "hostid": "10197",
    "type": 0,
    "interfaceid": "112",
    "delay": "30s"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}
```

フィルタを使用

結果をフィルタリングするための一連の条件を使用して LLD ルールを作成します。条件は、論理"and" 演算子を使用してグループ化されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
```

```

"params": {
  "name": "Filtered LLD rule",
  "key_": "lld",
  "hostid": "10116",
  "type": 0,
  "interfaceid": "13",
  "delay": "30s",
  "filter": {
    "evaltype": 1,
    "conditions": [
      {
        "macro": "#{MACRO1}",
        "value": "@regex1"
      },
      {
        "macro": "#{MACRO2}",
        "value": "@regex2",
        "operator": "9"
      },
      {
        "macro": "#{MACRO3}",
        "value": "",
        "operator": "12"
      },
      {
        "macro": "#{MACRO4}",
        "value": "",
        "operator": "13"
      }
    ]
  }
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}

```

マクロパスを使用した LLD ルールの作成

Request:

```

{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "LLD rule with LLD macro paths",
    "key_": "lld",
    "hostid": "10116",
    "type": 0,
    "interfaceid": "13",
    "delay": "30s",
    "lld_macro_paths": [
      {
        "lld_macro": "#{MACRO1}",

```

```

        "path": "$.path.1"
      },
      {
        "lld_macro": "#{MACRO2}",
        "path": "$.path.2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}

```

カスタム式フィルターの使用

カスタム式を使用して条件を評価するフィルターで LLD ルールを作成します。LLD ルールは "#{MACRO1}" マクロ値が正規表現 "regex1" と "regex2" の両方に一致し、 "#{MACRO2}" の値が "regex3" または "regex4" のいずれかに一致するオブジェクトのみを検出する必要があります。式 ID "A","B","C","D" は任意に選択されています。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Filtered LLD rule",
    "key_": "lld",
    "hostid": "10116",
    "type": 0,
    "interfaceid": "13",
    "delay": "30s",
    "filter": {
      "evaltype": 3,
      "formula": "(A and B) and (C or D)",
      "conditions": [
        {
          "macro": "#{MACRO1}",
          "value": "@regex1",
          "formulaid": "A"
        },
        {
          "macro": "#{MACRO1}",
          "value": "@regex2",
          "formulaid": "B"
        },
        {
          "macro": "#{MACRO2}",
          "value": "@regex3",
          "formulaid": "C"
        },
        {
          "macro": "#{MACRO2}",
          "value": "@regex4",
          "formulaid": "D"
        }
      ]
    }
  }
}

```

```

    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "27665"
    ]
  },
  "id": 1
}

```

カスタムクエリフィールドとヘッダーの使用

カスタムクエリフィールドとヘッダーを使用して LLD ルールを作成します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "hostid": "10257",
    "interfaceid": "5",
    "type": 19,
    "name": "API HTTP agent",
    "key_": "api_discovery_rule",
    "value_type": 3,
    "delay": "5s",
    "url": "http://127.0.0.1?discoverer.php",
    "query_fields": [
      {
        "mode": "json"
      },
      {
        "elements": "2"
      }
    ],
    "headers": {
      "X-Type": "api",
      "Authorization": "Bearer mF_A.B5f-2.1JcM"
    },
    "allow_traps": 1,
    "trapper_hosts": "127.0.0.1"
  },
  "auth": "d678e0b85688ce578ff061bd29a20d3b",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "28336"
    ]
  },
}

```

```
"id": 1
}
```

事前処理を使用した LLD ルールの作成

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Discovery rule with preprocessing",
    "key_": "lld.with.preprocessing",
    "hostid": "10001",
    "ruleid": "27665",
    "type": 0,
    "value_type": 3,
    "delay": "60s",
    "interfaceid": "1155",
    "preprocessing": [
      {
        "type": 20,
        "params": "20",
        "error_handler": 0,
        "error_handler_params": ""
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44211"
    ]
  },
  "id": 1
}
```

オーバーライドを使用した LLD ルールの作成

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Discover database host",
    "key_": "lld.with.overrides",
    "hostid": "10001",
    "type": 0,
    "value_type": 3,
    "delay": "60s",
    "interfaceid": "1155",
    "overrides": [
      {
        "name": "Discover MySQL host",
        "step": "1",
        "stop": "1",
        "filter": {
          "evaltype": "2",

```

```

        "conditions": [
            {
                "macro": "#{UNIT.NAME}",
                "operator": "8",
                "value": "~mysqld\\.service$"
            },
            {
                "macro": "#{UNIT.NAME}",
                "operator": "8",
                "value": "~mariadb\\.service$"
            }
        ]
    },
    "operations": [
        {
            "operationobject": "3",
            "operator": "2",
            "value": "Database host",
            "opstatus": {
                "status": "0"
            },
            "optemplate": [
                {
                    "templateid": "10170"
                }
            ],
            "optag": [
                {
                    "tag": "Database",
                    "value": "MySQL"
                }
            ]
        }
    ]
},
{
    "name": "Discover PostgreSQL host",
    "step": "2",
    "stop": "1",
    "filter": {
        "evaltype": "0",
        "conditions": [
            {
                "macro": "#{UNIT.NAME}",
                "operator": "8",
                "value": "~postgresql\\.service$"
            }
        ]
    },
    "operations": [
        {
            "operationobject": "3",
            "operator": "2",
            "value": "Database host",
            "opstatus": {
                "status": "0"
            },
            "optemplate": [
                {
                    "templateid": "10263"
                }
            ]
        }
    ],

```



```

        "optag": [
          {
            "tag": "Database",
            "value": "PostgreSQL"
          }
        ]
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "30980"
    ]
  },
  "id": 1
}

```

スクリプト LLD ルールを作成する

スクリプト LLD ルールを使用して単純なデータコレクションを作成します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "discoveryrule.create",
  "params": {
    "name": "Script example",
    "key_": "custom.script.lldrule",
    "hostid": "12345",
    "type": 21,
    "value_type": 4,
    "params": "var request = new CurlHttpRequest();\nreturn request.Post(\"https://postman-echo.com/post\");",
    "parameters": [
      {
        "name": "host",
        "value": "{HOST.CONN}"
      }
    ]
  },
  "timeout": "6s",
  "delay": "30s"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 2
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23865"
    ]
  },
  "id": 3
}

```

```
}
```

参照

- [LLD rule filter](#)
- [LLD macro paths](#)
- [LLD rule preprocessing](#)

ソース

CDiscoveryRule::create() in ui/include/classes/api/services/CDiscoveryRule.php.

discoveryrule.delete

説明

object discoveryrule.delete(array lldRuleIds)

このメソッドで LLD ルールを削除できます

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除する LLD ルールの ID

戻り値

(object) は ruleids プロパティの下にある削除された LLD ルールの ID を含むオブジェクトを返します。

例

複数の LLD ルール削除

2つの LLD ルールを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.delete",
  "params": [
    "27665",
    "27668"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "ruleids": [
      "27665",
      "27668"
    ]
  },
  "id": 1
}
```

ソース

CDiscoveryRule::delete() in ui/include/classes/api/services/CDiscoveryRule.php.

discoveryrule.get

説明

integer/array discoveryrule.get(object parameters)

このメソッドでは、指定されたパラメーターに従って LLD ルールを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
itemids	string/array	指定された ID を持つ LLD ルールのみを返します
groupids	string/array	指定されたグループのホストに属する LLD ルールのみを返します
hostids	string/array	指定されたホストに属する LLD ルールのみを返します
inherited	boolean	true に設定すると、テンプレートから継承された LLD ルールのみが返されます
interfaceids	string/array	指定されたホストインターフェイスを使用する LLD ルールのみを返します
monitored	boolean	true に設定すると、監視対象ホストに属する有効な LLD ルールのみが返されます
templated	boolean	true に設定すると、テンプレートに属する LLD ルールのみが返されます
templateids	string/array	指定されたテンプレートに属する LLD ルールのみを返します
selectFilter	query	LLD ルールで使用されるフィルターのデータを含む フィルター プロパティを返します。
selectGraphs	query	LLD ルールに属するグラフプロトタイプを含む グラフ プロパティを返します。
selectHostPrototypes	query	count をサポートします。 LLD ルールに属するホストプロトタイプを含む ホストプロトタイプ プロパティを返します。
selectHosts	query	count をサポートします。 LLD ルールが属するホストの配列を含む hosts プロパティを返します。
selectItems	query	LLD ルールに属するアイテムプロトタイプを含む items プロパティを返します。
selectTriggers	query	count をサポートします。 LLD ルールに属するトリガープロトタイプを含む triggers プロパティを返します。
selectLLDMacroPaths	query	count をサポートします。 lld_macro_paths プロパティを返し、LLD マクロのリストと、対応する各マクロに割り当てられた値へのパスを示します。

パラメータ	タイプ	説明
selectPreprocessing	query	<p>LLD ルールの事前処理オプションを使用して <code>preprocessing</code> プロパティを返します。</p> <p>次のプロパティがあります。 <code>type-(string)</code> 事前処理オプションタイプ: 5 - 正規表現のマッチング 11 - XML XPath 12 - JSONPath 15 - 正規表現のアンマッチ 16 - JSON のエラーチェック 17 - XML のエラーチェック 20 - ハートビートを変更せずに破棄 23 - プロメテウスから JSON 24 - CSV から JSON 25 - 置換 27 - XML から JSON</p> <p><code>params-(string)</code> 事前処理オプションで 사용되는追加のパラメータ。複数のパラメータは LF(\n) 文字で区切られます。 <code>error_handler-(string)</code> 事前処理ステップが失敗した場合に使用されるアクションタイプ: 0 - エラーメッセージを Zabbix サーバーによって設定 1 - 値を破棄 2 - カスタム値を設定 3 - カスタムエラーメッセージを設定</p> <p><code>error_handler_params-(string)</code> エラーハンドラパラメータ</p>
selectOverrides	<code>lld_rule_overrides</code> プロパティを、プロトタイプオブジェクトで実行されるオーバーライドフィルター、条件、および操作のリストとともに返します	
filter	object	<p>指定されたフィルターに完全に一致する結果のみを返します。</p> <p>キーがプロパティ名であり、値が単一の値または一致する値の配列である配列を受け入れます。 <code>
</code> 追加のフィルターをサポートします: <code>host</code> - LLD ルールが属するホストの技術名 副選択によって返されるレコードの数を制限します。</p>
limitSelects	integer	<p>次の副選択に適用されます: <code>selectItems</code> <code>selectGraphs</code> <code>selectTriggers</code></p> <p>指定されたプロパティで結果を並べ替えます</p>
sortfield	string/array	<p>指定されたプロパティで結果を並べ替えます</p>
countOutput	boolean	<p>利用可能な値は <code>itemid,name,key_,delay,type,status</code> です すべての <code>get</code> メソッドに共通するこれらのパラメータについては フランス解説 ページで詳しく説明されています。</p>
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ホストからディスカバリルールの取得

特定ホスト ID の全ディスカバリルールを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.get",
  "params": {
    "output": "extend",
    "hostids": "10202"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "27425",
      "type": "0",
      "snmp_oid": "",
      "hostid": "10202",
      "name": "Network interface discovery",
      "key_": "net.if.discovery",
      "delay": "1h",
      "status": "0",
      "trapper_hosts": "",
      "templateid": "22444",
      "valuemapid": "0",
      "params": "",
      "ipmi_sensor": "",
      "authtype": "0",
      "username": "",
      "password": "",
      "publickey": "",
      "privatekey": "",
      "interfaceid": "119",
      "description": "Discovery of network interfaces as defined in global regular expression \"Netw",
      "lifetime": "30d",
      "jmx_endpoint": "",
      "master_itemid": "0",
      "timeout": "3s",
      "url": "",
      "query_fields": [],
      "posts": "",
      "status_codes": "200",
      "follow_redirects": "1",
      "post_type": "0",
      "http_proxy": "",
      "headers": [],
      "retrieve_mode": "0",
      "request_method": "0",
      "ssl_cert_file": "",

```

```

    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": []
  },
  {
    "itemid": "27426",
    "type": "0",
    "snmp_oid": "",
    "hostid": "10202",
    "name": "Mounted filesystem discovery",
    "key_": "vfs.fs.discovery",
    "delay": "1h",
    "status": "0",
    "trapper_hosts": "",
    "templateid": "22450",
    "valuemapid": "0",
    "params": "",
    "ipmi_sensor": "",
    "authtype": "0",
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "interfaceid": "119",
    "description": "Discovery of file systems of different types as defined in global regular expr
    "lifetime": "30d",
    "jmx_endpoint": "",
    "master_itemid": "0",
    "timeout": "3s",
    "url": "",
    "query_fields": [],
    "posts": "",
    "status_codes": "200",
    "follow_redirects": "1",
    "post_type": "0",
    "http_proxy": "",
    "headers": [],
    "retrieve_mode": "0",
    "request_method": "0",
    "ssl_cert_file": "",
    "ssl_key_file": "",
    "ssl_key_password": "",
    "verify_peer": "0",
    "verify_host": "0",
    "allow_traps": "0",
    "uuid": "",
    "state": "0",
    "error": "",
    "parameters": []
  }
],
  "id": 1
}

```

フィルタ条件の取得

LLD ルール"24681" の名前とそのフィルター条件を取得します。フィルタは"and" 評価タイプを使用するため、formula プロパテ

イは空であり、eval_formula が自動的に生成されます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.get",
  "params": {
    "output": ["name"],
    "selectFilter": "extend",
    "itemids": ["24681"]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "24681",
      "name": "Filtered LLD rule",
      "filter": {
        "evaltype": "1",
        "formula": "",
        "conditions": [
          {
            "macro": "#{MACRO1}",
            "value": "@regex1",
            "operator": "8",
            "formulaid": "A"
          },
          {
            "macro": "#{MACRO2}",
            "value": "@regex2",
            "operator": "9",
            "formulaid": "B"
          },
          {
            "macro": "#{MACRO3}",
            "value": "",
            "operator": "12",
            "formulaid": "C"
          },
          {
            "macro": "#{MACRO4}",
            "value": "",
            "operator": "13",
            "formulaid": "D"
          }
        ],
        "eval_formula": "A and B and C and D"
      }
    }
  ],
  "id": 1
}
```

URL で LLD ルールを取得する

ルール URL フィールド値によってホストの LLD ルールを取得します。LLD ルールに定義された URL 文字列の完全一致のみがサポートされます。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "discoveryrule.get",
  "params": {
    "hostids": "10257",
    "filter": {
      "type": 19,
      "url": "http://127.0.0.1/discoverer.php"
    }
  },
  "id": 39,
  "auth": "d678e0b85688ce578ff061bd29a20d3b"
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "28336",
      "type": "19",
      "snmp_oid": "",
      "hostid": "10257",
      "name": "API HTTP agent",
      "key_": "api_discovery_rule",
      "delay": "5s",
      "status": "0",
      "trapper_hosts": "",
      "templateid": "0",
      "valuemapid": "0",
      "params": "",
      "ipmi_sensor": "",
      "authtype": "0",
      "username": "",
      "password": "",
      "publickey": "",
      "privatekey": "",
      "interfaceid": "5",
      "description": "",
      "lifetime": "30d",
      "jmx_endpoint": "",
      "master_itemid": "0",
      "timeout": "3s",
      "url": "http://127.0.0.1/discoverer.php",
      "query_fields": [
        {
          "mode": "json"
        },
        {
          "elements": "2"
        }
      ],
      "posts": "",
      "status_codes": "200",
      "follow_redirects": "1",
      "post_type": "0",
      "http_proxy": "",
      "headers": {
        "X-Type": "api",
        "Authorization": "Bearer mF_A.B5f-2.1JcM"
      },
      "retrieve_mode": "0",

```



```

        "request_method": "1",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "state": "0",
        "error": "",
        "parameters": []
    }
],
    "id": 39
}

```

オーバーライドを使用して LLD ルールを取得する

さまざまなオーバーライド設定を持つ 1 つの LLD ルールを取得します

Request:

```

{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 39,
    "auth": "d678e0b85688ce578ff061bd29a20d3b"
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "Discover database host",
            "overrides": [
                {
                    "name": "Discover MySQL host",
                    "step": "1",
                    "stop": "1",
                    "filter": {
                        "evaltype": "2",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mysqld\\.service$",
                                "formulaid": "A"
                            },
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mariadb\\.service$",
                                "formulaid": "B"
                            }
                        ],
                        "eval_formula": "A or B"
                    }
                }
            ]
        }
    ]
}

```

```

    "operations": [
      {
        "operationobject": "3",
        "operator": "2",
        "value": "Database host",
        "opstatus": {
          "status": "0"
        },
        "optag": [
          {
            "tag": "Database",
            "value": "MySQL"
          }
        ],
        "optemplate": [
          {
            "templateid": "10170"
          }
        ]
      }
    ],
  },
  {
    "name": "Discover PostgreSQL host",
    "step": "2",
    "stop": "1",
    "filter": {
      "evaltype": "0",
      "formula": "",
      "conditions": [
        {
          "macro": "{#UNIT.NAME}",
          "operator": "8",
          "value": "^postgresql\\.service$",
          "formulaid": "A"
        }
      ],
      "eval_formula": "A"
    },
    "operations": [
      {
        "operationobject": "3",
        "operator": "2",
        "value": "Database host",
        "opstatus": {
          "status": "0"
        },
        "optag": [
          {
            "tag": "Database",
            "value": "PostgreSQL"
          }
        ],
        "optemplate": [
          {
            "templateid": "10263"
          }
        ]
      }
    ]
  }
]

```

```

    }
  ],
  "id": 39
}

```

参照

- [Graph prototype](#)
- [Host](#)
- [Item prototype](#)
- [LLD rule filter](#)
- [Trigger prototype](#)

ソース

CDiscoveryRule::get() in ui/include/classes/api/services/CDiscoveryRule.php.

discoveryrule.update

説明

object discoveryrule.update(object/array lldRules)

このメソッドで、既存のローレベルディスカバリルールを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新される LLD ルールのプロパティ。

itemid プロパティは LLD ルールごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準の LLD ルールプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
filter	object	LLD ルールフィルターオブジェクトを使用して現在のフィルターを置き換えます
preprocessing	array	LLD ルール事前処理オプションを使用して、現在の事前処理オプションを置き換えます。
lld_macro_paths	array	LLD ルールlld_macro_pathオプション
overrides	array	LLD ルールオーバーライドオプション

戻り値

(object) itemids プロパティの下で更新された LLD ルールの ID を含むオブジェクトを返します。

例

LLD ルールへのフィルターの追加

{#FSTYPE} マクロの内容が @File systems for discovery の正規表現と一致するようにフィルターを追加します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "discoveryrule.update",
  "params": {
    "itemid": "22450",
    "filter": {
      "evaltype": 1,
      "conditions": [
        {
          "macro": "{#FSTYPE}",
          "value": "@File systems for discovery"
        }
      ]
    }
  }
}

```

```
    }
  ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "22450"
    ]
  },
  "id": 1
}
```

LLD マクロパスの追加

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.update",
  "params": {
    "itemid": "22450",
    "lld_macro_paths": [
      {
        "lld_macro": "#{MACRO1}",
        "path": "$.json.path"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "22450"
    ]
  },
  "id": 1
}
```

トラッピングを無効にする

ディスクバリルールの LLD トラッピングを無効にします

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.update",
  "params": {
    "itemid": "28336",
    "allow_traps": 0
  },
  "id": 36,
  "auth": "d678e0b85688ce578ff061bd29a20d3b"
}
```

```
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "28336"
    ]
  },
  "id": 36
}
```

LLD ルールの事前処理オプション更新

事前処理ルール"JSONPath" で LLD ルールを更新します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.update",
  "params": {
    "itemid": "44211",
    "preprocessing": [
      {
        "type": 12,
        "params": "$.path.to.json",
        "error_handler": 2,
        "error_handler_params": "5"
      }
    ]
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "44211"
    ]
  },
  "id": 1
}
```

LLD ルールスクリプトの更新

LLD ルールスクリプトを別のスクリプトで更新し、前のスクリプトで使用されていた不要なパラメーターを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "discoveryrule.update",
  "params": {
    "itemid": "23865",
    "parameters": [],
    "script": "Zabbix.Log(3, 'Log test');\nreturn 1;"
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "itemids": [
      "23865"
    ]
  },
  "id": 1
}
```

ソース

CDiscoveryRule::update() in ui/include/classes/api/services/CDiscoveryRule.php.

Maintenance

このクラスは、メンテナンスで機能するように設計されています。

オブジェクトリファレンス:

- [Maintenance](#)
- [Time period](#)

利用可能なメソッド:

- [maintenance.create](#) - 新しいメンテナンスの作成
- [maintenance.delete](#) - メンテナンスの削除
- [maintenance.get](#) - メンテナンスの取得
- [maintenance.update](#) - メンテナンスの更新

> Maintenance object

次のオブジェクトは、maintenanceAPI に直接関連しています。

メンテナンス

メンテナンスオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
maintenanceid	string	(読取専用) メンテナンスの ID
name (必須)	string	メンテナンス名
active_since (必須)	timestamp	メンテナンスがアクティブになる時間。
active_till (必須)	timestamp	指定された値は分に切り捨てられます。 メンテナンスがノンアクティブになる時間。
description	string	指定された値は分に切り捨てられます。 メンテナンスの説明
maintenance_type	integer	メンテナンスの種類
tags_evaltype	integer	利用可能な値: 0 - (デフォルト) データ収集あり 1 - データ収集なし 障害タグの評価方法
		可能な値: 0 - (デフォルト) And/Or 2 - Or

期間

期間オブジェクトは、メンテナンスを有効にする必要がある期間を定義するために使用されます。以下のプロパティがあります。

プロパティ	タイプ	説明
period	integer	メンテナンス期間 (秒単位) 指定された値は分で切り捨てられます
timeperiod_type	integer	デフォルト:3600 期間のタイプ 可能な値: 0 - (デフォルト)1 回のみ 2 - 毎日 3 - 毎週 4 - 毎月
start_date	timestamp	メンテナンス期間を有効にする日 1 つの期間にのみ使用されます。 指定された値は分で切り捨てられます。
start_time	integer	デフォルト: 現在の日付 メンテナンスが開始される時刻 (秒単位) 毎日、毎週、毎月の期間に使用されます。 指定された値は分で切り捨てられます。
every	integer	デフォルト:0 毎日、毎週、毎月の期間で使用されます。 毎日および毎週の期間では”すべて” はメンテナンスを有効にする必要がある日または週の間隔を定義します。 デフォルト:1 毎月の期間で、dayofweek プロパティに少なくとも 1 つの選択された曜日が含まれている場合、every プロパティはメンテナンスを有効にする必要がある月の週を定義します。 可能な値:< br>1 - (デフォルト) 第 1 週 2 - 第 2 週 3 - 第 3 週 4 - 第 4 週 5 - 先週
dayofweek	integer	メンテナンスを有効にする曜日 バイナリ形式で保存され、各ビットは対応する日を表します。たとえば 4 はバイナリで 100 に等しく、水曜日にメンテナンスが有効になることを意味します。 毎週および毎月の期間に使用されます。毎週のみ必須です。
day	integer	毎月の期間には、少なくとも 1 つの「dayofweek」または「day」を指定する必要があります。 メンテナンスを有効にする必要がある月の日 毎月にものみ使用されます 毎月の期間には、少なくとも 1 つの「曜日」または「日」を指定する必要があります。

プロパティ	タイプ	説明
month	integer	メンテナンスを有効にする月。 バイナリ形式で保存され、各ビットは対応する月を表します。たとえば5はバイナリで101に等しく、メンテナンスが1月と3月に有効になることを意味します。 毎月の期間のときのみ必須です

障害タグ

障害タグオブジェクトは、メンテナンスが有効になったときにどの障害を抑制しなければならないかを定義するために使用されます。以下のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	障害タグ名
operator	integer	条件演算子 可能な値: 0 - 等しい 2 - (デフォルト) 含む
value	string	障害タグ値

タグはデータ収集 ("maintenance_type ":0) を使用したメンテナンス期間にのみ指定できます。

maintenance.create

説明

`object maintenance.create(object/array maintenances)`

このメソッドで、新しいメンテナンスを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) メンテナンスの作成

このメソッドは標準のメンテナンスプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
groups	object/array	メンテナンスが行われるホスト groups ホストグループには <code>groupid</code> プロパティが定義されている必要があります。
hosts	object/array	groups または hosts のオブジェクトを少なくとも 1 つ指定する必要があります。 メンテナンスが行われる ホスト ホストには <code>hostid</code> プロパティが定義されている必要があります。
timeperiods (必須)	object/array	groups または hosts のオブジェクトを少なくとも 1 つ指定する必要があります。 メンテナンス 期間

パラメータ	タイプ	説明
tags	object/array	障害タグ どの障害を抑制するかを定義します。 タグが指定されていない場合、すべてのアクティブなメンテナンスホストの障害が抑制されます。

戻り値

(object) maintenanceids プロパティの下で作成されたメンテナンスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメンテナンスの順序と一致します。

例

メンテナンスの作成

ID"2" のホストグループのデータ収集と、障害タグ **service:mysqld** および **error** を使用してメンテナンスを作成します。2013 年 1 月 22 日から 2014 年 1 月 22 日までアクティブで、毎週日曜日の 18:00 に有効になり 1 時間持続します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "maintenance.create",
  "params": {
    "name": "Sunday maintenance",
    "active_since": 1358844540,
    "active_till": 1390466940,
    "tags_evaltype": 0,
    "groups": [
      {"groupid": "2"}
    ],
    "timeperiods": [
      {
        "period": 3600,
        "timeperiod_type": 3,
        "start_time": 64800,
        "every": 1,
        "dayofweek": 64
      }
    ],
    "tags": [
      {
        "tag": "service",
        "operator": "0",
        "value": "mysqlld"
      },
      {
        "tag": "error",
        "operator": "2",
        "value": ""
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "maintenanceids": [
      "3"
    ]
  }
}
```

```
    ],  
    },  
    "id": 1  
  }  
}
```

参照

- [Time period](#)

ソース

CMaintenance::create() in ui/include/classes/api/services/CMaintenance.php.

maintenance.delete

説明

object maintenance.delete(array maintenanceIds)

このメソッドを使用すると、メンテナンス期間を削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するメンテナンス期間の ID。

戻り値

(object) maintenanceids プロパティの下に削除されたメンテナンス期間の ID を含むオブジェクトを返します。

例

複数のメンテナンス期間の削除

2つのメンテナンス期間を削除します。

リクエスト:

```
{  
  "jsonrpc": "2.0",  
  "method": "maintenance.delete",  
  "params": [  
    "3",  
    "1"  
  ],  
  "auth": "3a57200802b24cda67c4e4010b50c065",  
  "id": 1  
}
```

レスポンス:

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "maintenanceids": [  
      "3",  
      "1"  
    ]  
  },  
  "id": 1  
}
```

ソース

CMaintenance::delete() in ui/include/classes/api/services/CMaintenance.php.

maintenance.get

説明

integer/array maintenance.get(object parameters)

このメソッドで、指定されたパラメーターに従ってメンテナンスを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
groupids	string/array	指定されたホストグループに割り当てられているメンテナンスのみを返します
hostids	string/array	指定されたホストに割り当てられているメンテナンスのみを返します
maintenanceids	string/array	指定された ID のメンテナンスのみを返します
selectGroups	query	メンテナンスに割り当てられたホストグループを持つ <code>groups</code> プロパティを返します。
selectHosts	query	メンテナンスに割り当てられたホストを持つ <code>hosts</code> プロパティを返します。
selectTags	query	メンテナンスの障害タグを含む <code>tags</code> プロパティを返します。
selectTimeperiods	query	メンテナンスの期間を含む <code>timeperiods</code> プロパティを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	利用可能な値は <code>maintenanceid,name,maintenance_type</code> です。すべての <code>get</code> メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得されたオブジェクトの数

例

メンテナンスの取得

構成されたすべてのメンテナンス、割り当てられたホストグループ、定義された期間、障害タグに関するデータを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "maintenance.get",
  "params": {
    "output": "extend",
```

```

    "selectGroups": "extend",
    "selectTimeperiods": "extend",
    "selectTags": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "3",
      "name": "Sunday maintenance",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1358844540",
      "active_till": "1390466940",
      "tags_evaltype": "0",
      "groups": [
        {
          "groupid": "4",
          "name": "Zabbix servers",
          "internal": "0"
        }
      ],
      "timeperiods": [
        {
          "timeperiod_type": "3",
          "every": "1",
          "month": "0",
          "dayofweek": "1",
          "day": "0",
          "start_time": "64800",
          "period": "3600",
          "start_date": "2147483647"
        }
      ],
      "tags": [
        {
          "tag": "service",
          "operator": "0",
          "value": "mysqld",
        },
        {
          "tag": "error",
          "operator": "2",
          "value": ""
        }
      ]
    }
  ],
  "id": 1
}

```

参照

- [Host](#)
- [Host group](#)
- [Time period](#)

ソース

CMaintenance::get() in ui/include/classes/api/services/CMaintenance.php.

maintenance.update

説明

object maintenance.update(object/array maintenances)

このメソッドで、既存のメンテナンスを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるメンテナンスプロパティ

maintenanceid プロパティは、メンテナンスごとに定義する必要があります。すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のメンテナンスプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
groups	object/array	ホスト groups を使用して、現在のグループを置き換えます。
hosts	object/array	ホストグループには groupid プロパティを定義する必要があります。 ホスト は現在のホストを置き換えます。
timeperiods	object/array	ホストには hostid プロパティが定義されている必要があります。 現在のメンテナンス期間を置き換えるための期間
tags	object/array	障害タグで現在のタグを置き換えます。

Attention:

メンテナンスごとに、少なくとも 1 つのホストまたはホストグループを定義する必要があります。

戻り値

(object) maintenanceids プロパティの下で更新されたメンテナンスの ID を含むオブジェクトを返します。

例

異なるホストの割り当て

現在メンテナンスに割り当てられているホストを 2 つの異なるホストに置き換えます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "maintenance.update",
  "params": {
    "maintenanceid": "3",
    "hosts": [
      {"hostid": "10085"},
      {"hostid": "10084"}
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "maintenanceids": [
      "3"
    ]
  },
  "id": 1
}
```

参照

- [Time period](#)

ソース

CMaintenance::update() in ui/include/classes/api/services/CMaintenance.php.

Map

このクラスは、マップを操作するように設計されています。

オブジェクトリファレンス:

- [Map](#)
- [Map element](#)
- [Map link](#)
- [Map URL](#)
- [Map user](#)
- [Map user group](#)
- [Map shape](#)
- [Map line](#)

利用可能なメソッド:

- [map.create](#) - 新しいマップの作成
- [map.delete](#) - マップの削除
- [map.get](#) - マップの取得
- [map.update](#) - マップの更新

> Map object

次のオブジェクトは map API に直接関連しています。

Map

Map オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
sysmapid	string	(読取専用) マップの ID
height (必須)	integer	マップの高さ (ピクセル)
name (必須)	string	マップの名前
width (必須)	integer	マップの横幅 (ピクセル)
backgroundid	string	マップの背景として使用される画像の ID
expand_macros	integer	マップの構成時にラベルのマクロを展開するかどうか

有効な値:

- 0 - (デフォルト) 展開しない
- 1 - 展開する

プロパティ	タイプ	説明
expandproblem	integer	1つの障害がある要素に対して障害のトリガーを表示するかどうか 有効な値: 0 - 常に問題の数を表示する
grid_align	integer	1 - (デフォルト) 障害が1つしかない場合は障害のトリガーを表示します グリッド配置 有効な値: 0 - 無効
grid_show	integer	1 - (デフォルト) 有効 地図上のグリッド表示 有効な値: 0 - 表示しない
grid_size	integer	1 - (デフォルト) 表示する 地図上のグリッドピクセルサイズ 有効な値:20,40,50,75,100
highlight	integer	デフォルト: 50 アイコンの強調表示 有効な値: 0 - しない
iconmapid	string	1 - (デフォルト) する マップで使用されるアイコンマップの ID
label_format	integer	高度なラベルの有効化 有効な値: 0 - (デフォルト) 無効
label_location	integer	1 - 有効 マップエレメントラベルの位置 有効な値: 0 - (デフォルト) 下
label_string_host	string	1 - 左 2 - 右 3 - 上 ホスト要素のカスタムラベル
label_string_hostgroup	string	カスタムホストラベルタイプのマップでは必須 ホストグループ要素のカスタムラベル
label_string_image	string	カスタムホストグループラベルタイプのマップでは必須 画像要素のカスタムラベル
label_string_map	string	カスタム画像ラベルタイプの地図では必須 マップ要素のカスタムラベル
label_string_trigger	string	カスタムマップラベルタイプのマップでは必須 トリガー要素のカスタムラベル
label_type	integer	カスタムトリガーラベルタイプのマップでは必須 マップ要素のラベルタイプ 有効な値: 0 - ラベル 1 - IP アドレス 2 - (デフォルト) 要素名 3 - ステータスのみ 4 - 無し

プロパティ	タイプ	説明
label_type_host	integer	<p>ホスト要素のラベルタイプ</p> <p>有効な値: 0 - ラベル 1 - IP アドレス 2 - (デフォルト) 要素名 3 - ステータスのみ 4 - 無し 5 - カスタム</p>
label_type_hostgroup	integer	<p>ホストグループ要素のラベルタイプ</p> <p>有効な値: 0 - ラベル 2 - (デフォルト) 要素名 3 - ステータスのみ 4 - 無し 5 - カスタム</p>
label_type_image	integer	<p>ホストグループ要素のラベルタイプ</p> <p>有効な値: 0 - ラベル 2 - (デフォルト) 要素名 4 - 無し 5 - カスタム</p>
label_type_map	integer	<p>マップ要素のラベルタイプ</p> <p>有効な値: 0 - ラベル 2 - (デフォルト) 要素名 3 - ステータスのみ 4 - なし 4 - 無し 5 - カスタム</p>
label_type_trigger	integer	<p>トリガー要素のラベルタイプ</p> <p>有効な値: 0 - ラベル 2 - (デフォルト) 要素名 3 - ステータスのみ 4 - 無し 5 - カスタム</p>
markelements	integer	<p>最近ステータスが変更されたマップ要素の強調表示</p> <p>有効な値: 0 - (デフォルト) 強調表示しない 1 - 要素を強調表示する</p>
severity_min	integer	<p>マップに表示されるトリガーの最小深刻度</p> <p>サポートされているトリガー深刻度のリストについては、トリガーの”深刻度”プロパティを参照してください</p>
show_unack	integer	<p>障害の表示方法</p> <p>有効な値: 0 - (デフォルト) すべての障害数を表示 1 - 未確認の障害数のみ表示 2 - 確認済みの障害数と未確認の障害数を別々に表示</p>
userid	string	<p>マップ所有者のユーザー ID</p>
private	integer	<p>マップの共有</p> <p>有効な値: 0 - 公開 1 - (デフォルト) 非公開</p>

プロパティ	タイプ	説明
show_suppressed	integer	抑制された障害の表示 有効な値: 0 - (デフォルト) 非表示 1 - 表示

Map element

map element オブジェクトは、マップ上に表示されるオブジェクトを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
selementid	string	(読取専用) マップ要素の ID
elements (必須)	array	要素データオブジェクト。ホスト、ホストグループ、トリガー、およびマップタイプ要素に必要です。
elementtype (必須)	integer	マップ要素のタイプ 有効な値: 0 - ホスト 1 - マップ 2 - トリガー 3 - ホストグループ 4 - 画像
iconid_off (必須)	string	デフォルト状態で要素を表示するために使用される画像の ID
areatype	integer	個別のホストグループホストを表示する方法。 有効な値: 0 - (デフォルト) ホストグループ要素がマップ全体を占有 1 - ホストグループ要素はサイズ固定
elementsubtype	integer	ホストグループ要素をマップ上に表示する方法 有効な値: 0 - (デフォルト) ホストグループを 1 つの要素として表示 1 - 各ホストを表示グループで別々に
evaltype	integer	マップ要素タグのフィルタリング条件評価方法 有効な値: 0 - (デフォルト) AND/OR 2 - OR
height	integer	固定サイズのホストグループ要素の高さ (ピクセル単位) デフォルト: 200
iconid_disabled	string	無効な地図要素を表示するために使用される画像の ID。画像要素には使用されません。
iconid_maintenance	string	メンテナンスでマップ要素を表示するために使用される画像の ID。画像要素には使用されません。
iconid_on	string	障害のあるマップ要素を表示するために使用される画像の ID。画像要素には使用されません。
label	string	要素のラベル。
label_location	integer	マップ要素ラベル位置 有効な値: -1 - (デフォルト) デフォルト位置 0 - 下; 1 - 左; 2 - 右; 3 - 上

プロパティ	タイプ	説明
permission	integer	権限レベル 有効な値: -1 - 無し 2 - 読取専用 3 - 読取/書込
sysmapid	string	(読取専用) 要素が属するマップの ID
urls	array	マップ要素の URL
use_iconmap	integer	マップ要素の URL オブジェクトについては、 以下で詳しく説明 します。 ホスト要素にアイコンマッピングを使用するか 有効な値: 0 - 使用しない 1 - (デフォルト) 使用する
viewtype	integer	ホストグループ要素配置アルゴリズム 可能な値: 0 - (デフォルト) グリッド
width	integer	固定サイズのホストグループ要素の幅 (ピクセル単位)
x	integer	デフォルト: 200 要素の X 座標 (ピクセル単位)
y	integer	デフォルト: 0 要素の Y 座標 (ピクセル単位)
		デフォルト: 0

Map element Host

map element Host オブジェクトは 1 つのホスト要素を定義します

プロパティ	タイプ	説明
hostid	string	ホスト ID

Map element Host group

map element Host group オブジェクトは、1 つのホストグループ要素を定義します。

プロパティ	タイプ	説明
groupid	string	ホストグループ ID

Map element Map

map element Map オブジェクトは、1 つのマップ要素を定義します。

プロパティ	タイプ	説明
sysmapid	string	マップ ID

Map element Trigger

map element Trigger オブジェクトは、1 つまたは複数のトリガー要素を定義します。

プロパティ	タイプ	説明
triggerid	string	トリガー ID

Map element tag

map element tag オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	マップ要素のタグ名
operator	string	要素タグの条件演算子をマップします 有効な値: 0 - (デフォルト) 含む 1 - 等しい 2 - 含まない 3 - 含まない等しい 4 - 存在する 5 - 存在しない
value	string	マップ要素のタグ値

Map element URL

map element URL オブジェクトは、特定のマップ要素で使用できるクリック可能なリンクを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
sysmapelementurlid	string	(読取専用) マップ要素 URL の ID
name (必須)	string	リンクのキャプション
url (必須)	string	リンク URL
selementid	string	URL が属するマップ要素の ID

Map link

Map link オブジェクトは 2 つのマップ要素間のリンクを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
linkid	string	(読取専用) マップリンクの ID
selementid1 (必須)	string	一方の端にリンクされている最初のマップ要素の ID
selementid2 (必須)	string	もう一方の端にリンクされている最初のマップ要素の ID
color	string	ラインの色を 16 進数のカラーコードで表します デフォルト: 000000
drawtype	integer	リンク線の描画スタイル 有効な値: 0 - (デフォルト) 線 2 - 太線 3 - 点線 4 - 破線
label	string	リンクラベル
linktriggers	array	リンクステータスインジケータとして使用するマップリンクトリガー マップリンクトリガーオブジェクトについては、 以下で詳しく説明 します。
permission	integer	権限レベル 有効な値: -1 - 無し 2 - 読取専用 3 - 読取/書込

プロパティ	タイプ	説明
sysmapid	string	リンクが属する地図の ID。

Map link trigger

The map link trigger オブジェクトはトリガーの状態に基づいてマップリンクステータスインジケータを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
linktriggerid	string	(読取専用) マップリンクトリガーの ID
triggerid (必須)	string	リンクインジケータとして使用されるトリガーの ID
color	string	インジケータの色を 16 進数のカラーコードで表したもの。
drawtype	integer	デフォルト: DD0000 インジケータの描画スタイル 有効な値: 0 - (デフォルト) 線 2 - 太線 3 - 点線 4 - 破線
linkid	string	リンクトリガーが属するマップリンクの ID

Map URL

The map URL オブジェクトはマップ上の特定タイプすべての要素で使用できるクリック可能なリンクを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
sysmapurlid	string	(読取専用) MapURL の ID
name (必須)	string	リンクキャプション
url (必須)	string	リンク URL
elementtype	integer	URL を使用できる地図要素のタイプ サポートされているタイプのリストについては、 地図要素の "type" プロパティ を参照してください。
sysmapid	string	デフォルト: 0 URL が属するマップの ID

Map user

ユーザーに基づくマップ権限のリスト。次のプロパティがあります。

プロパティ	タイプ	説明
sysmapuserid	string	(読取専用) マップユーザーの ID
userid (必須)	string	ユーザー ID
permission (必須)	integer	権限レベル 有効な値: 2 - 読取専用 3 - 読書可能

Map user group

ユーザーグループに基づくマップ権限のリスト。次のプロパティがあります。

プロパティ	タイプ	説明
sysmapusrgrpid	string	(読取専用) マップユーザーグループの ID
usrgrpid (必須)	string	ユーザーグループ ID
permission (必須)	integer	権限レベル 有効な値: 2 - 読取専用 3 - 読書可能

Map shapes

Map shapes オブジェクトは、マップ上に表示される幾何学的形状 (テキストの有無にかかわらず) を定義します。次のプロパティがあります。

プロパティ	タイプ	説明
sysmap_shapeid	string	(読取専用) マップシェイプエレメントの ID
type (必須)	integer	マップシェイプエレメントのタイプ 有効な値: 0 - 長方形 1 - 楕円形
x	integer	新しいシェイプを作成する場合はプロパティが必要です。 シェイプの X 座標 (ピクセル単位)
y	integer	デフォルト: 0 シェイプの Y 座標 (ピクセル単位)
width	integer	デフォルト: 0 シェイプの幅 (ピクセル単位)
height	integer	デフォルト: 200 シェイプの高さ (ピクセル単位)
text	string	デフォルト: 200 シェイプのテキスト
font	integer	シェイプ内のテキストのフォント 有効な値: 0 - Georgia, serif 1 - "Palatino Linotype", "Book Antiqua", Palatino, serif 2 - "Times New Roman", Times, serif 3 - Arial, Helvetica, sans-serif 4 - "Arial Black", Gadget, sans-serif 5 - "Comic Sans MS", cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - "Lucida Sans Unicode", "Lucida Grande", sans-serif 8 - Tahoma, Geneva, sans-serif 9 - "Trebuchet MS", Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - "Courier New", Courier, monospace 12 - "Lucida Console", Monaco, monospace
font_size	integer	デフォルト: 9 ピクセル単位のフォントサイズ
font_color	string	デフォルト: 11 フォントカラー デフォルト: '000000'

プロパティ	タイプ	説明
text_halign	integer	テキストの水平方向配置 有効な値: 0 - 中央 1 - 左 2 - 右
text_valign	integer	デフォルト: 0 テキストの垂直方向配置 有効な値: 0 - 中央 1 - 上 2 - 下
border_type	integer	デフォルト: 0 ボーダータイプ 可能な値: 0 - 無し 1 - _____ 2 - --- 3 - - - -
border_width	integer	デフォルト: 0 ボーダーの幅 (ピクセル単位)
border_color	string	デフォルト: 0 ボーダーカラー
background_color	string	デフォルト: '000000' 背景色 (塗りつぶし)
zindex	integer	デフォルト: (空) すべての形状と線の順序付けに使用される値 (z-index) デフォルト: 0

Map lines

map line オブジェクトは、マップに表示されるラインを定義します。次のプロパティがあります：

プロパティ	タイプ	説明
sysmap_shapeid	string	(読取専用) 地図形状要素の ID
x1	integer	ラインポイント 1 の X 座標 (ピクセル単位) デフォルト: 0
y1	integer	ラインポイント 1 の Y 座標 (ピクセル単位) デフォルト: 0
x2	integer	ラインポイント 2 の X 座標 (ピクセル単位) デフォルト: 200
y2	integer	ラインポイント 2 の Y 座標 (ピクセル単位) デフォルト: 200

プロパティ	タイプ	説明
line_type	integer	線のタイプ 使用可能な値: 0 - 無し 1 - _____ 2 - -- 3 - - - -
line_width	integer	デフォルト: 0 ラインの幅 (ピクセル単位)
line_color	string	デフォルト: 0 ラインの色
zindex	integer	デフォルト: '000000' すべての形と線の順序付けに使用される値 (z-index) デフォルト: 0

map.create

説明

object map.create(object/array maps)

このメソッドで、新しいマップを作成できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) マップの作成

このメソッドは標準のマッププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
links	array	マップ上に作成するマップリンク
selements	array	マップ上に作成するマップエレメント
urls	array	マップ上に作成するマップURLs
users	array	マップ上に作成するマップユーザー共有
userGroups	array	マップ上に作成するマップユーザーグループ共有
shapes	array	マップ上に作成するマップ形状
lines	array	マップ上に作成するマップライン

Note:

マップリンクを作成するには、マップ要素の selementid を任意の値に設定し、この値を使用してリンクの selementid1 または selementid2 プロパティでこの要素を参照する必要があります。要素が作成されるとこの値は Zabbix によって生成された正しい ID に置き換えられます。[例を参照](#)

戻り値

(object) sysmapids プロパティの下で作成されたマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたマップの順序と一致します。

例

空のマップを作成する

要素のないマップを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.create",
  "params": {
    "name": "Map",
    "width": 600,
    "height": 600
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "8"
    ]
  },
  "id": 1
}
```

ホストマップを作成する

2つのホスト要素とそれらの間のリンクを使用してマップを作成します。マップ要素を参照するために、マップリンクオブジェクトで一時的な「selementid1」および「selementid2」値を使用していることに注意してください。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.create",
  "params": {
    "name": "Host map",
    "width": 600,
    "height": 600,
    "selements": [
      {
        "selementid": "1",
        "elements": [
          {"hostid": "1033"}
        ],
        "elementtype": 0,
        "iconid_off": "2"
      },
      {
        "selementid": "2",
        "elements": [
          {"hostid": "1037"}
        ],
        "elementtype": 0,
        "iconid_off": "2"
      }
    ],
    "links": [
      {
        "selementid1": "1",
        "selementid2": "2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
}
```



```
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "9"
    ]
  },
  "id": 1
}
```

トリガーマップを作成する

2つのトリガーを含むトリガー要素を使用してマップを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.create",
  "params": {
    "name": "Trigger map",
    "width": 600,
    "height": 600,
    "selements": [
      {
        "elements": [
          {"triggerid": "12345"},
          {"triggerid": "67890"}
        ],
        "elementtype": 2,
        "iconid_off": "2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "10"
    ]
  },
  "id": 1
}
```

地図の共有

2種類の共有（ユーザーとユーザーグループ）マップを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.create",
  "params": {
    "name": "Map sharing",
    "width": 600,
    "height": 600,
```

```

    "users": [
      {
        "userid": "4",
        "permission": "3"
      }
    ],
    "userGroups": [
      {
        "usrgrpid": "7",
        "permission": "2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "9"
    ]
  },
  "id": 1
}

```

マップの形状

マップ名 title でマップを作成します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "map.create",
  "params": {
    "name": "Host map",
    "width": 600,
    "height": 600,
    "shapes": [
      {
        "type": 0,
        "x": 0,
        "y": 0,
        "width": 600,
        "height": 11,
        "text": "{MAP.NAME}"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "10"
    ]
  },
}

```

```
"id": 1
}
```

マップライン

マップラインを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.create",
  "params": {
    "name": "Map API lines",
    "width": 500,
    "height": 500,
    "lines": [
      {
        "x1": 30,
        "y1": 10,
        "x2": 100,
        "y2": 50,
        "line_type": 1,
        "line_width": 10,
        "line_color": "009900"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "11"
    ]
  },
  "id": 1
}
```

参照

- [Map element](#)
- [Map link](#)
- [Map URL](#)
- [Map user](#)
- [Map user group](#)
- [Map shape](#)
- [Map line](#)

ソース

CMap::create() in ui/include/classes/api/services/CMap.php.

map.delete

説明

object map.delete(array mapIds)

このメソッドで、マップを削除できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するマップの ID

戻り値

(object) sysmapids プロパティの下にある削除されたマップの ID を含むオブジェクトを返します。

例

複数のマップ削除

2 つのマップを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.delete",
  "params": [
    "12",
    "34"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "12",
      "34"
    ]
  },
  "id": 1
}
```

ソース

CMap::delete() in ui/include/classes/api/services/CMap.php.

map.get

説明

integer/array map.get(object parameters)

このメソッドで、指定されたパラメーターに従ってマップを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
sysmapids	string/array	指定された ID のマップのみを返します

パラメータ	タイプ	説明
userids	string/array	指定されたユーザー ID に属するマップのみを返します
expandUrls	flag	対応するマップ要素にグローバルマップ URL を追加し、すべてのマップ要素 URL のマクロを展開します
selectIconMap	query	マップで使用されているアイコンマップを含む <code>iconmap</code> プロパティを返します。
selectLinks	query	要素間のマップリンクを含む <code>links</code> プロパティを返します。
selectSelements	query	マップ要素とともに <code>selements</code> プロパティを返します。
selectUrls	query	マップ URL を含む <code>urls</code> プロパティを返します。
selectUsers	query	マップが共有されているユーザーの <code>users</code> プロパティを返します。
selectUserGroups	query	マップが共有されているユーザーグループを持つ <code>userGroups</code> プロパティを返します。
selectShapes	query	マップシェイプとともに <code>shapes</code> プロパティを返します。
selectLines	query	マップラインとともに <code>lines</code> プロパティを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	利用可能な値は <code>name,width,height</code> です。 すべての <code>get</code> メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

マップを取得する

マップ"3"に関するすべてのデータを取得します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "map.get",
  "params": {
    "output": "extend",
    "selectSelements": "extend",
    "selectLinks": "extend",
    "selectUsers": "extend",
    "selectUserGroups": "extend",
    "selectShapes": "extend",
    "selectLines": "extend",
    "sysmapids": "3"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "selements": [
        {
          "selementid": "10",
          "sysmapid": "3",
          "elementtype": "4",
          "evaltype": "0",
          "iconid_off": "1",
          "iconid_on": "0",
          "label": "Zabbix server",
          "label_location": "3",
          "x": "11",
          "y": "141",
          "iconid_disabled": "0",
          "iconid_maintenance": "0",
          "elementsubtype": "0",
          "areatype": "0",
          "width": "200",
          "height": "200",
          "tags": [
            {
              "tag": "service",
              "value": "mysqld",
              "operator": "0"
            }
          ],
          "viewtype": "0",
          "use_iconmap": "1",
          "urls": [],
          "elements": []
        },
        {
          "selementid": "11",
          "sysmapid": "3",
          "elementtype": "4",
          "evaltype": "0",
          "iconid_off": "1",
          "iconid_on": "0",
          "label": "Web server",
          "label_location": "3",
          "x": "211",
          "y": "191",
          "iconid_disabled": "0",
          "iconid_maintenance": "0",
          "elementsubtype": "0",
          "areatype": "0",
          "width": "200",
          "height": "200",
          "viewtype": "0",
          "use_iconmap": "1",
          "tags": [],
          "urls": [],
          "elements": []
        },
        {
          "selementid": "12",
          "sysmapid": "3",
          "elementtype": "0",
          "evaltype": "0",

```

```

        "iconid_off": "185",
        "iconid_on": "0",
        "label": "{HOST.NAME}\r\n{HOST.CONN}",
        "label_location": "0",
        "x": "111",
        "y": "61",
        "iconid_disabled": "0",
        "iconid_maintenance": "0",
        "elementsubtype": "0",
        "areatype": "0",
        "width": "200",
        "height": "200",
        "viewtype": "0",
        "use_iconmap": "0",
        "tags": [],
        "urls": [],
        "elements": [
            {
                "hostid": "10084"
            }
        ]
    },
    "links": [
        {
            "linkid": "23",
            "sysmapid": "3",
            "selementid1": "10",
            "selementid2": "11",
            "drawtype": "0",
            "color": "00CC00",
            "label": "",
            "linktriggers": []
        }
    ],
    "users": [
        {
            "sysmapuserid": "1",
            "userid": "2",
            "permission": "2"
        }
    ],
    "userGroups": [
        {
            "sysmapusrgrpid": "1",
            "usrgrpid": "7",
            "permission": "2"
        }
    ],
    "shapes": [
        {
            "sysmap_shapeid": "1",
            "type": "0",
            "x": "0",
            "y": "0",
            "width": "680",
            "height": "15",
            "text": "{MAP.NAME}",
            "font": "9",
            "font_size": "11",
            "font_color": "000000",
            "text_halign": "0",

```

```

        "text_valign": "0",
        "border_type": "0",
        "border_width": "0",
        "border_color": "000000",
        "background_color": "",
        "zindex": "0"
    }
],
"lines": [
    {
        "sysmap_shapeid": "2",
        "x1": 30,
        "y1": 10,
        "x2": 100,
        "y2": 50,
        "line_type": 1,
        "line_width": 10,
        "line_color": "009900",
        "zindex": "1"
    }
],
"sysmapid": "3",
"name": "Local network",
"width": "400",
"height": "400",
"backgroundid": "0",
"label_type": "2",
"label_location": "3",
"highlight": "1",
"expandproblem": "1",
"markelements": "0",
"show_unack": "0",
"grid_size": "50",
"grid_show": "1",
"grid_align": "1",
"label_format": "0",
"label_type_host": "2",
"label_type_hostgroup": "2",
"label_type_trigger": "2",
"label_type_map": "2",
"label_type_image": "2",
"label_string_host": "",
"label_string_hostgroup": "",
"label_string_trigger": "",
"label_string_map": "",
"label_string_image": "",
"iconmapid": "0",
"expand_macros": "0",
"severity_min": "0",
"userid": "1",
"private": "1",
"show_suppressed": "1"
}
],
"id": 1
}

```

参照

- [Icon map](#)
- [Map element](#)
- [Map link](#)
- [Map URL](#)

- [Map user](#)
- [Map user group](#)
- [Map shapes](#)
- [Map lines](#)

ソース

CMap::get() in ui/include/classes/api/services/CMap.php.

map.update

説明

object map.update(object/array maps)

このメソッドで、既存のマップを更新できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するマッププロパティ。

mapid プロパティはマップごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のマッピングプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
links	array	links をマップして、既存のリンクを置き換えます。
selements	array	elements をマップして、既存の要素を置き換えます。
urls	array	URLs をマップして、既存の URL を置き換えます。
users	array	user 共有をマップして、既存の要素を置き換えます
userGroups	array	user group の共有をマップして、既存の要素を置き換えます。
shapes	array	shapes をマップして、既存のシェイプを置き換えます。
lines	array	lines をマップして、既存の行を置き換えます。

Note:

新しいマップ要素間にマッピングリンクを作成するには、要素の `selementid` を任意の値に設定し、この値を使用して、リンクの `selementid1` または `selementid2` プロパティでこの要素を参照する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。[map.create](#) の例を参照してください

戻り値

(object) `sysmapids` プロパティの下で更新されたマップの ID を含むオブジェクトを返します。

例

マップのサイズ変更

マップのサイズを 1200x1200 ピクセルに変更

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.update",
  "params": {
    "sysmapid": "8",
    "width": 1200,
    "height": 1200
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "8"
    ]
  },
  "id": 1
}
```

マップの所有者変更

管理者とスーパー管理者のみが利用できます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "map.update",
  "params": {
    "sysmapid": "9",
    "userid": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 2
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "sysmapids": [
      "9"
    ]
  },
  "id": 2
}
```

参照

- [Map element](#)
- [Map link](#)
- [Map URL](#)
- [Map user](#)
- [Map user group](#)
- [Map shapes](#)
- [Map lines](#)

ソース

CMap::update() in ui/include/classes/api/services/CMap.php.

Media type

このクラスは、メディアタイプで機能するように設計されています。

オブジェクトリファレンス:

- [Media type](#)

利用可能なメソッド:

- [mediatype.create](#) - 新しいメディアタイプの作成
- [mediatype.delete](#) - メディアタイプの削除

- [mediatype.get](#) - メディアタイプの取得
- [mediatype.update](#) - メディアタイプの更新

> Media type object

次のオブジェクトは、mediatypeAPI に直接関連しています。

Media type

media type オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
mediatypeid	string	(読取専用) メディアタイプの ID
name (必須)	string	メディアタイプ名
type (必須)	integer	メディアタイプで使用する送信方法。 可能な値: 0 - 電子メール 1 - スクリプト 2 - SMS 4 - Webhook
exec_path	string	スクリプトメディアタイプの場合 exec_path には実行されたスクリプトの名前が含まれます。
gsm_modem	string	スクリプトメディアタイプで使用されます GSM モデムのシリアルデバイス名。
passwd	string	SMS メディアタイプで使用されます 認証パスワード
smtp_email	string	電子メールメディアタイプで使用されます 通知の送信元の電子メールアドレス。
smtp_helo	string	電子メールメディアタイプで使用されます SMTP HELO.
smtp_server	string	電子メールメディアタイプで使用されます SMTP サーバ
smtp_port	integer	電子メールメディアタイプで使用されます SMTP サーバ接続ポート
smtp_security	integer	SMTP 接続のセキュリティレベル。 利用可能な値: 0 - なし 1 - STARTTLS 2 - SSL/TLS
smtp_verify_host	integer	SMTP の SSL ホスト検証
smtp_verify_peer	integer	SMTP の SSL ピア検証 利用可能な値: 0 - いいえ 1 - はい
smtp_authentication	integer	SMTP 認証方法 利用可能な値: 0 - なし 1 - 通常のパスワード

プロパティ	タイプ	説明
status	integer	メディアタイプの有効無効 利用可能な値: 0 - (デフォルト) 有効 1 - 無効
username	string	ユーザー名
exec_params	string	電子メールメディアタイプで使用されます。 スクリプトパラメータ
maxsessions	integer	各パラメータは改行で終わります 並行して処理できるアラートの最大数 SMS で利用可能な値: 1 - (デフォルト)
maxattempts	integer	他のメディアタイプで利用可能な値: 0-100 アラート送信の最大試行回数 可能な値: 1-100
attempt_interval	string	デフォルト値: 3 再試行の間隔。接尾辞付きの秒と時間の単位を受け入れます。 可能な値: 0-1h
content_type	integer	デフォルト値: 10s メッセージフォーマット 可能な値: 0 - プレーンテキスト 1 - (デフォルト)html
script	string	メディアタイプ webhook スクリプト javascript 本体
timeout	string	メディアタイプの webhook スクリプトのタイムアウト。接尾辞付きの秒と時間の単位を受け入れます。 利用可能な値: 1-60s デフォルト値: 30s
process_tags	integer	Webhook スクリプトの応答をタグとして解釈し、これらのタグに関連するイベントに追加する必要があるかどうかを定義します。 可能な値: 0 - (デフォルト)Webhook スクリプトの応答を無視します 1 - Webhook スクリプトの応答をタグとして処理します
show_event_menu	integer	problem.get および event.get プロパティ urls にメディアタイプエントリを表示します。 利用可能な値: 0 - (デフォルト)urls エントリを追加しない 1 - メディアタイプを urls プロパティに追加する
event_menu_url	string	メディアタイプエントリの url プロパティを problem.get と event.get の urls プロパティに定義します。
event_menu_name	string	メディアタイプエントリの name プロパティを problem.get と event.get の urls プロパティに定義します。

プロパティ	タイプ	説明
parameters	array	webhook 入力パラメーターの配列
description	string	メディアタイプの説明

Webhook parameters

Webhook スクリプトが呼び出されたときに渡されるパラメーターには、次のプロパティがあります。

プロパティ	タイプ	説明
name (required)	string	パラメータ名
value	string	パラメータ値、サポートマクロ。ページで説明されているサポートされているマクロ。

Message template

message template オブジェクトは、通知を送信するためのアクション操作のデフォルトメッセージとして使用されるテンプレートを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
eventsources (必須)	integer	イベントソース 利用可能な値: 0 - トリガー 1 - ディスカバリ 2 - 自動登録 3 - インターナル 4 - サービス
recovery (必須)	integer	オペレーションモード 可能な値: 0 - 操作 1 - 回復操作 2 - 更新操作
subject	string	メッセージの件名
message	string	メッセージテキスト

mediatype.create

説明

`object mediatype.create(object/array mediaTypes)`

このメソッドで新しいメディアタイプを作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

パラメータ

(object/array) メディアタイプの作成

このメソッドは標準のメディアタイププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
parameters	array	メディアタイプに合わせて作成される Webhook パラメーター
message_templates	array	メディアタイプに合わせて作成される [メッセージテンプレート (object#message_template)]

戻り値

(object) mediatypeids プロパティの下に作成されたメディアタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメディアタイプの順序と一致します。

例

電子メールメディアタイプの作成

カスタム SMTP ポートとメッセージテンプレートを使用して、新しい電子メールメディアタイプを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "mediatype.create",
  "params": {
    "type": "0",
    "name": "E-mail",
    "smtp_server": "mail.example.com",
    "smtp_helo": "example.com",
    "smtp_email": "zabbix@example.com",
    "smtp_port": "587",
    "content_type": "1",
    "message_templates": [
      {
        "eventsourcing": "0",
        "recovery": "0",
        "subject": "Problem: {EVENT.NAME}",
        "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" started at {EVENT.TIME}."
      },
      {
        "eventsourcing": "0",
        "recovery": "1",
        "subject": "Resolved in {EVENT.DURATION}: {EVENT.NAME}",
        "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" has been resolved at {EVENT.TIME}."
      },
      {
        "eventsourcing": "0",
        "recovery": "2",
        "subject": "Updated problem in {EVENT.AGE}: {EVENT.NAME}",
        "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\""
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "mediatypeids": [
      "7"
    ]
  },
  "id": 1
}
```

スクリプトメディアタイプの作成

試行回数と試行間隔のカスタム値を使用して、新しいスクリプトメディアタイプを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "mediatype.create",
```

```

"params": {
  "type": "1",
  "name": "Push notifications",
  "exec_path": "push-notification.sh",
  "exec_params": "{ALERT.SENDTO}\n{ALERT.SUBJECT}\n{ALERT.MESSAGE}\n",
  "maxattempts": "5",
  "attempt_interval": "11s"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "mediatypeids": [
      "8"
    ]
  },
  "id": 1
}

```

webhook メディアタイプの作成

新しい webhook メディアタイプを作成します

Request:

```

{
  "jsonrpc": "2.0",
  "method": "mediatype.create",
  "params": {
    "type": "4",
    "name": "Webhook",
    "script": "var Webhook = {\r\n  token: null,\r\n  to: null,\r\n  subject: null,\r\n  messa",
    "parameters": [
      {
        "name": "Message",
        "value": "{ALERT.MESSAGE}"
      },
      {
        "name": "Subject",
        "value": "{ALERT.SUBJECT}"
      },
      {
        "name": "To",
        "value": "{ALERT.SENDTO}"
      },
      {
        "name": "Token",
        "value": "<Token>"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "mediatypeids": [

```

```
        "9"
    ],
},
"id": 1
}
```

ソース

CMediaType::create() in ui/include/classes/api/services/CMediaType.php.

mediatype.delete

説明

object mediatype.delete(array mediaTypeIds)

このメソッドで、メディアタイプを削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するメディアタイプの ID

戻り値

(object) mediatypeids プロパティの下で削除されたメディアタイプの ID を含むオブジェクトを返します。

例

複数のメディアタイプ削除

二つのメディアタイプを削除する

Request:

```
{
  "jsonrpc": "2.0",
  "method": "mediatype.delete",
  "params": [
    "3",
    "5"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "mediatypeids": [
      "3",
      "5"
    ]
  },
  "id": 1
}
```

ソース

CMediaType::delete() in ui/include/classes/api/services/CMediaType.php.

mediatype.get

説明

integer/array mediatype.get(object parameters)

このメソッドで、指定されたパラメーターに従ってメディアタイプを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
mediatypeids	string/array	指定された ID を持つメディアタイプのみを返します。
mediaids	string/array	指定されたメディアで使用されているメディアタイプのみを返します。
userid	string/array	指定されたユーザーが使用するメディアタイプのみを返します。
selectMessageTemplates	query	メディアタイプメッセージの配列を含む <code>message_templates</code> プロパティを返します。
selectUsers	query	メディアタイプを使用するユーザーの <code>users</code> プロパティを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	使用可能な値は <code>mediatypeid</code> です。 すべての <code>get</code> メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

メディアタイプの取得

構成されているすべてのメディアタイプを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "mediatype.get",
  "params": {
    "output": "extend",
    "selectMessageTemplates": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "mediatypeid": "1",
      "type": "0",
      "name": "Email",
      "smtp_server": "mail.example.com",
      "smtp_helo": "example.com",
      "smtp_email": "zabbix@example.com",
      "exec_path": "",
      "gsm_modem": "",
      "username": "",
      "passwd": "",
      "status": "0",
      "smtp_port": "25",
      "smtp_security": "0",
      "smtp_verify_peer": "0",
      "smtp_verify_host": "0",
      "smtp_authentication": "0",
      "exec_params": "",
      "maxsessions": "1",
      "maxattempts": "3",
      "attempt_interval": "10s",
      "content_type": "0",
      "script": "",
      "timeout": "30s",
      "process_tags": "0",
      "show_event_menu": "1",
      "event_menu_url": "",
      "event_menu_name": "",
      "description": "",
      "message_templates": [
        {
          "eventsourced": "0",
          "recovery": "0",
          "subject": "Problem: {EVENT.NAME}",
          "message": "Problem started at {EVENT.TIME} on {EVENT.DATE}\r\nProblem name: {EVENT.NA",
        },
        {
          "eventsourced": "0",
          "recovery": "1",
          "subject": "Resolved: {EVENT.NAME}",
          "message": "Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE",
        },
        {
          "eventsourced": "0",
          "recovery": "2",
          "subject": "Updated problem: {EVENT.NAME}",
          "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN",
        },
        {
          "eventsourced": "1",
          "recovery": "0",
          "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
          "message": "Discovery rule: {DISCOVERY.RULE.NAME}\r\n\r\nDevice IP: {DISCOVERY.DEVICE.",
        },
        {
          "eventsourced": "2",
          "recovery": "0",
          "subject": "Autoregistration: {HOST.HOST}",
          "message": "Host name: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.PORT}"
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "parameters": []
},
{
  "mediatypeid": "3",
  "type": "2",
  "name": "SMS",
  "smtp_server": "",
  "smtp_helo": "",
  "smtp_email": "",
  "exec_path": "",
  "gsm_modem": "/dev/ttyS0",
  "username": "",
  "passwd": "",
  "status": "0",
  "smtp_port": "25",
  "smtp_security": "0",
  "smtp_verify_peer": "0",
  "smtp_verify_host": "0",
  "smtp_authentication": "0",
  "exec_params": "",
  "maxsessions": "1",
  "maxattempts": "3",
  "attempt_interval": "10s",
  "content_type": "1",
  "script": "",
  "timeout": "30s",
  "process_tags": "0",
  "show_event_menu": "1",
  "event_menu_url": "",
  "event_menu_name": "",
  "description": "",
  "message_templates": [
    {
      "eventsourcing": "0",
      "recovery": "0",
      "subject": "",
      "message": "{EVENT.SEVERITY}: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
    },
    {
      "eventsourcing": "0",
      "recovery": "1",
      "subject": "",
      "message": "RESOLVED: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
    },
    {
      "eventsourcing": "0",
      "recovery": "2",
      "subject": "",
      "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}"
    },
    {
      "eventsourcing": "1",
      "recovery": "0",
      "subject": "",
      "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
    },
    {
      "eventsourcing": "2",
      "recovery": "0",
      "subject": "",
    }
  ]
}

```

```

        "message": "Autoregistration: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.P
    }
    ],
    "parameters": []
}
],
"id": 1
}

```

参照

- User

ソース

CMediaType::get() in ui/include/classes/api/services/CMediaType.php.

mediatype.update

説明

object mediatype.update(object/array mediaTypes)

このメソッドで、既存のメディアタイプを更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するメディアタイプのプロパティ

mediatypeid プロパティは、メディアタイプごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のメディアタイププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
parameters	array	Webhook パラメーター。現在の Webhook パラメーターを置き換えます。
message_templates	array	メッセージテンプレート。現在のメッセージテンプレートを置き換えます。

戻り値

(object) mediatypeids プロパティの下で更新されたメディアタイプの ID を含むオブジェクトを返します。

例

メディアタイプの有効化

ステータスを"0"に設定して、メディアタイプを有効にします。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "mediatype.update",
  "params": {
    "mediatypeid": "6",
    "status": "0"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "mediatypeids": [
      "6"
    ]
  },
  "id": 1
}
```

ソース

CMediaType::update() in ui/include/classes/api/services/CMediaType.php.

Problem

このクラスは、障害を処理するように設計されています。

オブジェクトリファレンス:

- **Problem**

利用可能なメソッド:

- **problem.get** - 障害の取得

> Problem object

Note:

障害は Zabbix サーバーによって作成され、API を介して変更することはできません。

Problem オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
eventid	string	障害のイベント ID
source	integer	障害のイベントタイプ
object	integer	<p>利用可能な値:</p> <p>0 - トリガーによって作成されたイベント</p> <p>3 - 内部イベント</p> <p>4 - サービスステータスの更新時に作成されたイベント</p> <p>障害のイベントに関連するオブジェクトのタイプ</p> <p>トリガーイベントの値:</p> <p>0 - トリガー</p> <p>内部イベントの値:</p> <p>0 - トリガー < br>4 - アイテム</p> <p>5 - LLD ルール</p> <p>サービスイベントの可能な値:</p> <p>6 - サービス</p>
objectid	string	関連オブジェクトの ID
clock	timestamp	障害のイベントが作成された時刻
ns	integer	障害のイベントが作成されたときのナノ秒
r_eventid	string	リカバリイベント ID
r_clock	timestamp	リカバリイベントが作成された時刻
r_ns	integer	リカバリイベントが作成されたときのナノ秒
correlationid	string	このイベントがグローバル相関ルールによって回復された場合の相関ルール ID
userid	string	障害が手動でクローズされた場合のユーザー ID

プロパティ	タイプ	説明
name	string	解決した障害名
acknowledged	integer	障害の状態を確認します
severity	integer	可能な値： 0 - 未確認 1 - 確認済 障害の現在深刻度
suppressed	integer	可能な値： 0 - 未分類 1 - 情報 2 - 警告 3 - 軽度の障害 4 - 重度の障害 5 - 致命的な障害 障害が抑制されているかどうか
opdata	string	可能な値： 0 - 障害は正常です 1 - 障害は抑制されています。 Whether the problem is suppressed.
urls	array of Media type URLs	Possible values: 0 - problem is in normal state; 1 - problem is suppressed. マクロが展開された運用データ アクティブメディアタイプの URL

Problem

Note:

Problems are created by the Zabbix server and cannot be modified via the API.

The problem object has the following properties.

Property	Type	Description
eventid	string	ID of the problem event.
source	integer	Type of the problem event.
object	integer	Possible values: 0 - event created by a trigger; 3 - internal event; 4 - event created on service status update. Type of object that is related to the problem event.
objectid	string	Possible values for trigger events: 0 - trigger. Possible values for internal events: 0 - trigger; 4 - item; 5 - LLD rule. Possible values for service events: 6 - service. ID of the related object.
clock	timestamp	Time when the problem event was created.
ns	integer	Nanoseconds when the problem event was created.
r_eventid	string	Recovery event ID.
r_clock	timestamp	Time when the recovery event was created.
r_ns	integer	Nanoseconds when the recovery event was created.

Property	Type	Description
correlationid	string	Correlation rule ID if this event was recovered by global correlation rule.
userid	string	User ID if the problem was manually closed.
name	string	Resolved problem name.
acknowledged	integer	Acknowledge state for problem. Possible values: 0 - not acknowledged; 1 - acknowledged.
severity	integer	Problem current severity. Possible values: 0 - not classified; 1 - information; 2 - warning; 3 - average; 4 - high; 5 - disaster.
suppressed	integer	Whether the problem is suppressed. Possible values: 0 - problem is in normal state; 1 - problem is suppressed.
opdata	string	Operational data with expanded macros.
urls	array of Media type URLs	Active media types URLs.

Problem tag

障害タグオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
tag	string	障害タグ名
value	string	障害タグ値

Media type URLs

メディアタイプが url のオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
name	string	メディアタイプで定義された URL 名
url	string	メディアタイプで定義された URL 値

結果には、イベントメニューエントリが有効になっているアクティブなメディアタイプのエントリのみが含まれます。プロパティで使用されるマクロは展開されますが、プロパティの 1 つに展開されていないマクロが含まれている場合、両方のプロパティが結果から除外されます。サポートされているマクロについては[ページ](#)に記載しています。

problem.get

説明

`integer/array problem.get(object parameters)`

このメソッドで、指定されたパラメーターに従って障害を取得できます。

このメソッドは、未解決の障害を取得するためのものです。指定されている場合は、最近解決された障害を追加で取得することもできます。“最近”の古さを決定する期間は、管理 →* **一般設定*** で定義されています。その期間より前に解決された障害は、障害テーブルに保持されません。過去にさらに解決された障害を取得するには、**event.get**メソッドを使用します。

Attention:

これらの障害がハウスキーパーによってまだ削除されていない場合、このメソッドは削除済みの障害を返す可能性があります。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
eventids	string/array	指定された ID の障害のみを返します。
groupids	string/array	指定されたホストグループに属するオブジェクトによって作成された障害のみを返します。
hostids	string/array	指定されたホストに属するオブジェクトによって作成された障害のみを返します。
objectids	string/array	指定されたオブジェクトによって作成された障害のみを返します。
source	integer	指定されたタイプの障害のみを返します。
		サポートされているイベントタイプのリストについては、 障害イベントオブジェクトページ を参照してください。
object	integer	デフォルト：0 - トリガーによって作成された障害 指定されたタイプのオブジェクトによって作成された障害のみを返します。
		サポートされているオブジェクトタイプのリストについては 障害イベントオブジェクトページ を参照してください。
acknowledged	boolean	デフォルト：0 - トリガー true - 確認済みの障害を返します false - 未確認の障害を返します
suppressed	boolean	true - 表示が抑制された障害を返します false - 通常の状態での障害を返します
severities	integer/array	指定されたイベントの深刻度に関する障害を返します。オブジェクトがトリガーの場合にのみ適用されます。
evaltype	integer	タグ検索のルール
tags	array of objects	利用可能な値： 0 - (デフォルト)And/Or 2 - Or 指定されたタグの障害のみを返します。タグによる完全一致、および値と演算子による大文字と小文字を区別しない検索。 フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...] 空の配列はすべての障害を返します。
		可能な演算子タイプ： 0 - (デフォルト)Like 1 - 等しい 2 - Not Like 3 - 等しくない 4 - 存在する 5 - 存在しない
recent	boolean	true - 障害と最近解決された障害を返します（「最近」は [Display OK] トリガーの N 秒表示に依存します） デフォルト：false-未解決の問題のみ
eventid_from	string	指定された ID 以上の ID を持つ障害を返します。

パラメータ	タイプ	説明
eventid_till	string	指定された ID 以下の ID の障害を返します。
time_from	timestamp	指定された時間以降に作成された障害を返します。
time_till	timestamp	指定された時間以前に作成された障害を返します。
selectAcknowledges	query	障害の更新とともに <code>acknowledges</code> プロパティを返します。障害の更新は、時系列の逆順に並べ替えられます。
		障害の更新オブジェクトには次のプロパティがあります。 <code>acknowledgeid</code> - (string) 更新の ID <code>userid</code> - (string) イベントを更新したユーザーの ID <code>eventid</code> - (文字列) 更新されたイベントの ID <code>clock</code> - (タイムスタンプ) イベントが更新された時刻 <code>message</code> - (string) メッセージのテキスト <code>action</code> - (integer) 更新アクションのタイプ (<code>event.acknowledge</code> を参照) <code>old_severity</code> - (integer) この更新アクション前のイベントの深刻度 <code>new_severity</code> - (integer) この更新アクション後のイベントの深刻度
selectTags	query	<code>count</code> をサポートします。 障害のあるタグを含む <code>tags</code> プロパティを返します。フォーマット： <code>[{"tag": "<tag>", "value": "<value>"}, ...]</code>
selectSuppressionData	query	メンテナンスのリストを含む <code>suppression_data</code> プロパティを返します。 <code>maintenanceid</code> - (string) メンテナンスの ID <code>suppress_until</code> - (integer) 問題が抑制されるまでの時間。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	可能な値は次のとおりです： <code>eventid</code> すべての <code>get</code> メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

トリガーの障害イベント取得

トリガー“15112”から最近のイベントを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "problem.get",
  "params": {
    "output": "extend",
    "selectAcknowledges": "extend",
    "selectTags": "extend",
    "selectSuppressionData": "extend",
```

```

    "objectids": "15112",
    "recent": "true",
    "sortfield": ["eventid"],
    "sortorder": "DESC"
  },
  "auth": "67f45d3eb1173338e1b1647c4bdc1916",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "eventid": "1245463",
      "source": "0",
      "object": "0",
      "objectid": "15112",
      "clock": "1472457242",
      "ns": "209442442",
      "r_eventid": "1245468",
      "r_clock": "1472457285",
      "r_ns": "125644870",
      "correlationid": "0",
      "userid": "1",
      "name": "Zabbix agent on localhost is unreachable for 5 minutes",
      "acknowledged": "1",
      "severity": "3",
      "opdata": "",
      "acknowledges": [
        {
          "acknowledgeid": "14443",
          "userid": "1",
          "eventid": "1245463",
          "clock": "1472457281",
          "message": "problem solved",
          "action": "6",
          "old_severity": "0",
          "new_severity": "0"
        }
      ],
      "suppression_data": [
        {
          "maintenanceid": "15",
          "suppress_until": "1472511600"
        }
      ],
      "suppressed": "1",
      "tags": [
        {
          "tag": "test tag",
          "value": "test value"
        }
      ]
    }
  ],
  "id": 1
}

```

参照

- Alert
- Item

- Host
- LLD rule
- Trigger

ソース

CEvent::get() in ui/include/classes/api/services/CProblem.php.

Proxy

このクラスは、プロキシで機能するように設計されています。

オブジェクトリファレンス:

- Proxy
- Proxy interface

利用可能なメソッド:

- **proxy.create** - 新しいプロキシの作成
- **proxy.delete** - プロキシの削除
- **proxy.get** - プロキシの取得
- **proxy.update** - プロキシの更新

> Proxy object

次のオブジェクトは、proxyAPI に直接関連しています。

Proxy

proxy オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
proxyid	string	(読取専用) プロキシ ID
host (必須)	string	プロキシ名
status (必須)	integer	プロキシのタイプ 使用可能な値: 5 - アクティブプロキシ 6 - パッシブプロキシ
description	text	プロキシの説明
lastaccess	timestamp	(読取専用) プロキシが最後にサーバーに接続した時刻。
tls_connect	integer	ホストへの接続 利用可能な値: 1 - (デフォルト) 暗号化なし 2 - PSK 4 - 証明書
tls_accept	integer	ホストからの接続。 利用可能なビットマップ値: 1 - (デフォルト) 暗号化なし 2 - PSK 4 - 証明書
tls_issuer	string	証明書の発行者
tls_subject	string	証明書の件名
tls_psk_identity	string	(書込専用) PSK アイデンティティ。tls_connect または tls_acceptのPSKが有効になっている場合に必要です。機密情報を PSKID に入れないでください。機密情報は暗号化されずにネットワーク経由で送信され、使用する PSK を受信者に通知します。

プロパティ	タイプ	説明
tls_psk	string	(書込専用) 事前共有キー。少なくとも 32 桁の 16 進数。tls_connect または tls_accept のいずれかで PSK が有効になっている場合に必要です。
proxy_address	string	アクティブな Zabbix プロキシのカンマ区切りの IP アドレスまたは DNS 名。
auto_compress	integer	(読取専用) Zabbix サーバーとプロキシ間の通信が圧縮されるか。 利用可能な値: 0 - 圧縮なし 1 - 圧縮有効

Proxy interface

proxy interface オブジェクトは、パッシブプロキシへの接続に使用されるインターフェイスを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
dns (required)	string	接続する DNS 名 IP アドレスで接続する場合は、空にすることができます。
ip (required)	string	接続する IP アドレス
port (required)	string	DNS 名で接続する場合は、空にすることができます。 接続ポート番号
useip (required)	integer	IP アドレスを介して接続するかどうか 利用可能な値: 0 - DNS 名で接続 1 - IP アドレスで接続

proxy.create

説明

object proxy.create(object/array proxies)

このメソッドで新しいプロキシを作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) プロキシの作成

このメソッドは標準プロキシプロパティに加えて、次のパラメータを受け入れます。

パラメータ	タイプ	説明
hosts	array	プロキシによって監視されるホスト。ホストがすでに別のプロキシによって監視されている場合は、現在のプロキシに再割り当てされます。
interface	object	ホストには hostid プロパティが定義されている必要があります。 パッシブプロキシ用に作成されるホスト インターフェイス パッシブプロキシで使用されます。

戻り値

(object) proxyids プロパティの下に作成されたプロキシの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシの順序と一致します。

例

アクティブプロキシ作成

アクションプロキシ"Active proxy" を作成し、それによって監視されるホストを割り当てます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "proxy.create",
  "params": {
    "host": "Active proxy",
    "status": "5",
    "hosts": [
      {
        "hostid": "10279"
      }
    ]
  },
  "auth": "ab9638041ec6922cb14b07982b268f47",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "proxyids": [
      "10280"
    ]
  },
  "id": 1
}
```

パッシブプロキシ作成

パッシブプロキシ"Passive proxy" を作成し、それによって監視される 2 つのホストを割り当てます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "proxy.create",
  "params": {
    "host": "Passive proxy",
    "status": "6",
    "interface": {
      "ip": "127.0.0.1",
      "dns": "",
      "useip": "1",
      "port": "10051"
    }
  },
  "hosts": [
    {
      "hostid": "10192"
    },
    {
      "hostid": "10139"
    }
  ]
},
  "auth": "ab9638041ec6922cb14b07982b268f47",
  "id": 1
}
```

```
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "proxyids": [
      "10284"
    ]
  },
  "id": 1
}
```

参照

- [Host](#)
- [Proxy interface](#)

ソース

CProxy::create() in ui/include/classes/api/services/CProxy.php.

proxy.delete

説明

object proxy.delete(array proxies)

このメソッドで、プロキシを削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するプロキシの ID

戻り値

(object) proxyids プロパティの下にある削除されたプロキシの ID を含むオブジェクトを返します。

例

複数のプロキシ削除

2つのプロキシを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "proxy.delete",
  "params": [
    "10286",
    "10285"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "proxyids": [
      "10286",
      "10285"
    ]
  }
}
```

```

    ],
    },
    "id": 1
}

```

ソース

CProxy::delete() in ui/include/classes/api/services/CProxy.php.

proxy.get

説明

integer/array proxy.get(object parameters)

このメソッドで、指定されたパラメーターに従ってプロキシを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
proxyids	string/array	指定された ID を持つプロキシのみを返します
selectHosts	query	プロキシによって監視されているホストで hosts プロパティを返します。
selectInterface	query	パッシブプロキシが使用するプロキシインターフェイスで interface プロパティを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	利用可能な値は <code>hostid,host,status</code> です。 すべての <code>get</code> メソッドに共通するこれらのパラメーターについては ファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのプロキシを取得する

構成済みのすべてのプロキシとそのインターフェイスを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "proxy.get",
  "params": {
    "output": "extend",
    "selectInterface": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "host": "Active proxy",
      "status": "5",
      "lastaccess": "0",
      "description": "",
      "tls_connect": "1",
      "tls_accept": "1",
      "tls_issuer": "",
      "tls_subject": "",
      "proxy_address": "",
      "auto_compress": "0",
      "proxyid": "30091",
      "interface": []
    },
    {
      "host": "Passive proxy",
      "status": "6",
      "lastaccess": "0",
      "description": "",
      "tls_connect": "1",
      "tls_accept": "1",
      "tls_issuer": "",
      "tls_subject": "",
      "proxy_address": "",
      "auto_compress": "0",
      "proxyid": "30092",
      "interface": {
        "interfaceid": "30109",
        "hostid": "30092",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10051"
      }
    }
  ],
  "id": 1
}
```

参照

- [Host](#)
- [Proxy interface](#)

ソース

CProxy::get() in ui/include/classes/api/services/CProxy.php.

proxy.update

説明

object proxy.update(object/array proxies)

このメソッドで、既存のプロキシを更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるプロキシプロパティ。

proxyid プロパティはプロキシごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準プロキシプロパティに加えて、次のパラメータを受け入れます。

パラメータ	タイプ	説明
hosts	array	プロキシによって監視されるHosts。ホストがすでに別のプロキシによって監視されている場合は、現在のプロキシに再割り当てされます。
interface	object	ホストには hostid プロパティが定義されている必要があります。ホストinterfaceを使用して、バッチプロキシの既存インターフェースを置き換えます。

戻り値

(object) proxyids プロパティの下で更新されたプロキシの ID を含むオブジェクトを返します。

例

プロキシによって監視されているホストを変更する

プロキシを更新して指定された2つのホストを監視します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "proxy.update",
  "params": {
    "proxyid": "10293",
    "hosts": [
      {
        "hostid": "10294"
      },
      {
        "hostid": "10295"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "proxyids": [
      "10293"
    ]
  }
}
```

```
    },  
    "id": 1  
  }  
}
```

プロキシステータスの変更

プロキシをアクティブプロキシに変更し、名前を"Active proxy" に変更します。

Request:

```
{  
  "jsonrpc": "2.0",  
  "method": "proxy.update",  
  "params": {  
    "proxyid": "10293",  
    "host": "Active proxy",  
    "status": "5"  
  },  
  "auth": "038e1d7b1735c6a5436ee9eae095879e",  
  "id": 1  
}
```

Response:

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "proxyids": [  
      "10293"  
    ]  
  },  
  "id": 1  
}
```

参照

- [Host](#)
- [Proxy interface](#)

ソース

CProxy::update() in ui/include/classes/api/services/CProxy.php.

Regular expression

このクラスは、グローバル正規表現で機能するように設計されています。

オブジェクトリファレンス:

- [Regular expression](#)

利用可能なメソッド:

- [regexp.create](#) - 新しい正規表現の作成
- [regexp.delete](#) - 正規表現の削除
- [regexp.get](#) - 正規表現の取得
- [regexp.update](#) - 正規表現の更新

> Regular expression object

次のオブジェクトは、`regexpAPI` に直接関連しています。

Regular expression

global regular expression オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
regexpid	string	(読取専用) 正規表現の ID
name (必須)	string	正規表現名
test_string	string	テスト用文字列

Expressions object

expressions オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
expression (必須)	string	正規表現
expression_type (必須)	integer	正規表現のタイプ 利用可能な値： 0 - 文字列が含まれている 1 - 任意の文字列が含まれている 2 - 文字列が含まれていない 3 - 結果が TRUE 4 - 結果が FALSE
exp_delimiter	string	式の区切り文字。expression_type の場合のみ任意の文字列が含まれます。 デフォルト値,
case_sensitive	integer	利用可能な値：, , / 大文字と小文字の区別 デフォルト値 0 可能な値： 0 - 大文字と小文字を区別しない 1 - 大文字と小文字を区別する

regexp.create

説明

object regexp.create(object/array regularExpressions)

このメソッドで、新しいグローバル正規表現を作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成する正規表現

このメソッドは標準プロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
expressions	array	Expressions オプション

戻り値

(object) regexpids プロパティの下で作成された正規表現の ID を含むオブジェクトを返します。

例

新しいグローバル正規表現の作成

Request:

```
{
  "jsonrpc": "2.0",
  "method": "regexp.create",
  "params": {
    "name": "Storage devices for SNMP discovery",
    "test_string": "/boot",
    "expressions": [
      {
        "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$",
        "expression_type": "4",
        "case_sensitive": "1"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "regexprids": [
      "16"
    ]
  },
  "id": 1
}
```

ソース

CRegexp::create() in ui/include/classes/api/services/CRegexp.php.

regexp.delete

説明

object regexp.delete(array regexprids)

このメソッドで、グローバル正規表現を削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除する正規表現の ID

戻り値

(object) regexprids プロパティの下にある削除された正規表現の ID を含むオブジェクトを返します。

例

複数のグローバル正規表現を削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "regexp.delete",
  "params": [
    "16",
  ],
}
```

```

    "17"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "regexpids": [
      "16",
      "17"
    ]
  },
  "id": 1
}

```

ソース

CRegex::delete() in ui/include/classes/api/services/CRegex.php.

regexp.get

説明

integer/array regexp.get(object parameters)

このメソッドで、指定されたパラメーターに従ってグローバル正規表現を取得できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします

パラメータ	タイプ	説明
regexpids	string/array	指定された ID を持つ正規表現のみを返します
selectExpressions	query	expressions プロパティを返します
sortfield	string/array	指定されたプロパティで結果を並べ替えます
countOutput	boolean	利用可能な値は、 regexpid と name です。 すべての get メソッドに共通するこれらのパラメーターについては フレックス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

グローバル正規表現の取得

Request:

```
{
  "jsonrpc": "2.0",
  "method": "regexp.get",
  "params": {
    "output": ["regexpid", "name"],
    "selectExpressions": ["expression", "expression_type"],
    "regexpids": [1, 2],
    "preservekeys": true
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32)",
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Network interfaces for discovery",
      "expressions": [
        {
          "expression": "^Software Loopback Interface",
          "expression_type": "4"
        },
        {
          "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
          "expression_type": "4"
        },
        {
          "expression": "^NULL[0-9.*]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ll]o[0-9.*]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ss]ystem$",
          "expression_type": "4"
        },
        {
          "expression": "^Nu[0-9.*]*$",
          "expression_type": "4"
        }
      ]
    }
  ]
}
```

```
    }  
  },  
  "id": 1  
}
```

ソース

CRegexp::get() in ui/include/classes/api/services/CRegexp.php.

regexp.update

説明

object regexp.update(object/array regularExpressions)

このメソッドで、既存のグローバル正規表現を更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新される正規表現のプロパティ

regexpid プロパティはオブジェクトごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準プロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
expressions	array	Expressions オプション

戻り値

(object) regexpid プロパティの下にある更新された正規表現の ID を含むオブジェクトを返します。

例

ファイルシステムディスカバリのグローバル正規表現更新

Request:

```
{  
  "jsonrpc": "2.0",  
  "method": "regexp.update",  
  "params": {  
    "regexpid": "1",  
    "name": "File systems for discovery",  
    "test_string": "",  
    "expressions": [  
      {  
        "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",  
        "expression_type": "3",  
        "exp_delimiter": ",",  
        "case_sensitive": "0"  
      },  
      {  
        "expression": "^(ntfs|fat32|fat16)$",  
        "expression_type": "3",  
        "exp_delimiter": ",",  
        "case_sensitive": "0"  
      }  
    ]  
  },  
  "auth": "700ca65537074ec963db7efabda78259",  
}
```

```
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "regexpids": [
      "1"
    ]
  },
  "id": 1
}
```

ソース

CRegexp::update() in ui/include/classes/api/services/CRegexp.php.

Report

このクラスは、スケジュールされたレポートを処理するように設計されています。

オブジェクトリファレンス:

- [Report](#)
- [Users](#)
- [User groups](#)

利用可能なメソッド:

- [report.create](#) - 新しいスケジュールされたレポートの作成
- [report.delete](#) - スケジュールされたレポートの削除
- [report.get](#) - スケジュールされたレポートの取得
- [report.update](#) - スケジュールされたレポートの更新

> Report object

次のオブジェクトは、reportAPI に直接関連しています。

Report

report オブジェクトは以下のプロパティを持ちます。

プロパティ	タイプ	説明
reportid	string	(読取専用) レポートの ID
userid (必須)	string	レポートを作成したユーザーの ID
name (必須)	string	ユニークなレポート名
dashboardid (必須)	string	レポートの基になっているダッシュボードの ID
period	integer	レポートが作成される期間

使用可能な値：
0 - (デフォルト) 前日
1 - 前週
2 - 前月
3 - 前年

プロパティ	タイプ	説明
cycle	integer	<p>期間の繰り返しスケジュール</p> <p>使用可能な値： 0 - (デフォルト) 毎日 1 - 毎週 2 - 毎月 3 - 毎年</p>
start_time	integer	<p>レポートの送信準備が行われる時刻 (秒単位)</p>
weekdays	integer	<p>デフォルト : 0 レポートを送信する曜日</p> <p>週次レポートにのみ必要</p> <p>曜日はバイナリ形式で保存され、各ビットは対応する曜日を表します。 たとえば 12 はバイナリで 1100 に等しく、レポートが毎週水曜日と木曜日に送信されることを意味します。</p>
active_since	string	<p>開始日</p> <p>使用可能な値： 空の文字列 - (デフォルト) 指定なし (0 として保存) YYYY-MM-DD 形式の特定日 (時刻は (00:00:00) として保存されます)</p>
active_till	string	<p>終了日</p> <p>使用可能な値： 空の文字列 - (デフォルト) 指定なし (0 として保存) YYYY-MM-DD 形式の特定日 (時刻は (23:59:59) として保存されます)</p>
subject	string	<p>レポートメッセージの件名</p>
message	string	<p>レポートメッセージのテキスト</p>
status	integer	<p>レポートが有効か無効か</p> <p>使用可能な値： 0 - 無効 1 - (デフォルト) 有効</p>
description	text	<p>レポートの説明</p>
state	integer	<p>(読取専用) レポートの状態</p> <p>使用可能な値： 0 - (デフォルト) レポートは未処理 1 - レポートが生成され、すべての受信者に正常送信済 2 - レポートの生成に失敗。"info" にはエラー情報が含まれています。 3 - レポートは生成されましたが、一部 (またはすべて) の受信者への送信に失敗。"info" にはエラー情報が含まれています。</p>
lastsent	timestamp	<p>(読取専用) 最後に正常送信されたレポートの Unix タイムスタンプ</p>
info	string	<p>(読取専用) エラーの説明または追加情報</p>

Users

users オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
userid (必須)	string	レポート送信先のユーザー ID
access_userid	string	レポートが生成されるユーザー ID
		0 - (デフォルト) 受信者ごとにレポートを生成

プロパティ	タイプ	説明
exclude	integer	ユーザーをメーリングリストから除外するか 可能な値： 0 - (デフォルト) 含める 1 - 除外する

User groups

user groups オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
usrgrp_id (必須)	string	レポート送信先のユーザーグループ ID
access_userid	string	レポートが生成されるユーザー ID。 0 - (デフォルト) 受信者ごとにレポートを生成

report.create

説明

`object report.create(object/array reports)`

このメソッドで、新しいスケジュールされたレポートを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するスケジュールされたレポート

このメソッドは標準のスケジュールされたレポートプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
users	object/array of objects	レポート送信先のユーザー
user_groups	object/array of objects	レポート送信先のユーザーグループ

戻り値

(object) `reportids` プロパティの下で作成されたスケジュールされたレポートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスケジュール済みレポートの順序と一致します。

例

スケジュールされたレポートの作成

2021-04-01 から 2021-08-31 までの毎週月曜日から金曜日の 12:00 に前週に準備した週次レポートを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "report.create",
  "params": {
    "userid": "1",
    "name": "Weekly report",
    "dashboardid": "1",
    "period": "1",
    "cycle": "1",
    "start_time": "43200",
    "weekdays": "31",
  }
}
```

```

    "active_since": "2021-04-01",
    "active_till": "2021-08-31",
    "subject": "Weekly report",
    "message": "Report accompanying text",
    "status": "1",
    "description": "Report description",
    "users": [
        {
            "userid": "1",
            "access_userid": "1",
            "exclude": "0"
        },
        {
            "userid": "2",
            "access_userid": "0",
            "exclude": "1"
        }
    ],
    "user_groups": [
        {
            "usrgrpid": "7",
            "access_userid": "0"
        }
    ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}

```

参照

- [Users](#)
- [User groups](#)

ソース

CReport::create() in ui/include/classes/api/services/CReport.php.

report.delete

説明

object report.delete(array reportids)

このメソッドで、スケジュールされたレポートを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するスケジュール済みレポートの ID

戻り値

(object) reportids プロパティの下で削除されたスケジュールされたレポートの ID を含むオブジェクトを返します。

例

複数のスケジュールされたレポートの削除

2 つのスケジュールされたレポートを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "report.delete",
  "params": [
    "1",
    "2"
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "reportids": [
      "1",
      "2"
    ]
  },
  "id": 1
}
```

ソース

CReport::delete() in ui/include/classes/api/services/CReport.php.

report.get

説明

integer/array report.get(object parameters)

このメソッドで、指定されたパラメーターに従ってスケジュールされたレポートを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
reportids	string/array	指定されたレポート ID を持つスケジュールされたレポートのみを返します
expired	boolean	true : 期限切れのスケジュール済みレポートのみ false : アクティブなスケジュール済みレポートのみ
selectUsers	query	レポートの送信先として構成されている <code>users</code> プロパティを返します
selectUserGroups	query	レポートの送信先として構成されている <code>user_groups</code> プロパティを返します。

パラメータ	タイプ	説明
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	可能な値は reportid,name,status です。 すべての get メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

レポートデータの取得

Request:

```
{
  "jsonrpc": "2.0",
  "method": "report.get",
  "params": [
    "output": "extend",
    "selectUsers": "extend",
    "selectUserGroups": "extend",
    "reportids": ["1", "2"]
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "reportid": "1",
      "userid": "1",
      "name": "Weekly report",
      "dashboardid": "1",
      "period": "1",
      "cycle": "1",
      "start_time": "43200",
      "weekdays": "31",
      "active_since": "2021-04-01",
      "active_till": "2021-08-31",
      "subject": "Weekly report",
      "message": "Report accompanying text",
      "status": "1",
      "description": "Report description",
      "state": "1",
      "lastsent": "1613563219",
    }
  ]
}
```

```

        "info": "",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            },
            {
                "userid": "2",
                "access_userid": "0",
                "exclude": "1"
            }
        ],
        "user_groups": [
            {
                "usrgrpid": "7",
                "access_userid": "0"
            }
        ]
    },
    {
        "reportid": "2",
        "userid": "1",
        "name": "Monthly report",
        "dashboardid": "2",
        "period": "2",
        "cycle": "2",
        "start_time": "0",
        "weekdays": "0",
        "active_since": "2021-05-01",
        "active_till": "",
        "subject": "Monthly report",
        "message": "Report accompanying text",
        "status": "1",
        "description": "",
        "state": "0",
        "lastsent": "0",
        "info": "",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            }
        ],
        "user_groups": []
    }
],
    "id": 1
}

```

参照

- [Users](#)
- [User groups](#)

ソース

CReport::get() in ui/include/classes/api/services/CReport.php.

report.update

説明

object report.update(object/array reports)

このメソッドで、既存のスケジュールされたレポートを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるスケジュールされたレポートプロパティ

reportid プロパティは、スケジュールされたレポートごとに定義する必要があります。すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のスケジュールされたレポートプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
users	object/array of objects	スケジュールされたレポートに割り当てられている現在のユーザーを置き換える ユーザー
user_groups	object/array of objects	スケジュールされたレポートに割り当てられている現在のユーザーグループを置き換える ユーザーグループ

戻り値

(object) reportids プロパティの下で更新されたスケジュールされたレポートの ID を含むオブジェクトを返します。

例

スケジュールされたレポートの無効化

Request:

```
{
  "jsonrpc": "2.0",
  "method": "report.update",
  "params": {
    "reportid": "1",
    "status": "0"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "reportids": [
      "1"
    ]
  },
  "id": 1
}
```

参照

- [Users](#)
- [User groups](#)

ソース

CReport::update() in ui/include/classes/api/services/CReport.php.

Role

このクラスは、ユーザーロールを操作するように設計されています。

オブジェクトリファレンス:

- [Role](#)
- [Role rules](#)
- [UI element](#)
- [Service](#)
- [Service tag](#)
- [Module](#)
- [Action](#)

利用可能なメソッド:

- [role.create](#) - 新しいユーザーロールの作成
- [role.delete](#) - ユーザーロールの削除
- [role.get](#) - ユーザーロールの取得
- [role.update](#) - ユーザーロールの更新

> Role object

次のオブジェクトは、roleAPI に直接関連しています。

Role

role オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
roleid	string	(読取専用) ロール ID
name (必須)	string	ロール名
type (必須)	integer	ユーザータイプ 使用可能な値: 1 - (デフォルト) ユーザー 2 - 管理者 3 - スーパー管理者
readonly	integer	(読取専用) ロールが読み取り専用かどうか 使用可能な値: 0 - (デフォルト) いいえ 1 - はい

Role rules

role rules オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
ui	array	UI element オブジェクトの配列
ui.default_access	integer	新しい UI エlementへのアクセスを有効にするかどうか。 使用可能な値 : 0 - 無効 1 - (デフォルト) 有効
services.read.mode	integer	サービスへの読み取り専用アクセス。 使用可能な値 : 0 - <code>services.read.list</code> で指定されるか、 <code>services.read.tag</code> プロパティで一致するサービスへの読み取り専用アクセス。 1 - (デフォルト) すべてのサービスへの読み取り専用アクセス。

プロパティ	タイプ	説明
services.read.list	array	<p>Service オブジェクトの配列。</p> <p>子サービスを含む指定されたサービスには、ユーザーロールへの読み取り専用アクセスが付与されます。読み取り専用アクセスは、サービスへの読み取り/書き込みアクセスをオーバーライドしません。</p> <p>services.read.mode が 0 に設定されている場合にのみ使用されます。</p>
services.read.tag	object	<p>Service tag オブジェクトの配列。</p> <p>タグに一致するサービス (子サービスを含む) には、ユーザーロールへの読み取り専用アクセスが付与されます。読み取り専用アクセスは、サービスへの読み取り/書き込みアクセスをオーバーライドしません。</p> <p>services.read.mode が 0 に設定されている場合にのみ使用されます。</p>
services.write.mode	integer	<p>サービスへの読み取り/書き込みアクセス。</p> <p>使用可能な値:</p> <p>0 - (デフォルト) サービスへの読み書きアクセス。 services.write.list で指定されるか、services.write.tag プロパティで一致します。</p> <p>1 - すべてのサービスへの読み取り/書き込みアクセス。</p>
services.write.list	array	<p>Service オブジェクトの配列。</p> <p>子サービスを含む指定されたサービスには、ユーザーロールへの読み取り/書き込みアクセスが付与されます。読み取り/書き込みアクセスは、サービスへの読み取り専用アクセスをオーバーライドします。</p> <p>services.write.mode が 0 に設定されている場合にのみ使用されません。</p>
services.write.tag	object	<p>Service tag オブジェクトの配列。</p> <p>タグに一致するサービス (子サービスを含む) には、ユーザーロールへの読み取り/書き込みアクセスが付与されます。読み取り/書き込みアクセスは、サービスへの読み取り専用アクセスをオーバーライドします。</p> <p>services.write.mode が 0 に設定されている場合にのみ使用されません。</p>
modules	array	module オブジェクトの配列
modules.default_access	integer	新しいモジュールへのアクセスを有効にするか
api.access	integer	<p>使用可能な値 :</p> <p>0 - 無効</p> <p>1 - (デフォルト) 有効</p> <p>API へのアクセスが有効か</p>
api.mode	integer	<p>使用可能な値 :</p> <p>0 - 無効</p> <p>1 - (デフォルト) 有効</p> <p>api プロパティにリストされた API メソッドを処理するためのモード</p>
api.actions	array	API メソッドの配列。
	array	action オブジェクトの配列。

プロパティ	タイプ	説明
actions.default_access	integer	新しいアクションへのアクセスを有効にするか 使用可能な値： 0 - 無効 1 - (デフォルト) 有効

UI element

UI element オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
name (必須)	string	UI エlement 名 使用可能な値： monitoring.dashboard - 監視データ → ダッシュボード; monitoring.problems - 監視データ → 障害; monitoring.hosts - 監視データ → ホスト; monitoring.latest_data - 監視データ → 最新データ; monitoring.maps - 監視データ → マップ; services.services - サービス → サービス; services.sla_report - サービス → SLA レポート; inventory.overview - インベントリ → 概要; inventory.hosts - インベントリ → ホスト; reports.availability_report - レポート → 稼働レポート; reports.top_triggers - レポート → 障害発生数上位 100 項目. 管理者およびスーパー管理者ユーザータイプのみ使用可能な値: monitoring.discovery - 監視データ → ディスカバリ; services.actions - サービス → サービスアクション; services.sla - サービス → SLA; reports.scheduled_reports - レポート → 定期レポート; reports.notifications - レポート → 通知レポート; configuration.host_groups - 設定 → ホストグループ; configuration.templates - 設定 → テンプレート; configuration.hosts - 設定 → ホスト; configuration.maintenance - 設定 → メンテナンス; configuration.actions - 設定 → アクション; configuration.discovery - 設定 → ディスカバリ. スーパー管理者ユーザータイプのみ使用可能な値: reports.system_info - レポート → システム情報; reports.audit - レポート → 監査; reports.action_log - レポート → アクションログ; configuration.event_correlation - 設定 → イベント相関関係; administration.general - 管理 → 一般設定; administration.proxies - 管理 → プロキシ; administration.authentication - 管理 → 認証; administration.user_groups - 管理 → ユーザーグループ; administration.user_roles - 管理 → ユーザーロール; administration.users - 管理 → ユーザー; administration.media_types - 管理 → メディアタイプ; administration.scripts - 管理 → スクリプト; administration.queue - 管理 → キュー
status	integer	UI 要素へのアクセスが有効か 使用可能な値： 0 - 無効 1 - (デフォルト) 有効

プロパティ	タイプ	説明
serviceid (必須)	string	サービス ID

Service tag

プロパティ	タイプ	説明
tag (必須)	string	タグ名 空の文字列が指定されている場合、サービスタグはサービスマッチングに使用されません。
value	string	タグ値 値がないか空の文字列の場合、サービスの照合にはタグ名のみが使用されます。

Module

module オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
moduleid (必須)	string	モジュール ID
status	integer	モジュールへのアクセスが有効か 使用可能な値： 0 - 無効 1 - (デフォルト) 有効

Action

action オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
name (必須)	string	アクション名 任意のタイプのユーザーに可能な値： edit_dashboards - ダッシュボードの作成と編集 edit_maps - マップの作成と編集 add_problem_comments - 障害のコメント追加 change_severity - 障害の深刻度を変更 acknowledge_problems - 障害を確認 close_problems - 障害をクローズします execute_scripts - スクリプトを実行 manage_api_tokens - API トークンを管理
status	integer	Admin および Superadmin ユーザータイプのユーザーにのみ可能な値： edit_maintenance - メンテナンスの作成および編集 manage_scheduled_reports - レポートのスケジュール管理 アクションを実行するためのアクセスが有効か 使用可能な値： 0 - 無効 1 - (デフォルト) 有効

role.create

説明

object role.create(object/array roles)

このメソッドで、新しいロールを作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するロール

このメソッドは標準のロールプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
rules	array	ロール用に作成されるロールルール

戻り値

(object) roleids プロパティの下に作成されたロールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたロールの順序と一致します。

例

ロールの作成

タイプが"ユーザー"で、2つのUI要素へのアクセスを拒否するロールを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "role.create",
  "params": {
    "name": "Operator",
    "type": "1",
    "rules": {
      "ui": [
        {
          "name": "monitoring.hosts",
          "status": "0"
        },
        {
          "name": "monitoring.maps",
          "status": "0"
        }
      ]
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "roleids": [
      "5"
    ]
  },
  "id": 1
}
```

参照

- [Role rules](#)
- [UI element](#)
- [Module](#)
- [Action](#)

ソース

CRole::create() in ui/include/classes/api/services/CRole.php.

role.delete

説明

object role.delete(array roleids)

このメソッドでロールを削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するロールの ID

戻り値

(object) roleids プロパティの下にある削除されたロールの ID を含むオブジェクトを返します。

例

複数のユーザーロールの削除

2つのユーザーロールを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "role.delete",
  "params": [
    "4",
    "5"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "roleids": [
      "4",
      "5"
    ]
  },
  "id": 1
}
```

ソース

CRole::delete() in ui/include/classes/api/services/CRole.php.

role.get

説明

integer/array role.get(object parameters)

このメソッドで、指定されたパラメーターに従ってロールを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします

パラメータ	タイプ	説明
roleids	string/array	指定された ID を持つロールのみを返します
selectRules	query	<code>rules</code> プロパティでロールルールを返します
selectUsers	query	この役割が割り当てられているユーザーを選択します
sortfield	string/array	指定されたプロパティで結果を並べ替えます
countOutput	boolean	利用可能な値は <code>roleid,name</code> です すべての <code>get</code> メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ロールデータの取得

"スーパー管理者ロール" のロールデータとそのアクセスルールを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "role.get",
  "params": {
    "output": "extend",
    "selectRules": "extend",
    "roleids": "3"
  },
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "roleid": "3",
      "name": "Super admin role",
    }
  ]
}
```

```
"type": "3",
"readonly": "1",
"rules": {
  "ui": [
    {
      "name": "inventory.hosts",
      "status": "1"
    },
    {
      "name": "inventory.overview",
      "status": "1"
    },
    {
      "name": "monitoring.dashboard",
      "status": "1"
    },
    {
      "name": "monitoring.hosts",
      "status": "1"
    },
    {
      "name": "monitoring.latest_data",
      "status": "1"
    },
    {
      "name": "monitoring.maps",
      "status": "1"
    },
    {
      "name": "monitoring.problems",
      "status": "1"
    },
    {
      "name": "reports.availability_report",
      "status": "1"
    },
    {
      "name": "reports.top_triggers",
      "status": "1"
    },
    {
      "name": "services.services",
      "status": "1"
    },
    {
      "name": "services.sla_report",
      "status": "1"
    },
    {
      "name": "configuration.actions",
      "status": "1"
    },
    {
      "name": "configuration.discovery",
      "status": "1"
    },
    {
      "name": "configuration.host_groups",
      "status": "1"
    },
    {
      "name": "configuration.hosts",
```

```

    "status": "1"
  },
  {
    "name": "configuration.maintenance",
    "status": "1"
  },
  {
    "name": "configuration.templates",
    "status": "1"
  },
  {
    "name": "monitoring.discovery",
    "status": "1"
  },
  {
    "name": "reports.notifications",
    "status": "1"
  },
  {
    "name": "reports.scheduled_reports",
    "status": "1"
  },
  {
    "name": "services.actions",
    "status": "1"
  },
  {
    "name": "services.sla",
    "status": "1"
  },
  {
    "name": "administration.authentication",
    "status": "1"
  },
  {
    "name": "administration.general",
    "status": "1"
  },
  {
    "name": "administration.media_types",
    "status": "1"
  },
  {
    "name": "administration.proxies",
    "status": "1"
  },
  {
    "name": "administration.queue",
    "status": "1"
  },
  {
    "name": "administration.scripts",
    "status": "1"
  },
  {
    "name": "administration.user_groups",
    "status": "1"
  },
  {
    "name": "administration.user_roles",
    "status": "1"
  },
}

```



```

    {
      "name": "administration.users",
      "status": "1"
    },
    {
      "name": "configuration.event_correlation",
      "status": "1"
    },
    {
      "name": "reports.action_log",
      "status": "1"
    },
    {
      "name": "reports.audit",
      "status": "1"
    },
    {
      "name": "reports.system_info",
      "status": "1"
    }
  ],
  "ui.default_access": "1",
  "services.read.mode": "1",
  "services.read.list": [],
  "services.read.tag": {
    "tag": "",
    "value": ""
  },
  "services.write.mode": "1",
  "services.write.list": [],
  "services.write.tag": {
    "tag": "",
    "value": ""
  },
  "modules": [],
  "modules.default_access": "1",
  "api.access": "1",
  "api.mode": "0",
  "api": [],
  "actions": [
    {
      "name": "edit_dashboards",
      "status": "1"
    },
    {
      "name": "edit_maps",
      "status": "1"
    },
    {
      "name": "acknowledge_problems",
      "status": "1"
    },
    {
      "name": "close_problems",
      "status": "1"
    },
    {
      "name": "change_severity",
      "status": "1"
    },
    {
      "name": "add_problem_comments",

```

```

        "status": "1"
    },
    {
        "name": "execute_scripts",
        "status": "1"
    },
    {
        "name": "manage_api_tokens",
        "status": "1"
    },
    {
        "name": "edit_maintenance",
        "status": "1"
    },
    {
        "name": "manage_scheduled_reports",
        "status": "1"
    },
    {
        "name": "manage_sla",
        "status": "1"
    }
    ],
    "actions.default_access": "1"
}
}
],
"id": 1
}

```

参照

- [Role rules](#)
- [User](#)

ソース

CRole::get() in ui/include/classes/api/services/CRole.php.

role.update

説明

object role.update(object/array roles)

このメソッドで、既存の役割を更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるロールプロパティ

roleid プロパティはルールごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のロールプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
rules	array	ルールに割り当てられている現在のアクセスルールを置き換えるアクセスルール

戻り値

(object) roleids プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

スクリプトを実行する機能を無効にする

ID"5" でロールを更新し、スクリプトを実行する機能を無効にします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "role.update",
  "params": [
    {
      "roleid": "5",
      "rules": {
        "actions": [
          {
            "name": "execute_scripts",
            "status": "0"
          }
        ]
      }
    }
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "roleids": [
      "5"
    ]
  },
  "id": 1
}
```

API へのアクセスを制限する

ID が"5" のロールを更新し、"create","update","delete" メソッドの呼び出しを拒否します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "role.update",
  "params": [
    {
      "roleid": "5",
      "rules": {
        "api.access": "1",
        "api.mode": "0",
        "api": ["*.create", "/*.update", "/*.delete"]
      }
    }
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
```

```

    "result": {
      "roleids": [
        "5"
      ]
    },
    "id": 1
  }
}

```

ソース

CRole::update() in ui/include/classes/api/services/CRole.php.

Script

このクラスは、スクリプトを操作するように設計されています。

オブジェクトリファレンス:

- [Script](#)
- [Webhook parameters](#)
- [Debug](#)
- [Log entry](#)

利用可能なメソッド:

- [script.create](#) - 新しいスクリプトの作成
- [script.delete](#) - スクリプトの削除
- [script.execute](#) - スクリプトの実行
- [script.get](#) - スクリプトの取得
- [script.getscriptsbyhosts](#) - ホストのスクリプトを取得
- [script.update](#) - スクリプトの更新

> Script object

次のオブジェクトは、scriptAPIに直接関連しています。

Script

script オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
scriptid	string	(読取専用) スクリプトの ID
name (必須)	string	スクリプト名
type (必須)	integer	スクリプトタイプ 使用可能な値： 0 - Script 1 - IPMI 2 - SSH 3 - Telnet 5 - (デフォルト) Webhook
command (必須)	string	実行コマンド
scope	integer	スクリプトのスコープ 使用可能な値： 1 - デフォルトアクション操作 2 - 手動ホストアクション 4 - 手動イベントアクション

プロパティ	タイプ	説明
execute_on	integer	スクリプトの実行場所 type が 0(スクリプト) の場合に使用されます 使用可能な値： 0 - Zabbix エージェントで実行 1 - Zabbix サーバーで実行 2 - (デフォルト) Zabbix サーバー (プロキシ) で実行
menu_path	string	ホストまたはイベントをクリックしたときにフロントエンドのナビゲーションのようなメニューを形成する基となる、スラッシュで区切られたフォルダー
authtype	integer	scope が 2 または 4 の場合に使用されます SSH スクリプトタイプに使用される認証方法 type が 2 の場合に使用されます 使用可能な値： 0 - パスワード 1 - 公開鍵
username	string	認証に使用されるユーザー名
password	string	type が 2 または 3 の場合に必要です パスワード認証付きの SSH スクリプトと Telnet スクリプトに使用されるパスワード
publickey	string	type が 2 で authtype が 0 または type が 3 の場合に使用され ます 公開鍵認証を使用する SSH スクリプトに使用される公開鍵ファイルの名前
privatekey	string	type が 2 で authtype が 1 の場合に必要 公開鍵認証を使用する SSH スクリプトに使用される秘密鍵ファイルの名前
port	string	type が 2 で authtype が 1 の場合に必要 SSH および Telnet スクリプトに使用されるポート番号
groupid	string	タイプが 2 または 3 の場合に必要 スクリプトを実行できるホストグループの ID。0 に設定するとスクリプトはすべてのホストグループで使用できます。
usrgrp_id	string	デフォルト：0 スクリプトの実行を許可されるユーザーグループの ID。0 に設定するとスクリプトはすべてのユーザーグループで使用できます。 scope が 2 または 4 の場合に使用されます。
host_access	integer	デフォルト：0 スクリプトの実行に必要なホストの権限 scope が 2 または 4 の場合に使用されます 使用可能な値： 2 - (デフォルト) 読み取り 3 - 書き込み
confirmation	string	確認ポップアップテキスト。Zabbix フロントエンドからスクリプトを実行しようとする、ポップアップが表示されます。 scope が 2 または 4 の場合に使用されます
timeout	string	Webhook スクリプトの実行タイムアウト (秒単位)。時間の接尾辞がサポートされています。30s,1m type が 5 の場合は必須 使用可能な値： 1 - 60s
parameters	array	デフォルト値： 30s webhook 入力パラメーターの配列 type が 5 の場合に使用されます
description	string	スクリプトの説明

Webhook parameters

Webhook スクリプトが呼び出されたときに渡されるパラメーターには、次のプロパティがあります。

プロパティ	タイプ	説明
name (必須)	string	パラメータ名
value	string	パラメータ値。マクロをサポート

デバッグ

実行された webhook スクリプトのデバッグ情報。デバッグオブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
logs	array	ログエントリの配列
ms	string	スクリプトの実行時間 (ミリ秒単位)

Log entry

log entry オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
level	integer	ログレベル
ms	string	スクリプトが実行されてからログエントリが追加されるまでの経過時間 (ミリ秒単位)
message	string	ログメッセージ

script.create

説明

object script.create(object/array scripts)

このメソッドで新しいスクリプトを作成できます

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受け入れます

戻り値

(object) scriptids プロパティの下で作成されたスクリプトの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスクリプトの順序と一致します。

例

Webhook スクリプトを作成する

HTTP リクエストを外部サービスに送信する Webhook スクリプトを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "script.create",
  "params": {
    "name": "Webhook script",
    "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co",
    "type": 5,
    "timeout": "40s",
```

```

    "parameters": [
      {
        "name": "token",
        "value": "${WEBHOOK.TOKEN}"
      },
      {
        "name": "host",
        "value": "${HOST.HOST}"
      },
      {
        "name": "v",
        "value": "2.2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "scriptids": [
      "3"
    ]
  },
  "id": 1
}

```

SSH スクリプトを作成する

コンテキストメニューを備えた公開鍵認証を使用してホスト上で実行できる SSH スクリプトを作成します

Request:

```

{
  "jsonrpc": "2.0",
  "method": "script.create",
  "params": {
    "name": "SSH script",
    "command": "my script command",
    "type": 2,
    "username": "John",
    "publickey": "pub.key",
    "privatekey": "priv.key",
    "password": "secret",
    "port": "12345",
    "scope": 2,
    "menu_path": "All scripts/SSH",
    "usrgrpid": "7",
    "groupid": "4"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "scriptids": [
      "5"
    ]
  }
}

```

```
    },
    "id": 1
  }
}
```

カスタムスクリプトを作成する

サーバーを再起動するカスタムスクリプトを作成します。スクリプトはホストへの書き込みアクセスを必要とし、フロントエンドで実行する前に設定メッセージを表示します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "script.create",
  "params": {
    "name": "Reboot server",
    "command": "reboot server 1",
    "confirmation": "Are you sure you would like to reboot the server?",
    "scope": 2,
    "type": 0
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "scriptids": [
      "4"
    ]
  },
  "id": 1
}
```

ソース

CScript::create() in ui/include/classes/api/services/CScript.php.

script.delete

説明

object script.delete(array scriptIds)

このメソッドでスクリプトを削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するスクリプトの ID

戻り値

(object) scriptids プロパティの下で削除されたスクリプトの ID を含むオブジェクトを返します。

例

複数のスクリプトを削除

2つのスクリプト削除

Request:


```
{
  "jsonrpc": "2.0",
  "method": "script.delete",
  "params": [
    "3",
    "4"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "scriptids": [
      "3",
      "4"
    ]
  },
  "id": 1
}
```

ソース

CScript::delete() in ui/include/classes/api/services/CScript.php.

script.execute

説明

object script.execute(object parameters)

このメソッドで、ホストまたはイベントでスクリプトを実行できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 実行するスクリプトの ID と、ホスト ID またはイベント ID のいずれかを含むパラメーター。

パラメータ	タイプ	説明
scriptid (必須)	string	実行するスクリプトの ID
hostid	string	スクリプトを実行するホストの ID
eventid	string	スクリプトを実行するイベントの ID

戻り値

(object) スクリプトの実行結果を返します。

プロパティ	タイプ	説明
response	string	スクリプトが正常に実行されたか。
value	string	可能な値 - success スクリプトの出力
debug	object	Webhook スクリプトが実行される場合、 デバッグオブジェクト が含まれます。他のスクリプトタイプの場合、空のオブジェクトが含まれます。

例

Webhook スクリプトを実行する

HTTP リクエストを外部サービスに送信する Webhook スクリプトを実行します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "script.execute",
  "params": {
    "scriptid": "4",
    "hostid": "30079"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "response": "success",
    "value": "{\"status\":\"sent\",\"timestamp\":\"1611235391\"}",
    "debug": {
      "logs": [
        {
          "level": 3,
          "ms": 480,
          "message": "[Webhook Script] HTTP status: 200."
        }
      ],
      "ms": 495
    }
  },
  "id": 1
}
```

カスタムスクリプト実行

ホストで“ping” スクリプトを実行します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "script.execute",
  "params": {
    "scriptid": "1",
    "hostid": "30079"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "response": "success",
    "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_req=1 tt",
    "debug": []
  },
  "id": 1
}
```

ソース

CScript::execute() in ui/include/classes/api/services/CScript.php.

script.get

説明

integer/array script.get(object parameters)

このメソッドで、指定されたパラメーターに従ってスクリプトを取得できます。

Note:

すべての get メソッドに共通するこれらのパラメーターについては[リファレンス解説](#)ページで詳しく説明されています。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします

パラメータ	タイプ	説明
groupids	string/array	指定されたホストグループで実行できるスクリプトのみを返します
hostids	string/array	指定されたホストで実行できるスクリプトのみを返します
scriptids	string/array	指定された ID のスクリプトのみを返します
usrgrpids	string/array	指定されたユーザーグループのユーザーが実行できるスクリプトのみを返します
selectGroups	query	スクリプトを実行できるホストグループを含む groups プロパティを返します
selectHosts	query	スクリプトを実行できるホストを含む [hosts][hosts](/manual/api/reference/host/object) プロパティを返します
selectActions	query	スクリプトが関連付けられているアクションを含む actions プロパティを返します
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	使用可能な値は scriptid と name です すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得されたオブジェクトの数

例

すべてのスクリプトを取得する

構成済みのすべてのスクリプトを取得します

Request:

```

{
  "jsonrpc": "2.0",
  "method": "script.get",
  "params": {
    "output": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "scriptid": "1",
      "name": "Ping",
      "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
      "host_access": "2",
      "usrgrpuid": "0",
      "groupid": "0",
      "description": "",
      "confirmation": "",
      "type": "0",
      "execute_on": "1",
      "timeout": "30s",
      "parameters": []
    },
    {
      "scriptid": "2",
      "name": "Traceroute",
      "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
      "host_access": "2",
      "usrgrpuid": "0",
      "groupid": "0",
      "description": "",
      "confirmation": "",
      "type": "0",
      "execute_on": "1",
      "timeout": "30s",
      "parameters": []
    },
    {
      "scriptid": "3",
      "name": "Detect operating system",
      "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
      "host_access": "2",
      "usrgrpuid": "7",
      "groupid": "0",
      "description": "",
      "confirmation": "",
      "type": "0",
      "execute_on": "1",
      "timeout": "30s",
      "parameters": []
    },
    {
      "scriptid": "4",
      "name": "Webhook",
      "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
      "host_access": "2",
      "usrgrpuid": "7",

```

```

        "groupid": "0",
        "description": "",
        "confirmation": "",
        "type": "5",
        "execute_on": "1",
        "timeout": "30s",
        "parameters": [
            {
                "name": "token",
                "value": "${WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "${HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}

```

参照

- [Host](#)
- [Host group](#)

ソース

CScript::get() in ui/include/classes/api/services/CScript.php.

script.getscriptsbyhosts

説明

object script.getscriptsbyhosts(array hostIds)

このメソッドで、特定のホストで使用可能なスクリプトを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(string/array) スクリプトを返すホストの ID

戻り値

(object) プロパティとしてホスト ID を持ち、値として利用可能なスクリプトの配列を持つオブジェクトを返します。

Note:

このメソッドは confirmation テキストのマクロを自動的に展開します

例

ホスト ID でスクリプトを取得する

ホスト "30079" および "30073" で使用可能なすべてのスクリプトを取得します

Request:

```

{
  "jsonrpc": "2.0",
  "method": "script.getscriptsbyhosts",

```

```
"params": [
  "30079",
  "30073"
],
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "30079": [
      {
        "scriptid": "3",
        "name": "Detect operating system",
        "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
        "host_access": "2",
        "usrgrpuid": "7",
        "groupid": "0",
        "description": "",
        "confirmation": "",
        "type": "0",
        "execute_on": "1",
        "hostid": "10001"
      },
      {
        "scriptid": "1",
        "name": "Ping",
        "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
        "host_access": "2",
        "usrgrpuid": "0",
        "groupid": "0",
        "description": "",
        "confirmation": "",
        "type": "0",
        "execute_on": "1",
        "hostid": "10001"
      },
      {
        "scriptid": "2",
        "name": "Traceroute",
        "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
        "host_access": "2",
        "usrgrpuid": "0",
        "groupid": "0",
        "description": "",
        "confirmation": "",
        "type": "0",
        "execute_on": "1",
        "hostid": "10001"
      }
    ],
    "30073": [
      {
        "scriptid": "3",
        "name": "Detect operating system",
        "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
        "host_access": "2",
        "usrgrpuid": "7",
        "groupid": "0",
        "description": "",

```

```

        "confirmation": "",
        "type": "0",
        "execute_on": "1",
        "hostid": "10001"
    },
    {
        "scriptid": "1",
        "name": "Ping",
        "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
        "host_access": "2",
        "usrgrpuid": "0",
        "groupid": "0",
        "description": "",
        "confirmation": "",
        "type": "0",
        "execute_on": "1",
        "hostid": "10001"
    },
    {
        "scriptid": "2",
        "name": "Traceroute",
        "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
        "host_access": "2",
        "usrgrpuid": "0",
        "groupid": "0",
        "description": "",
        "confirmation": "",
        "type": "0",
        "execute_on": "1",
        "hostid": "10001"
    }
    ]
},
"id": 1
}

```

ソース

CScript::getScriptsByHosts() in ui/include/classes/api/services/CScript.php.

script.update

説明

object script.update(object/array scripts)

このメソッドでは、既存のスクリプトを更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) **スクリプトのプロパティ**が更新されます。

scriptid プロパティはスクリプトごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。例外は type プロパティが 5 (Webhook) からその他に変更された場合です。その場合は parameters プロパティはクリーンアップされます。

戻り値

(object) scriptids プロパティの下で更新されたスクリプトの ID を含むオブジェクトを返します。

例

スクリプトコマンドの変更

スクリプトのコマンドを"/bin/ping -c 10 {HOST.CONN} 2>&1" に変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "script.update",
  "params": {
    "scriptid": "1",
    "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "scriptids": [
      "1"
    ]
  },
  "id": 1
}
```

ソース

CScript::update() in ui/include/classes/api/services/CScript.php.

Service

このクラスは、IT インフラストラクチャ/ビジネスサービスと連携するように設計されています。

オブジェクトリファレンス:

- [Service](#)
- [Status rule](#)
- [Service tag](#)
- [Service alarm](#)
- [Problem tag](#)

利用可能なメソッド:

- [service.create](#) - 新しいサービスの作成
- [service.delete](#) - サービスの削除
- [service.get](#) - サービスの取得
- [service.update](#) - サービスの更新

> Service object

次のオブジェクトは、`serviceAPI` に直接関連しています。

Service

service オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
serviceid	string	(読取専用) サービス ID

プロパティ	タイプ	説明
algorithm (必須)	integer	ステータス計算ルール。子サービスが存在する場合にのみ適用されます。 使用可能な値： 0 - ステータスを OK に設定 1 - すべての子サービスに問題がある場合、最も高い深刻度に設定 2 - 子サービスで最も高い深刻度に設定
name (必須)	string	サービス名
sortorder (r 必須)	integer	並べ替えに使用されるサービスの位置。 使用可能な値：0~999
weight	integer	サービスの重み 使用可能な値：0-1000000
propagation_rule	integer	デフォルト：0 ステータス伝播ルール。propagation_value と一緒に設定する必要があります。 使用可能な値： 0 - (デフォルト) サービスステータスをそのまま伝播します - 変更なし 1 - 伝播されたステータスを指定された値だけ増やします propagation_value(1 から 5 の深刻度) 2 - 指定された propagation_value(1 から 5 の深刻度) だけ伝播ステータスを減らします 3 - このサービスを無視します - ステータスは親サービスに伝播されません 4 - 指定された propagation_value の値でサービスステータスを設定します
propagation_value	integer	ステータス伝播値。propagation_rule と一緒に設定する必要があります。 propagation_rule の値が 0 および 3 の時に使用可能な値：0 propagation_rule の値が 1 および 2 の時に使用可能な値：1-5 propagation_rule の値が 4 の時に使用可能な値： -1 - OK 0 - 未分類 1 - 情報 2 - 警告; 3 - 軽度の障害 4 - 重度の障害 5 - 致命的な障害
status	integer	(読取専用) サービスが OK か障害状態か サービスが障害状態にある場合 status は次のいずれかに等しくなります。 -最も重大な障害の深刻度 -障害のある状態の子サービスの一番重大なステータス サービスが OK 状態の場合、status は-1 になります。
description	string	サービスの説明
uuid	string	ユニバーサル一意識別子。更新の場合、このフィールドは読み取り専用です。
created_at	integer	サービスが作成されたときの Unix タイムスタンプ。

プロパティ	タイプ	説明
readonly	boolean	(読取専用) サービスへのアクセス。 使用可能な値： 0 - 読取/書込 1 - 読取専用

Status rule

status rule オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
type (必須)	integer	(New status) ステータスを設定するための条件 使用可能な値： 0 - 少なくとも (N) 個の子サービスが (Status) ステータス以上の場合 1 - 少なくとも (N%) の子サービスが (Status) ステータス以上の場合 2 - (N) 未満の子サービスのステータスが (Status) 以下の場合 3 - 子サービスの (N%) 未満が (Status) ステータス以下の場合 4 - (Status) ステータス以上の子サービスの重みが少なくとも (W) である場合 5 - (Status) ステータス以上の子サービスの重みが少なくとも (N%) である場合 6 - (Status) ステータス以下の子サービスの重みが (W) 未満の場合 7 - (Status) ステータス以下の子サービスの重みが (N%) 未満の場合 読み替え： - (N) と (W) は limit_value です - (Status) は limit_status です - (New status) は new_status です
limit_value (必須)	integer	制限値
limit_status (必須)	integer	制限ステータス 使用可能な値： - N-W: 1-100000; - N%: 1-100.
new_status (必須)	integer	新しいステータス値 使用可能な値： 0 - 未分類 1 - 情報 2 - 警告; 3 - 軽度の障害 4 - 重度の障害 5 - 致命的な障害

Service tag

service tag オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	サービスタグ名
value	string	サービスタグ値

Service alarm

Note:

Zabbix API を介して Service alarm を直接作成、更新、削除することはできません。

service alarm オブジェクトは、サービスの状態変化を表します。以下の特性があります。

プロパティ	タイプ	説明
clock	timestamp	サービス状態の変更が発生した時刻
value	integer	サービスのステータス 使用可能な値のリストについては サービスステータスプロパティ を参照してください。

Problem tag

Problem tag を使用すると、サービスを障害イベントにリンクできます。Problem tag オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	障害タグ名
operator	integer	マッピング条件演算子 使用可能な値： 0 - (デフォルト) 等しい 2 - like
value	string	障害タグ値

service.create

説明

`object service.create(object/array services)`

このメソッドで、新しいサービスを作成できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するサービス

標準サービスプロパティに加えて、このメソッドは次のパラメーターを受け入れます。

パラメータ	タイプ	説明
children	array	サービスにリンクされる子サービス 子には <code>serviceid</code> プロパティが定義されている必要があります
parents	array	サービスにリンクされる親サービス 親には <code>serviceid</code> プロパティが定義されている必要があります
tags	array	サービス用に作成されるサービスタグ
problem_tags	array	サービス用に作成される障害タグ
status_rules	サービス用に作成されるステータスルール	

戻り値

(object) `serviceids` プロパティの下で作成されたサービスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたサービスの順序と一致します。

例

サービスの作成

少なくとも 1 つの子で障害があった場合に、障害状態に切り替わるサービスを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.create",
  "params": {
    "name": "Server 1",
    "algorithm": 1,
    "sortorder": 1
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "serviceids": [
      "5"
    ]
  },
  "id": 1
}
```

ソース

CService::create() in ui/include/classes/api/services/CService.php.

service.delete

説明

object `service.delete`(array `serviceIds`)

このメソッドで、サービスを削除できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するサービスの ID

戻り値

(object) `serviceids` プロパティの下にある削除されたサービスの ID を含むオブジェクトを返します。

例

複数サービスの削除

2 つのサービスを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.delete",
  "params": [
```

```

        "4",
        "5"
    ],
    "auth": "3a57200802b24cda67c4e4010b50c065",
    "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "serviceids": [
      "4",
      "5"
    ]
  },
  "id": 1
}

```

ソース

CService::delete() in ui/include/classes/api/services/CService.php.

service.get

説明

integer/array service.get(object parameters)

このメソッドで、指定されたパラメーターに従ってサービスを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
serviceids	string/array	指定された ID のサービスのみを返します
parentids	string/array	指定された親サービスにリンクされているサービスのみを返します
deep_parentids	flag	すべての直接および間接的な子サービスを返します。parentids と一緒に使用されます
childids	string/array	指定された子サービスにリンクされているサービスのみを返します
evaltype	integer	タグ検索のルール

利用可能な値：

0 - (デフォルト) And/Or

2 - Or

パラメータ	タイプ	説明
tags	object/array of objects	<p>指定されたタグを持つサービスのみを返します。タグによる完全一致、演算子の値に応じたタグの大文字と小文字を区別する検索、大文字と小文字を区別しない検索。</p> <p>フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]</p> <p>空配列の場合はすべてのサービスを返します</p> <p>使用可能な演算子値:</p> <ul style="list-style-type: none"> 0 - (デフォルト) 含む
1 - 等しい 2 - 含まない 3 - 等しくない 4 - 存在する 5 - 存在しない
problem_tags	object/array of objects	<p>指定された障害タグを持つサービスのみを返します。タグによる完全一致、演算子の値に応じた大文字と小文字を区別する検索、大文字と小文字を区別しない検索。</p> <p>フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...].</p> <p>空配列の場合はすべてのサービスを返します</p> <p>使用可能な演算子値:</p> <ul style="list-style-type: none"> 0 - (デフォルト) 含む 1 - 等しい 2 - 含まない 3 - 等しくない 4 - 存在する 5 - 存在しない
without_problem_tags	flag	障害タグのないサービスのみを返します
slays	string/array	特定の SLA にリンクされているサービスのみを返します。
selectChildren	query	子サービスで children プロパティを返します。
selectParents	query	count をサポートします。 親サービスで parents プロパティを返します。
selectTags	query	count をサポートします。 tags プロパティをサービスタグとともに返します。
selectProblemEvents	query	<p>count をサポートします。 障害のあるイベントオブジェクトの配列を含む problem_events プロパティを返します。</p> <p>障害のあるイベントオブジェクトには次のプロパティがあります。</p> <ul style="list-style-type: none"> eventid - (string) イベント ID severity - (string) 現在のイベントの深刻度 name - (string) 解決されたイベント名
selectProblemTags	query	count をサポートします 障害のあるタグを含む problem_tags プロパティを返します。
selectStatusRules	query	count をサポートします。 ステータスルールを含む status_rules プロパティを返します。
		count をサポートします。

パラメータ	タイプ	説明
selectStatusTimeline	object/array of objects	<p>指定された期間のサービス状態の変更を含む status_timeline プロパティを返します。</p> <p>フォーマット [{"period_from": "<period_from>", "period_to": "<period_to>"}, ...] - period_from は指定期間の開始日（その日を含む整数のタイムスタンプ）であり、period_to は指定期間の終了日（その日を含まない整数のタイムスタンプ）です。</p> <p>指定された期間内の状態変化について start_value プロパティと alarms 配列を含むエントリの配列を返します。</p>
sortfield	string/array	指定されたプロパティで結果を並べ替えます
countOutput	boolean	指定可能な値は serviceid,name,status,sortorder,created_at です すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのサービスを取得する

すべてのサービスとその依存関係に関するすべてのデータを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.get",
  "params": {
    "output": "extend",
    "selectChildren": "extend",
    "selectParents": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "serviceid": "1",
      "name": "My Service - 0001",

```

```

        "status": "-1",
        "algorithm": "2",
        "sortorder": "0",
        "weight": "0",
        "propagation_rule": "0",
        "propagation_value": "0",
        "description": "My Service Description 0001.",
        "uuid": "dfa4daeaea754e3a95c04d6029182681",
        "created_at": "946684800",
        "readonly": false,
        "parents": [],
        "children": []
    },
    {
        "serviceid": "2",
        "name": "My Service - 0002",
        "status": "-1",
        "algorithm": "2",
        "sortorder": "0",
        "weight": "0",
        "propagation_rule": "0",
        "propagation_value": "0",
        "description": "My Service Description 0002.",
        "uuid": "20ea0d85212841219130abeaca28c065",
        "created_at": "946684800",
        "readonly": false,
        "parents": [],
        "children": []
    }
],
    "id": 1
}

```

ソース

CService::get() in ui/include/classes/api/services/CService.php.

service.update

説明

object service.update(object/array services)

このメソッドで、既存のサービスを更新できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるサービスプロパティ

serviceid プロパティはサービスごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準サービスプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
children	array	現在のサービスの子サービスを置き換えるサービス。
parents	array	子には serviceid プロパティを定義する必要があります。 現在のサービスの親サービスを置き換えるサービス。 親には serviceid プロパティを定義する必要があります。

パラメータ	タイプ	説明
tags	array	現在のサービスタグを置き換えるサービスタグ
problem_tags	array	現在の障害タグを置き換える障害タグ
status_rules	array	現在のステータスルールを置き換えるステータスルール

戻り値

(object) serviceids プロパティの下で更新されたサービスの ID を含むオブジェクトを返します。

例

親サービスを設定する

ID"3" のサービスを ID"5" のサービスの親にします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.update",
  "params": {
    "serviceid": "5",
    "parents": [
      {
        "serviceid": "3"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "serviceids": [
      "5"
    ]
  },
  "id": 1
}
```

スケジュールされたダウンタイムの追加

ID"4" のサービスに毎週月曜日 22:00 から火曜日 10:00 でスケジュールされたダウンタイムを追加します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.update",
  "params": {
    "serviceid": "4",
    "times": [
      {
        "type": "1",
        "ts_from": "165600",
        "ts_to": "201600"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "serviceids": [
      "4"
    ]
  },
  "id": 1
}
```

ソース

CService::update() in ui/include/classes/api/services/CService.php.

Settings

このクラスは、一般的な管理設定で機能するように設計されています。

オブジェクトリファレンス:

- [Settings](#)

利用可能なメソッド:

- [settings.get](#) - 設定を取得します
- [settings.update](#) - 設定を更新します

> Settings object

次のオブジェクトは、[settingsAPI](#) に直接関連しています。

Settings

settings オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
default_lang	string	デフォルトの言語
default_timezone	string	デフォルト: en_GB デフォルトのタイムゾーン デフォルト: system - システムデフォルト
default_theme	string	サポートされているタイムゾーンの完全なリストについては PHP のドキュメント を参照してください。 デフォルトのテーマ
search_limit	integer	使用可能な値: blue-theme - (デフォルト) 青 dark-theme - ダーク hc-light - ハイコントラストライト hc-dark - ハイコントラストダーク 検索/フィルターの項目の上限値
max_overview_table_size	integer	デフォルト: 1000 データの概要とトリガーの概要のダッシュボードウィジェットの列と行の最大数。
max_in_table	integer	デフォルト: 50 テーブルセル内に表示する要素の最大数
		デフォルト: 50

プロパティ	タイプ	説明
server_check_interval	integer	Zabbix サーバー停止時にワーニングを表示 使用可能な値: 0 - ワーニングを表示しない 10 - (デフォルト) ワーニングを表示する
work_period	string	ワーキングタイム デフォルト: 1-5,09:00-18:00.
show_technical_errors	integer	技術的エラーを表示。スーパー管理者以外のユーザーおよびデバッグモードが有効になっているユーザーグループに属していないユーザーに技術的なエラー (PHP/SQL) を表示します。 使用可能な値: 0 - (デフォルト) エラーを表示しない 1 - エラーを表示する
history_period	string	ヒストリの最大表示期間。最新データ、Web、およびデータ概要ダッシュボードウィジェットにヒストリデータを表示する最大期間。接尾辞付きの秒と時間の単位を受け入れます。
period_default	string	デフォルト: 24h フィルタリングするデフォルト期間。月と年をサポートする接尾辞付きの秒と時間の単位を受け入れます (30s,1m,2h,1d,1M,1y)
max_period	string	デフォルト: 1h. タイムセレクターの最大期間。月と年をサポートする接尾辞付きの秒と時間の単位を受け入れます (30s,1m,2h,1d,1M,1y)
severity_color_0	string	デフォルト: 2y 深刻度"未分類"の色を 16 進数のカラーコードで指定
severity_color_1	string	デフォルト: 97AAB3. 深刻度"情報"の色を 16 進数のカラーコードで指定
severity_color_2	string	デフォルト: 7499FF. 深刻度"警告"の色を 16 進数のカラーコードで指定
severity_color_3	string	デフォルト: FFC859. 深刻度"軽度の障害"の色を 16 進数のカラーコードで指定
severity_color_4	string	デフォルト: FFA059. 深刻度"重度の障害"の色を 16 進数のカラーコードで指定
severity_color_5	string	デフォルト: E97659. 深刻度"致命的な障害"の色を 16 進数のカラーコードで指定
severity_name_0	string	デフォルト: E45959. 深刻度"未分類"の名前
severity_name_1	string	Default: "未分類" 深刻度"情報"の名前
severity_name_2	string	Default: "情報" 深刻度"警告"の名前
severity_name_3	string	Default: "警告" 深刻度"軽度の障害"の名前
severity_name_4	string	Default: "軽度の障害" 深刻度"重度の障害"の名前 Default: "重度の障害"

プロパティ	タイプ	説明
severity_name_5	string	深刻度”致命的な障害”の名前
custom_color	integer	Default: ”致命的な障害” イベントのステータスの文字色を変更。カスタムイベントステータスカラーの使用有無。 使用可能な値: 0 - (デフォルト) 使用しない 1 - 使用する
ok_period	string	正常イベントの表示期間。接尾辞付きの秒と時間の単位を受け入れます。
blink_period	string	デフォルト: 5m ステータスが変化したイベントの点滅期間。接尾辞付きの秒と時間の単位を受け入れます。
problem_unack_color	string	デフォルト: 2m 未確認の障害イベントの色を 16 進数のカラーコードで指定
problem_ack_color	string	デフォルト: CC0000. 確認済障害イベントの色を 16 進数のカラーコードで指定
ok_unack_color	string	デフォルト: CC0000. 未確認の解決済イベントの色を 16 進数のカラーコードで指定
ok_ack_color	string	デフォルト: 009900. 確認済の解決済イベントの色を 16 進数のカラーコードで指定
problem_unack_style	integer	デフォルト: 009900. 未確認の障害イベントを点滅させるか 使用可能な値 0 - させない 1 - (デフォルト) させる
problem_ack_style	integer	確認済障害イベントを点滅させるか 使用可能な値 0 - させない 1 - (デフォルト) させる
ok_unack_style	integer	未確認の解決済イベントを点滅させるか 使用可能な値 0 - させない 1 - (デフォルト) させる
ok_ack_style	integer	確認済の解決済を点滅させるか 使用可能な値 0 - させない 1 - (デフォルト) させる
url	string	フロントエンドの URL
discovery_groupid	integer	ディスカバリで見つけられたホストのグループ ID
default_inventory_mode	integer	デフォルトのホストインベントリモード 使用可能な値: -1 - (デフォルト) 無効 0 - マニュアル 1 - 自動
alert_usrgrp_id	integer	データベース停止メッセージの送信先グループ ID。空白に設定するとアラームメッセージは送信されません。

プロパティ	タイプ	説明
snmptrap_logging	integer	マッチしない SNMP トラップをログに記録するか 使用可能な値 0 - 記録しない 1 - (デフォルト) 記録する
login_attempts	integer	ログインの試行
login_block	string	デフォルト: 5 ログインブロックの間隔。接尾辞付きの秒と時間の単位を受け入れます。
validate_uri_schemes	integer	デフォルト: 30s. URI スキーマの検証
uri_valid_schemes	string	使用可能な値: 0 - 検証しない 1 - (デフォルト) 検証する 有効な URI スキーマ
x_frame_options	string	デフォルト: http,https,ftp,file,mailto,tel,ssh. X-Frame-Options HTTP ヘッダ
iframe_sandboxing_enabled	integer	デフォルト: SAMEORIGIN. iframe サンドボックスを使用する
iframe_sandboxing_exceptions	string	使用可能な値: 0 - 使用しない 1 - (デフォルト) 使用する Iframe サンドボックスの例外
connect_timeout	string	Zabbix server との接続タイムアウト
socket_timeout	string	デフォルト: 3s. Zabbix server とのネットワークタイムアウト
media_type_test_timeout	string	デフォルト: 3s. メディアテストのためのネットワークタイムアウト
item_test_timeout	string	デフォルト: 65s. アイテムテストのためのネットワークタイムアウト
script_timeout	string	デフォルト: 60s. スクリプト実行のためのネットワークタイムアウト
report_test_timeout	string	デフォルト: 60s. 定期レポートテストのためのネットワークタイムアウト
auditlog_enabled	integer	デフォルト: 60s. 監査ログの有効化
ha_failover_delay	string	使用可能な値: 0 - 無効 1 - (デフォルト) 有効 フェイルオーバー遅延を秒単位で指定
		デフォルト: 1m.

プロパティ	タイプ	説明
geomaps_tile_provider	string	ジオマップタイルプロバイダー 使用可能な値： OpenStreetMap.Mapnik - (デフォルト) OpenStreetMap Mapnik; OpenTopoMap - OpenTopoMap; Stamen.TonerLite - Stamen Toner Lite; Stamen.Terrain - Stamen Terrain; USGS.USTopo - USGS US Topo; USGS.USImagery - USGS US Imagery
geomaps_tile_url	string	空の文字列を許可。フロントエンドのその他に相当。その場合は geomaps_tile_url, geomaps_max_zoom, geomaps_attribution にカスタム値を指定します。
geomaps_max_zoom	integer	geomaps_tile_provider が空の文字列 (その他) に設定されている場合のジオマップの最大ズームレベル。最大ズームは 0 から 30 の範囲でなければなりません。
geomaps_attribution	string	geomaps_tile_provider が空の文字列 (その他) に設定されている場合のジオマップ属性テキスト

settings.get

説明

object settings.get(object parameters)

このメソッドで、指定されたパラメーターに従って設定オブジェクトを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは [ユーザーの役割](#) を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは 1 つのパラメーターのみをサポートします

パラメータ	タイプ	説明
output	query	すべての get メソッドに共通するこのパラメーターについては リファレンス解説 ページで詳しく説明されています。

戻り値

(object) 設定オブジェクトを返します

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "settings.get",
  "params": {
    "output": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "default_theme": "blue-theme",
    "search_limit": "1000",
    "max_in_table": "50",
    "server_check_interval": "10",
    "work_period": "1-5,09:00-18:00",
    "show_technical_errors": "0",
    "history_period": "24h",
    "period_default": "1h",
    "max_period": "2y",
    "severity_color_0": "97AAB3",
    "severity_color_1": "7499FF",
    "severity_color_2": "FFC859",
    "severity_color_3": "FFA059",
    "severity_color_4": "E97659",
    "severity_color_5": "E45959",
    "severity_name_0": "Not classified",
    "severity_name_1": "Information",
    "severity_name_2": "Warning",
    "severity_name_3": "Average",
    "severity_name_4": "High",
    "severity_name_5": "Disaster",
    "custom_color": "0",
    "ok_period": "5m",
    "blink_period": "2m",
    "problem_unack_color": "CC0000",
    "problem_ack_color": "CC0000",
    "ok_unack_color": "009900",
    "ok_ack_color": "009900",
    "problem_unack_style": "1",
    "problem_ack_style": "1",
    "ok_unack_style": "1",
    "ok_ack_style": "1",
    "discovery_groupid": "5",
    "default_inventory_mode": "-1",
    "alert_usrgrpid": "7",
    "snmptrap_logging": "1",
    "default_lang": "en_GB",
    "default_timezone": "system",
    "login_attempts": "5",
    "login_block": "30s",
    "validate_uri_schemes": "1",
    "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
    "x_frame_options": "SAMEORIGIN",
    "iframe_sandboxing_enabled": "1",
    "iframe_sandboxing_exceptions": "",
    "max_overview_table_size": "50",
    "connect_timeout": "3s",
    "socket_timeout": "3s",
    "media_type_test_timeout": "65s",
    "script_timeout": "60s",
    "item_test_timeout": "60s",
    "url": "",
    "report_test_timeout": "60s",
    "auditlog_enabled": "1",
    "ha_failover_delay": "1m",
    "geomaps_tile_provider": "OpenStreetMap.Mapnik",
    "geomaps_tile_url": "",
    "geomaps_max_zoom": "0",
    "geomaps_attribution": ""
  }
}

```

```
    },  
    "id": 1  
}
```

ソース

CSettings::get() in ui/include/classes/api/services/CSettings.php.

settings.update

説明

object settings.update(object settings)

このメソッドで、既存の共通設定を更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 更新する設定プロパティ

戻り値

(array) 更新されたパラメータの名前を含む配列を返します

例

Request:

```
{  
  "jsonrpc": "2.0",  
  "method": "settings.update",  
  "params": {  
    "login_attempts": "1",  
    "login_block": "1m"  
  },  
  "auth": "038e1d7b1735c6a5436ee9eae095879e",  
  "id": 1  
}
```

Response:

```
{  
  "jsonrpc": "2.0",  
  "result": [  
    "login_attempts",  
    "login_block"  
  ],  
  "id": 1  
}
```

ソース

CSettings::update() in ui/include/classes/api/services/CSettings.php.

SLA

このクラスは、IT インフラストラクチャとビジネスサービスのパフォーマンスを推定するために使用される SLA（サービスレベルアグリーメント）オブジェクトと連携するように設計されています。

オブジェクトリファレンス:

- [SLA](#)
- [SLA schedule](#)

- SLA excluded downtime
- SLA service tag

利用可能なメソッド:

- `sla.create` - 新しい SLAs の作成
- `sla.delete` - SLAs の削除
- `sla.get` - SLAs の取得
- `sla.getsli` - サービスレベルインジケータ (SLI) として可用性情報を取得する
- `sla.update` - SLAs の更新

> SLA object

The following objects are directly related to the `sla` (Service Level Agreement) API.

SLA

SLA オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
<code>slaid</code>	string	(読取専用) SLA の ID
<code>name</code> (必須)	string	SLA 名
<code>period</code> (必須)	integer	SLA のレポート期間 使用可能な値： 0 - 毎日 1 - 毎週 2 - 毎月 3 - 四半期ごと 4 - 毎年
<code>slo</code> (必須)	float	パーセントで表された最小許容サービスレベル目標。サービスレベルインジケータ (SLI) が低くなると、SLA は障害がある/稼働率が低い状態であると見なされます。
<code>effective_date</code>	integer	可能な値：0~100 (最大 4 桁) SLA の発効日
<code>timezone</code> (必須)	文字列	使用可能な値：UTC での日付タイムスタンプ レポートのタイムゾーン。例： Europe/London,UTC サポートされているタイムゾーンの完全なリストについては、 PHP ドキュメント を参照してください。
<code>status</code>	integer	SLA のステータス 使用可能な値： 0 - (デフォルト) SLA 無効 1 - SLA 有効
<code>description</code>	string	SLA の説明

SLA Schedule

SLA Schedule オブジェクトは、接続されたサービスが正常に機能するようにスケジュールされている期間を定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
period_from (必須)	integer	繰り返される毎週の期間の開始時刻 (指定日含む)
period_to (必須)	integer	使用可能な値 : (日曜日から数えた) 秒数 繰り返される毎週の期間の終了時間 (指定日含まない)
		可能な値 : (日曜日から数えた) 秒数

SLA excluded downtime

excluded downtime オブジェクトは、接続されたサービスが SLI に影響を与えることなく正常に機能しないようにスケジュールされている期間を定義します。例えば、計画的なメンテナンスを受けるなどです。

プロパティ	タイプ	説明
name (必須)	string	除外されたダウンタイムの名前
period_from (必須)	integer	除外されたダウンタイムの開始時刻 (指定日含む) 可能な値 : タイムスタンプ
period_to (必須)	integer	除外されたダウンタイムの終了時間 (指定日含まず) 可能な値 : タイムスタンプ

SLA service tag

SLA service tag オブジェクトは、SLA の計算に含めるサービスをリンクします。以下のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	SLA サービスタグ名
operator	integer	SLA サービスタグオペレーター
		使用可能な値 : 0 - (デフォルト) イコール 2 - like
value	string	SLA サービスタグ値

sla.create

説明

object sla.create(object/array SLAs)

このメソッドで、新しい SLA オブジェクトを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成する SLA オブジェクト

このメソッドは標準の SLA プロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
service_tags (必須)	array	作成される SLA 用のサービスタグ。 少なくとも 1 つのサービスタグを指定する必要があります。
schedule	array	作成される SLA 用のスケジュール。 空のパラメーターを指定すると、24 時間年中無休のスケジュールとして解釈されます。 Default: 24 時間年中無休

パラメータ	タイプ	説明
excluded_downtimes	array	作成される SLA から除外されたダウンタイム

戻り値

(object) slaid のプロパティの下で作成された SLA の ID を含むオブジェクトを返します。返される ID の順序は、渡された SLA の順序と一致します。

例

SLA の作成

次の SLA エントリを作成するように指示します。* SQL エンジン関連サービスの稼働時間を追跡 * 土曜日の最後の時間を除くすべての平日のカスタムスケジュール * 2022 年の最終発効日 * 7 月 4 日の深夜から始まる 1 時間 15 分の計画されたダウンタイム * SLA ウィークリーレポートの計算をオンに * 最小許容 SLO を 99.9995% に

Request:

```
{
  "jsonrpc": "2.0",
  "method": "sla.create",
  "params": [
    {
      "name": "Database Uptime",
      "slo": "99.9995",
      "period": "1",
      "timezone": "America/Toronto",
      "description": "Provide excellent uptime for main database engines.",
      "effective_date": 1672444800,
      "status": 1,
      "schedule": [
        {
          "period_from": 0,
          "period_to": 601200
        }
      ],
      "service_tags": [
        {
          "tag": "Database",
          "operator": "0",
          "value": "MySQL"
        },
        {
          "tag": "Database",
          "operator": "0",
          "value": "PostgreSQL"
        }
      ],
      "excluded_downtimes": [
        {
          "name": "Software version upgrade rollout",
          "period_from": "1648760400",
          "period_to": "1648764900"
        }
      ]
    }
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
```

```
    "result": {
      "slaid": [
        "5"
      ]
    },
    "id": 1
  }
}
```

ソース

CSla::create() in ui/include/classes/api/services/CSla.php.

sla.delete

説明

object sla.delete(array slaid)

このメソッドで、SLA エントリを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除する SLA の ID

戻り値

(object) slaid プロパティの下にある削除された SLA の ID を含むオブジェクトを返します。

例

複数の SLA を削除

2つの SLA エントリを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "sla.delete",
  "params": [
    "4",
    "5"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "slaid": [
      "4",
      "5"
    ]
  },
  "id": 1
}
```

ソース

CSla::delete() in ui/include/classes/api/services/CSla.php.

sla.get

説明

integer/array sla.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従って SLA オブジェクトを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
slaid	string/array	指定された ID を持つ SLA のみを返します
serviceids	string/array	特定のサービスに一致する SLA のみを返します
selectSchedule	query	SLA スケジュールを含む schedule プロパティを返します。
selectExcludedDowntimes	query	count をサポートします。 SLA が除外されたダウンタイムを含む excluded_downtimes プロパティを返します。
selectServiceTags	query	count をサポートします。 SLA サービスタグを含む service_tags プロパティを返します。 count をサポートします。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	可能な値は slaid,name,period、slo,effective_date,timezone,status,description です すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

すべての SLA を取得する

すべての SLA とそのプロパティに関するすべてのデータを取得します

Request:

```

{
  "jsonrpc": "2.0",
  "method": "sla.get",
  "params": {
    "output": "extend",
    "selectSchedule": ["period_from", "period_to"],
    "selectExcludedDowntimes": ["name", "period_from", "period_to"],
    "selectServiceTags": ["tag", "operator", "value"],
    "preservekeys": true
  },
  "auth": "85dd04b94cbfad794616eb923be13c71",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "slaid": "1",
      "name": "Database Uptime",
      "period": "1",
      "slo": "99.9995",
      "effective_date": "1672444800",
      "timezone": "America/Toronto",
      "status": "1",
      "description": "Provide excellent uptime for main SQL database engines.",
      "service_tags": [
        {
          "tag": "Database",
          "operator": "0",
          "value": "MySQL"
        },
        {
          "tag": "Database",
          "operator": "0",
          "value": "PostgreSQL"
        }
      ],
      "schedule": [
        {
          "period_from": "0",
          "period_to": "601200"
        }
      ],
      "excluded_downtimes": [
        {
          "name": "Software version upgrade rollout",
          "period_from": "1648760400",
          "period_to": "1648764900"
        }
      ]
    }
  },
  "id": 1
}

```

ソース

CSla::get() in ui/include/classes/api/services/CSla.php.

sla.getsli

説明

`object sla.getsli(object parameters)`

このメソッドで、サービスレベルインジケータ (SLI) データを計算できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) SLI を計算するための、SLA ID、レポート期間、オプションでサービスの ID を含むパラメーター

パラメータ	タイプ	説明
slaid (必須)	string	可用性情報を返すサービスの ID
period_from	integer	SLI 報告開始日 (指定日含む)
period_to	integer	使用可能な値: タイムスタンプ SLI 報告終了日 (指定日含まない)
periods	array	使用可能な値: タイムスタンプ 報告すべき望ましい期間数 使用可能な値: 1-100
serviceids	string/array	SLI を返すサービスの ID

期間の分割

以下は、パラメーターの組み合わせに基づいて返される period slices の配置を示しています。

パラメータ	説明		
period_from	period_to	periods	
-	-	-	過去 20 期間 (現在の期間を含む) で、SLA の発効日に基づいて利用可能な最初の期間を過ぎていない
-	-	指定	periods パラメータで指定された最後の期間。
-	指定	-	指定された日付より前の最後の 20 期間で、SLA の発効日に基づく最初の利用可能な期間を過ぎていない
-	指定	指定	指定された日付より前の periods パラメータで指定された最後の期間。
指定	-	-	最初の 20 期間 (現在の期間を含む) であるが、現在の期間よりも前。
指定	-	指定	指定された日付から始まる periods パラメータで指定された最初の期間。
指定	指定	-	指定された日付範囲内の期間のうち、100 を超えず、SLA の発効日に基づいて最初に利用可能な期間を
指定	指定	指定	指定された日付範囲内の期間かつ、指定された期間数を超えず、SLA の発効日に基づいた最初の利用可

戻り値

(object) 計算結果を返します。

プロパティ	タイプ	説明
periods	array	報告された期間のリスト 報告された各期間は、次で構成されるオブジェクトとして表されます。 - period_from - 報告された期間の開始日 (タイムスタンプ) - period_to - 終了報告された期間の日付 (タイムスタンプ) 期間は period_from フィールドの昇順で並べ替えられます。
serviceids	array	レポートされた期間のサービス ID のリスト。
sli	array	serviceids パラメータが <code>sla.getsli</code> メソッドに渡された場合でも、リストの並べ替え順序は定義されていません。 (** 2 次元配列で) 報告された各期間とサービスの SLI データ。 periods プロパティのインデックスは、 sli プロパティの 1 次元目として使用されます。 serviceids プロパティのインデックスは、 sli プロパティの二次元目 ** として使用されます。

SLI data

SLI data 報告された期間ごとに返され、サービスは次のもので構成されます。

プロパティ	タイプ	説明
uptime	integer	スケジュールされた稼働時間中にサービスが <code>_OK_</code> 状態だった時間から、除外されたダウンタイムを差し引いたもの。
downtime	integer	スケジュールされた稼働時間中にサービスが <code>_not OK_</code> 状態だった時間から、除外されたダウンタイムを差し引いたもの。
sli	float	稼働時間とダウンタイムに基づく SLI (総稼働時間のパーセンテージ)
error_budget	integer	SLI および SLO に基づくエラーバジェット (秒単位)
excluded_downtime	arrays	このレポート期間で除外されたダウンタイムの配列。

各オブジェクトには、次のパラメータが含まれます。

- name - 除外されたダウンタイムの名前
- period_from - 開始日時 (指定日含む)
- period_to - 除外されたダウンタイムの終了日時 (指定日含まず)

除外されたダウンタイムは、period_from フィールドの昇順で並べ替えられます。

例

SLI の計算

2021 年 11 月 1 日から 3 期間、ID が "5" の SLA にリンクされている ID が "50,60,70" のサービスの SLI を取得します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "sla.getsli",
  "params": {
    "slaid": "5",
    "serviceids": [
      50,
      60,
      70
    ],
    "periods": 3,
    "period_from": "1635724800"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "periods": [
      {
        "period_from": 1635724800,
        "period_to": 1638316800
      },
      {
        "period_from": 1638316800,
        "period_to": 1640995200
      },
      {
        "period_from": 1640995200,
        "period_to": 1643673600
      }
    ]
  },
  "serviceids": [
```



```

50,
60,
70
],
"sli": [
  [
    {
      "uptime": 1186212,
      "downtime": 0,
      "sli": 100,
      "error_budget": 0,
      "excluded_downtimes": [
        {
          "name": "Excluded Downtime - 1",
          "period_from": 1637836212,
          "period_to": 1638316800
        }
      ]
    },
    {
      "uptime": 1186212,
      "downtime": 0,
      "sli": 100,
      "error_budget": 0,
      "excluded_downtimes": [
        {
          "name": "Excluded Downtime - 1",
          "period_from": 1637836212,
          "period_to": 1638316800
        }
      ]
    },
    {
      "uptime": 1186212,
      "downtime": 0,
      "sli": 100,
      "error_budget": 0,
      "excluded_downtimes": [
        {
          "name": "Excluded Downtime - 1",
          "period_from": 1637836212,
          "period_to": 1638316800
        }
      ]
    }
  ],
  [
    {
      "uptime": 1147548,
      "downtime": 0,
      "sli": 100,
      "error_budget": 0,
      "excluded_downtimes": [
        {
          "name": "Excluded Downtime - 1",
          "period_from": 1638439200,
          "period_to": 1639109652
        }
      ]
    },
    {
      "uptime": 1147548,

```

```

        "downtime": 0,
        "sli": 100,
        "error_budget": 0,
        "excluded_downtimes": [
            {
                "name": "Excluded Downtime - 1",
                "period_from": 1638439200,
                "period_to": 1639109652
            }
        ]
    },
    {
        "uptime": 1147548,
        "downtime": 0,
        "sli": 100,
        "error_budget": 0,
        "excluded_downtimes": [
            {
                "name": "Excluded Downtime - 1",
                "period_from": 1638439200,
                "period_to": 1639109652
            }
        ]
    }
],
[
    {
        "uptime": 1674000,
        "downtime": 0,
        "sli": 100,
        "error_budget": 0,
        "excluded_downtimes": []
    },
    {
        "uptime": 1674000,
        "downtime": 0,
        "sli": 100,
        "error_budget": 0,
        "excluded_downtimes": []
    },
    {
        "uptime": 1674000,
        "downtime": 0,
        "sli": 100,
        "error_budget": 0,
        "excluded_downtimes": []
    }
]
],
    "id": 1
}

```

ソース

CSla::getSli() in ui/include/classes/api/services/CSla.php

sla.update

説明

object sla.update(object/array slaid)

このメソッドで、既存の SLA エントリを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新される SLA プロパティ

slaid プロパティは SLA ごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準の SLA プロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
service_tags	array	SLA service tags を使用して、現在の SLA サービスタグを置き換えます。
schedule	array	少なくとも 1 つのサービスタグを指定する必要があります。 SLA schedule を使用して、現在のパラメーターを置き換えます。
excluded_downtimes	array	パラメーターを空にすると、24 時間年中無休のスケジュールとして設定されます SLA excluded downtimes を使用して、現在のダウンタイムを置き換えます。

戻り値

(object) slaid プロパティの下にある更新された SLA の ID を含むオブジェクトを返します。

例

サービスタグの更新

ID"5" の SLA を、スケジュールを変更したりダウンタイムを除外したりせずに、NoSQL 関連サービスの月間隔で計算するようにします。SLO を 95% に設定します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "sla.update",
  "params": [
    {
      "slaid": "5",
      "name": "NoSQL Database engines",
      "slo": "95",
      "period": 2,
      "service_tags": [
        {
          "tag": "Database",
          "operator": "0",
          "value": "Redis"
        },
        {
          "tag": "Database",
          "operator": "0",
          "value": "MongoDB"
        }
      ]
    }
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "slaid": [
      "5"
    ]
  },
  "id": 1
}
```

SLA のスケジュールの変更

ID"5" の SLA を 24 時間年中無休のスケジュールに切り替えます。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.update",
  "params": {
    "slaid": "5",
    "schedule": []
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "slaid": [
      "5"
    ]
  },
  "id": 1
}
```

SLA の除外されたダウンタイムの変更

ID"5" の SLA に対して 7 月 4 日に計画された既存のソフトウェアアップグレードを維持しながら、2022 年 4 月 6 日に計画された 4 時間の RAM アップグレードダウンタイムを追加します（新たに定義する必要があります）。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "service.update",
  "params": {
    "slaid": "5",
    "excluded_downtimes": [
      {
        "name": "Software version upgrade rollout",
        "period_from": "1648760400",
        "period_to": "1648764900"
      },
      {
        "name": "RAM upgrade",
        "period_from": "1649192400",
        "period_to": "1649206800"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "slaid": [
      "5"
    ]
  },
  "id": 1
}
```

ソース

CSla::update() in ui/include/classes/api/services/CSla.php.

Task

このクラスは、タスク（構成の再ロードなしで項目をチェックしたり、低レベルのディスカバリルールをチェックしたりするなど）で機能するように設計されています。

オブジェクトリファレンス:

- [Task](#)
- ['Check now' request object](#)
- ['Diagnostic information' request object](#)
- [Statistic request object](#)
- [Statistic result object](#)

利用可能なメソッド:

- [task.create](#) - 新しいタスクの作成
- [task.get](#) - タスクの取得

> Task object

次のオブジェクトは task API に直接関連しています。

タスクオブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
taskid	string	(読取専用) タスクの ID
type (必須)	integer	タスクのタイプ 使用可能な値: 1 - 診断情報 6 - 今すぐ確認
status	integer	(読取専用) タスクのステータス 使用可能な値: 1 - 新しいタスク 2 - 進行中のタスク; 3 - 完了タスク 4 - 期限切れタスク。
clock	timestamp	(読取専用) タスクが作成された時刻
ttl	integer	(読取専用) タスクが期限切れになるまでの秒数
proxy_hostid	string	診断情報の統計が収集されるプロキシの ID。 type が今すぐ確認では無視されます。
request (必須)	object	タスク種別に応じたタスク依頼オブジェクト: 診断情報タスクオブジェクトの 詳細 今すぐ確認タスクオブジェクトの 詳細

プロパティ	タイプ	説明
result	object	(読取専用) 診断情報タスクの結果オブジェクト。結果がまだ準備できていない場合は、NULL が含まれる場合があります。結果オブジェクトについては以下で詳しく説明します

'Check now' request object

'Check now' task request オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
itemid	string	アイテムの ID とローレベルディスクバリエーション

'Diagnostic information' request object

diagnostic information task request オブジェクトには次のプロパティがあります。すべてのタイプのプロパティの統計リクエストオブジェクトについては以下で詳しく説明します。

プロパティ	タイプ	説明
historycache	object	履歴キャッシュ統計リクエスト。サーバーとプロキシで利用可能。
valuecache	object	アイテムキャッシュ統計リクエスト。サーバーで利用可能。
preprocessing	object	マネージャ統計リクエストの事前処理。サーバーとプロキシで利用可能。
alerting	object	アラートマネージャの統計要求。サーバーで利用可能。
lld	object	LLD マネージャー統計要求。サーバーで利用可能。

Statistic request object

Statistic request オブジェクトは、サーバー/プロキシの内部プロセスに関して収集する必要がある情報の種類を定義するために使用されます。次のプロパティがあります。

プロパティ	タイプ	説明
stats	query	返される統計オブジェクトのプロパティ。診断情報統計の各タイプで使用可能なフィールドのリストについては以下で詳しく説明します。
top	object	デフォルト: extend 利用可能なすべての統計フィールドを返します。返された統計値を並べ替えて制限するオブジェクト。診断情報統計の各タイプで使用可能なフィールドのリストについては以下で詳しく説明します。 例: { "source.alerts": 10 }

診断情報要求の各タイプで使用可能な統計フィールドのリスト

次の統計フィールドは、診断情報リクエストプロパティのタイプごとにリクエストできます。

診断タイプ	使用可能なフィールド	説明
historycache	items	キャッシュされたアイテムの数
	values	キャッシュされた値の数
	memory	共有メモリ統計 (空き容量、使用済みチャンク数、空きチャンク数、空きチャンクの最大サイズ)
	memory.data	ヒストリー・データ・キャッシュの共有メモリー統計
	memory.index	ヒストリー索引キャッシュ共有メモリー統計
valuecache	items	キャッシュされたアイテムの数
	values	キャッシュされた値の数
	memory	共有メモリ統計 (空き容量、使用済みチャンク数、空きチャンク数、空きチャンクの最大サイズ)
	mode	値キャッシュモード
preprocessing	values	キューに入れられた値の数
	preproc.values	事前処理ステップでキューに入れられた値の数
alerting	alerts	キューに入れられたアラート数

診断タイプ	使用可能なフィールド	説明
lld	rules	キューに入れられたルールの数
	values	キューに入れられた値の数

診断情報要求の各タイプで使用できる並べ替えフィールドのリスト

次の統計フィールドを使用して、要求された情報をソートおよび制限できます。

診断タイプ	使用可能なフィールド	タイプ
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Statistic result object

Statistic result オブジェクトは、タスクオブジェクトの `result` フィールドで取得されます。

プロパティ	タイプ	説明
status	integer	(読取専用) タスク結果のステータス
data	string/object	使用可能な値: -1 - タスク実行中にエラー 0 - タスク実行結果作成 特定の診断情報タスクの統計要求オブジェクトに応じた結果。タスクの実行中にエラーが発生した場合は、エラーメッセージ文字列が含まれません。

task.create

説明

`object task.create(object/array tasks)`

このメソッドでは新しいタスクを作成できます (診断データの収集、アイテムのチェック、設定の再読み込みをしないローレベルディスカバリルールなど)

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については [ユーザーの役割](#) を参照してください。

パラメータ

(object/array) タスクの作成

このメソッドは、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
type (必須)	integer	タスクタイプ 可能な値: 1 - 診断情報 6 - 即時確認
request (必須)	object	タスクタイプに応じたタスクリクエストオブジェクト。リクエストオブジェクトの正しい形式は タスクオブジェクト セクションで説明されています

パラメータ	タイプ	説明
proxy_hostid	integer	'diagnostic information' タスクがデータを収集するプロキシ "即時確認" タスクでは無視されます。

"即時確認" タスクは次の種類のアイテム/ディスクバリルルールに対してのみ作成できることに注意してください。

- Zabbix agent
- SNMPv1/v2/v3 agent
- Simple check
- Internal check
- External check
- Database monitor
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- Calculated check
- JMX agent

戻り値

(object) taskids プロパティの下で作成されたタスクの ID を含むオブジェクトを返します。アイテムおよびローレベルディスクバリルルールごとに 1 つのタスクが作成されます。返される ID の順序は、渡された itemids の順序と一致します。

例

タスクの作成

2 つのアイテムのタスク'即時確認'を作成します。1 つはアイテムで、もう 1 つはローレベルディスクバリルルールです。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "task.create",
  "params": [
    {
      "type": 6,
      "request": {
        "itemid": "10092"
      }
    },
    {
      "type": "6",
      "request": {
        "itemid": "10093"
      }
    }
  ],
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "taskids": [
      "1",
      "2"
    ]
  },
  "id": 1
}
```


タスク'診断情報'を作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "task.create",
  "params": [
    {
      "type": 1,
      "request": {
        "alerting": {
          "stats": [
            "alerts"
          ],
          "top": {
            "media.alerts": 10
          }
        },
        "lld": {
          "stats": "extend",
          "top": {
            "values": 5
          }
        }
      },
      "proxy_hostid": 0
    }
  ],
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 2
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "taskids": [
      "3"
    ]
  },
  "id": 2
}
```

参照

- [Task](#)
- ['Check now' request object](#)
- ['Diagnostic information' request object](#)
- [Statistic request object](#)

ソース

CTask::create() in ui/include/classes/api/services/CTask.php.

task.get

説明

integer/array task.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従ってタスクを取得できます。メソッドは'診断情報' タスクに関する詳細のみを返します。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
taskids	string/array	指定された ID を持つタスクのみを返します
output	query	すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています
preservekeys	boolean	

戻り値

(integer/array) オブジェクトの配列を返します

例

ID でタスクを取得

ID"1" のタスクに関するすべてのデータを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "task.get",
  "params": {
    "output": "extend",
    "taskids": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "taskid": "1",
      "type": "7",
      "status": "3",
      "clock": "1601039076",
      "ttl": "3600",
      "proxy_hostid": null,
      "request": {
        "alerting": {
          "stats": [
            "alerts"
          ],
          "top": {
            "media.alerts": 10
          }
        },
        "lld": {
          "stats": "extend",
          "top": {
            "values": 5
          }
        }
      }
    }
  ],
}
```

```

    "result": {
      "data": {
        "alerting": {
          "alerts": 0,
          "top": {
            "media.alerts": []
          },
          "time": 0.000663
        },
        "lld": {
          "rules": 0,
          "values": 0,
          "top": {
            "values": []
          },
          "time": 0.000442
        }
      },
      "status": "0"
    }
  ],
  "id": 1
}

```

参照

- [Task](#)
- [Statistic result object](#)

ソース

CTask::get() in ui/include/classes/api/services/CTask.php.

Template

このクラスは、テンプレートを操作するように設計されています。

オブジェクトリファレンス:

- [Template](#)

利用可能なメソッド:

- [template.create](#) - 新しいテンプレートの作成
- [template.delete](#) - テンプレートの削除
- [template.get](#) - テンプレートの取得
- [template.massadd](#) - テンプレートへ関連オブジェクトの追加
- [template.massremove](#) - テンプレートから関連オブジェクトの削除
- [template.massupdate](#) - テンプレートの関連オブジェクト置換または削除
- [template.update](#) - テンプレートの更新

> Template object

次のオブジェクトは `template` API に直接関連しています。

Template

template オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
<code>templateid</code>	string	(読取専用) テンプレートの ID
host (必須)	string	テンプレートの技術的名称

プロパティ	タイプ	説明
description	text	テンプレートの説明
name	string	テンプレートの表示名
uuid	string	デフォルト: host プロパティの値 インポートされたテンプレートを既存のテンプレートにリンクするために使用されるユニバーサル意識別子。指定されていない場合は自動生成されます。

更新操作の場合、このフィールドは 読取専用です。

Template tag

template tag オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
tag (必須)	string	テンプレートのタグ名
value	string	テンプレートのタグ値

template.create

説明

object template.create(object/array templates)

このメソッドで、新しいテンプレートを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するテンプレート

このメソッドは標準のテンプレートプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
groups (必須)	object/array	テンプレートを追加するホストグループ
tags	object/array	ホストグループには groupid プロパティが定義されている必要があります。 テンプレートタグ
templates	object/array	テンプレートタグ テンプレートをテンプレートにリンクします。
macros	object/array	テンプレートには templateid プロパティが定義されている必要があります。 ユーザーマクロをテンプレート用に作成します。

戻り値

(object) templateids プロパティの下で作成されたテンプレートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートの順序と一致します。

例

テンプレートの作成

タグ付きのテンプレートを作成し、2つのテンプレートをこのテンプレートにリンクします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.create",
  "params": {
    "host": "Linux template",
    "groups": {
      "groupid": 1
    },
    "templates": [
      {
        "templateid": "11115"
      },
      {
        "templateid": "11116"
      }
    ],
    "tags": [
      {
        "tag": "Host name",
        "value": "{HOST.NAME}"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
      "11117"
    ]
  },
  "id": 1
}
```

ソース

CTemplate::create() in ui/include/classes/api/services/CTemplate.php.

template.delete

説明

object template.delete(array templateIds)

このメソッドで、テンプレートを削除できます。

テンプレートを削除すると、すべてのテンプレートエンティティ (アイテム、トリガー、グラフなど) が削除されます。テンプレートエンティティをホストに残しつつ、テンプレート自体を削除するには `template.update`、`template.massupdate`、`host.update`、`host.massupdate` のいずれかを使用して、ホストからテンプレートのリンクを解除します。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については [ユーザーの役割](#) を参照してください。

パラメータ

(array) 削除するテンプレートの ID

戻り値

(object) `templateids` プロパティの下で削除されたテンプレートの ID を含むオブジェクトを返します。

例

複数のテンプレート削除

2つのテンプレートを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.delete",
  "params": [
    "13",
    "32"
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
      "13",
      "32"
    ]
  },
  "id": 1
}
```

ソース

CTemplate::delete() in ui/include/classes/api/services/CTemplate.php.

template.get

説明

integer/array template.get(object parameters)

このメソッドで、指定されたパラメーターに従ってテンプレートを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメーター

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートしています。

パラメータ	タイプ	説明
templateids	string/array	指定されたテンプレート ID を持つテンプレートのみを返します
groupids	string/array	指定されたホストグループに属するテンプレートのみを返します
parentTemplateids	string/array	指定されたテンプレートの親であるテンプレートのみを返します
hostids	string/array	指定されたホスト/テンプレートにリンクされているテンプレートのみを返します
graphids	string/array	指定されたグラフを含むテンプレートのみを返します。
itemids	string/array	指定された項目を含むテンプレートのみを返します。
triggerids	string/array	指定されたトリガーを含むテンプレートのみを返します。
with_items	flag	アイテムを持つテンプレートのみを返します。
with_triggers	flag	トリガーを持つテンプレートのみを返します。
with_graphs	flag	グラフを持つテンプレートのみを返します。
with_httptests	flag	Web シナリオを含むテンプレートのみを返します。

パラメータ	タイプ	説明
evaltype	integer	タグ検索のルール
tags	array/object	<p>使用可能な値: 0 - (デフォルト) And/Or 2 - Or</p> <p>タグによる完全一致と、演算子の値に応じたタグ値による大文字と小文字の区別または区別なしで指定したタグを持つテンプレートのみを返します。</p> <p>フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]</p> <p>配列が空の時はすべてのテンプレートを返します。</p> <p>可能な演算子の値: 0 - (デフォルト) 含む 1 - 等しい 2 - 含まない 3 - 等しくない 4 - 存在する 5 - 存在しない</p>
selectGroups	query	groups プロパティでテンプレートが属するホストグループを返します。
selectTags	query	tags プロパティでテンプレートタグを返します。
selectHosts	query	hosts プロパティでテンプレートにリンクされているホストを返します。
selectTemplates	query	<p>count をサポートします。</p> <p>templates プロパティで、テンプレートが子であるテンプレートを返します。</p>
selectParentTemplates	query	<p>count をサポートします。</p> <p>parent Templates プロパティで、テンプレートが親であるテンプレートを返します。</p>
selectHttpTests	query	<p>count をサポートします。</p> <p>httpTests プロパティのテンプレートから Web シナリオを返します。</p>
selectItems	query	<p>count をサポートします。</p> <p>items プロパティのテンプレートからアイテムを返します。</p>
selectDiscoveries	query	<p>count をサポートします。</p> <p>discoveries プロパティでテンプレートからローレベルディスカバリを返します。</p>
selectTriggers	query	<p>count をサポートします。</p> <p>triggers プロパティでテンプレートからトリガーを返します。</p>
selectGraphs	query	<p>count をサポートします。</p> <p>graphs プロパティでテンプレートからグラフを返します。</p>
selectMacros	query	count をサポートします。
selectDashboards	query	<p>macros プロパティでテンプレートからマクロを返します</p> <p>dashboards プロパティのテンプレートからダッシュボードを返します。</p>
selectValueMaps	query	<p>count をサポートします。</p> <p>テンプレート値マップを含む valuemaps プロパティを返します。</p>

パラメータ	タイプ	説明
limitSelects	integer	副問い合わせによって返されるレコードの数を制限します。 次の副問い合わせに適用されます。 selectTemplates - 結果は name でソートされます。 selectHosts - host でソート。 selectParentTemplates - host でソート。 selectItems - name でソート。 selectDiscoveries - name でソート。 selectTriggers - description でソート。 selectGraphs - name でソート。 selectDashboards - name でソート。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	使用可能な値は次のとおりです: <code>hostid</code> 、 <code>host</code> 、 <code>name</code> 、 <code>status</code> 。 すべての <code>get</code> メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

名前によるテンプレートの取得

"Linux" と "Windows" という名前の 2 つのテンプレートに関するすべてのデータを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.get",
  "params": {
    "output": "extend",
    "filter": {
      "host": [
        "Linux",
        "Windows"
      ]
    }
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
```



```

    "proxy_hostid": "0",
    "host": "Linux",
    "status": "3",
    "disable_until": "0",
    "error": "",
    "available": "0",
    "errors_from": "0",
    "lastaccess": "0",
    "ipmi_authtype": "0",
    "ipmi_privilege": "2",
    "ipmi_username": "",
    "ipmi_password": "",
    "ipmi_disable_until": "0",
    "ipmi_available": "0",
    "snmp_disable_until": "0",
    "snmp_available": "0",
    "maintenanceid": "0",
    "maintenance_status": "0",
    "maintenance_type": "0",
    "maintenance_from": "0",
    "ipmi_errors_from": "0",
    "snmp_errors_from": "0",
    "ipmi_error": "",
    "snmp_error": "",
    "jmx_disable_until": "0",
    "jmx_available": "0",
    "jmx_errors_from": "0",
    "jmx_error": "",
    "name": "Linux",
    "flags": "0",
    "templateid": "10001",
    "description": "",
    "tls_connect": "1",
    "tls_accept": "1",
    "tls_issuer": "",
    "tls_subject": "",
    "tls_psk_identity": "",
    "tls_psk": "",
    "uuid": "282ffe33afc74cccaf1524d9aa9dc502"
  },
  {
    "proxy_hostid": "0",
    "host": "Windows",
    "status": "3",
    "disable_until": "0",
    "error": "",
    "available": "0",
    "errors_from": "0",
    "lastaccess": "0",
    "ipmi_authtype": "0",
    "ipmi_privilege": "2",
    "ipmi_username": "",
    "ipmi_password": "",
    "ipmi_disable_until": "0",
    "ipmi_available": "0",
    "snmp_disable_until": "0",
    "snmp_available": "0",
    "maintenanceid": "0",
    "maintenance_status": "0",
    "maintenance_type": "0",
    "maintenance_from": "0",
    "ipmi_errors_from": "0",

```

```

        "snmp_errors_from": "0",
        "ipmi_error": "",
        "snmp_error": "",
        "jmx_disable_until": "0",
        "jmx_available": "0",
        "jmx_errors_from": "0",
        "jmx_error": "",
        "name": "Windows",
        "flags": "0",
        "templateid": "10081",
        "description": "",
        "tls_connect": "1",
        "tls_accept": "1",
        "tls_issuer": "",
        "tls_subject": "",
        "tls_psk_identity": "",
        "tls_psk": "",
        "uuid": "522d17e1834049be879287b7c0518e5d"
    }
],
    "id": 1
}

```

Retrieving hosts by template

Retrieve hosts that have the "10001" (Linux by Zabbix agent) template linked to them.

Request:

```

{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}

```

```
}
```

テンプレートタグによる検索

タグ"Host name" が"{HOST.NAME}" に等しいテンプレートを取得します。

Retrieve templates that have tag "Host name" equal to "{HOST.NAME}".

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.get",
  "params": {
    "output": ["hostid"],
    "selectTags": "extend",
    "evaltype": 0,
    "tags": [
      {
        "tag": "Host name",
        "value": "{HOST.NAME}",
        "operator": 1
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10402",
      "tags": [
        {
          "tag": "Host name",
          "value": "{HOST.NAME}"
        }
      ]
    }
  ],
  "id": 1
}
```

参照

- [Host group](#)
- [Template](#)
- [User macro](#)
- [Host interface](#)

ソース

CTemplate::get() in ui/include/classes/api/services/CTemplate.php.

template.massadd

説明

object template.massadd(object parameters)

このメソッドで、指定されたテンプレートに複数の関連オブジェクトを同時に追加できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 更新するテンプレートの ID とテンプレートに追加するオブジェクトを含むパラメーター

このメソッドは、次のパラメーターを受け入れます

パラメータ	タイプ	説明
templates (必須)	object/array	更新するテンプレート。 テンプレートには <code>templateid</code> プロパティが定義されている必要があります。
groups	object/array	指定されたテンプレートを追加するホストグループ。 ホストグループには <code>groupid</code> プロパティが定義されている必要があります。
macros templates_link	object/array object/array	指定されたテンプレート用に作成されるユーザーマクロ。 指定されたテンプレートにリンクするテンプレート。 テンプレートには <code>templateid</code> プロパティが定義されている必要があります。

戻り値

(object) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

グループをテンプレートにリンクする

2つのテンプレートにホストグループ"2"を追加します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "template.massadd",
  "params": {
    "templates": [
      {
        "templateid": "10085"
      },
      {
        "templateid": "10086"
      }
    ],
    "groups": [
      {
        "groupid": "2"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
      "10085",

```

```
"10086"  
],  
},  
"id": 1  
}
```

2つのテンプレートを1つのテンプレートにリンクする

テンプレート"10106"と"10104"をテンプレートにリンクします。

リクエスト:

```
{  
  "jsonrpc": "2.0",  
  "method": "template.massadd",  
  "params": {  
    "templates": [  
      {  
        "templateid": "10073"  
      }  
    ],  
    "templates_link": [  
      {  
        "templateid": "10106"  
      },  
      {  
        "templateid": "10104"  
      }  
    ]  
  },  
  "auth": "038e1d7b1735c6a5436ee9eae095879e",  
  "id": 1  
}
```

レスポンス:

```
{  
  "jsonrpc": "2.0",  
  "result": {  
    "templateids": [  
      "10073"  
    ]  
  },  
  "id": 1  
}
```

参照

- [template.update](#)
- [Host](#)
- [Host group](#)
- [User macro](#)

ソース

CTemplate::massAdd() in ui/include/classes/api/services/CTemplate.php.

template.massremove

説明

object template.massremove(object parameters)

このメソッドで、関連するオブジェクトを複数のテンプレートから削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 更新するテンプレートの ID と削除する必要があるオブジェクトを含むパラメーター。

パラメータ	タイプ	説明
templateids (必須)	string/array	更新するテンプレートの ID。
groupids	string/array	特定のテンプレートを削除するホストグループ。
macros	string/array	指定されたテンプレートから削除するユーザーマクロ。
templateids_clear	string/array	指定されたテンプレート (上流) からリンクを解除してクリアするテンプレート。
templateids_link	string/array	指定されたテンプレート (上流) からリンクを解除するテンプレート。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

グループからのテンプレートの削除

グループ"2" から 2 つのテンプレートを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.massremove",
  "params": {
    "templateids": [
      "10085",
      "10086"
    ],
    "groupids": "2"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
      "10085",
      "10086"
    ]
  },
  "id": 1
}
```

ホストからのテンプレートのリンク解除

テンプレート"10085" からテンプレート"10106"、"10104" のリンクを解除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.massremove",
  "params": {
    "templateids": "10085",
    "templateids_link": [

```

```

        "10106",
        "10104"
    ]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
      "10085"
    ]
  },
  "id": 1
}

```

参照

- [template.update](#)
- [User macro](#)

ソース

CTemplate::massRemove() in ui/include/classes/api/services/CTemplate.php.

template.massupdate

説明

object template.massupdate(object parameters)

このメソッドで、関連するオブジェクトを同時に置換または削除して複数のテンプレートのプロパティを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object) 更新するテンプレートの ID とテンプレートを置き換えるオブジェクトを含むパラメーター

このメソッドは、次のパラメーターを受け入れます

パラメータ	タイプ	説明
templates (必須)	object/array	更新するテンプレート。 テンプレートには <code>templateid</code> プロパティが定義されている必要があります。
groups	object/array	テンプレートが属する現在のホストグループを置換するホストグループ。 ホストグループには <code>groupid</code> プロパティが定義されている必要があります。
macros	object/array	指定されたテンプレートの現在のユーザーマクロを置換するユーザーマクロ。
templates_clear	object/array	指定されたテンプレートからリンクを解除してクリアするテンプレート。 テンプレートには <code>templateid</code> プロパティが定義されている必要があります。

パラメータ	タイプ	説明
templates_link	object/array	現在リンクされているテンプレートを置き換えるテンプレート。 テンプレートには <code>templateid</code> プロパティが定義されている必要があります。

戻り値

(object) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

ホストグループの置換

指定されたテンプレートからテンプレート"10091" のリンクを解除してクリアします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.massupdate",
  "params": {
    "templates": [
      {
        "templateid": "10085"
      },
      {
        "templateid": "10086"
      }
    ],
    "templates_clear": [
      {
        "templateid": "10091"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
      "10085",
      "10086"
    ]
  },
  "id": 1
}
```

参照

- [template.update](#)
- [template.massadd](#)
- [Host group](#)
- [User macro](#)

ソース

CTemplate::massUpdate() in ui/include/classes/api/services/CTemplate.php.

template.update

説明

object template.update(object/array templates)

このメソッドで、既存のテンプレートを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するテンプレートプロパティ

テンプレートごとに templateid プロパティを定義する必要があります。すべてのプロパティはオプションです。指定されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のテンプレートプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
groups	object/array	ホストグループを使用して、テンプレートが属する現在のホストグループを置き換えます。 ホストグループには groupid プロパティが定義されている必要があります。
tags	object/array	テンプレートタグを使用して、現在のテンプレートタグを置き換えます。
macros	object/array	ユーザーマクロを使用して、特定のテンプレートの現在のユーザーマクロを置き換えます。
templates	object/array	テンプレートを使用して、現在リンクされているテンプレートを置き換えます。リンクが渡されなかったテンプレートは、リンクのみ解除されます。
templates_clear	object/array	テンプレートには templateid プロパティが定義されている必要があります。 テンプレートで指定されたテンプレートからリンクを解除してクリアします。 テンプレートには templateid プロパティが定義されている必要があります。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートの名前変更

テンプレートの名前を "Template OS Linux" に変更します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.update",
  "params": {
    "templateid": "10086",
    "name": "Template OS Linux"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "templateids": [
```

```
        "10086"
    ]
},
"id": 1
}
```

テンプレートタグの更新

すべてのテンプレートタグを新しいものに置き換えます

Request:

```
{
  "jsonrpc": "2.0",
  "method": "template.update",
  "params": {
    "templateid": "10086",
    "tags": [
      {
        "tag": "Host name",
        "value": "{HOST.NAME}"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostids": [
      "10086"
    ]
  },
  "id": 1
}
```

ソース

CTemplate::update() in ui/include/classes/api/services/CTemplate.php.

Template dashboard

このクラスは、テンプレートダッシュボードで動作するように設計されています。

オブジェクトリファレンス:

- [Template dashboard](#)
- [Template dashboard page](#)
- [Template dashboard widget](#)
- [Template dashboard widget field](#)

利用可能なメソッド:

- [templatedashboard.create](#) - 新しいテンプレートダッシュボードの作成
- [templatedashboard.delete](#) - テンプレートダッシュボードの削除
- [templatedashboard.get](#) - テンプレートダッシュボードの取得
- [templatedashboard.update](#) - テンプレートダッシュボードの更新

> Template dashboard object

次のオブジェクトは templatedashboard API に直接関連しています。

Template dashboard

template dashboard オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
dashboardid	string	(読取専用) テンプレートダッシュボードの ID
name (必須)	string	テンプレートダッシュボード名
templateid (必須)	string	ダッシュボードが属するテンプレートの ID
display_period	integer	デフォルトのページ表示期間 (秒単位) 有効な値: 10,30,60,120,600,1800,3600
auto_start	integer	デフォルト: 30 スライドショーの自動開始 有効な値: 0 - しない 1 - (デフォルト) する
uuid	string	インポートされたテンプレートダッシュボードを既存のものにリンクするために使用されるユニバーサル一意識別子。指定されていない場合は自動生成されます。 更新操作の場合このフィールドは 読取専用です。

一部のメソッド (update, delete) では、必須/オプションのパラメーターの組み合わせが異なることに注意してください。

Template dashboard page

template dashboard page オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
dashboard_pageid	string	(読取専用) ダッシュボードページの ID
name	string	ダッシュボードページ名
display_period	integer	デフォルト: 空の文字列 ダッシュボードページの表示期間 (秒単位) 有効な値: 0,10,30,60,120,600,1800,3600
widgets	array	デフォルト: 0 テンプレートダッシュボードウィジェットオブジェクトの配列

Template dashboard widget

template dashboard widget オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
widgetid	string	(読取専用) ダッシュボードウィジェットの ID
type (必須)	string	ダッシュボードウィジェットのタイプ 利用可能な値: clock - クロック graph - グラフ (クラシック) graphprototype - グラフプロトタイプ item - アイテム値 plaintext - プレーンテキスト url - URL
name	string	カスタムウィジェット名
x	integer	ダッシュボードの左側からの水平位置。 有効な値の範囲は 0 から 23 です。

プロパティ	タイプ	説明
y	integer	ダッシュボードの上部からの垂直位置。
width	integer	有効な値の範囲は 0 ~ 62 です。 ウィジェットの幅。
height	integer	有効な値の範囲は 1 ~ 24 です。 ウィジェットの高さ。
view_mode	integer	有効な値の範囲は 2 から 32 です。 ウィジェットビューモード
fields	array	有効な値: 0 - (デフォルト) デフォルトのウィジェットビュー 1 - ヘッダー非表示 テンプレートダッシュボードウィジェットフィールド オブジェクトの配列。

Template dashboard widget field

template dashboard widget field オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
type (必須)	integer	ウィジェットフィールドのタイプ 使用可能な値: 0 - 整数 1 - 文字列 4 - アイテム 5 - アイテムプロトタイプ 6 - グラフ 7 - グラフのプロトタイプ
name	string	ウィジェットフィールド名
value (必須)	mixed	タイプに応じたウィジェットフィールドの値

templatedashboard.create

説明

`object templatedashboard.create(object/array templateDashboards)`

このメソッドで、新しいテンプレートダッシュボードを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するテンプレートダッシュボード

このメソッドは標準のテンプレートダッシュボードプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
pages (必須)	array	ダッシュボード用に作成されるテンプレートダッシュボードページ。ダッシュボードページは指定した順序で並べられます。pages プロパティには少なくとも 1 つのダッシュボードページオブジェクトが必要です。

戻り値

(object) dashboardids プロパティの下に作成されたテンプレートダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートダッシュボードの順序と一致します。

例

テンプレートダッシュボードの作成

1つのダッシュボードページに1つのグラフウィジェットを含む“Graphs”という名前のテンプレートダッシュボードを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "templatedashboard.create",
  "params": {
    "templateid": "10318",
    "name": "Graphs",
    "pages": [
      {
        "widgets": [
          {
            "type": "graph",
            "x": 0,
            "y": 0,
            "width": 12,
            "height": 5,
            "view_mode": 0,
            "fields": [
              {
                "type": 6,
                "name": "graphid",
                "value": "1123"
              }
            ]
          }
        ]
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "32"
    ]
  },
  "id": 1
}
```

参照

- [Template dashboard page](#)
- [Template dashboard widget](#)
- [Template dashboard widget field](#)

ソース

CTemplateDashboard::create() in ui/include/classes/api/services/CTemplateDashboard.php.

templatedashboard.delete

説明

`object templatedashboard.delete(array templateDashboardIds)`

このメソッドで、テンプレートダッシュボードを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するテンプレートダッシュボードの ID

戻り値

(object) dashboardids プロパティの下で削除されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

複数のテンプレートダッシュボードを削除

2つのテンプレートダッシュボードを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "templatedashboard.delete",
  "params": [
    "45",
    "46"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "45",
      "46"
    ]
  },
  "id": 1
}
```

ソース

CTemplateDashboard::delete() in ui/include/classes/api/services/CTemplateDashboard.php.

templatedashboard.get

説明

`integer/array templatedashboard.get(object parameters)`

このメソッドで、指定されたパラメーターに従ってテンプレートダッシュボードを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートしています

パラメータ	タイプ	説明
dashboardids	string/array	指定された ID を持つテンプレートダッシュボードのみを返します
templateids	string/array	指定されたテンプレートに属するテンプレートダッシュボードのみを返します
selectPages	query	テンプレートダッシュボードページを正しく並べたページプロパティを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	可能な値は次のとおりです: dashboardid,name すべての get メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

テンプレートダッシュボードの取得

指定されたテンプレートのウィジェットを含むすべてのテンプレートダッシュボードを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "templatedashboard.get",
  "params": {
    "output": "extend",
    "selectPages": "extend",
    "templateids": "10001"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "dashboardid": "23",
      "name": "Docker overview",
      "templateid": "10001",
      "display_period": "30",
      "auto_start": "1",
      "uuid": "6dfcbe0bc5ad400ea9c1c2dd7649282f",
      "pages": [
        {
```

```

"dashboard_pageid": "1",
"name": "",
"display_period": "0",
"widgets": [
  {
    "widgetid": "220",
    "type": "graph",
    "name": "",
    "x": "0",
    "y": "0",
    "width": "12",
    "height": "5",
    "view_mode": "0",
    "fields": [
      {
        "type": "6",
        "name": "graphid",
        "value": "1125"
      }
    ]
  },
  {
    "widgetid": "221",
    "type": "graph",
    "name": "",
    "x": "12",
    "y": "0",
    "width": "12",
    "height": "5",
    "view_mode": "0",
    "fields": [
      {
        "type": "6",
        "name": "graphid",
        "value": "1129"
      }
    ]
  },
  {
    "widgetid": "222",
    "type": "graph",
    "name": "",
    "x": "0",
    "y": "5",
    "width": "12",
    "height": "5",
    "view_mode": "0",
    "fields": [
      {
        "type": "6",
        "name": "graphid",
        "value": "1128"
      }
    ]
  },
  {
    "widgetid": "223",
    "type": "graph",
    "name": "",
    "x": "12",
    "y": "5",
    "width": "12",

```



```

        "height": "5",
        "view_mode": "0",
        "fields": [
            {
                "type": "6",
                "name": "graphid",
                "value": "1126"
            }
        ]
    },
    {
        "widgetid": "224",
        "type": "graph",
        "name": "",
        "x": "0",
        "y": "10",
        "width": "12",
        "height": "5",
        "view_mode": "0",
        "fields": [
            {
                "type": "6",
                "name": "graphid",
                "value": "1127"
            }
        ]
    }
]
}
],
"id": 1
}

```

参照

- [Template dashboard page](#)
- [Template dashboard widget](#)
- [Template dashboard widget field](#)

ソース

CTemplateDashboard::get() in ui/include/classes/api/services/CTemplateDashboard.php.

templatedashboard.update

説明

object templatedashboard.update(object/array templateDashboards)

このメソッドで、既存のテンプレートダッシュボードを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するテンプレートダッシュボードプロパティ

ダッシュボードごとに dashboardid プロパティを指定する必要があります。すべてのプロパティはオプションです。指定したプロパティのみが更新されます。

このメソッドは標準のテンプレートダッシュボードプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
pages	array	<p>既存のダッシュボードページを置き換えるテンプレートダッシュボードページ。</p> <p>ダッシュボードページは <code>dashboard_pageid</code> プロパティによって更新されます。<code>dashboard_pageid</code> プロパティがないオブジェクトに対しては新しいダッシュボードページが作成され、既存のダッシュボードページが再利用されない場合は削除されます。ダッシュボードページは、指定した順序で並べられます。ダッシュボードページの指定されたプロパティのみが更新されます。<code>pages</code> プロパティには少なくとも 1 つのダッシュボードページオブジェクトが必要です。</p>

戻り値

(object) `dashboardids` プロパティの下で更新されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

テンプレートダッシュボードの名前変更

テンプレートダッシュボードの名前を "Performance graphs" に変更します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "templatedashboard.update",
  "params": {
    "dashboardid": "23",
    "name": "Performance graphs"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "dashboardids": [
      "23"
    ]
  },
  "id": 1
}
```

テンプレートダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換え、3 番目のページとして新しいページを追加します。他のすべてのダッシュボードページを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "templatedashboard.update",
  "params": {
    "dashboardid": "2",
    "pages": [
      {
        "dashboard_pageid": 1,
        "name": 'Renamed Page'
      },
      {
        "dashboard_pageid": 2,
        "widgets": [

```

```

        {
            "type": "clock",
            "x": 0,
            "y": 0,
            "width": 4,
            "height": 3
        }
    ],
    {
        "display_period": 60
    }
]
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 2
}

```

参照

- [Template dashboard widget](#)
- [Template dashboard widget field](#)

ソース

CTemplateDashboard::update() in ui/include/classes/api/services/CTemplateDashboard.php.

Token

このクラスは、トークンを処理できます。

オブジェクトリファレンス:

- [Token](#)

利用可能なメソッド:

- [token.create](#) - 新しいトークンの作成
- [token.delete](#) - トークンの削除
- [token.get](#) - トークンの取得
- [token.update](#) - トークンの更新
- [token.generate](#) - トークンの生成

> Token object

次のオブジェクトは、tokenAPI に直接関連しています。

Token

token オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tokenid	string	(読取専用) トークンの ID
name (必須)	string	トークンの名前
description	text	トークンの説明
userid	string	(update の時は読取専用) トークンが割り当てられているユーザー
lastaccess	timestamp	デフォルト: 現在のユーザー (読取専用) トークンが認証された最新の日時
status	integer	トークンが認証されたことがない場合はゼロ トークンのステータス
expires_at	timestamp	使用可能な値: 0 - (デフォルト) 有効なトークン 1 - 無効なトークン トークンの有効期限日時
created_at	timestamp	有効期限切れが無いトークンの場合はゼロ (読取専用) トークンの作成日時
creator_userid	string	(読取専用) トークンの作成者ユーザー

token.create

説明

object token.create(object/array tokens)

このメソッドでは、新しいトークンを作成できます。

Note:

他のユーザーのトークンを管理できるのは Super admin ユーザータイプのみです。

Note:

このメソッドで作成されたトークンは、使用する前に **generated** する必要があります。

パラメータ

(object/array) トークンの作成

このメソッドは **標準トークンプロパティ** を持つトークンを受け入れます。

戻り値

(object) tokenids プロパティの下で作成されたトークンの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトークンの順序と一致します。

例

トークンを作成する

有効期限が切れることのない有効なトークンを作成し、ID2 のユーザーを認証します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "token.create",
  "params": {
    "name": "Your token",
    "userid": "2"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "tokenids": [
      "188"
    ]
  },
  "id": 1
}
```

2021年1月21日に有効期限が切れるトークンを作成します。このトークンは現在のユーザーを認証します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "token.create",
  "params": {
    "name": "Your token",
    "status": "1",
    "expires_at": "1611238072"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "tokenids": [
      "189"
    ]
  },
  "id": 1
}
```

ソース

CToken::create() in ui/include/classes/api/services/CToken.php.

token.delete

説明

object token.delete(array tokenids)

このメソッドでは、トークンを削除できます。

Note:

他のユーザーのトークンを管理できるのは Super admin ユーザータイプのみです。

パラメータ

(array) 削除するトークンの ID

戻り値

(object) tokenids プロパティの下にある削除されたトークンの ID を含むオブジェクトを返します。

例

複数のトークンを削除

二個のトークンを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "token.delete",
  "params": [
    "188",
    "192"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "tokenids": [
      "188",
      "192"
    ]
  },
  "id": 1
}
```

ソース

CToken::delete() in ui/include/classes/api/services/CToken.php.

token.generate

説明

object token.generate(array tokenids)

このメソッドではトークンを生成できます。

Note:

他のユーザーのトークンを管理できるのは Super admin ユーザータイプのみです。

パラメータ

(array) 生成するトークンの ID

戻り値

(array) tokenId プロパティで生成されたトークンの ID と token プロパティで生成された認証文字列を含むオブジェクトの配列を返します。

プロパティ	タイプ	説明
tokenId	string	トークン ID
token	string	このトークンに対して生成された認証文字列

例

複数のトークンを生成

2個のトークンを生成します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "token.generate",
  "params": [
    "1",
    "2"
  ]
}
```

```

    ],
    "auth": "3a57200802b24cda67c4e4010b50c065",
    "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "tokenid": "1",
      "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
    },
    {
      "tokenid": "2",
      "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
    }
  ],
  "id": 0
}

```

ソース

CToken::generate() in ui/include/classes/api/services/CToken.php.

token.get

説明

integer/array token.get(object parameters)

このメソッドでは、指定されたパラメーターに従ってトークンを取得できます。

Note:

Super admin ユーザータイプのみが他のユーザーのトークンを表示できます。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします

パラメータ	タイプ	説明
tokenids	string/array	指定された ID のトークンのみを返します
userids	string/array	指定されたユーザー用に作成されたトークンのみを返します
token	string	指定された Auth トークン用に作成されたトークンのみを返します。
valid_at	timestamp	指定された日時に有効な（有効期限が切れていない）トークンのみを返します。
expired_at	timestamp	指定された日時に無効な（期限切れになった）トークンのみを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます
countOutput	boolean	指定可能な値は <code>tokenid,name,lastaccess,status,expires_at</code> および <code>created_at</code> です。
excludeSearch	boolean	すべての <code>get</code> メソッドに共通するこれらのパラメーターについては フレックス解説 ページで詳しく説明されています。
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

パラメータ	タイプ	説明
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トークンを取得する

ID"2" のトークンのデータをすべて取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "token.get",
  "params": {
    "output": "extend",
    "tokenids": "2"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "tokenid": "1",
      "name": "The Token",
      "description": "",
      "userid": "1",
      "lastaccess": "0",
      "status": "0",
      "expires_at": "1609406220",
      "created_at": "1611239454",
      "creator_userid": "1"
    }
  ],
  "id": 1
}
```

ソース

CToken::get() in ui/include/classes/api/services/CToken.php.

token.update

説明

object token.update(object/array tokens)

このメソッドでは、既存のトークンを更新できます。

Note:

他のユーザーのトークンを管理できるのは、Super admin ユーザータイプのみです。

パラメータ

(object/array) 更新されるトークンプロパティ

tokenid プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) tokenids プロパティの下で更新されたトークンの ID を含むオブジェクトを返します。

例

トークンの有効期限変更

トークンから有効期限を削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "token.update",
  "params": {
    "tokenid": "2",
    "expires_at": "0"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "tokenids": [
      "2"
    ]
  },
  "id": 1
}
```

ソース

CToken::update() in ui/include/classes/api/services/CToken.php.

Trend

このクラスはトレンドデータを処理できます。

オブジェクトリファレンス:

- **Trend**

利用可能なメソッド:

- **trend.get** - トレンドの取得

> Trend object

次のオブジェクトは、trendAPI に直接関連しています。

Note:

トレンドオブジェクトは、アイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API を介して変更することはできません。

Float trend

float trend オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値が計算された時間のタイムスタンプ。例: 04:00:00 の場合、04:00:00-04:59:59 の間で計算された値であることを意味します。
itemid	integer	関連アイテムの ID
num	integer	その時間に使用可能だった値の数
value_min	float	1 時間ごとの最小値
value_avg	float	1 時間ごとの平均値
value_max	float	1 時間ごとの最大値

Integer trend

integer trend オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
clock	timestamp	値が計算された時間のタイムスタンプ。例: 04:00:00 の場合、04:00:00-04:59:59 の間で計算された値であることを意味します。
itemid	integer	関連アイテムの ID
num	integer	その時間に使用可能だった値の数
value_min	integer	1 時間ごとの最小値
value_avg	integer	1 時間ごとの平均値
value_max	integer	1 時間ごとの最大値

trend.get

説明

`integer/array trend.get(object parameters)`

このメソッドでは、指定されたパラメーターを基にトレンドデータを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
itemids	string/array	指定されたアイテム ID のトレンドのみを返します
time_from	timestamp	指定された時間以降またはその時点で収集された値のみを返します
time_till	timestamp	指定された時間より前またはその時点で収集された値のみを返します
countOutput	boolean	取得したオブジェクトの数を数えます
limit	integer	取得するオブジェクトの量を制限します
output	query	出力フィールドを設定します

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

アイテムのトレンドデータ取得

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trend.get",
  "params": {
    "output": [
      "itemid",
      "clock",
      "num",
      "value_min",
      "value_avg",
      "value_max",
    ],
    "itemids": [
      "23715"
    ],
    "limit": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "itemid": "23715",
      "clock": "1446199200",
      "num": "60",
      "value_min": "0.165",
      "value_avg": "0.2168",
      "value_max": "0.35",
    }
  ],
  "id": 1
}
```

ソース

CTrend::get() in ui/include/classes/api/services/CTrend.php.

Trigger

このクラスは、トリガーを操作するように設計されています。

オブジェクトリファレンス:

- [Trigger](#)

利用可能なメソッド:

- [trigger.adddependencies](#) - 新しいトリガー依存関係の追加
- [trigger.create](#) - 新しいトリガーの作成
- [trigger.delete](#) - トリガーの削除
- [trigger.deletedependencies](#) - トリガー依存関係の削除
- [trigger.get](#) - トリガーの取得
- [trigger.update](#) - トリガーの更新

> Trigger object

次のオブジェクトは `trigger` API に直接関連しています。

Trigger

trigger オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
triggerid	string	(読取専用) トリガー ID
description (必須)	string	トリガー名
expression (必須)	string	トリガーの条件式
event_name	string	トリガーによって生成されたイベント名
opdata	string	運用データ
comments	string	トリガーの説明
error	string	(読取専用) トリガーの状態を更新するときに問題が発生した場合のエラーテキスト。
flags	integer	(読取専用) トリガーの作成元 使用可能な値： 0 - (デフォルト) 単純なトリガー 4 - ディスカバリされたトリガー
lastchange	timestamp	(読取専用) トリガーが最後に状態を変更した時刻
priority	integer	トリガーの深刻度 使用可能な値： 0 - (デフォルト) 未分類 1 - 情報 2 - 警告 3 - 軽度の障害 < br>4 - 重度の障害 5 - 致命的な障害
state	integer	(読取専用) トリガーの状態 使用可能な値： 0 - (デフォルト) トリガーの状態は最新 1 - 現在のトリガーの状態は不明
status	integer	トリガーの有効無効 有効な値： 0 - (デフォルト) 有効 1 - 無効
templateid	string	(読取専用) 親テンプレートトリガーの ID
type	integer	トリガーが複数の障害イベントを生成できるかどうか 使用可能な値： 0 - (デフォルト) 生成不可 1 - 生成可能
url	string	トリガーに関連付けられた URL
value	integer	(読取専用) トリガーの状態が OK か障害か 使用可能な値： 0 - (デフォルト) OK 1 - 障害
recovery_mode	integer	正常イベントの生成モード 使用可能な値： 0 - (デフォルト) 式 1 - 回復式 2 - なし
recovery_expression	string	トリガー復帰の条件式
correlation_mode	integer	正常イベントのクローズ 使用可能な値： 0 - (デフォルト) 全ての障害 1 - タグの値が一致したすべての障害

プロパティ	タイプ	説明
correlation_tag	string	マッチング用タグ
manual_close	integer	手動クローズの許可
uuid	string	使用可能な値： 0 - (デフォルト) 許可しない 1 - 許可する インポートされたトリガーを既存のものにリンクするために使用されるユニバーサル意識別子。テンプレートのトリガーにのみ使用されます。指定されていない場合は自動生成されます。

更新の場合、このフィールドは読み取り専用です。

Trigger tag

trigger tag オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	トリガータグ名
value	string	トリガータグ値

trigger.adddependencies

説明

object trigger.adddependencies(object/array triggerDependencies)

このメソッドで、新しいトリガー依存関係を作成できます

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するトリガーの依存関係

各トリガーの依存関係には、次のパラメーターがあります。

パラメータ	タイプ	説明
triggerid (必須)	string	依存トリガーの ID
dependsOnTriggerid (必須)	string	トリガーが依存するトリガーの ID

戻り値

(object) triggerids プロパティの下で依存するトリガーの ID を含むオブジェクトを返します。

例

トリガーの依存関係を追加する

トリガー"13565" に依存するトリガー"14092" を作成します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.adddependencies",
  "params": {
    "triggerid": "14092",
```

```
    "dependsOnTriggerid": "13565"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "14092"
    ]
  },
  "id": 1
}
```

参照

- [trigger.update](#)
- [Trigger dependencies](#)

ソース

CTrigger::addDependencies() in ui/include/classes/api/services/CTrigger.php.

trigger.create

説明

object trigger.create(object/array triggers)

このメソッドで、新しいトリガーを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するトリガー

このメソッドは[標準トリガープロパティ](#)に加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
dependencies	array	トリガーが依存するトリガー トリガーには triggerid プロパティが定義されている必要があります。
tags	array	トリガータグ

Attention:

トリガー式は、展開された形式で指定する必要があります。

戻り値

(object) triggerids プロパティの下で作成されたトリガーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガーの順序と一致します。

例

トリガーの作成

単一のトリガー依存関係を持つトリガーを作成します

Request:

```

{
  "jsonrpc": "2.0",
  "method": "trigger.create",
  "params": [
    {
      "description": "Processor load is too high on {HOST.NAME}",
      "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
      "dependencies": [
        {
          "triggerid": "17367"
        }
      ]
    },
    {
      "description": "Service status",
      "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
      "dependencies": [
        {
          "triggerid": "17368"
        }
      ],
      "tags": [
        {
          "tag": "service",
          "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
        },
        {
          "tag": "error",
          "value": ""
        }
      ]
    }
  ]
},
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "17369",
      "17370"
    ]
  },
  "id": 1
}

```

ソース

CTrigger::create() in ui/include/classes/api/services/CTrigger.php.

trigger.delete

説明

object trigger.delete(array triggerIds)

このメソッドで、トリガーを削除できます

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するトリガーの ID

戻り値

(object) triggerids プロパティの下で削除されたトリガーの ID を含むオブジェクトを返します。

例

複数のトリガー削除

2個のトリガーを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.delete",
  "params": [
    "12002",
    "12003"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "12002",
      "12003"
    ]
  },
  "id": 1
}
```

ソース

CTrigger::delete() in ui/include/classes/api/services/CTrigger.php.

trigger.deletedependencies

説明

object trigger.deletedependencies(string/array triggers)

このメソッドで、指定されたトリガーからすべてのトリガー依存関係を削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(string/array) トリガーの依存関係を削除するトリガー

戻り値

(object) triggerids プロパティの下で影響を受けるトリガーの ID を含むオブジェクトを返します。

例

複数のトリガーからの依存関係の削除

2つのトリガーからすべての依存関係を削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.deleteDependencies",
  "params": [
    {
      "triggerid": "14544"
    },
    {
      "triggerid": "14545"
    }
  ],
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "14544",
      "14545"
    ]
  },
  "id": 1
}
```

参照

- [trigger.update](#)

ソース

CTrigger::deleteDependencies() in ui/include/classes/api/services/CTrigger.php.

trigger.get

説明

integer/array trigger.get(object parameters)

このメソッドで、指定されたパラメーターに従ってトリガーを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートしています。

パラメータ	タイプ	説明
triggerids	string/array	指定された ID を持つトリガーのみを返します。
groupids	string/array	指定されたホストグループのホストに属するトリガーのみを返します。
templateids	string/array	指定されたテンプレートに属するトリガーのみを返します。
hostids	string/array	指定されたホストに属するトリガーのみを返します。
itemids	string/array	指定されたアイテムを含むトリガーのみを返します。

パラメータ	タイプ	説明
functions	string/array	指定された関数を使用するトリガーのみを返します。 サポートされている関数のリストは、 サポートされている関数 ページを参照してください。
group	string	指定された名前のホストグループのホストに属するトリガーのみを返します。
host	string	指定された名前のホストに属するトリガーのみを返します。
inherited	boolean	true に設定すると、テンプレートから継承されたトリガーのみが返されます。
templated	boolean	true に設定すると、テンプレートに属するトリガーのみが返されます。
dependent	boolean	true に設定すると依存関係のあるトリガーのみが返されます。false に設定すると依存関係のないトリガーのみが返されます。
monitored	flag	監視対象ホストに属し、有効なアイテムのみを含む有効なトリガーのみを返します。
active	flag	監視対象ホストに属する有効なトリガーのみを返します。
maintenance	boolean	true に設定すると、メンテナンス中のホストに属する有効なトリガーのみが返されます。
withUnacknowledgedEvents	flag	未確認のイベントがあるトリガーのみを返します。
withAcknowledgedEvents	flag	すべてのイベントが確認されたトリガーのみを返します。
withLastEventUnacknowledged	flag	Return only triggers with the last event unacknowledged.
skipDependent	flag	他のトリガーに依存する障害状態のトリガーをスキップします。無効になっている場合、アイテムが無効になっている場合、またはアイテムホストが無効になっている場合、他のトリガーは無視されます。
lastChangeSince	timestamp	指定された時間後に状態が変化したトリガーのみを返します。
lastChangeTill	timestamp	指定された時間より前に状態が変化したトリガーのみを返します。
only_true	flag	最近、障害状態になったトリガーのみを返します。
min_severity	integer	指定された深刻度以上の深刻度を持つトリガーのみを返します。
evaltype	integer	タグ検索のルール
		使用可能な値: 0 - (デフォルト) And/Or 2 - Or
tags	array of objects	指定されたタグを持つトリガーのみを返します。タグによる完全一致と、演算子の値に応じたタグ値による大文字と小文字の区別または区別なしの検索。 フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...] 空の配列はすべてのトリガーを返します。 可能な演算子の種類: 0 - (デフォルト) Like; 1 - 等しい 2 - Not Like 3 - 等しくない 4 - 存在する 5 - 存在しない
expandComment	flag	トリガーの説明でマクロを展開します。
expandDescription	flag	トリガーの名前でマクロを展開します。
expandExpression	flag	トリガー式で関数とマクロを展開します。
selectGroups	query	groups プロパティでトリガーが属するホストグループを返します
selectHosts	query	hosts プロパティでトリガーが属するホストを返します
selectItems	query	items プロパティのトリガーに含まれるアイテムを返します
selectFunctions	query	functions プロパティのトリガーで使用される関数を返します。
		関数オブジェクトは、トリガー式で使用される関数を表し、次のプロパティがあります。 functionid - (文字列) 関数の ID。 itemid - (文字列) 関数で使用されるアイテムの ID。 function - (文字列) 関数の名前; parameter - (string) 関数に渡されるパラメータ。クエリパラメータは、返された文字列で \$ 記号に置き換えられます。

パラメータ	タイプ	説明
selectDependencies	query	dependencies プロパティでトリガーが依存するトリガーを返します。
selectDiscoveryRule	query	トリガーを作成したローレベルディスカバリルールを返します。
selectLastEvent	query	lastEvent プロパティで最後の重要なトリガーイベントを返します。
selectTags	query	tags プロパティでトリガータグを返します。
selectTriggerDiscovery	query	triggerDiscovery プロパティでトリガーディスカバリオブジェクトを返します。トリガーディスカバリオブジェクトは、トリガーを作成元のトリガープロトタイプにリンクします。
filter	object	<p>次のプロパティがあります。</p> <p>parent_triggerid - (string) トリガーが作成されたトリガープロトタイプの ID</p> <p>指定されたフィルタに正確に一致する結果のみを返します。</p> <p>配列を受け入れます。キーはプロパティ名で、値は単一の値または照合する値の配列です。</p> <p>
 追加のフィルターをサポート:</p> <p>host - トリガーが属するホストの技術名;</p> <p>hostid - トリガーが属するホストの ID</p>
limitSelects	integer	subselects によって返されるレコードの数を制限します。
sortfield	string/array	<p>次の subselects に適用:</p> <p>selectHosts - 結果は host でソートされます。</p> <p>指定されたプロパティで結果を並べ替えます。</p>
countOutput	boolean	<p>使用可能な値は次のとおりです:</p> <p>triggerid,description,status,priority,lastchange,hostname</p> <p>すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。</p>
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トリガー ID によるデータの取得

トリガー "14062" で使用されるすべてのデータと関数を取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.get",
  "params": {
    "triggerids": "14062",
    "output": "extend",
    "selectFunctions": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
```

```
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "triggerid": "14062",
      "expression": "{13513}<10m",
      "description": "{HOST.NAME} has been restarted (uptime < 10m)",
      "url": "",
      "status": "0",
      "value": "0",
      "priority": "2",
      "lastchange": "0",
      "comments": "The host uptime is less than 10 minutes",
      "error": "",
      "templateid": "10016",
      "type": "0",
      "state": "0",
      "flags": "0",
      "recovery_mode": "0",
      "recovery_expression": "",
      "correlation_mode": "0",
      "correlation_tag": "",
      "manual_close": "0",
      "opdata": "",
      "functions": [
        {
          "functionid": "13513",
          "itemid": "24350",
          "triggerid": "14062",
          "parameter": "$",
          "function": "last"
        }
      ]
    }
  ],
  "id": 1
}
```

障害状態のトリガーの取得

障害状態にあるすべてのトリガーの ID、名前、深刻度を取得し、深刻度の降順で並べ替えます。Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.get",
  "params": {
    "output": [
      "triggerid",
      "description",
      "priority"
    ],
    "filter": {
      "value": 1
    },
    "sortfield": "priority",
    "sortorder": "DESC"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "triggerid": "13907",
      "description": "Zabbix self-monitoring processes < 100% busy",
      "priority": "4"
    },
    {
      "triggerid": "13824",
      "description": "Zabbix discoverer processes more than 75% busy",
      "priority": "3"
    }
  ],
  "id": 1
}
```

タグを使用して特定のトリガーを取得する

タグ付きの特定のトリガーを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.get",
  "params": {
    "output": [
      "triggerid",
      "description"
    ],
    "selectTags": "extend",
    "triggerids": [
      "17578"
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "triggerid": "17370",
      "description": "Service status",
      "tags": [
        {
          "tag": "service",
          "value": "{{ITEM.VALUE}}.regex(\"Service (.*) has stopped\", \"\\1\")"
        },
        {
          "tag": "error",
          "value": ""
        }
      ]
    }
  ],
  "id": 1
}
```

参照

- [Discovery rule](#)
- [Item](#)
- [Host](#)
- [Host group](#)

ソース

CTrigger::get() in ui/include/classes/api/services/CTrigger.php.

trigger.update

説明

`object trigger.update(object/array triggers)`

このメソッドを使用すると、既存のトリガーを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するトリガープロパティ

triggerid プロパティはトリガーごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トリガープロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
dependencies	array	トリガーが依存するトリガー。 トリガーには triggerid プロパティが定義されている必要があります。
tags	array	トリガータグ

Attention:

トリガー式は、展開された形式で指定する必要があります。

戻り値

(object) triggerids プロパティの下で更新されたトリガーの ID を含むオブジェクトを返します。

例

トリガーの有効化

ステータスを 0 に設定して、トリガーを有効にします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.update",
  "params": {
    "triggerid": "13938",
    "status": 0
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "13938"
    ]
  },
  "id": 1
}
```

トリガータグ置換

トリガーのタグを置換します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "trigger.update",
  "params": {
    "triggerid": "13938",
    "tags": [
      {
        "tag": "service",
        "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
      },
      {
        "tag": "error",
        "value": ""
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "13938"
    ]
  },
  "id": 1
}
```

参照

- [trigger.adddependencies](#)
- [trigger.deletedependencies](#)

ソース

CTrigger::update() in ui/include/classes/api/services/CTrigger.php.

Trigger prototype

このクラスは、トリガープロトタイプで動作するように設計されています

オブジェクトリファレンス:

- [Trigger prototype](#)

利用可能なメソッド:

- `triggerprototype.create` - 新しいトリガープロトタイプを作成
- `triggerprototype.delete` - トリガープロトタイプの削除
- `triggerprototype.get` - トリガープロトタイプの取得
- `triggerprototype.update` - トリガープロトタイプの更新

> Trigger prototype object

次のオブジェクトは `triggerprototype` API に直接関連しています。

Trigger prototype

`trigger prototype` オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
<code>triggerid</code>	string	(読取専用) トリガープロトタイプの ID
description (必須)	string	トリガープロトタイプ名
expression (必須)	string	トリガープロトタイプの条件式
<code>event_name</code>	string	トリガーによって生成されたイベント名
<code>opdata</code>	string	運用データ
<code>comments</code>	string	トリガープロトタイプの説明
<code>status</code>	integer	integer トリガープロトタイプの有効無効
		有効な値： 0 - (デフォルト) 有効 1 - 無効
<code>templateid</code>	string	(読取専用) 親テンプレートトリガープロトタイプの ID
<code>type</code>	integer	integer トリガープロトタイプが複数の障害イベントを生成できるかどうか
		使用可能な値： 0 - (デフォルト) 生成不可 1 - 生成可能
<code>url</code>	string	トリガープロトタイプに関連付けられた URL
<code>recovery_mode</code>	integer	integer 正常イベントの生成モード
		使用可能な値： 0 - (デフォルト) 式 1 - 回復式 2 - なし
<code>recovery_expression</code>	string	トリガー復帰の条件式
<code>correlation_mode</code>	integer	integer 正常イベントのクローズ
		使用可能な値： 0 - (デフォルト) 全ての障害 1 - タグの値が一致したすべての障害
<code>correlation_tag</code>	string	マッチング用タグ
<code>manual_close</code>	integer	integer 手動クローズの許可
		使用可能な値： 0 - (デフォルト) 許可しない 1 - 許可する
<code>discover</code>	integer	integer トリガープロトタイプの検出ステータス。
		有効な値： 0 - (デフォルト) 新しいトリガーが検出されます。 1 - 新しいトリガーは検出されず、既存のトリガーは失われたものとしてマークされます。

プロパティ	タイプ	説明
uuid	string	インポートされたトリガープロトタイプを既存のものにリンクするために使用されるユニバーサル一意識別子。テンプレートのトリガープロトタイプにのみ使用されます。指定されていない場合は自動生成されます。 更新の場合、このフィールドは読み取り専用です。

Trigger prototype tag

trigger prototype tag オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	トリガープロトタイプタグ名
value	string	トリガープロトタイプタグ値

triggerprototype.create

説明

object triggerprototype.create(object/array triggerPrototypes)

このメソッドを使用すると、新しいトリガープロトタイプを作成できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成されるトリガープロトタイプ「

このメソッドは標準のトリガープロトタイププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
dependencies	array	トリガーおよびトリガープロトタイプが依存するトリガープロトタイプ。 トリガーには triggerid プロパティが定義されている必要があります。
tags	array	トリガープロトタイプタグ

Attention:

トリガー式は展開された状態で指定する必要があり、少なくとも 1 つのアイテムプロトタイプを含める必要があります。

戻り値

(object) triggerids プロパティの下に作成されたトリガープロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガープロトタイプの順序と一致します。

例

トリガープロトタイプの作成

ファイルシステムの空きディスク容量が 20% 未満になったことを検出するトリガープロトタイプを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "triggerprototype.create",
  "params": {
    "description": "Free disk space is less than 20% on volume {#FSNAME}",
    "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME}],pfree)<20",
  }
}
```

```

    "tags": [
      {
        "tag": "volume",
        "value": "#{FSNAME}"
      },
      {
        "tag": "type",
        "value": "#{FSTYPE}"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "17372"
    ]
  },
  "id": 1
}

```

ソース

CTriggerPrototype::create() in ui/include/classes/api/services/CTriggerPrototype.php.

triggerprototype.delete

説明

object triggerprototype.delete(array triggerPrototypeIds)

このメソッドを使用すると、トリガープロトタイプを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するトリガープロトタイプの ID

戻り値

(object) triggerids プロパティの下で削除されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

複数のトリガープロトタイプ削除

2つのトリガープロトタイプを削除します。

Request:

```

{
  "jsonrpc": "2.0",
  "method": "triggerprototype.delete",
  "params": [
    "12002",
    "12003"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
}

```

```
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "12002",
      "12003"
    ]
  },
  "id": 1
}
```

ソース

CTriggerPrototype::delete() in ui/include/classes/api/services/CTriggerPrototype.php.

triggerprototype.get

説明

integer/array triggerprototype.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従ってトリガープロトタイプを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートしています。

パラメータ	タイプ	説明
active	flag	監視対象ホストに属する有効なトリガープロトタイプのみを返します。
discoveryids	string/array	指定された LLD ルールに属するトリガープロトタイプのみを返します。
functions	string/array	指定された関数を使用するトリガーのみを返します。
group	string	サポートされている関数のリストについては、 サポートされているトリガー関数 ページを参照してください。 指定された名前のホストグループのホストに属するトリガープロトタイプのみを返します。
groupids	string/array	指定されたホストグループのホストに属するトリガープロトタイプのみを返します。
host	string	指定された名前のホストに属するトリガープロトタイプのみを返します。
hostids	string/array	指定されたホストに属するトリガープロトタイプのみを返します。
inherited	boolean	"true" に設定すると、テンプレートから継承されたトリガープロトタイプのみが返されます。
maintenance	boolean	"true" に設定すると、メンテナンス中のホストに属する有効なトリガープロトタイプのみが返されます。
min_severity	integer	指定された深刻度以上の深刻度を持つトリガープロトタイプのみを返します。
monitored	flag	監視対象ホストに属し、有効なアイテムのみを含む有効なトリガープロトタイプのみを返します。
templated	boolean	"true" に設定すると、テンプレートに属するトリガープロトタイプのみが返されます。
templateids	string/array	指定されたテンプレートに属するトリガープロトタイプのみを返します。
triggerids	string/array	指定された ID を持つトリガープロトタイプのみを返します。

パラメータ	タイプ	説明
expandExpression	flag	トリガー式で関数とマクロを展開します。
selectDependencies	query	トリガープロトタイプと、トリガープロトタイプが依存するトリガーを <code>dependencies</code> プロパティで返します。
selectDiscoveryRule	query	トリガープロトタイプが属する LLD ルール を返します。
selectFunctions	query	<code>functions</code> プロパティのトリガープロトタイプで使用される関数を返します。
		関数オブジェクトは、トリガー式で使用される関数を表し、次のプロパティがあります: <code>functionid</code> - (string) ID 関数の; <code>itemid</code> - (string) 関数で使用されるアイテムの ID; <code>function</code> - (string) 関数の名前; <code>parameter</code> - * (文字列)* 関数に渡されるパラメーター。クエリパラメータは、返された文字列で \$ 記号に置き換えられます。
selectGroups	query	<code>groups</code> プロパティで、トリガープロトタイプが属するホストグループを返します。
selectHosts	query	<code>hosts</code> プロパティでトリガープロトタイプが属するホストを返します。
selectItems	query	<code>items</code> プロパティでトリガープロトタイプを使用したアイテムとアイテムプロトタイプを返します。
selectTags	query	<code>tags</code> プロパティでトリガープロトタイプタグを返します。
filter	object	指定されたフィルタに正確に一致する結果のみを返します。
		配列を受け入れます。キーはプロパティ名で、値は単一の値または照合する値の配列です。 追加のフィルターをサポート: <code>host</code> - トリガープロトタイプが属するホストの技術名; <code>hostid</code> - トリガープロトタイプが属するホストの ID サブ <code>selects</code> によって返されるレコードの数を制限します。
limitSelects	integer	次のサブ <code>selects</code> に適用されます: <code>selectHosts</code> - 結果は <code>host</code> でソートされます。 指定されたプロパティで結果を並べ替えます。
sortfield	string/array	使用可能な値: <code>triggerid,description,status,priority</code> すべての <code>get</code> メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
countOutput	boolean	
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

LLD ルールからトリガープロトタイプを取得する

すべてのトリガープロトタイプとその関数を LLD ルールから取得します。

Request:

```
{
  "jsonrpc": "2.0",
```

```

"method": "triggerprototype.get",
"params": {
  "output": "extend",
  "selectFunctions": "extend",
  "discoveryids": "22450"
},
"auth": "038e1d7b1735c6a5436ee9eae095879e",
"id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "triggerid": "13272",
      "expression": "{12598}<20",
      "description": "Free inodes is less than 20% on volume {#FSNAME}",
      "url": "",
      "status": "0",
      "priority": "2",
      "comments": "",
      "templateid": "0",
      "type": "0",
      "flags": "2",
      "recovery_mode": "0",
      "recovery_expression": "",
      "correlation_mode": "0",
      "correlation_tag": "",
      "manual_close": "0",
      "opdata": "",
      "discover": "0",
      "functions": [
        {
          "functionid": "12598",
          "itemid": "22454",
          "triggerid": "13272",
          "parameter": "$",
          "function": "last"
        }
      ]
    },
    {
      "triggerid": "13266",
      "expression": "{13500}<20",
      "description": "Free disk space is less than 20% on volume {#FSNAME}",
      "url": "",
      "status": "0",
      "priority": "2",
      "comments": "",
      "templateid": "0",
      "type": "0",
      "flags": "2",
      "recovery_mode": "0",
      "recovery_expression": "",
      "correlation_mode": "0",
      "correlation_tag": "",
      "manual_close": "0",
      "opdata": "",
      "discover": "0",
      "functions": [
        {

```

```

        "functionid": "13500",
        "itemid": "22686",
        "triggerid": "13266",
        "parameter": "$",
        "function": "last"
    }
]
},
"id": 1
}

```

タグを使用して特定のトリガープロトタイプを取得する

Request:

```

{
  "jsonrpc": "2.0",
  "method": "triggerprototype.get",
  "params": {
    "output": [
      "triggerid",
      "description"
    ],
    "selectTags": "extend",
    "triggerids": [
      "17373"
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": [
    {
      "triggerid": "17373",
      "description": "Free disk space is less than 20% on volume {#FSNAME}",
      "tags": [
        {
          "tag": "volume",
          "value": "{#FSNAME}"
        },
        {
          "tag": "type",
          "value": "{#FSTYPE}"
        }
      ]
    }
  ],
  "id": 1
}

```

参照

- [Discovery rule](#)
- [Item](#)
- [Host](#)
- [Host group](#)

ソース

CTriggerPrototype::get() in ui/include/classes/api/services/CTriggerPrototype.php.

triggerprototype.update

説明

object triggerprototype.update(object/array triggerPrototypes)

このメソッドで、トリガープロトタイプを更新できます

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新されるトリガープロトタイププロパティ

triggerid プロパティは、トリガープロトタイプごとに定義する必要があります。すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準のトリガープロトタイププロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
dependencies	array	トリガーおよびトリガープロトタイプが依存するトリガープロトタイプ。 トリガーには triggerid プロパティが定義されている必要があります。
tags	array	トリガープロトタイプタグ

Attention:

トリガー式は展開された状態で指定する必要があり、少なくとも 1 つのアイテムプロトタイプを含める必要があります。

戻り値

(object) triggerids プロパティの下で更新されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

トリガープロトタイプの有効化

ステータスを 0 に設定してトリガープロトタイプを有効にします

Request:

```
{
  "jsonrpc": "2.0",
  "method": "triggerprototype.update",
  "params": {
    "triggerid": "13938",
    "status": 0
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "13938"
    ]
  },
  "id": 1
}
```

トリガープロトタイプタグの置き換え

1つのトリガープロトタイプのタグを置き換えます

Request:

```
{
  "jsonrpc": "2.0",
  "method": "triggerprototype.update",
  "params": {
    "triggerid": "17373",
    "tags": [
      {
        "tag": "volume",
        "value": "#{FSNAME}"
      },
      {
        "tag": "type",
        "value": "#{FSTYPE}"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "triggerids": [
      "17373"
    ]
  },
  "id": 1
}
```

ソース

CTriggerPrototype::update() in ui/include/classes/api/services/CTriggerPrototype.php.

User

このクラスはユーザーと連携するように設計されています

オブジェクトリファレンス:

- [User](#)

利用可能なメソッド:

- [user.checkauthentication](#) - ユーザーセッションの確認と延長
- [user.create](#) - 新しいユーザーの作成
- [user.delete](#) - ユーザーの削除
- [user.get](#) - ユーザーの取得
- [user.login](#) - API にログイン
- [user.logout](#) - API からログアウト
- [user.unblock](#) - ユーザーブロックを解除
- [user.update](#) - ユーザーの更新

> User object

次のオブジェクトは `userAPI` に直接関連しています

User

User オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
userid	string	(読取専用) ユーザー ID
username (必須)	string	ユーザー名
attempt_clock	timestamp	(読取専用) 最後に失敗したログイン試行の時刻
attempt_failed	integer	(読取専用) 直近の失敗したログイン試行回数
attempt_ip	string	(読取専用) 最後に失敗したログイン試行の送信元 IP アドレス
autologin	integer	自動ログインを有効にするか 利用可能な値: 0 - (デフォルト) 自動ログイン無効 1 - 自動ログイン有効
autologout	string	ユーザーセッションの存続期間。接尾辞付きの秒と時間の単位を受け入れます。0 に設定すると、セッションは無期限に。
lang	string	デフォルト:15m。 ユーザーの言語コード。たとえば en_GB。
name	string	デフォルト:default - システムデフォルト
refresh	string	ユーザー名 自動更新期間。接尾辞付きの秒と時間の単位を受け入れます。
rows_per_page	integer	デフォルト:30 秒。 1 ページに表示するオブジェクト行数
surname	string	デフォルト:50 ユーザー名
theme	string	ユーザーのテーマ 利用可能な値: default - (デフォルト) システムデフォルト blue-theme - 青 dark-theme - 暗い
url	string	ログイン後にユーザーをリダイレクトするページの URL
timezone	string	ユーザーのタイムゾーン (例:Europe/London,UTC) デフォルト:default - システムデフォルト
roleid (必須)	string	サポートされているタイムゾーンの完全なリストについては、 PHP ドキュメント 参照 ユーザーロール ID

Media

The media オブジェクトには次のプロパティがあります

プロパティ	タイプ	説明
mediatypeid (必須)	string	メディアが使用するメディアタイプの ID
sendto (必須)	string/array	受信者のアドレス、ユーザー名、またはその他の識別子 メディアタイプが電子メールの場合、値は配列として表されます。他のタイプのメディアタイプの場合、値は文字列として表されます。

プロパティ	タイプ	説明
active	integer	メディアの有効無効 利用可能な値: 0 - (デフォルト) 有効 1 - 無効
severity	integer	深刻度をトリガーとして通知を送信します 深刻度はバイナリ形式で保存され、各ビットは対応する深刻度を表します。たとえば 12 はバイナリで 1100 に等しく、深刻度でいう警告と平均を伴うトリガーから通知が送信されることを意味します。 サポートされているトリガーの深刻度リストは トリガーオブジェクトのページ を参照してください。
period	string	デフォルト:63 通知を送信できる時間。期間またはセミコロンで区切ったユーザーマクロで指定 デフォルト: 1-7,00:00-24:00

user.checkAuthentication

説明

object user.checkAuthentication

このメソッドは、ユーザーセッションをチェックして延長します。

Attention:

パラメータ `sessionid` を設定して `user.checkAuthentication` メソッドを呼び出すと、デフォルトでユーザーセッションが延長されます。

パラメータ

このメソッドは、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
<code>extend</code>	boolean	ユーザーセッションを延長するかどうか。 デフォルト値: "true"。値を "false" に設定すると、ユーザーセッションを延長せずにセッションを確認できます。
<code>sessionid</code>	string	"sessionid" が設定されている場合に使用されます。 ユーザーセッション ID
<code>token</code>	string	"token" が設定されていない場合は必須 ユーザーAPI トークン "sessionid" が設定されていない場合は必須

戻り値

(object) ユーザーに関する情報を含むオブジェクトを返します。

例

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.checkAuthentication",
```

```
"params": {
  "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
},
"id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userid": "1",
    "username": "Admin",
    "name": "Zabbix",
    "surname": "Administrator",
    "url": "",
    "autologin": "1",
    "autologout": "0",
    "lang": "ru_RU",
    "refresh": "0",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "127.0.0.1",
    "attempt_clock": "1355919038",
    "rows_per_page": "50",
    "timezone": "Europe/Riga",
    "roleid": "3",
    "type": 3,
    "sessionid": "673b8ba11562a35da902c66cf5c23fa2",
    "debug_mode": 0,
    "userip": "127.0.0.1",
    "gui_access": 0
  },
  "id": 1
}
```

Note:

レスポンスは"userData" パラメーターが true に設定された **User.login** メソッドの結果に似ています。(違いはユーザーデータがユーザー名/パスワードではなくセッション ID によって取得されることです)

ソース

CUser::checkAuthentication() in ui/include/classes/api/services/CUser.php.

user.create

説明

object user.create(object/array users)

このメソッドで新しいユーザーを作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については [ユーザーの役割](#) を参照してください。

Note:

ユーザーパスワードの強度は AuthenticationAPI によって定義されたパスワードポリシールールに従って検証されます。詳細については [Authentication API](#) を参照してください。

パラメータ

(object/array) ユーザー作成

このメソッドは標準ユーザープロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
passwd (必須)	string	ユーザーのパスワード LDAP アクセス権を持つグループにのみユーザーが追加される場合は省略できます
usrgrps (必須)	array	ユーザーを追加するユーザーグループ ユーザーグループには <code>usrgrpid</code> プロパティを定義する必要があります。
medias	array	作成するユーザー <code>media</code>

戻り値

(object) `userids` プロパティの下で作成されたユーザーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーの順序と一致します。

例

ユーザー作成

新しいユーザーを作成し、ユーザーグループに追加して、新しいメディアを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.create",
  "params": {
    "username": "John",
    "passwd": "Doe123",
    "roleid": "5",
    "usrgrps": [
      {
        "usrgrpid": "7"
      }
    ],
    "medias": [
      {
        "mediatypeid": "1",
        "sendto": [
          "support@company.com"
        ],
        "active": 0,
        "severity": 63,
        "period": "1-7,00:00-24:00"
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userids": [
      "12"
    ]
  },
  "id": 1
}
```

参照

- [Authentication](#)
- [Media](#)
- [User group](#)
- [Role](#)

ソース

CUser::create() in ui/include/classes/api/services/CUser.php.

user.delete

説明

object user.delete(array users)

このメソッドで、ユーザーを削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するユーザーの ID

戻り値

(object) userids プロパティの下で削除されたユーザーの ID を含むオブジェクトを返します。

例

複数のユーザー削除

2 人のユーザーを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.delete",
  "params": [
    "1",
    "5"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userids": [
      "1",
      "5"
    ]
  },
  "id": 1
}
```

ソース

CUser::delete() in ui/include/classes/api/services/CUser.php.

user.get

説明

integer/array user.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従ってユーザー情報を取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
mediaids	string/array	指定されたメディアを使用するユーザーのみを返します
mediatypesids	string/array	指定されたメディアタイプを使用するユーザーのみを返します
userid	string/array	指定された ID を持つユーザーのみを返します
usrgrpsids	string/array	指定されたユーザーグループに属するユーザーのみを返します
getAccess	flag	ユーザー権限に関する追加情報
		ユーザーごとに次のプロパティを追加
		gui_access - (integer) ユーザーのフロントエンド認証方法。値のリストは ユーザーグループオブジェクト の gui_access プロパティを参照してください。
		debug_mode - (integer) ユーザーに対してデバッグが有効になっているかどうか。値:0 - 無効,1 - 有効
		users_status - (integer) ユーザーが無効になっているかどうか。値:0 - 有効,1 - 無効
selectMedias	query	medias プロパティでユーザーが使用したメディアを返します
selectMediatypes	query	mediatypes プロパティでユーザーが使用するメディアタイプを返します
selectUsrgrps	query	usrgrps プロパティで、ユーザーが属するユーザーグループを返します
selectRole	query	role プロパティでユーザーロールを返します
sortfield	string/array	指定されたプロパティで結果を並べ替えます
countOutput	boolean	利用可能な値は userid と username です すべての get メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ユーザーの取得

設定済みの全ユーザーを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.get",
  "params": {
    "output": "extend"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "1",
      "username": "Admin",
      "name": "Zabbix",
      "surname": "Administrator",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "en_GB",
      "refresh": "0s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
      "attempt_clock": "0",
      "rows_per_page": "50",
      "timezone": "default",
      "roleid": "3"
    },
    {
      "userid": "2",
      "username": "guest",
      "name": "",
      "surname": "",
      "url": "",
      "autologin": "0",
      "autologout": "15m",
      "lang": "default",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
      "attempt_clock": "0",
      "rows_per_page": "50",
      "timezone": "default",
      "roleid": "4"
    },
    {
      "userid": "3",
      "username": "user",
      "name": "Zabbix",
      "surname": "User",
      "url": "",
      "autologin": "0",
      "autologout": "0",
      "lang": "ru_RU",
      "refresh": "15s",

```

```

        "theme": "dark-theme",
        "attempt_failed": "0",
        "attempt_ip": "",
        "attempt_clock": "0",
        "rows_per_page": "100",
        "timezone": "default",
        "roleid": "1"
    }
],
    "id": 1
}

```

ユーザーデータ取得

ID"12" のユーザーのデータを取得します

Request:

```

{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}

```

Response:

```

{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "12",
            "username": "John",
            "role": {
                "roleid": "5",
                "name": "Operator",
                "type": "1",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}

```

参照

- [Media](#)
- [Media type](#)
- [User group](#)
- [Role](#)

ソース

CUser::get() in ui/include/classes/api/services/CUser.php.

user.login

説明

string/object user.login(object parameters)

このメソッドを使用すると、API にログインして認証トークンを生成できます

Warning:

このメソッドを使用する場合は`user.logout`を実行して、開いているセッションレコードが大量に生成されないようにする必要があります。

Attention:

このメソッドは、認証されていないユーザーのみが使用でき、JSON-RPC リクエストで `auth` パラメーターを指定せずに呼び出す必要があります。

パラメータ

(object) ユーザー名とパスワードを含むパラメーター

このメソッドは、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
password (必須)	string	ユーザーパスワード
username (必須)	string	ユーザー名
userData	flag	認証されたユーザーに関する情報
user (非推奨)	string	このパラメータは非推奨です。代わりに <code>username</code> を使用してください。 ユーザー名

戻り値

(string/object) `userData` パラメータが使用されている場合、認証されたユーザーに関する情報を含むオブジェクトを返します。

標準ユーザープロパティに加えて、次の情報が返されます。

プロパティ	タイプ	説明
<code>debug_mode</code>	boolean	ユーザーに対してデバッグモードが有効になっているかどうか
<code>gui_access</code>	integer	フロントエンドに対するユーザーの認証方法 値のリストはユーザーグループオブジェクトの <code>gui_access</code> プロパティを参照してください
<code>sessionid</code>	string	認証トークン。次の API リクエストで使用する必要があります。
<code>type</code>	integer	ユーザータイプ 使用可能な値のリストは <code>Role</code> オブジェクトの <code>type</code> プロパティを参照してください。
<code>userip</code>	string	ユーザーの IP アドレス

Note:

試行が 1 回以上失敗した後でユーザーが正常に認証された場合、メソッドは `attempt_clock`, `attempt_failed` および `attempt_ip` プロパティの現在の値を返し、それらをリセットします。

`userData` パラメータが使用されていない場合、メソッドは認証トークンを返します。

Note:

生成された認証トークンは記憶され、次の JSON-RPC リクエストの `auth` パラメータで使用される必要があります。HTTP 認証を使用する場合にも必要です。

例

ユーザー認証

ユーザー認証

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "username": "Admin",
    "password": "zabbix"
  },
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": "0424bd59b807674191e7d77572075f33",
  "id": 1
}
```

ユーザー認証と情報要求

ユーザーを認証して追加情報を返します。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "username": "Admin",
    "password": "zabbix",
    "userData": true
  },
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userid": "1",
    "username": "Admin",
    "name": "Zabbix",
    "surname": "Administrator",
    "url": "",
    "autologin": "1",
    "autologout": "0",
    "lang": "ru_RU",
    "refresh": "0",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "127.0.0.1",
    "attempt_clock": "1355919038",
    "rows_per_page": "50",
    "timezone": "Europe/Riga",
    "roleid": "3",
    "type": 3,
    "userip": "127.0.0.1",
    "debug_mode": 0,
    "gui_access": 0,
    "sessionid": "5b56eee8be445e98f0bd42b435736e42"
  },
  "id": 1
}
```

参照

- [user.logout](#)

ソース

CUser::login() in ui/include/classes/api/services/CUser.php.

user.logout

説明

string/object user.logout(array)

このメソッドを使用すると、API からログアウトして、現在の認証トークンを無効にすることができます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(array) このメソッドは空の配列を受け入れます。

戻り値

(boolean) ユーザーが正常にログアウトされた場合は `true` を返します

例

ログアウト

API からログアウト

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.logout",
  "params": [],
  "id": 1,
  "auth": "16a46baf181ef9602e1687f3110abf8a"
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 1
}
```

参照

- [user.login](#)

ソース

CUser::login() in ui/include/classes/api/services/CUser.php.

user.unblock

説明

object user.unblock(array userids)

このメソッドで、ユーザーブロックを解除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) ユーザーブロックを解除する ID

戻り値

(object) `userid` プロパティの下にあるブロックされていないユーザーの ID を含むオブジェクトを返します。

例

複数ユーザーのブロックを解除します

2 人のユーザーブロックを解除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.unblock",
  "params": [
    "1",
    "5"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userid": [
      "1",
      "5"
    ]
  },
  "id": 1
}
```

ソース

`CUser::unblock()` in `ui/include/classes/api/services/CUser.php`.

user.update

説明

`object user.update(object/array users)`

このメソッドでは既存のユーザーを更新できます

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

Note:

ユーザーパスワードの強度は Authentication API によって定義されたパスワードポリシールールに従って検証されます。詳細については[Authentication API](#)を参照してください。

パラメータ

(object/array) 更新するユーザープロパティ

`userid` プロパティはユーザーごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準ユーザープロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
passwd	string	ユーザーのパスワード
usrgrps	array	ユーザーが LDAP アクセス権を持つグループに属しているまたは LDAP アクセス権を持つグループのみに移動する場合は、空文字にすることができます。 ユーザーグループを使用して既存のユーザーグループを置き換えます。
medias	array	ユーザーグループには <code>usrgrpid</code> プロパティを定義する必要があります。 既存のメディアを置き換えるユーザーメディア

戻り値

(object) `userid` プロパティの下で更新されたユーザーの ID を含むオブジェクトを返します。

例

ユーザー名変更

ユーザー名を John Doe に変更

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.update",
  "params": {
    "userid": "1",
    "name": "John",
    "surname": "Doe"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userid": [
      "1"
    ]
  },
  "id": 1
}
```

ユーザーの役割を変更

ユーザーの役割を変更

Request:

```
{
  "jsonrpc": "2.0",
  "method": "user.update",
  "params": {
    "userid": "12",
    "roleid": "6"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "userids": [
      "12"
    ]
  },
  "id": 1
}
```

参照

- [Authentication](#)

ソース

CUser::update() in ui/include/classes/api/services/CUser.php.

User group

このクラスはユーザーグループと連携するように設計されています。

オブジェクトリファレンス:

- [User group](#)

利用可能なメソッド:

- [usergroup.create](#) - 新しいユーザーグループの作成
- [usergroup.delete](#) - ユーザーグループの削除
- [usergroup.get](#) - ユーザーグループの取得
- [usergroup.update](#) - ユーザーグループの更新

> User group object

次のオブジェクトは、usergroupAPI に直接関連しています。

User group

user group オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
usrgrpId	string	(読取専用) ユーザーグループの ID
name (必須)	string	ユーザーグループの名前
debug_mode	integer	デバッグモードの有効無効
		値は次のとおりです 0 - (デフォルト) 無効 1 - 有効
gui_access	integer	グループ内ユーザーのフロントエンド認証方法
		値は次のとおりです 0 - (デフォルト) システムのデフォルト認証方法を使用 1 - 内部認証を使用 2 - LDAP 認証を使用 3 - フロントエンドへのアクセス無効
users_status	integer	ユーザーグループの有効無効
		値は次のとおりです 0 - (デフォルト) 有効 1 - 無効

Permission

permission オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
id (必須)	string	アクセス許可を追加するホストグループの ID
permission (必須)	integer	ホストグループへのアクセスレベル 可能な値： 0 - アクセス拒否 2 - 読み取り専用 3 - 読み取り/書き込み可

Tag based permission

tag based permission オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
groupid (必須)	string	アクセス許可を追加するホストグループ ID
tag	string	タグ名
value	string	タグ値

usergroup.create

説明

object usergroup.create(object/array userGroups)

このメソッドでは、新しいユーザーグループを作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) ユーザーグループ作成

このメソッドは標準ユーザーグループプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
rights	object/array	グループに割り当てる権限
tag_filters	array	グループに割り当てるタグベースの権限
users	object/array	Usersを使用して、ユーザーグループに追加します。 ユーザーには userid プロパティを定義する必要があります。

戻り値

(object) usrgroups プロパティの下で作成されたユーザーグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーグループの順序と一致します。

例

ユーザーグループの作成

ホストグループ"2" へアクセスできないユーザーグループを作成し、それにユーザーを追加します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usergroup.create",
  "params": {
    "name": "Operation managers",
    "rights": {
      "permission": 0,
      "id": "2"
    },
    "users": [
      {"userid": "12"}
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "usrgrpids": [
      "20"
    ]
  },
  "id": 1
}
```

参照

- [Permission](#)

ソース

CUserGroup::create() in ui/include/classes/api/services/CUserGroup.php.

usergroup.delete

説明

object usergroup.delete(array userGroupIds)

このメソッドではユーザーグループを削除できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するユーザーグループの ID

戻り値

(object) usrgrpids プロパティの下にある削除されたユーザーグループの ID を含むオブジェクトを返します。

例

複数のユーザーグループ削除

二つのユーザーグループを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usergroup.delete",
  "params": [
```



```

    "20",
    "21"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "usrgrpids": [
      "20",
      "21"
    ]
  },
  "id": 1
}

```

ソース

CUserGroup::delete() in ui/include/classes/api/services/CUserGroup.php.

usergroup.get

説明

integer/array usergroup.get(object parameters)

このメソッドでは、指定されたパラメーターに従ってユーザーグループを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
status	integer	指定されたステータスのユーザーグループのみを返します。 サポートされているステータスのリストについては ユーザーグループページ を参照してください。
userid	string/array	指定されたユーザーを含むユーザーグループのみを返します
usrgrpids	string/array	指定された ID を持つユーザーグループのみを返します
selectTagFilters	query	tag_filters プロパティでユーザーグループタグベースの権限を返します。 次のプロパティがあります： groupid - (string) ホストグループの ID tag - (string) タグ名 value - (string) タグ値
selectUsers	query	users プロパティのユーザーグループからユーザーを返します
selectRights	query	rights プロパティでユーザーグループの権限を返します。 次のプロパティがあります permission - (integer) アクセスレベル id - (string) ホストグループの ID
limitSelects	integer	ホストグループへのアクセスレベルのリストについては ユーザーグループページ を参照してください。 副選択によって返されるレコードの数を制限します

パラメータ	タイプ	説明
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	指定可能な値は <code>usrgrpid</code> と <code>name</code> です。 すべての <code>get</code> メソッドに共通するこれらのパラメータについては リファレンス解説 ページで詳しく説明されています
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

有効なユーザーグループの取得

有効なすべてのユーザーグループを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usergroup.get",
  "params": {
    "output": "extend",
    "status": 0
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "usrgrpid": "7",
      "name": "Zabbix administrators",
      "gui_access": "0",
      "users_status": "0",
      "debug_mode": "1"
    },
    {
      "usrgrpid": "8",
      "name": "Guests",
      "gui_access": "0",
      "users_status": "0",
      "debug_mode": "0"
    },
    {
      "usrgrpid": "11",
      "name": "Enabled debug mode",

```

```

        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "1"
    },
    {
        "usrgrpid": "12",
        "name": "No access to the frontend",
        "gui_access": "2",
        "users_status": "0",
        "debug_mode": "0"
    },
    {
        "usrgrpid": "14",
        "name": "Read only",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0"
    },
    {
        "usrgrpid": "18",
        "name": "Deny",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0"
    }
],
    "id": 1
}

```

参照

- [User](#)

ソース

CUserGroup::get() in ui/include/classes/api/services/CUserGroup.php.

usergroup.update

説明

object usergroup.update(object/array userGroups)

このメソッドでは既存のユーザーグループを更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するユーザーグループのプロパティ

usrgrpid プロパティは、ユーザーグループごとに定義する必要があります。すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準ユーザーグループプロパティに加えて次のパラメーターを受け入れます。

パラメータ	タイプ	説明
rights	object/array	ユーザーグループに割り当てられている現在の権限を置き換えます
tag_filters	array	タグベースの権限をグループに割り当てます
users	object/array	Usersを使用して、ユーザーグループにユーザーを追加します。
		ユーザーには、userid プロパティを定義する必要があります

戻り値

(object) `usrgrpids` プロパティの下で更新されたユーザーグループの ID を含むオブジェクトを返します。

例

ユーザーグループの無効化

ユーザーグループを無効にします

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usergroup.update",
  "params": {
    "usrgrp_id": "17",
    "users_status": "1"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "usrgrpids": [
      "17"
    ]
  },
  "id": 1
}
```

参照

- [Permission](#)

ソース

`CUserGroup::update()` in `ui/include/classes/api/services/CUserGroup.php`.

User macro

このクラスは、ホストおよびグローバルマクロで動作するように設計されています。

オブジェクトリファレンス:

- [Global macro](#)
- [Host macro](#)

利用可能なメソッド:

- [usermacro.create](#) - 新しいホストマクロの作成
- [usermacro.createglobal](#) - 新しいグローバルマクロの作成
- [usermacro.delete](#) - ホストマクロの削除
- [usermacro.deleteglobal](#) - グローバルマクロの削除
- [usermacro.get](#) - ホストマクロとグローバルマクロの取得
- [usermacro.update](#) - ホストマクロの更新
- [usermacro.updateglobal](#) - グローバルマクロの更新

> User macro object

次のオブジェクトは、`usermacro` API に直接関連しています。

Global macro

global macro オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
globalmacroid	string	(読取専用) グローバルマクロの ID。
macro (必須)	string	マクロ文字列
value (必須)	string	マクロの値
type	integer	マクロのタイプ 有効な値: 0 - (デフォルト) テキストマクロ 1 - シークレットマクロ 2 - Vault シークレット
description	string	マクロの説明

Host macro

host macro object はホスト、ホストプロトタイプ、またはテンプレートで使用可能なマクロを定義します。次のプロパティがあります。

プロパティ	タイプ	説明
hostmacroid	string	(読取専用) ホストマクロの ID
hostid (必須)	string	マクロが属するホストの ID
macro (必須)	string	マクロ文字列
value (必須)	string	マクロの値
type	integer	マクロのタイプ 有効な値: 0 - (デフォルト) テキストマクロ 1 - シークレットマクロ 2 - Vault シークレット
description	string	マクロの説明

usermacro.create

説明

object usermacro.create(object/array hostMacros)

この方法では、新しいホストマクロを作成できます

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するホストマクロ

このメソッドは標準のホストマクロプロパティを持つホストマクロを受け入れます。

戻り値

(object) hostmacroids プロパティの下で作成されたホストマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストマクロの順序と一致します。

例

ホストマクロの作成

ホスト "10198" で値 "public" を持つホストマクロ "\${SNMP_COMMUNITY}" を作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.create",
  "params": {
    "hostid": "10198",
    "macro": "{$SNMP_COMMUNITY}",
    "value": "public"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostmacroids": [
      "11"
    ]
  },
  "id": 1
}
```

ソース

CUserMacro::create() in ui/include/classes/api/services/CUserMacro.php.

usermacro.createglobal

説明

object usermacro.createglobal(object/array globalMacros)

このメソッドを使用すると、新しいグローバルマクロを作成できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 作成するグローバルマクロ

このメソッドは標準のグローバルマクロプロパティを持つグローバルマクロを受け入れます。

戻り値

(object) globalmacroids プロパティの下で作成されたグローバルマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグローバルマクロの順序と一致します。

例

グローバルマクロの作成

値が "public" のグローバルマクロ "{\$SNMP_COMMUNITY}" を作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.createglobal",
  "params": {
    "macro": "{$SNMP_COMMUNITY}",
    "value": "public"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
}
```

```
    "id": 1
  }
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "globalmacroids": [
      "6"
    ]
  },
  "id": 1
}
```

ソース

CUserMacro::createGlobal() in ui/include/classes/api/services/CUserMacro.php.

usermacro.delete

説明

object usermacro.delete(array hostMacroIds)

このメソッドを使用すると、ホストマクロを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するホストマクロの ID

戻り値

(object) hostmacroids プロパティの下で削除されたホストマクロの ID を含むオブジェクトを返します。

例

複数のホストマクロ削除

2つのホストマクロを削除します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.delete",
  "params": [
    "32",
    "11"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostmacroids": [
      "32",
      "11"
    ]
  },
}
```

```
"id": 1
}
```

ソース

CUserMacro::delete() in ui/include/classes/api/services/CUserMacro.php.

usermacro.deleteglobal

説明

object usermacro.deleteglobal(array globalMacroIds)

このメソッドを使用すると、グローバルマクロを削除できます

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するグローバルマクロの ID

戻り値

(object) globalmacroids プロパティの下で削除されたグローバルマクロの ID を含むオブジェクトを返します。

例

複数のグローバルマクロ削除

2つのグローバルマクロを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.deleteglobal",
  "params": [
    "32",
    "11"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "globalmacroids": [
      "32",
      "11"
    ]
  },
  "id": 1
}
```

ソース

CUserMacro::deleteGlobal() in ui/include/classes/api/services/CUserMacro.php.

usermacro.get

説明

integer/array usermacro.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従ってホストマクロとグローバルマクロを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター

このメソッドは、次のパラメーターをサポートしています

パラメータ	タイプ	説明
globalmacro	flag	ホストマクロの代わりにグローバルマクロを返します
globalmacroids	string/array	指定された ID を持つグローバルマクロのみを返します
groupids	string/array	指定されたホストグループのホストまたはテンプレートに属するホストマクロのみを返します
hostids	string/array	指定されたホストまたはテンプレートに属するマクロのみを返します
hostmacroids	string/array	指定された ID を持つホストマクロのみを返します
inherited	boolean	"true" に設定すると、テンプレートから継承されたホストプロトタイプユーザーマクロのみが返されます
selectGroups	query	groups プロパティでホストマクロが属するホストグループを返します。
selectHosts	query	ホストマクロを取得する場合にのみ使用されます。 hosts マクロがホストプロパティに属するホストを返します。
selectTemplates	query	ホストマクロを取得する場合にのみ使用されます。 templates プロパティでホストマクロが属するテンプレートを返します。
sortfield	string/array	ホストマクロを取得する場合にのみ使用されます。 指定されたプロパティで結果を並べ替えます。
countOutput	boolean	使用可能な値: macro すべての get メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- **countOutput** パラメーターが使用されている場合、取得されたオブジェクトの数

例

ホストのホストマクロの取得

ホスト "10198" に定義されているすべてのホストマクロを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.get",
  "params": {
    "output": "extend",
```

```
    "hostids": "10198"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostmacroid": "9",
      "hostid": "10198",
      "macro": "{$INTERFACE}",
      "value": "eth0",
      "description": "",
      "type": "0"
    },
    {
      "hostmacroid": "11",
      "hostid": "10198",
      "macro": "{$SNMP_COMMUNITY}",
      "value": "public",
      "description": "",
      "type": "0"
    }
  ],
  "id": 1
}
```

グローバルマクロの取得

すべてのグローバルマクロを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.get",
  "params": {
    "output": "extend",
    "globalmacro": true
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "globalmacroid": "6",
      "macro": "{$SNMP_COMMUNITY}",
      "value": "public",
      "description": "",
      "type": "0"
    }
  ],
  "id": 1
}
```

ソース

CUserMacro::get() in ui/include/classes/api/services/CUserMacro.php.

usermacro.update

説明

object usermacro.update(object/array hostMacros)

このメソッドを使用すると、既存のホストマクロを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するホストマクロプロパティ

ホストマクロごとに hostmacroid プロパティを定義する必要があります。他のプロパティはすべてオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) hostmacroids プロパティの下で更新されたホストマクロの ID を含むオブジェクトを返します。

例

ホストマクロの値変更

ホストマクロの値を "public" に変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.update",
  "params": {
    "hostmacroid": "1",
    "value": "public"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "hostmacroids": [
      "1"
    ]
  },
  "id": 1
}
```

ソース

CUserMacro::update() in ui/include/classes/api/services/CUserMacro.php.

usermacro.updateglobal

説明

object usermacro.updateglobal(object/array globalMacros)

このメソッドを使用すると、既存のグローバルマクロを更新できます。

Note:

このメソッドは Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新するグローバルマクロプロパティ

グローバルマクロごとに globalmacroid プロパティを定義する必要があります。他のプロパティはすべてオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) globalmacroids プロパティの下で更新されたグローバルマクロの ID を含むオブジェクトを返します。

例

グローバルマクロの値変更

グローバルマクロの値を "public" に変更します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "usermacro.updateglobal",
  "params": {
    "globalmacroid": "1",
    "value": "public"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "globalmacroids": [
      "1"
    ]
  },
  "id": 1
}
```

ソース

CUserMacro::updateGlobal() in ui/include/classes/api/services/CUserMacro.php.

Value map

このクラスでは、バリュemapを操作できます。

オブジェクトリファレンス:

- [Value map](#)

利用可能なメソッド:

- [valuemap.create](#) - 新しいバリュemapの作成
- [valuemap.delete](#) - バリュemapの削除
- [valuemap.get](#) - バリュemapの取得
- [valuemap.update](#) - バリュemapの更新

> Value map object

次のオブジェクトは、valuemapAPIに直接関連しています。

Value map

value map オブジェクトには次のプロパティがあります。

プロパティ	タイプ	説明
valuemapid	string	(読み専用) バリュemapの ID
hostid (必須)	id	バリュemapのホスト ID
name (必須)	string	バリュemapの名前
mappings (必須)	array	現在のバリュemapのバリュemapマッピング。マッピングオブジェクトは以下で詳細に説明します
uuid	string	インポートされたバリュemapを既存のバリュemapにリンクするために使用されるユニバーサル一意識別子。テンプレートのバリュemapにのみ使用されます。指定されていない場合は自動生成されます。

更新操作の場合、このフィールドは読み取り専用です。

Value mappings

value mappings オブジェクトは、バリュemapのバリュemapマッピングを設定します。以下のプロパティがあります。

プロパティ	タイプ	説明
type	integer	マッピングマッチタイプ。タイプが 0、1、2、3、4 の場合、value フィールドを空にすることはできません。タイプ 5 の場合、value フィールドは空にする必要があります。
		可能な値： 0 - (デフォルト) 完全一致 1 - 値が大きいか等しい場合にマッピングが適用 ¹ ; 2 - 値が小さいか等しい場合にマッピングが適用 ¹ ; 3 - カンマ文字で区切られた複数の範囲を定義でき、値が範囲内の場合にマッピングが適用 ^ 1 ^; 4 - 値が正規表現と一致する場合マッピングが適用 ² ; 5 - デフォルト値。マッピングは一致するものが見つからなかった場合に適用
value (必須)	string	元の値
newvalue (必須)	string	タイプ " デフォルト " のマッピングには不要 元の値がマップされる値

¹ 値のタイプが "numeric unsigned", "numeric float" のアイテムでのみサポートされます。

² 値のタイプが "character" のアイテムでのみサポートされます。

valuemap.create

説明

```
object valuemap.create(object/array valuemaps)
```

このメソッドでは新しいバリュemapを作成できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については [ユーザーの役割](#) を参照してください。

パラメータ

(object/array) バリュemap作製

このメソッドは、[標準バリュemapのプロパティ](#)を持つバリュemapを受け入れます。

戻り値

(object) `valuemapids` プロパティの下に作成されたバリュemapの ID を含むオブジェクトを返します。返される ID の順序は、渡されたバリュemapの順序と一致します。

例

バリュemapの作成

2つのマッピングを使用して1つのバリュemapを作成します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "valuemap.create",
  "params": {
    "hostid": "50009",
    "name": "Service state",
    "mappings": [
      {
        "type": "1",
        "value": "1",
        "newvalue": "Up"
      },
      {
        "type": "5",
        "newvalue": "Down"
      }
    ]
  },
  "auth": "57562fd409b3b3b9a4d916d45207bbcb",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "valuemapids": [
      "1"
    ]
  },
  "id": 1
}
```

ソース

`CValueMap::create()` in `ui/include/classes/api/services/CValueMap.php`.

valuemap.delete

説明

`object valuemap.delete(array valuemapids)`

このメソッドではバリュemapを削除できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除するバリュemapの ID。

戻り値

(object) `valuemapids` プロパティの下にある削除されたバリュemapの ID を含むオブジェクトを返します。

例

複数のバリュemap削除

二つのバリュemapを削除

Request:

```
{
  "jsonrpc": "2.0",
  "method": "valuemap.delete",
  "params": [
    "1",
    "2"
  ],
  "auth": "57562fd409b3b3b9a4d916d45207bbcb",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "valuemapids": [
      "1",
      "2"
    ]
  },
  "id": 1
}
```

ソース

CValueMap::delete() in ui/include/classes/api/services/CValueMap.php.

valuemap.get

説明

integer/array valuemap.get(object parameters)

このメソッドでは、指定されたパラメーターに従ってバリュemapを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメーター。

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
valuemapids	string/array	指定された ID を持つバリュemapのみを返します
selectMappings	query	mappings プロパティで現在のバリュemapのバリュemapマッピングを返します。
sortfield	string/array	count をサポートします。 指定されたプロパティで結果を並べ替えます。
countOutput	boolean	可能な値は valuemapid と name です。 すべての get メソッドに共通するこれらのパラメーターについては フランス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	

パラメータ	タイプ	説明
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

バリュemapの取得

構成済みのすべてのバリュemapを取得します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "valuemap.get",
  "params": {
    "output": "extend"
  },
  "auth": "57562fd409b3b3b9a4d916d45207bbcb",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "valuemapid": "4",
      "name": "APC Battery Replacement Status"
    },
    {
      "valuemapid": "5",
      "name": "APC Battery Status"
    },
    {
      "valuemapid": "7",
      "name": "Dell Open Manage System Status"
    }
  ],
  "id": 1
}
```

Retrieve one value map with its mappings.

Request:

```
{
  "jsonrpc": "2.0",
  "method": "valuemap.get",
  "params": {
    "output": "extend",
    "selectMappings": "extend",
  }
}
```



```
    "valuemapids": ["4"]
  },
  "auth": "57562fd409b3b3b9a4d916d45207bbcb",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "valuemapid": "4",
      "name": "APC Battery Replacement Status",
      "mappings": [
        {
          "type": "0",
          "value": "1",
          "newvalue": "unknown"
        },
        {
          "type": "0",
          "value": "2",
          "newvalue": "notInstalled"
        },
        {
          "type": "0",
          "value": "3",
          "newvalue": "ok"
        },
        {
          "type": "0",
          "value": "4",
          "newvalue": "failed"
        },
        {
          "type": "0",
          "value": "5",
          "newvalue": "highTemperature"
        },
        {
          "type": "0",
          "value": "6",
          "newvalue": "replaceImmediately"
        },
        {
          "type": "0",
          "value": "7",
          "newvalue": "lowCapacity"
        }
      ]
    }
  ],
  "id": 1
}
```

ソース

CValueMap::get() in ui/include/classes/api/services/CValueMap.php.

valuemap.update

説明

object valuemap.update(object/array valuemaps)

このメソッドでは既存のバリュemapを更新できます。

Note:

このメソッドは Superadmin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) バリュemapのプロパティが更新されます。

valuemapid プロパティは、バリュemapごとに設定する必要があります。すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

戻り値

(object) valuemaps プロパティの下にある更新されたバリュemapの ID を含むオブジェクトを返します。

例

バリュemap名の変更

バリュemapの名前を "Device status" に変更する

Request:

```
{
  "jsonrpc": "2.0",
  "method": "valuemap.update",
  "params": {
    "valuemapid": "2",
    "name": "Device status"
  },
  "auth": "57562fd409b3b3b9a4d916d45207bbcb",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "valuemapids": [
      "2"
    ]
  },
  "id": 1
}
```

1つのバリュemapのマッピングを変更します。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "valuemap.update",
  "params": {
    "valuemapid": "2",
    "mappings": [
      {
        "type": "0",
        "value": "0",
        "newvalue": "Online"
      },
      {
        "type": "0",
        "value": "1",
        "newvalue": "Offline"
      }
    ]
  }
}
```

```

    ],
    },
    "auth": "57562fd409b3b3b9a4d916d45207bbcb",
    "id": 1
}

```

Response:

```

{
  "jsonrpc": "2.0",
  "result": {
    "valuemapids": [
      "2"
    ]
  },
  "id": 1
}

```

ソース

CValueMap::update() in ui/include/classes/api/services/CValueMap.php.

Web scenario

このクラスは、Web シナリオで機能するように設計されています。

オブジェクトリファレンス:

- [Web scenario](#)
- [Scenario step](#)

利用可能なメソッド:

- [httpstest.create](#) - 新しい Web シナリオの作成
- [httpstest.delete](#) - Web シナリオの削除
- [httpstest.get](#) - Web シナリオの取得
- [httpstest.update](#) - Web シナリオの更新

> Web scenario object

次のオブジェクトは、webcheckAPI に直接関連しています。

Web scenario

Web scenario オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
httpstestid	string	(読取専用)Web シナリオの ID
hostid (必須)	string	Web シナリオが属するホストの ID
name (必須)	string	Web シナリオ名
agent	string	Web シナリオで使用されるユーザーエージェント文字列
authentication	integer	デフォルト:Zabbix Web シナリオで使用される認証方法

利用可能な値:
 0 - (デフォルト) なし
 1 - basic HTTP 認証
 2 - NTLM 認証

プロパティ	タイプ	説明
delay	string	Web シナリオの実行間隔。秒、接尾辞付きの時間単位、およびユーザーマクロを受け入れます。
headers	array of HTTP fields	デフォルト:1m リクエストの実行時に送信される HTTP ヘッダー
http_password	string	basic HTTP 認証または NTLM 認証に使用されるパスワード。
http_proxy	string	次のように指定された Web シナリオで使用されるプロキシ: シ:http://[username[:password]@]proxy.example.com[:port]
http_user	string	basic HTTP 認証または NTLM 認証に使用されるユーザー名
nextcheck	timestamp	(読取専用) 次の Web シナリオの実行時間
retries	integer	Web シナリオが失敗する前に各ステップの実行を試行する回数
ssl_cert_file	string	デフォルト:1 クライアント認証に使用される SSL 証明書ファイルの名前 (PEM 形式である必要があります)
ssl_key_file	string	クライアント認証に使用される SSL 秘密鍵ファイルの名前 (PEM 形式である必要があります)
ssl_key_password	string	SSL 秘密鍵のパスワード
status	integer	Web シナリオが有効かどうか
templateid	string	利用可能な値: 0 - (デフォルト) 有効 1 - 無効 (読取専用) 親テンプレート Web シナリオの ID
variables	array of HTTP fields	Web シナリオ変数
verify_host	integer	SSL 証明書で指定されたホスト名がシナリオで使用されているものと一致することを確認するかどうか
verify_peer	integer	利用可能な値は次のとおり 0 - (デフォルト) ホストの検証をスキップ 1 - ホストを確認する Web サーバーの SSL 証明書を検証するかどうか
uuid	string	利用可能な値は次のとおり 0 - (デフォルト) ピア検証をスキップ 1 - ピア検証を実施 (既存の Web シナリオの場合は読み取り専用) インポートされた Web シナリオを既存のシナリオにリンクするために使用されるグローバル意識別子。テンプレートの Web シナリオにのみ使用されます。

Web scenario のタグ

Web scenario タグオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
tag (必須)	string	Web シナリオタグ名
value	string	Web シナリオのタグ値

Scenario step

Scenario step オブジェクトは、特定の Web シナリオチェックを定義します。以下の特性があります。

プロパティ	タイプ	説明
httpstepid	string	(読取専用) シナリオステップの ID
name (必須)	string	シナリオステップ名

プロパティ	タイプ	説明
no (必須)	integer	Web シナリオステップのシーケンス番号
url (必須)	string	チェックする URL
follow_redirects	integer	HTTP リダイレクトするか 利用可能な値は次のとおり 0 - リダイレクトしない 1 - (デフォルト) リダイレクトする
headers	array of HTTP fields	リクエストの実行時に送信される HTTP ヘッダー。シナリオステップヘッダーは、Web シナリオに指定されたヘッダーを上書きします
httptestid	string	(読取専用) ステップが属する Web シナリオの ID
posts	string array of HTTP fields	文字列 (生の POST データ) または HTTP フィールド (form field data) の配列としての HTTPPOST 変数
required	string	応答に存在する必要のあるテキスト
retrieve_mode	integer	シナリオステップで取得する必要のある HTTP 応答の一部 利用可能な値は次のとおり 0 - (デフォルト) 本文のみ 1 - ヘッダーのみ 2 - ヘッダーと本文
status_codes	string	コンマで区切られた必要な HTTP ステータスコードの範囲
timeout	string	秒単位のリクエストタイムアウト。秒、接尾辞付きの時間単位、およびユーザーマクロを受け入れます。
variables	array of HTTP fields	デフォルト:15s 最大:1h 最小:1s シナリオステップ変数
query_fields	array of HTTP fields	クエリフィールド - リクエストの実行時に URL に追加される HTTP フィールド の配列

HTTP field

HTTP field オブジェクトは、クエリフィールドデータの変数、HTTP ヘッダー、POST フォームフィールドデータを指定するために使用される名前と値を定義します。

プロパティ	タイプ	説明
name (必須)	string	ヘッダー/変数/POST または GET フィールドの名前
value (必須)	string	ヘッダー/変数/POST または GET フィールドの値

httptest.create

説明

`object httptest.create(object/array webScenarios)`

このメソッドでは、新しい Web シナリオを作成できます。

Note:

Web シナリオを作成すると、**Web モニタリングアイテム**のセットが自動的に作成されます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については**ユーザーの役割**を参照してください。

パラメータ

(object/array) Web シナリオの作成

このメソッドは標準の Web シナリオプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
steps (必須)	array	Web シナリオの ステップ 。
tags	array	Web シナリオの タグ

戻り値

(object) httpstestids プロパティの下で作成された Web シナリオの ID を含むオブジェクトを返します。返される ID の順序は、渡された Web シナリオの順序と一致します。

例

Web シナリオの作成

ホームページを監視するための Web シナリオを作成します。シナリオには、ホームページと「バージョン情報」ページをチェックして HTTP ステータスコード 200 が返されることを確認する 2 つのステップがあります。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "httpstest.create",
  "params": {
    "name": "Homepage check",
    "hostid": "10085",
    "steps": [
      {
        "name": "Homepage",
        "url": "http://example.com",
        "status_codes": "200",
        "no": 1
      },
      {
        "name": "Homepage / About",
        "url": "http://example.com/about",
        "status_codes": "200",
        "no": 2
      }
    ]
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "httpstestids": [
      "5"
    ]
  },
  "id": 1
}
```

参照

- [Scenario step](#)

ソース

CHttpTest::create() in ui/include/classes/api/services/CHttpTest.php.

httptest.delete

説明

object httptest.delete(array webScenarioIds)

このメソッドでは、Web シナリオを削除できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(array) 削除する Web シナリオの ID

戻り値

(object) httptestids プロパティの下で削除された Web シナリオの ID を含むオブジェクトを返します。

例

複数の Web シナリオの削除

2つの Web シナリオを削除します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "httptest.delete",
  "params": [
    "2",
    "3"
  ],
  "auth": "3a57200802b24cda67c4e4010b50c065",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "httptestids": [
      "2",
      "3"
    ]
  },
  "id": 1
}
```

ソース

CHttpRequest::delete() in ui/include/classes/api/services/CHttpRequest.php.

httptest.get

説明

integer/array httptest.get(object parameters)

このメソッドを使用すると、指定されたパラメーターに従って Web シナリオを取得できます。

Note:

このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは[ユーザーの役割](#)を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ

このメソッドは、次のパラメーターをサポートします。

パラメータ	タイプ	説明
groupids	string/array	指定されたホストグループに属する Web シナリオのみを返します
hostids	string/array	指定されたホストに属する Web シナリオのみを返します
httptestids	string/array	指定された ID を持つ Web シナリオのみを返します
inherited	boolean	true に設定すると、テンプレートから継承された Web シナリオのみが返されます
monitored	boolean	true に設定すると、監視対象ホストに属する有効な Web シナリオのみが返されます
templated	boolean	true に設定すると、テンプレートに属する Web シナリオのみが返されます
templateids	string/array	指定されたテンプレートに属する Web シナリオのみを返します。
expandName	flag	Web シナリオの名前でマクロを展開します
expandStepName	flag	シナリオステップの名前でマクロを展開します
evaltype	integer	タグ検索ルール。
tags	array of objects	<p>可能な値: 0 - (デフォルト) And/Or 2 - Or 指定されたタグを持つ Web シナリオのみを返します。タグによる完全一致または演算子値に応じた大文字と小文字を区別する検索または大文字と小文字を区別しない検索。 フォーマット: [{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...] 何も指定しない場合、すべての Web シナリオを返します。</p> <p>利用可能な演算子: 0 - (デフォルト) Like; 1 - Equal; 2 - Not like; 3 - Not equal 4 - Exists; 5 - Not exists.</p>
selectHosts	query	Web シナリオが属するホストを hosts プロパティの配列として返します。
selectSteps	query	steps プロパティで Web シナリオのステップを返します。
selectTags	query	count をサポートします tags プロパティで Web シナリオタグを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。
countOutput	boolean	指定可能な値は httptestid と name です すべての get メソッドに共通するこれらのパラメーターについては リファレンス解説 ページで詳しく説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列

- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

Web シナリオの取得

Web シナリオ"4" に関するすべてのデータを取得します

Request:

```
{
  "jsonrpc": "2.0",
  "method": "httptest.get",
  "params": {
    "output": "extend",
    "selectSteps": "extend",
    "httptestids": "9"
  },
  "auth": "038e1d7b1735c6a5436ee9eae095879e",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "httptestid": "9",
      "name": "Homepage check",
      "nextcheck": "0",
      "delay": "1m",
      "status": "0",
      "variables": [],
      "agent": "Zabbix",
      "authentication": "0",
      "http_user": "",
      "http_password": "",
      "hostid": "10084",
      "templateid": "0",
      "http_proxy": "",
      "retries": "1",
      "ssl_cert_file": "",
      "ssl_key_file": "",
      "ssl_key_password": "",
      "verify_peer": "0",
      "verify_host": "0",
      "headers": [],
      "steps": [
        {
          "httpstepid": "36",
          "httptestid": "9",
          "name": "Homepage",
          "no": "1",
          "url": "http://example.com",
          "timeout": "15s",
          "posts": "",
          "required": "",
          "status_codes": "200",
          "variables": [
            {
              "name": "{var}",
              "value": "12"
            }
          ],
          "follow_redirects": "1",

```

```

        "retrieve_mode": "0",
        "headers": [],
        "query_fields": []
    },
    {
        "httpstepid": "37",
        "httptestid": "9",
        "name": "Homepage / About",
        "no": "2",
        "url": "http://example.com/about",
        "timeout": "15s",
        "posts": "",
        "required": "",
        "status_codes": "200",
        "variables": [],
        "follow_redirects": "1",
        "retrieve_mode": "0",
        "headers": [],
        "query_fields": []
    }
]
},
"id": 1
}

```

参照

- [Host](#)
- [Scenario step](#)

ソース

CHttpTest::get() in ui/include/classes/api/services/CHttpTest.php.

httpstest.update

説明

object httpstest.update(object/array webScenarios)

このメソッドでは、既存の Web シナリオを更新できます。

Note:

このメソッドは Admin および Super admin ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については[ユーザーの役割](#)を参照してください。

パラメータ

(object/array) 更新される Web シナリオのプロパティ

httpstestid プロパティは Web シナリオごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準の Web シナリオプロパティに加えて、次のパラメーターを受け入れます。

パラメータ	タイプ	説明
steps	array	既存のステップを置き換えるシナリオステップ
tags	array	Web シナリオのタグ

戻り値

(object) httpstestid プロパティの下で更新された Web シナリオの ID を含むオブジェクトを返します。

例

Web シナリオの有効化

ステータスを"0" に設定して Web シナリオを有効にします。

Request:

```
{
  "jsonrpc": "2.0",
  "method": "httptest.update",
  "params": {
    "httptestid": "5",
    "status": 0
  },
  "auth": "700ca65537074ec963db7efabda78259",
  "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "httptestids": [
      "5"
    ]
  },
  "id": 1
}
```

参照

- [Scenario step](#)

ソース

`CHttpTest::update()` in `ui/include/classes/api/services/CHttpTest.php`.

6.0 での Zabbix API 変更内容

6.0.19 アクション

変更点:

[ZBX-21804](#) `action.create`, `action.get`, `action.update`, `action.delete`: ホスト、ホストグループ、テンプレート、トリガー、プロキシの書き込み権限の要件を削除しました。

6.0.15 スクリプト

変更点:

[ZBX-19466](#) 'script' オブジェクトの検証を、'name' と 'menu_path' の 2 つのプロパティの組み合わせによって一意になるように変更しました。

6.0.14 ユーザー

変更点:

[ZBXNEXT-8012](#) `user.checkAuthentication`: パラメーター `token` を追加しました。

6.0.13 設定

変更点:

[ZBXNEXT-7951](#) `configuration.import`, `configuration.importcompare`: 'templateLinkage' のオプション 'deleteMissing: true' は、不足しているテンプレートの (リンクを解除してクリアするのではなく) リンクを解除するようになりました。

ディスクバリルール

変更点:

[ZBXNEXT-7951](#) `discoveryrule.update`: パラメーター `uuid` が更新できるようになりました。

グラフ

変更点:

[ZBXNEXT-7951](#) `graph.update`: パラメーター `uuid` が更新できるようになりました。

グラフプロトタイプ

変更点:

[ZBXNEXT-7951](#) `graphprototype.update`: パラメーター `uuid` が更新できるようになりました。

ホストグループ

変更点:

[ZBXNEXT-7951](#) `hostgroup.update`: パラメーター `uuid` が更新できるようになりました。

ホストプロトタイプ

変更点:

[ZBXNEXT-7951](#) `hostprototype.update`: パラメーター `uuid` が更新できるようになりました。

httptest

変更点:

[ZBXNEXT-7951](#) `httptest.update`: パラメーター `uuid` が更新できるようになりました。

アイテム

変更点:

[ZBXNEXT-7951](#) `item.update`: パラメーター `uuid` が更新できるようになりました。

アイテムプロトタイプ

変更点:

[ZBXNEXT-7951](#) `itemprototype.update`: パラメーター `uuid` が更新できるようになりました。

テンプレート

変更点:

[ZBXNEXT-7951](#) `template.update`: パラメーター `uuid` が更新できるようになりました。

テンプレートダッシュボード

変更点:

[ZBXNEXT-7951](#) `templatedashboard.update`: パラメーター `uuid` が更新できるようになりました。

トリガー

変更点:

[ZBXNEXT-7951](#) `trigger.update`: パラメーター `uuid` が更新できるようになりました。

トリガープロトタイプ

変更点:

[ZBXNEXT-7951](#) `triggerprototype.update`: パラメーター `uuid` が更新できるようになりました。

バリュemap

変更点:

[ZBXNEXT-7951](#) `valuemap.update`: パラメーター `uuid` が更新できるようになりました。

6.0.9 ユーザー

変更点:

[ZBXNEXT-7971](#) `user.create`, `user.update`: `"url"` フィールドの最大長を 2048 文字に増やしました。

6.0.7 グラフ

変更点:

[ZBX-7706](#) `graph.get`: グラフの可用性は、グラフの `"ymin_itemid"` フィールドと `"ymax_itemid"` フィールドで指定されたアイテムへのアクセス許可に依存しません。

アクセスできないアイテムにリンクされた MIN/MAX Y 軸を持つグラフは引き続きアクセスできますが、MIN/MAX Y 軸は、指定された計算方法が `"計算済み"` である場合と同じように機能します。

グラフプロトタイプ

変更点:

[ZBX-7706](#) `graphprototype.get`: グラフプロトタイプの可用性は、グラフプロトタイプの `"ymin_itemid"` フィールドと `"ymax_itemid"` フィールドで指定されたアイテムへのアクセス許可に依存しません。

6.0.3 ディスカバリルール

バグフィックス:

[ZBX-19118](#) `discoveryrule.create`, `discoveryrule.update`: HTTP エージェントタイプの LLD ルールを作成/更新するために、プロパティ `interfaceid` は不要になりました。

アイテム

バグフィックス:

[ZBX-19118](#) `item.create`, `item.update`: プロパティ `interfaceid` は、HTTP エージェントタイプのアイテムを作成/更新するために必要なくなりました。

アイテムプロトタイプ

バグフィックス:

[ZBX-19118](#) `itemprototype.create`, `itemprototype.update`: プロパティ `interfaceid` は、HTTP エージェントタイプのアイテムプロトタイプを作成/更新するために必要なくなりました。

付録 1. 参考文献の解説

表記法 Data types

Zabbix API は以下のデータ型を入力としてサポートしています:

型	説明
boolean	A boolean value, accepts either true or false.
flag	The value is considered to be true if it is passed and not equal to null and false otherwise.
integer	A whole number.
float	A floating point number.
string	A text string.
text	A longer text string.
timestamp	A Unix timestamp.
array	An ordered sequence of values, that is, a plain array.
object	An associative array.

型	説明
query	A value which defines, what data should be returned. Can be defined as an array of property names to return only specific properties, or as one of the predefined values: extend - returns all object properties; count - returns the number of retrieved records, supported only by certain subselects.

Attention:

Zabbix API は常に文字列または配列としてのみ値を返します。

プロパティラベル

いくつかのオブジェクトのプロパティは、その動作を説明するために短いラベルでマークされています。
 以下のラベルが使用されます:

- readonly - プロパティの値は自動的に設定され、クライアントによって定義または変更されることはありません。
- constant - プロパティの値は、オブジェクトの作成時に設定することができますが、その後変更することはできません。

予約 ID 値"0" 予約 ID 値"0" は、要素のフィルタリングや参照されているオブジェクトの削除に使用することができます。
 例えば、ホストから参照されている proxy を削除するには、proxy_hostid を 0 に設定し ("proxy_hostid": "0")、
 サーバオプションの proxyids で監視するホストをフィルターするには、proxyids を 0 に設定する必要があります ("proxyids": "0")

"get" メソッドの共通パラメータ 以下のパラメータは、すべての get メソッドでサポートされています:

パラメータ	Type	説明
countOutput	boolean	Return the number of records in the result instead of the actual data.
editable	boolean	If set to true return only objects that the user has write permissions to. Default: false.
excludeSearch	boolean	Return results that do not match the criteria given in the search parameter.
filter	object	Return only those results that exactly match the given filter. Accepts an array, where the keys are property names, and the values are either a single value or an array of values to match against. Doesn't work for text fields.
limit	integer	Limit the number of records returned.
output	query	Object properties to be returned. Default: extend.
preservekeys	boolean	Use IDs as keys in the resulting array.
search	object	Return results that match the given wildcard search (case-insensitive). Accepts an array, where the keys are property names, and the values are strings to search for. If no additional options are given, this will perform a LIKE "%...%" search.
searchByAny	boolean	Works only for string and text fields. If set to true return results that match any of the criteria given in the filter or search parameter instead of all of them.
searchWildcardsEnabled	boolean	Default: false. If set to true enables the use of "*" as a wildcard character in the search parameter.
sortfield	string/array	Default: false. Sort the result by the given properties. Refer to a specific API get method description for a list of properties that can be used for sorting. Macros are not expanded before sorting. If no value is specified, data will be returned unsorted.

パラメータ	Type	説明
sortorder	string/array	Order of sorting. If an array is passed, each value will be matched to the corresponding property given in the sortfield parameter. Possible values are: ASC - (default) ascending; DESC - descending.
startSearch	boolean	The search parameter will compare the beginning of fields, that is, perform a LIKE "...%" search instead. Ignored if searchWildcardsEnabled is set to true.

例 ユーザー権限チェック

ユーザは、名前が"MySQL" または"Linux" で始まるホストへの書き込み権限を持っていますか？

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "countOutput": true,
    "search": {
      "host": ["MySQL", "Linux"]
    },
    "editable": true,
    "startSearch": true,
    "searchByAny": true
  },
  "auth": "766b71ee543230a1182ca5c44d353e36",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": "0",
  "id": 1
}
```

Note:

結果が"0" の場合は、読み取り/書き込み権限を持つホストが存在しないことを意味します。

ミスマッチカウント

名前に"ubuntu " という部分文字列が含まれていないホストの数を数える

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "countOutput": true,
    "search": {
      "host": "ubuntu"
    },
    "excludeSearch": true
  },
  "auth": "766b71ee543230a1182ca5c44d353e36",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": "44",
  "id": 1
}
```

ワイルドカードを用いたホストの検索

ホスト名に"server" という単語が含まれ、かつインタフェースポートが"10050 " または"10071 " であるホストを検索する。
その結果をホスト名の降順でソートし、5つのホストに限定する。

リクエスト:

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["hostid", "host"],
    "selectInterfaces": ["port"],
    "filter": {
      "port": ["10050", "10071"]
    },
    "search": {
      "host": "*server*"
    },
    "searchWildcardsEnabled": true,
    "searchByAny": true,
    "sortfield": "host",
    "sortorder": "DESC",
    "limit": 5
  },
  "auth": "766b71ee543230a1182ca5c44d353e36",
  "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "50003",
      "host": "WebServer-Tomcat02",
      "interfaces": [
        {
          "port": "10071"
        }
      ]
    },
    {
      "hostid": "50005",
      "host": "WebServer-Tomcat01",
      "interfaces": [
        {
          "port": "10071"
        }
      ]
    },
    {
      "hostid": "50004",
      "host": "WebServer-Nginx",
      "interfaces": [
        {
          "port": "10071"
        }
      ]
    }
  ]
}
```



```

    ]
  },
  {
    "hostid": "99032",
    "host": "MySQL server 01",
    "interfaces": [
      {
        "port": "10050"
      }
    ]
  },
  {
    "hostid": "99061",
    "host": "Linux server 01",
    "interfaces": [
      {
        "port": "10050"
      }
    ]
  }
],
"id": 1
}

```

“preservekeys” を使ったワイルドカードによるホスト検索

前のリクエストにパラメータ“preservekeys”を追加すると、結果はオブジェクトの id をキーとする連想配列として返されます。

リクエスト:

```

{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": ["hostid", "host"],
    "selectInterfaces": ["port"],
    "filter": {
      "port": ["10050", "10071"]
    },
    "search": {
      "host": "*server*"
    },
    "searchWildcardsEnabled": true,
    "searchByAny": true,
    "sortfield": "host",
    "sortorder": "DESC",
    "limit": 5,
    "preservekeys": true
  },
  "auth": "766b71ee543230a1182ca5c44d353e36",
  "id": 1
}

```

レスポンス:

```

{
  "jsonrpc": "2.0",
  "result": {
    "50003": {
      "hostid": "50003",
      "host": "WebServer-Tomcat02",
      "interfaces": [
        {
          "port": "10071"
        }
      ]
    }
  }
}

```

```

    ],
    "50005": {
      "hostid": "50005",
      "host": "WebServer-Tomcat01",
      "interfaces": [
        {
          "port": "10071"
        }
      ]
    },
    "50004": {
      "hostid": "50004",
      "host": "WebServer-Nginx",
      "interfaces": [
        {
          "port": "10071"
        }
      ]
    },
    "99032": {
      "hostid": "99032",
      "host": "MySQL server 01",
      "interfaces": [
        {
          "port": "10050"
        }
      ]
    },
    "99061": {
      "hostid": "99061",
      "host": "Linux server 01",
      "interfaces": [
        {
          "port": "10050"
        }
      ]
    }
  ],
  "id": 1
}

```

付録 2 . 5.4 から 6.0 への変更点

後方互換性のない変更点 アクション

変更点:

ZBXNEXT-6920 action.create, action.update: メソッドパラメータのバリデーションを強化しました。

ZBXNEXT-6755 action.create, action.update: パラメーターの名称を acknowledge_operations から update_operations に変更しました。

ZBXNEXT-6755 action.get: パラメーターの名称を selectAcknowledgeOperations から selectUpdateOperations に変更しました。

監査ログ

変更点:

ZBXNEXT-6715 監査ログオブジェクト: プロパティ note のサポートを終了しました。

ZBXNEXT-6715 監査ログオブジェクト: resourcetype の値 2 (Zabbix の設定) と 7 (グラフ要素) のサポートを終了しました。

ZBXNEXT-6715 監査ログオブジェクト: action の値 5 (有効) と 6 (無効) のサポートを終了しました。

ZBXNEXT-6715 auditlog.get: パラメーター selectDetails のサポートを終了しました。

[ZBXNEXT-6718](#) 監査ログオブジェクト: action の値 3 (ログイン) のサポートを終了しました。

ホストグループ

変更点:

[ZBXNEXT-6868](#) hostgroup.massupdate: hosts と templates のフィールド入力が必須になりました。

[ZBXNEXT-6868](#) hostgroup.massadd, hostgroup.massupdate, hostgroup.massremove: メソッドパラメーターのバリデーションが強化されました。

ホストプロトタイプ

変更点:

[ZBXNEXT-6959](#) hostprototype.get: グループリンクとグループプロトタイプ API オブジェクトのプロパティ group_prototypeid, hostid, templateid のサポートを終了しました。

[ZBXNEXT-6959](#) hostprototype.update: 継承されたホストプロトタイプの読み込み専用フィールド host, name, custom_interfaces, interfaces, groupLinks, groupPrototypes, templates, tags, macros, inventory_mode を編集不可にしました。

[ZBXNEXT-6959](#) hostprototype.create, hostprototype.update, hostprototype.delete: メソッドパラメーターのバリデーションを強化しました。

アイコンマップ

変更点:

[ZBXNEXT-6914](#) iconmap.create, iconmap.update: iconmap オブジェクトのプロパティ sortorder のサポートを終了しました。

メンテナンス

変更点:

[ZBXNEXT-6890](#) maintenance.create, maintenance.update, maintenance.delete: メソッドパラメーターのバリデーションを強化しました。

[ZBXNEXT-6890](#) maintenance.get, maintenance.update: タイムピリオドオブジェクトのパラメーター timeperiodid のサポートを終了しました。

メディアタイプ

変更点:

[ZBXNEXT-6885](#) mediatype.create, mediatype.update: メソッドパラメーターのバリデーションを強化しました。

ロール

変更点:

[ZBXNEXT-6787](#) name プロパティの値 manage_services のサポートを終了しました。

[ZBXNEXT-3022](#) UI オブジェクトの name プロパティの値 configuration.services のサポートを終了しました。

サービス

変更点:

[ZBXNEXT-6999](#) プロパティ uuid, description, created_at を追加しました。

[ZBXNEXT-6999](#) プロパティ showsla, goodsla, times のサポートを終了しました。

[ZBXNEXT-6800](#) status の値 "0" の意味を "OK" から "未分類" に変更しました。

[ZBXNEXT-3022](#) service.adddependencies, service.addtimes, service.deletedependencies, service.deletetimes のサポートを終了しました。

[ZBXNEXT-6674](#) プロパティ triggerid のサポートを終了しました。

[ZBXNEXT-6999](#) service.get: パラメーター showsla, selectAlarms, selectTimes のサポートを終了しました。

[ZBXNEXT-6999](#) service.getsla: メソッドのサポートを終了しました。

[ZBXNEXT-6999](#) メソッド sla.get, sla.create, sla.update, sla.delete, sla.getsli が追加されました。

[ZBXNEXT-6999](#) service.get: serviceid, status, created_at でのソート機能のサポートを開始しました。

[ZBXNEXT-6999](#) service.get: slais パラメーターのサポートを開始しました。また、uuid でのフィルタリング機能のサポートを開始しました。

[ZBXNEXT-6999](#) service.create, serevice.update: パラメーター showsla, goodsla, times のサポートを終了しました。

[ZBXNEXT-2406](#) `service.getsla: intervals` パラメーターによるリクエストへの応答から、`status`, `problems` のプロパティを削除しました。

[ZBXNEXT-3022](#) `service.create`, `service.update`: パラメーター `dependencies`, `parentid` のサポートを終了しました。

[ZBXNEXT-3022](#) `service.get`: パラメーター `selectParent`, `selectDependencies`, `selectParentDependencies` のサポートを終了しました。

[ZBXNEXT-6674](#) `service.get`: パラメーター `selectTrigger` のサポートを終了しました。

テンプレート

変更点:

[ZBXNEXT-6867](#) `template.create`, `template.update`, `template.delete`, `template.massadd`, `template.massupdate`, `template.massremove`: メソッドパラメーターのバリデーションを強化しました。

[ZBXNEXT-6867](#) `template.create`, `template.update`, `template.massadd`, `template.massupdate`: パラメーター `hosts` のサポートを終了しました。

[ZBXNEXT-6867](#) `template.massremove`: パラメーター `hostids` のサポートを終了しました。

トリガー

変更点:

[ZBXNEXT-6867](#) `trigger.adddependencies`, `trigger.deletedependencies`: 継承されたトリガーの依存関係を編集不可にしました。

その他の変更点およびバグフィックス アクション

変更点:

[ZBXNEXT-6755](#) `conditiontype` の値 27 (サービス) と 28 (サービスの名前) のサポートを開始しました。

[ZBXNEXT-6250](#) `action.get`, `action.create`, `action.update`: 新規のプロパティ `notify_if_canceled` を追加しました。

監査ログ

変更点:

[ZBXNEXT-6999](#) 新規の `resourcetype` (48 - SLA) を追加しました。

[ZBXNEXT-6923](#) 新規の `resourcetype` (47 - High availability node) を追加しました。

[ZBXNEXT-6718](#) `action` の値 (8 - Login, 9 - Failed login, 10 - History clear) のサポートを開始しました。

[ZBXNEXT-6715](#) プロパティ `username`, `recordsetid`, `details` のサポートを開始しました。

認証

変更点:

[ZBXNEXT-4029](#) パスワードポリシーのフィールド `passwd_min_length`, `passwd_check_rules` を追加しました。

ダッシュボード

変更点:

[ZBXNEXT-6999](#) ウィジェットタイプ `slareport` と、ウィジェットのフィールドタイプ (9 - Service, 10 - SLA) を追加しました。

[ZBXNEXT-6966](#) ウィジェットタイプ `item` を追加しました。

履歴

変更点:

[ZBXNEXT-6714](#) API メソッド `history.clear` を追加しました。

ハウスキーピング

変更点:

[ZBXNEXT-6755](#) プロパティ `hk_events_service` のサポートを開始しました。

アイテムプロトタイプ

変更点:

[ZBXNEXT-7049](#) `itemprototype.get`, `itemprototype.create`, `itemprototype.update`: Prometheus パターンの前処理ステップに第三パラメーターを追加しました。第二パラメーターで集計メソッドを決定します: `value`, `label`, `function`。第三パラメ

ーターは、集計メソッド label の Prometheus 出力または集計メソッド function の集計関数となります。

メンテナンス

変更点:

[ZBXNEXT-6890](#) maintenance.create, maintenance.update: パラメーター groupids が廃止されました。代わりに groups を使用してください。

[ZBXNEXT-6890](#) maintenance.create, maintenance.update: パラメーター hostids が廃止されました。代わりに hosts を使用してください。

[ZBXNEXT-6890](#) maintenance.create, maintenance.update: パラメーター groups, hosts, timeperiods, tags のタイプを array から object/array に変更しました。

メディアタイプ

変更点:

[ZBXNEXT-6755](#) メッセージテンプレートオブジェクト: conditiontype の値: 4 - (event created on service status update) のサポートを開始しました。

プロキシ

変更点:

[ZBXNEXT-6889](#) proxy.create, proxy.update: プロパティ interface.interfaceid, interface.hostid のサポートを終了しました。

正規表現

変更点:

[ZBXNEXT-6717](#) regexp.get, regexp.create, regexp.update, regexp.delete API を追加しました。

ロール

変更点:

[ZBXNEXT-6787](#) プロパティ services.read.mode, services.read.list, services.read.tag, services.write.mode, services.write.list, services.write.tag のサポートを開始しました。

[ZBXNEXT-3022](#) action オブジェクトの name プロパティの値 manage_services のサポートを開始しました。

[ZBXNEXT-6999](#) action オブジェクトの name プロパティの値 manage_sla のサポートを開始しました。

サービス

変更点:

[ZBXNEXT-6787](#) プロパティ readonly を追加しました。

[ZBXNEXT-6800](#) プロパティ weight, propagation_rule, propagation_value のサポートを開始しました。

[ZBXNEXT-6800](#) status の値 (-1 - OK) のサポートを開始しました。

[ZBXNEXT-2406](#) service.get: パラメーター deep_parentids, selectProblemEvents のサポートを開始しました。

[ZBXNEXT-6800](#) service.create, service.update: パラメーター status_rules のサポートを開始しました。

[ZBXNEXT-6800](#) service.get: パラメーター selectStatusRules のサポートを開始しました。

[ZBXNEXT-6800](#) service.get: パラメーター selectAlarms の count のサポートを開始しました。

[ZBXNEXT-3022](#) service.create, service.update: パラメーター children, parents, tags のサポートを開始しました。

[ZBXNEXT-3022](#) service.get: パラメーター evaltype, tags, selectChildren, selectParents, selectTags のサポートを開始しました。

[ZBXNEXT-3022](#) service.get: パラメーター problem_tags, without_problem_tags, selectProblemTags のサポートを開始しました。

[ZBXNEXT-6674](#) service.create, service.update: パラメーター problem_tags のサポートを開始しました。

設定

変更点:

[ZBXNEXT-6945](#) settings.get, settings.update: パラメーター geomaps_tile_provider, geomaps_tile_url, geomaps_max_zoom, geomaps_attribution のサポートを開始しました。

[ZBXNEXT-6715](#) settings.get, settings.update: パラメーター auditlog_enabled のサポートを開始しました。

sla

変更点:

[ZBXNEXT-6999](#) メソッド `sla.create`, `sla.delete`, `sla.get`, `sla.getsli`, `sla.update` に API `sla` を追加しました。

テンプレートダッシュボード

変更点:

[ZBXNEXT-6966](#) ウィジェットタイプ `item` のサポートを開始しました。

ユーザー

変更点:

[ZBXNEXT-6718](#) メソッド `user.unblock` を追加しました。

[ZBXNEXT-4029](#) `user.create`, `user.update`: パスワードポリシーに従ってパスワード強度の検証を実装しました。

ユーザーグループ

変更点:

[ZBXNEXT-6866](#) `usergroup.create`, `usergroup.update`: `userid` パラメーターが廃止されました。代わりに `users` を使用してください。

アイテム

変更点:

[ZBXNEXT-7049](#) `item.get`, `item.create`, `item.update`: Prometheus パターンの前処理ステップに第三パラメーターを追加しました。第二パラメーターで集計方法を決定します: `value`, `label`, `function`。第三パラメーターは、集計メソッド `label` の Prometheus 出力または集計メソッド `function` の集計関数となります。

20. モジュール

概要 Zabbix フロントエンドは、サードパーティモジュールの追加や独自モジュールの開発により、
 Zabbix のソースコードを変更することなく、機能を拡張することができます。

モジュールコードは、Zabbix のソースコードと同じ権限で実行されることに注意してください。
 これはつまり、以下のことを意味します。

- サードパーティモジュールは有害である可能性があります。インストールするモジュールを信頼する必要があります。
- サードパーティモジュールのコードにエラーがあると、フロントエンドがクラッシュすることがあります。
 そのような場合は、当該のモジュールコードをフロントエンドから削除してください。
 Zabbix フロントエンドをリロードするとすぐに、いくつかのモジュールが存在しないことを示すというメモが表示されます。
 Module administration (Administration → General → Modules) にアクセスし、**Scan directory** を再度クリックして、
 存在しないモジュールをデータベースから削除します。

インストール方法 各モジュールのインストールマニュアルを必ずお読みください。
 また新しいモジュールは、故障を未然に防ぐため、1 つずつインストールすることをお勧めします。

モジュールをインストールする前に:

- 信頼できるソースからモジュールをダウンロードしたことを確認してください。
 有害なコードをインストールすると、データ損失のような結果につながる可能性があります。

- 同じモジュールの異なるバージョン (同じ ID) を並行してインストールすることができます。
 ただし、一度に有効化できるのは 1 つのバージョンだけです。

モジュールをインストールする手順:

- Zabbix フロントエンドの `modules` フォルダにあるモジュール専用のフォルダに解凍します。
- モジュールフォルダに少なくとも `manifest.json` が含まれていることを確認してください。
- **Module administration** に移動し、
 Scan directory ボタンをクリックします。
- 新しいモジュールは、バージョン、作者、説明、ステータスとともにリストに表示されます。
- そのステータスをクリックして、モジュールを有効にします。

トラブルシューティング

問題点	解決策
モジュールが一覧に表示されない	Zabbix フロントエンドの modules/your-module/ フォルダに manifest.json ファイルが存在することを確認してください。存在する場合、そのモジュールは現在の Zabbix のバージョンに対応していないことを意味します。manifest.json ファイルが存在しない場合、間違ったディレクトリで解凍された可能性があります。
* フロントエンドがクラッシュしました	現在の Zabbix のバージョンまたはサーバ構成とモジュールコードが互換性がありません。モジュールファイルを削除し、フロントエンドを再読み込みしてください。いくつかのモジュールが存在しないという通知が表示されます。 Module administration に移動し、Scan directory をもう一度クリックして、存在しないモジュールをデータベースから削除してください。
* 同一の名前空間、ID、アクションに関するエラーメッセージが表示されます。 技術的なエラーメッセージが表示される	新しいモジュールを有効にする前に、競合するモジュール（エラーメッセージに記載）を無効にしてください。 モジュールの開発者にエラーを報告してください。

モジュール開発 モジュールは PHP 言語で記述されます。モデル・ビュー・コントローラ (MVC) ソフトウェアパターンデザインは、Zabbix のフロントエンドでも使用されており、開発が容易になるため、推奨されています。PHP の strict typing も歓迎されますが、必須ではありません。

モジュールを使用することで、Zabbix フロントエンドに新しいメニューアイテムやそれぞれの ビュー、アクションを簡単に追加することができます。現在のところ、モジュールを使って新しい API を登録したり、新しいデータベーステーブルを作成することはできません。

モジュールの構造

各モジュールは、(modules ディレクトリの中にある) ディレクトリで、サブディレクトリに、コントローラ、ビュー、その他のコードが格納されています。

example_module_directory/	(required)	
manifest.json	(required)	Metadata and action definition.
Module.php		Module initialization and event handling.
actions/		Action controller files.
SomethingView.php		
SomethingCreate.php		
SomethingDelete.php		
data_export/		
ExportAsXml.php		
ExportAsExcel.php		
views/		View files.
example.something.view.php		
example.something.delete.php		
js/		JavaScript files used in views.
example.something.view.js.php		
partials/		View partial files.
example.something.reusable.php		
js/		JavaScript files used in partials.
example.something.reusable.js.php		

見てわかるように、カスタムモジュールディレクトリ内の唯一の必須ファイルは manifest.json です。このファイルがないと、モジュールは登録されません。Module.php は、メニューアイテムの登録と、'onBeforeAction' や 'onTerminate' などのイベントの処理を担当します。actions、views、partials ディレクトリには、モジュールのアクションに必要な PHP と JavaScript のコードが含まれています。

命名規則

モジュールを作成する前に、ディレクトリやファイルなど、モジュールの各項目の命名規則について合意しておくことが重要です。また、上記の例 **Module structure** の項を参照してください。

Item	命名ルール	例
Module directory	Lowercase [a-z], underscore and decimal digits	example_v2

Item	命名ルール	例
Action subdirectories	Lowercase [a-z] and underscore character	data_export
Action files	CamelCase, ending with action type	SomethingView.php
View and partial files	Lowercase [a-z] Words separated with dot Prefixed by module, followed by module name Ending with action type and .php file extension	module.example.something.view.php
Javascript files	The same rules apply as for view and partial files, except the .js.php file extension.	module.example.something.view.js.php

Zabbix コアのビューまたはパーシャルを上書きする必要がある場合を除き、ビューとパーシャルのファイル名には 'module' の
プレフィックスと名前のインクルージョンが必須であることに注意してください。
ただし、このルールはアクションファイル名には適用されません。

マニフェストの準備

各モジュールは、次のようなフィールドを持つ JSON 形式の manifest.json ファイルを用意する必要があります。

Parameter	必須?	型	デフォルト値	説明
manifest_version	Yes	Double	-	モジュールのマニフェスト・バージョン。現在サポートされているバージョンは 1 です。
id	Yes	String	-	モジュール ID。同時に有効にできるのは、指定された ID を持つ 1 つのモジュールのみ。
name	Yes	String	-	管理セクションに表示されるモジュール名
version	Yes	String	-	管理セクションに表示されるモジュールのバージョン
namespace	Yes	String	-	Module.php およびアクションクラス用の PHP 名前空間
author	No	String	""	管理セクションに表示されるモジュール作成者
url	No	String	""	管理セクションに表示されるモジュール URL
description	No	String	""	管理セクションに表示されるモジュールの説明
actions	No	Object	{}	このモジュールに登録するアクション。アクションを参照
config	No	Object	{}	モジュール設定

参考までに、[Reference](#)セクションの manifest.json の例をご覧ください。

アクション

このモジュールは、manifest.json ファイル内の actions オブジェクトで定義された
フロントエンドのアクションを制御します。このようにして、新しいアクションを定義することができます。
同じように、既存のアクションを再定義することもできます。アクションの各キーはアクション名を表し、
対応する値には class を含み、オプションで layout と view のキーを含みます。

1 つのアクションは、名前、コントローラ、ビュー、レイアウトの 4 つの対応関係で定義されます。
データの検証や準備は一般的にコントローラで行われ、出力のフォーマットはビューまたはパーシャルで行われます。
レイアウトは、ページをメニュー、ヘッダ、フッタなどの要素で装飾する役割を果たします。

モジュールのアクションは、manifest.json ファイルに actions オブジェクトとして定義する必要があります。

パラメータ	必須?	型	デフォルト値	説明
key	Yes	String	-	Action の名前、小文字の [a-z], ドット (.) で区切られる
class	Yes	String	-	アクションクラスの名前。actions ディレクトリ内のサブディレクトリのパス (使用される)
layout	No	String	"layout.htmlpage"	アクションレイアウト
view	No	String	null	アクションビュー

あらかじめ定義されたレイアウトがいくつかあり、例えば layout.json のような定義済みのレイアウトがいくつかあります。
これらは、HTML とは異なる結果を生成するアクションのために用意されています。
app/views/ディレクトリにある定義済みレイアウトを探したり、自分でレイアウトを作成することもできます。

時には、アクションのビュー部分のみを再定義する必要がある場合もあります。
そのような場合は、コントローラはそのまま必要なビューや部分ファイルをモジュールの views ディレクトリに配置します。

参考までに、[Reference](#)セクションのアクションコントローラファイルの例をご覧ください。
また Zabbix ソースコードの現在のアクションは、app/ ディレクトリにあります。

Module.php

このオプションの PHP ファイルは、モジュールの初期化およびイベント処理に使用されます。
 クラス 'Module' は、ベースクラス CoreCModule を拡張して、このファイルで定義されることが期待されます。
 Module クラスは manifest.json ファイルで指定された名前空間内で定義する必要があります。


```
<?php

namespace Modules\Example;
use Core\CModule as BaseModule;

class Module extends BaseModule {
    ...
}
```

参考までに、[参考セクション](#)の Module.php の例をご覧ください。

参考 このセクションには、前のセクションで紹介したさまざまなモジュール要素の基本バージョンがあります。

manifest.json

```
{
  "manifest_version": 1.0,
  "id": "example_module",
  "name": "Example module",
  "version": "1.0",
  "namespace": "Example",
  "author": "John Smith",
  "url": "http://module.example.com",
  "description": "Short description of the module.",
  "actions": {
    "example.something.view": {
      "class": "SomethingView",
      "view": "module.example.something.view"
    },
    "example.something.create": {
      "class": "SomethingCreate",
      "layout": null
    },
    "example.something.delete": {
      "class": "SomethingDelete",
      "layout": null
    },
    "example.something.export.xml": {
      "class": "data_export/ExportAsXml",
      "layout": null
    },
    "example.something.export.excel": {
      "class": "data_export/ExportAsExcel",
      "layout": null
    }
  },
  "config": {
    "username": "john_smith"
  }
}
```

Module.php

```
<?php declare(strict_types = 1);

namespace Modules\Example;

use APP;
use CController as CAction;

/**
 * Please see Core\CModule class for additional reference.
```

```

*/
class Module extends \Core\CModule {

    /**
     * Initialize module.
     */
    public function init(): void {
        // Initialize main menu (CMenu class instance).
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Reports'))
                ->getSubmenu()
                    ->add((new \CMenuItem(_('Example wide report'))
                        ->setAction('example.report.wide.php')
                    ))
                    ->add((new \CMenuItem(_('Example narrow report'))
                        ->setAction('example.report.narrow.php')
                    ))
    };
}

/**
 * Event handler, triggered before executing the action.
 *
 * @param CAction $action Action instance responsible for current request.
 */
public function onBeforeAction(CAction $action): void {
}

/**
 * Event handler, triggered on application exit.
 *
 * @param CAction $action Action instance responsible for current request.
 */
public function onTerminate(CAction $action): void {
}
}

```

Action controller

```

<?php declare(strict_types = 1);

namespace Modules\Example\Actions;

use CControllerResponseData;
use CControllerResponseFatal;
use CController as CAction;

/**
 * Example module action.
 */
class SomethingView extends CAction {

    /**
     * Initialize action. Method called by Zabbix core.
     *
     * @return void
     */
    public function init(): void {
        /**
         * Disable SID (Session ID) validation. Session ID validation should only be used for actions which
         * modification, such as update or delete actions. In such case Session ID must be presented in the
         * the URL would expire as soon as the session expired.
         */
        $this->disableSIDvalidation();
    }
}

```

```

}

/**
 * Check and sanitize user input parameters. Method called by Zabbix core. Execution stops if false is
 *
 * @return bool true on success, false on error.
 */
protected function checkInput(): bool {
    $fields = [
        'name' => 'required|string',
        'email' => 'required|string',
        'phone' => 'string'
    ];

    // Only validated data will further be available using $this->hasInput() and $this->getInput().
    $ret = $this->validateInput($fields);

    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }

    return $ret;
}

/**
 * Check if the user has permission to execute this action. Method called by Zabbix core.
 * Execution stops if false is returned.
 *
 * @return bool
 */
protected function checkPermissions(): bool {
    $permit_user_types = [USER_TYPE_ZABBIX_ADMIN, USER_TYPE_SUPER_ADMIN];

    return in_array($this->getUserType(), $permit_user_types);
}

/**
 * Prepare the response object for the view. Method called by Zabbix core.
 *
 * @return void
 */
protected function doAction(): void {
    $contacts = $this->getInput('email');

    if ($this->hasInput('phone')) {
        $contacts .= ', ' . $this->getInput('phone');
    }

    $data = [
        'name' => $this->getInput('name'),
        'contacts' => $contacts
    ];

    $response = new CControllerResponseData($data);

    $this->setResponse($response);
}
}

```

Action view

```
<?php declare(strict_types = 1);
```

```

/**
 * @var CView $this
 */

$this->includeJsFile('example.something.view.js.php');

(new CWidget())
    ->setTitle(_('Something view'))
    ->addItem(new CDiv($data['name']))
    ->addItem(new CPartial('module.example.something.reusable', [
        'contacts' => $data['contacts']
    ]))
    ->show();

```

21. 付録

サイドバーを使用して付録の各コンテンツにアクセスしてください。

1 FAQ / トラブルシューティング

よくある質問または FAQ

- Q : キューは、フラッシュ/クリアできますか？ ([管理]→[キュー] で記述されているように) A : いいえ、できません。
- Q : 1 つのデータベースから他のデータベースにマイグレーションするにはどうすればいいですか？ A : データだけをダンプ (MySQL では `-t` フラグまたは `--no-create-info` フラグを使用) して、Zabbix からスキーマファイルを使用して新しいデータベースを作成し、データをインポートします。
- Q : 古いバージョンでは有効でしたが、(さまざまな理由からアイテムキーへの一括更新で) バージョン 3.0 からアイテムキーの中で空白文字が無効になったので、アイテムキーの中のすべての空白文字をアンダースコアに置換したいのですが、どのようにすればいいですか？ また、何に注意すればいいでしょうか？ A : アイテムキー内のすべての空白文字をアンダースコアに変換するには、データベースのクエリを使用できます。 `update items set key_=replace(key,' ','_');` トリガーでは、何も編集しなくても以下のアイテムを使用できますが、このロケーションでのアイテムの参照は変更しなければならないかもしれません。
 - 通知 (アクション)
 - マップの項目とリンクのラベル
 - 計算のアイテム式
- Q : 私のグラフでは、線のかわりにドットや空白があります。なぜそうなっているのでしょうか？ A : データが欠落しています。このことは、さまざまな理由 - Zabbix データベース、Zabbix サーバ、ネットワーク、監視デバイス ... などのパフォーマンスの障害で発生し得ます。
- Q : 「Listener failed with error: socket() for [[:10050]] failed with error 22: Invalid argument.」 というメッセージが出て、Zabbix デーモンの開始に失敗します。 A : このエラーは、バージョン 2.6.27 またはそれ以上のバージョンでコンパイルした Zabbix エージェントを、2.6.26 またはそれ以下のバージョンのカーネルを搭載しているプラットフォームで実行しようとしたときに発生します。古いバージョンのカーネル上で `SOCK_CLOEXEC` フラグをサポートしていないのが、システムコール `socket()` なので、この場合はスタティックリンクは役に立たないことに注意してください。 [ZBX-3395](#)
- Q : `$1` のような位置のパラメータを使用するコマンドを使って、そのパラメータを許容する例外ユーザーのパラメータを設定しようとしていますが、機能しません。(かわりにアイテムのパラメータを使用しています) どのようにしてこれを解決すればいいですか？ A : `$$1` のように、2 重の `$` マークを使用してください。
- Q : Opera 11 では、すべてのドロップダウンリストはスクロールバーがついていて醜いです。なぜそうなるのでしょうか？ A : それは、Opera 11.00 と 11.01 で知られているバグです。詳細な情報は [Zabbix イシュートラッカー](#) を参照してください。
- Q : カスタムテーマ内で、グラフの背景の色を変更するにはどうすればいいですか？ A : データベース内の `graphs_colours` の表と [theming guide](#) を参照してください。
- Q : DebugLevel 4 では、サーバー/プロキシログに「Trappergot [0]len0」というメッセージが表示されます。これは何ですか？ A : ほとんどの場合、これはフロントエンドで発生し、サーバーがまだ実行されているかどうかを接続して確認しています。
- Q : 私のシステムには未来時が設定されていて、現在データが入っていません。これをどのように解決できますか？ A : データベースフィールドの `hosts.disable_until*`、`drules.nextcheck`、`httpstest.nextcheck` の値をクリアし、サーバー/プロキシを再起動します。
- Q : フロントエンドのテキストアイテム値 (`{ITEM.VALUE}` マクロを使用する場合など) は、20 個のシンボルにカット/トリミングされます。それは仕様ですか？ A : はい、現在 `include/items.inc.php` にハードコードされた制限があります。

質問に対する答えが見つからない場合は、[Zabbix フォーラム](#)を確認してみてください。

2 インストールとセットアップ

1 データベースの作成

概要

Zabbix サーバーまたはプロキシのインストール中に Zabbix データベースを作成する必要があります。

このセクションでは、Zabbix データベースを作成する手順について説明します。サポートされているデータベースごとの命令セットがあります。

UTF-8 は、Zabbix でサポートされている唯一のエンコーディングです。セキュリティの問題なしに動作します。他のエンコーディングを使用する場合、ユーザーは既知のセキュリティ上の問題があることに注意する必要があります。

Note:

Zabbix Git [リポジトリ](#) からインストールする場合は、次のステップに進む前に以下を実行する必要があります。\$ make dbschema

MySQL

Zabbix サーバー/プロキシが MySQL データベースで正しく動作するために、文字セット utf8 (別名 utf8mb3) および utf8mb4 がサポートされています。(それぞれ utf8_bin および utf8mb4_bin を使用) 新規インストールには utf8mb4 を使用することをお勧めします。

Zabbix 6.0.11 以降では、スキーマのインポート中に決定論的トリガーを作成する必要があります。MySQL と MariaDB では、バイナリロギングが有効であり、スーパーユーザー権限がなく、MySQL 構成ファイルに log_bin_trust_function_creators = 1 が設定されていない場合、GLOBAL log_bin_trust_function_creators = 1 を設定する必要があります。

Zabbix パッケージからインストールする場合は、お使いのプラットフォームの[手順](#)を確認してください。

ソースから Zabbix をインストールする場合：

- データベースとユーザーの作成と設定

```
shell> mysql -uroot -p<password>
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- データをデータベースにインポートします。Zabbix プロキシデータベースの場合、schema.sql のみをインポートする必要があります。(images.sql も data.sql もインポートしないでください)

```
shell> cd database/mysql
shell> mysql -uzabbix -p<password> zabbix < schema.sql
# Zabbix プロキシ用のデータベースを作成している場合は、ここで停止します
shell> mysql -uzabbix -p<password> zabbix < images.sql
shell> mysql -uzabbix -p<password> zabbix < data.sql
```

スキーマを正常にインポートした後、log_bin_trust_function_creators を無効にすることができます：

```
shell> mysql -uroot -p<password>
mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

データベースオブジェクトを作成する権限を持つデータベースユーザーが必要です。

Zabbix パッケージからインストールする場合は、お使いのプラットフォームの[手順](#)を確認してください。

Zabbix をソースからインストールする場合：

- データベースユーザーの作成

次のシェルコマンドは、ユーザー zabbix を作成します。プロンプトが表示されたらパスワードを指定し、パスワードを繰り返し入力します (最初に sudo のパスワード入力を求められる場合があります)

```
shell> sudo -u postgres createuser --pwprompt zabbix
```

- データベースの作成

次に以前に作成したユーザーを所有者 (-O zabbix) としてデータベース zabbix (最後のパラメーター) を設定します。

```
shell> sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- 初期スキーマとデータをインポートします (Zabbix ソースのルートディレクトリにいと仮定します)。

Zabbix プロキシデータベースの場合、schema.sql のみをインポートする必要があります (images.sql も data.sql もインポートしないでください)。

```
shell> cd database/postgresql
shell> cat schema.sql | sudo -u zabbix psql zabbix
# Zabbix プロキシ用のデータベースを作成している場合は、ここで停止します
shell> cat images.sql | sudo -u zabbix psql zabbix
shell> cat data.sql | sudo -u zabbix psql zabbix
```

Attention:

上記のコマンドは、ほとんどの GNU/Linux インストールで機能する例として提供されています。他にもさまざまなコマンドを使用できますが (例: 「psql -U <ユーザー名>」)、システム/データベースの構成方法によって異なります。データベースの設定で問題が発生した場合は、データベース管理者に相談してください。

TimescaleDB

TimescaleDB の作成と設定の手順は、別のセクションに記載されています。

Oracle

Oracle データベースの作成と設定の手順は、別のセクションに記載されています。

SQLite

SQLite の使用は **Zabbix** プロキシでのみサポートされています。

データベースが存在しない場合は自動的に作成されます。

インストールのセクションに戻ります。

2 Zabbix データベースのキャラクターセットと照合の修正

MySQL/MariaDB

歴史的に、MySQL とその派生物は utf8mb3 のエイリアスとして「utf8」を使用していました。これは MySQL 独自の標準 UTF8 の 3 バイト実装であり、4 バイトです。MySQL 8.0.28 および MariaDB 10.6.1 以降、「utf8mb3」キャラクターセットは非推奨になりました。ある時点でサポートが終了し、「utf8」は「utf8mb4」への参照になります。Zabbix 6.0 以降、「utf8mb4」がサポートされています。将来の問題を回避するために、「utf8mb4」を使用することを強くお勧めします。「utf8mb4」に切り替えるもう 1 つの利点は、補足 Unicode 文字をサポートすることです。

Warning:

Zabbix 6.0 より前のバージョンは utf8mb4 を認識しないため、utf8mb4 変換を実行する前に、まず Zabbix サーバーと DB スキーマを 6.0.x にアップグレードしてください。

1. データベースのキャラクターセットと照合を確認してください。

実行例:

```
mysql> SELECT @@character_set_database, @@collation_database;
+-----+-----+
| @@character_set_database | @@collation_database |
+-----+-----+
| latin2                   | latin2_general_ci   |
+-----+-----+
```

または:

```
mysql> SELECT @@character_set_database, @@collation_database;
+-----+-----+
| @@character_set_database | @@collation_database |
+-----+-----+
| utf8                     | utf8_bin              |
+-----+-----+
```

+-----+-----+

ご覧のとおり、キャラクターセットは「utf8mb4」ではなく、照合も「utf8mb4_bin」ではないため、修正する必要があります。

2.Zabbix を停止します。

3. データベースのバックアップコピーを作成します。

4. データベースレベルでキャラクターセットと照合を修正します。

```
alter database <データベース名> character set utf8mb4 collate utf8mb4_bin;
```

修正値の確認:

```
mysql> SELECT @@character_set_database, @@collation_database;
```

```
+-----+-----+
| @@character_set_database | @@collation_database |
+-----+-----+
| utf8mb4                  | utf8mb4_bin          |
+-----+-----+
```

5.スクリプトをロードして、テーブルおよび列レベルでの文字セットと照合を修正します。

```
mysql <データベース名> < utf8mb4_convert.sql
```

6. スクリプトを実行します。

```
SET @ZABBIX_DATABASE='<データベース名>';
MariaDBの場合→set innodb_strict_mode = OFF;
CALL zbx_convert_utf8();
MariaDBの場合→set innodb_strict_mode = ON;
drop procedure zbx_convert_utf8;
```

'utf8mb4' はわずかに多くのディスクスペースを消費すると予想されることに注意してください。

7. エラーがない場合 - 修正したデータベースのバックアップコピーを作成することをお勧めします。

8.Zabbix を起動します。

3 データベースの主キーのアップグレード

概要

Zabbix 6.0 以降、新規インストールではすべてのテーブルに主キーが使用されます。

このセクションでは既存の履歴テーブルを主キー使用に手動でアップグレードする手順について説明します。

手順は次のとおりです。

- MySQL
- PostgreSQL
- TimescaleDB
- Oracle

Attention:

The instructions provided on this page are designed for advanced users. Note that these instructions might need to be adjusted for your specific configuration.

重要事項

- アップグレードの前に、必ずデータベースをバックアップしてください。
- データベースがパーティションを使用している場合は、DB 管理者または Zabbix サポートチームにお問い合わせください。
- アップグレード中は Zabbix サーバーを停止することを強くお勧めします。ただし、どうしても必要な場合には、サーバー実行中にアップグレードを実行する方法があります (TimescaleDB を使用しない MySQL、MariaDB、および PostgreSQL のみ)。
- CSV ファイルは、プライマリキーへのアップグレードが成功した後に削除できます。
- 必要に応じて、Zabbix フロントエンドをメンテナンスモードに切り替えることができます。
- Zabbix サーバーを 6.0 にアップグレードした後、主キーへのアップグレードを行う必要があります。
- プロキシでは、使用されていない履歴テーブルは history_pk_prepare.sql を実行してアップグレードできます。

MySQL

セッションがドロップされないように、エクスポートとインポートは tmux/screen で実行する必要があります。

参照: [重要事項](#)

mysqlsh を使用した MySQL 8.0+

この方法は実行中の Zabbix サーバーで使用できますが、アップグレード中はサーバーを停止することをお勧めします。MySQL シェル (mysqlsh) が [インストール](#)され、DB に接続できる必要があります。

- root (推奨) または FILE 権限を持つ任意のユーザーとして MySQL コンソールにログインします。
- `local_infile` 変数を有効にして MySQL を起動します。
- `history_pk_prepare.sql` を実行して、古いテーブルの名前を変更し、新しいテーブルを作成します。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix-sql-scripts/mysql/history_pk_prepare.sql
```

- データのエクスポートとインポート

mysqlsh 経由で接続します。ソケット接続を使用している場合は、パスの指定が必要になる場合があります。

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

実行 (CSVPATH は必要に応じて変更してください):

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

- [移行後の手順](#) に従って、古いテーブルを削除します。

mysqlsh を使用しない MariaDB/MySQL 8.0+

このアップグレード方法は時間がかかるため、mysqlsh によるアップグレードが不可能な場合にのみ使用してください。

テーブルアップグレード

- root (推奨) または FILE 権限を持つ任意のユーザーとして MySQL コンソールにログインします。
- `local_infile` 変数を有効にして MySQL を起動します。
- `history_pk_prepare.sql` を実行して、古いテーブルの名前を変更し、新しいテーブルを作成します。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix-sql-scripts/mysql/history_pk_prepare.sql
```

停止したサーバーで移行

移行中のタイムアウトを避けるために、データを移行する前に `max_execution_time` を無効にする必要があります。

```
SET @@max_execution_time=0;

INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

[移行後の手順](#) に従って、古いテーブルを削除します。

実行中のサーバーで移行

インポート/エクスポートが有効になっているパスを確認します。


```
mysql> SELECT @@secure_file_priv;
+-----+
| @@secure_file_priv |
+-----+
| /var/lib/mysql-files/ |
+-----+
```

secure_file_priv 値がディレクトリへのパスである場合、そのディレクトリ内のファイルに対してエクスポート/インポートが実行されます。この場合、それに応じてクエリ内のファイルへのパスを編集するか、アップグレード時に secure_file_priv 値を空の文字列に設定します。

secure_file_priv 値が空の場合、エクスポート/インポートは任意の場所から実行できます。

secure_file_priv 値が NULL の場合は、エクスポートされたテーブルデータを含むパス (上記の例では'/var/lib/mysql-files/') に設定します。

詳細については、[MySQL ドキュメント](#) を参照してください。

エクスポート中のタイムアウトを避けるため、データをエクスポートする前に max_execution_time を無効にする必要があります。

```
SET @@max_execution_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\n'
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\n'
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\n'
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\n'
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\n'
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
```

移行後の手順 に従って、古いテーブルを削除します。

PostgreSQL

セッションがドロップされないように、エクスポートとインポートは tmux/screen で実行する必要があります。TimescaleDB を使用してインストールする場合は、このセクションをスキップして [PostgreSQL + TimescaleDB](#) に進んでください。

参照: [重要事項](#)

テーブルのアップグレード

- history_pk_prepare.sql を使用してテーブルの名前を変更します。

```
sudo -u zabbix psql zabbix < /usr/share/zabbix-sql-scripts/postgresql/history_pk_prepare.sql
```

停止したサーバーでの移行

- 現在の履歴をエクスポートして一時テーブルにインポートし、新しいテーブルにデータを重複を無視して挿入します。

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

```
INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

```
INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

```
INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

```
INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

INSERT パフォーマンスを改善するためのヒントを参照してください: [PostgreSQL: 膨大な量のデータの一括読み込み、チェックポイント距離および WAL の量](#)。

- **移行後の手順** に従って、古いテーブルを削除します。

実行中のサーバーでの移行

- 現在の履歴をエクスポートして一時テーブルにインポートし、重複を無視してデータを新しいテーブルに挿入します。

```
\copy history_old TO '/tmp/history.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history (
    itemid          bigint          NOT NULL,
    clock           integer         DEFAULT '0'      NOT NULL,
    value           DOUBLE PRECISION DEFAULT '0.0000' NOT NULL,
    ns              integer         DEFAULT '0'      NOT NULL
);
\copy temp_history FROM '/tmp/history.csv' DELIMITER ',' CSV
INSERT INTO history SELECT * FROM temp_history ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_uint_old TO '/tmp/history_uint.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_uint (
    itemid          bigint          NOT NULL,
    clock           integer         DEFAULT '0'      NOT NULL,
    value           numeric(20)     DEFAULT '0'      NOT NULL,
    ns              integer         DEFAULT '0'      NOT NULL
);
\copy temp_history_uint FROM '/tmp/history_uint.csv' DELIMITER ',' CSV
INSERT INTO history_uint SELECT * FROM temp_history_uint ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_str_old TO '/tmp/history_str.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_str (
    itemid          bigint          NOT NULL,
    clock           integer         DEFAULT '0'      NOT NULL,
    value           varchar(255)    DEFAULT ''       NOT NULL,
    ns              integer         DEFAULT '0'      NOT NULL
);
\copy temp_history_str FROM '/tmp/history_str.csv' DELIMITER ',' CSV
INSERT INTO history_str (itemid,clock,value,ns) SELECT * FROM temp_history_str ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_log_old TO '/tmp/history_log.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_log (
    itemid          bigint          NOT NULL,
    clock           integer         DEFAULT '0'      NOT NULL,
    timestamp       integer         DEFAULT '0'      NOT NULL,
    source          varchar(64)     DEFAULT ''       NOT NULL,
    severity        integer         DEFAULT '0'      NOT NULL,
    value           text            DEFAULT ''       NOT NULL,
    logeventid     integer         DEFAULT '0'      NOT NULL,
    ns              integer         DEFAULT '0'      NOT NULL
);
\copy temp_history_log FROM '/tmp/history_log.csv' DELIMITER ',' CSV
INSERT INTO history_log SELECT * FROM temp_history_log ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_text_old TO '/tmp/history_text.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_text (
    itemid          bigint          NOT NULL,
    clock           integer         DEFAULT '0'      NOT NULL,
    value           text            DEFAULT ''       NOT NULL,
    ns              integer         DEFAULT '0'      NOT NULL
);
\copy temp_history_text FROM '/tmp/history_text.csv' DELIMITER ',' CSV
INSERT INTO history_text SELECT * FROM temp_history_text ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

- **移行後の手順**に従って、古いテーブルを削除します。

PostgreSQL + TimescaleDB

セッションがドロップされないように、エクスポートとインポートは tmux/screen で実行する必要があります。アップグレード中は、Zabbix サーバーがダウンしている必要があります。

参照: [重要事項](#)

- `history_pk_prepare.sql` を使用してテーブルの名前を変更します。

```
sudo -u zabbix psql zabbix < /usr/share/zabbix-sql-scripts/postgresql/history_pk_prepare.sql
```

- 圧縮設定に基づいて、TimescaleDB ハイパーテーブル移行スクリプト (TSDB v2.x と v1.x の両方のバージョンと互換性があります) を実行します。
 - 圧縮が有効になっている場合 (デフォルトインストール) `database/postgresql/tsdb_history_pk_upgrade_with_compression` からスクリプトを実行します。

```
cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade_with_compression/history_pk.sql
cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade_with_compression/history_pk_uint
cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade_with_compression/history_pk_log.
cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade_with_compression/history_pk_str.
cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade_with_compression/history_pk_text
```

- 圧縮が無効になっている場合は、`database/postgresql/tsdb_history_pk_upgrade_no_compression` からスクリプトを実行します {bash}


```
cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade_no_compression
| sudo -u zabbix psql zabbix cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade
| sudo -u zabbix psql zabbix cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade
| sudo -u zabbix psql zabbix cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade
| sudo -u zabbix psql zabbix cat /usr/share/zabbix-sql-scripts/postgresql/tsdb_history_pk_upgrade
| sudo -u zabbix psql zabbix
```

INSERT のパフォーマンスを向上させるための [ヒント](#) も参照してください。

- [移行後の手順](#) に従って、古いテーブルを削除します。

Oracle

セッションがドロップされないように、エクスポートとインポートは `tmux/screen` で実行する必要があります。アップグレード中は、Zabbix サーバーがダウンしている必要があります。

参照: [重要事項](#)

テーブルのアップグレード

- Oracle Data Pump をインストールします ([Instant Client Tools パッケージ](#) で入手可能)。

パフォーマンスのヒントについては、Oracle Data Pump [ドキュメント](#) を参照してください。

- `history_pk_prepare.sql` を使用してテーブルの名前を変更します。

```
cd /usr/share/zabbix/zabbix-sql-scripts/database/oracle
sqlplus zabbix/password@oracle_host/service
sqlplus> @history_pk_prepare.sql
```

ヒストリテーブルの一括移行

- Data Pump 用のディレクトリを準備します。

Data Pump には、これらのディレクトリに対する読み取り権限と書き込み権限が必要です。

例:

```
mkdir -pv /export/history
chown -R oracle:oracle /export
```

- ディレクトリオブジェクトを作成し、Zabbix 認証に使用するユーザー (以下の例では 'zabbix') に、このオブジェクトへの読み取りと書き込みのアクセス許可を付与します。sysdba ロールの下で、次を実行します。

```
create directory history as '/export/history';
grant read,write on directory history to zabbix;
```

- テーブルをエクスポートします。N を目的のスレッド数に置き換えます。

```
expdp zabbix/password@oracle_host/service \
  DIRECTORY=history \
  TABLES=history_old,history_uint_old,history_str_old,history_log_old,history_text_old \
  PARALLEL=N
```

- テーブルをインポートします。N を目的のスレッド数に置き換えます。

```
impdp zabbix/password@oracle_host/service \
  DIRECTORY=history \
```

```
TABLES=history_uint_old \  
REMAP_TABLE=history_old:history,history_uint_old:history_uint,history_str_old:history_str,history_log_old:history_log \  
data_options=SKIP_CONSTRAINT_ERRORS table_exists_action=APPEND PARALLEL=N CONTENT=data_only
```

- **移行後の手順**に従って、古いテーブルを削除します。

ヒストリテーブルの個別移行

- ヒストリテーブルごとに Data Pump 用のディレクトリを用意してください。Data Pump には、これらのディレクトリに対する読み取り権限と書き込み権限が必要です。

例:

```
mkdir -pv /export/history /export/history_uint /export/history_str /export/history_log /export/history_text \  
chown -R oracle:oracle /export
```

- ディレクトリオブジェクトを作成し、Zabbix 認証に使用するユーザー (以下の例では'zabbix') に、このオブジェクトへの読み取りおよび書き込み権限を付与します。sysdba ロールの下で、次を実行します。

```
create directory history as '/export/history'; \  
grant read,write on directory history to zabbix; \  
  
create directory history_uint as '/export/history_uint'; \  
grant read,write on directory history_uint to zabbix; \  
  
create directory history_str as '/export/history_str'; \  
grant read,write on directory history_str to zabbix; \  
  
create directory history_log as '/export/history_log'; \  
grant read,write on directory history_log to zabbix; \  
  
create directory history_text as '/export/history_text'; \  
grant read,write on directory history_text to zabbix;
```

- 各テーブルをエクスポートおよびインポートします。N を目的のスレッド数に置き換えます。

```
expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history TABLES=history_old PARALLEL=N \  
impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history TABLES=history_old REMAP_TABLE=history_old:history \  
expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_uint TABLES=history_uint_old PARALLEL=N \  
impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_uint TABLES=history_uint_old REMAP_TABLE=history_uint_old:history_uint \  
expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_str TABLES=history_str_old PARALLEL=N \  
impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_str TABLES=history_str_old REMAP_TABLE=history_str_old:history_str \  
expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_log TABLES=history_log_old PARALLEL=N \  
impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_log TABLES=history_log_old REMAP_TABLE=history_log_old:history_log \  
expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_text TABLES=history_text_old PARALLEL=N \  
impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_text TABLES=history_text_old REMAP_TABLE=history_text_old:history_text
```

- **移行後の手順**に従って、古いテーブルを削除します。

移行後の手順

すべてのデータベースで、移行完了後に次の手順を実施します。

- すべてが想定どおりに機能することを確認します。
- 古いテーブルを削除:

```
DROP TABLE history_old; \  
DROP TABLE history_uint_old; \  
DROP TABLE history_str_old;
```

```
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

4 データベースへの安全な接続

概要

このセクションでは、以下の組み合わせでの安全な TLS 接続のため、Zabbix セットアップ手順と構成例を提供します。

データベース	Zabbix コンポーネント
MySQL	Zabbix Web インターフェース、Zabbix サーバー、Zabbix プロキシ
PostgreSQL	Zabbix Web インターフェース、Zabbix サーバー、Zabbix プロキシ

DBMS 内で接続暗号化を設定するには、詳細について以下のベンダー公式ドキュメントを参照してください：

- [MySQL](#): ソースおよびレプリカレプリケーションデータベースサーバー
- [MySQL](#): グループレプリケーションなどのデータベースサーバー
- [PostgreSQL](#): 暗号化オプション

すべての例は、AlmaLinux 8 の公式リポジトリから入手できる MySQL CE (8.0) および PgSQL (13) の GA リリースに基づいています。

要件

暗号化を設定するには、次が必要です。

- デベロッパーサポートされている OS と OpenSSL1.1.X 以上または OpenSSL の代用品。

Note:

特に新規インストールの場合は、サポート終了した OS は避けることをお勧めします。

- デベロッパーが提供する公式リポジトリからインストールおよび保守されるデータベースエンジン (RDBMS) やオペレーティングシステムには、暗号化サポートが実装されていない古いバージョンのデータベースソフトウェアが付属していることがよくあります。たとえば、RHEL7 ベースのシステムでの PostgreSQL9.2 や暗号化サポートのない MariaDB5.5 などです。

専門用語

このオプションを設定すると、Zabbix サーバー/プロキシやフロントエンドからのデータベース接続には TLS 接続を使用します。

- `required` - ID チェックなしでトランスポートモードとして TLS を使用して接続します。
- `verify_ca` - TLS を使用して接続し、証明書を検証します。
- `verify_full` - TLS を使用して接続し、証明書を検証し、DBHost によって指定されたデータベース ID (CN) がその証明書と一致することを確認します。

Zabbix の設定

Web インターフェースからデータベース

Web インターフェースのインストール中に、データベースへの安全な接続を設定できます。

- トランスポート暗号化を有効にするには **データベース接続設定** ステップでデータベースの TLS 暗号化チェックボックスをオンにします。
- 証明書による暗号化を有効にするには、TLS 暗号化フィールドがオンになっている場合に表示されるデータベースの証明書確認チェックボックスをオンにします。

Note:

MySQL の場合、データベースホストが localhost に設定されていると、ソケットファイル (Unix の場合) または共有メモリ (Windows の場合) を使用する接続を暗号化できないため、データベースの TLS 暗号化チェックボックスが無効になります。PostgreSQL の場合、データベースホストフィールドの値がスラッシュで始まるか、フィールドが空の場合、TLS 暗号化チェックボックスは無効になります。

次のパラメータは、証明書モードの TLS 暗号化で使用可能になります。(両方のチェックボックスがマークされている場合)

パラメータ	説明
データベースの TLS CA ファイル	有効な TLS 認証局 (CA) ファイルへのフルパスを指定します。


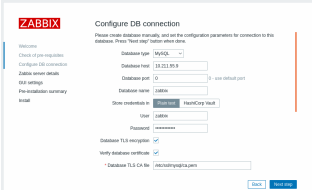
パラメータ	説明
データベースの TLS キーファイル データベースの TLS 証明書ファイル データベースのホスト検証	有効な TLS キーファイルへのフルパスを指定します。 有効な TLS 証明書ファイルへのフルパスを指定します。 このチェックボックスをオンにすると、ホスト検証がアクティブになります。 PHP MySQL ライブラリではピア証明書の検証手順をスキップできないため、MySQL では無効になっています。
データベースの TLS 暗号化方式リスト	有効な暗号のカスタムリストを指定します。暗号リストの形式は、OpenSSL 標準に準拠している必要があります。 MySQL でのみ使用できます。

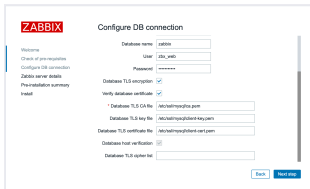
Attention:

TLS パラメータは有効なファイルを指している必要があります。存在しないファイルまたは無効なファイルを指している場合は、認証エラーが発生します。証明書ファイルが書き込み可能である場合、Web インターフェースはシステム情報レポートに「データベースの TLS 証明書ファイルは読み込み専用にしてください」という警告を表示します。(PHP ユーザーが証明書の所有者である場合にのみ表示されます)。パスワードで保護された証明書はサポートされていません。

ユースケース

Zabbix フロントエンドは GUI インターフェースを使用してオプションを定義します:required,verify_ca, verify_full. インストールウィザードの DB 接続設定で必要なオプションを指定します。これらのオプションは設定ファイル (zabbix.conf.php) に次のようにマップされます。

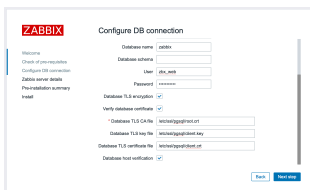
GUI の設定	設定ファイル	説明	結果
	<pre>... // Used for TLS connection. \$DB['ENCRYPTION'] = true; \$DB['KEY_FILE'] = ""; \$DB['CERT_FILE'] = ""; \$DB['CA_FILE'] = ""; \$DB['VERIFY_HOST'] = false; \$DB['CIPHER_LIST'] = ""; ... </pre>	データベースの TLS 暗号化を確認するデータベース証明書の確認をオフのままにします	'required' モードが有効
	<pre>... \$DB['ENCRYPTION'] = true;\n \$DB['KEY_FILE'] = "";\n \$DB['CERT_FILE'] = "";\n \$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';\n \$DB['VERIFY_HOST'] = false;\n \$DB['CIPHER_LIST'] = "";\n ... </pre>	<ol style="list-style-type: none"> データベース TLS 暗号化とデータベース証明書の確認をチェックします。 データベース TLSCA ファイルへのパスを指定します。 	'verify_ca' モードが有効



```
...
// Used for TLS connection with strictly
defined Cipher list.
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '<key_file_path>';
$DB['CERT_FILE'] = '<key_file_path>';
$DB['CA_FILE'] = '<key_file_path>';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '<cipher_list>';
...
```

Or:

```
...
// Used for TLS connection without Cipher
list defined - selected by MySQL server
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '<key_file_path>';
$DB['CERT_FILE'] = '<key_file_path>';
$DB['CA_FILE'] = '<key_file_path>';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = "";
...
```



```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '<key_file_path>';
$DB['CERT_FILE'] = '<key_file_path>';
$DB['CA_FILE'] = '<key_file_path>';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

1. データベース TLS 暗号化とデータベース証明書の確認をチェックします。
 2. データベース TLS キーファイルへのパスを指定します。
 3. データベース TLSCA ファイルへのパスを指定します。
 4. データベース TLS 証明書ファイルへのパスを指定します
 6. TLS 暗号リストを指定します (オプション)
- MySQL
で'verify_full'モードが有効

1. データベース TLS 暗号化とデータベース証明書の確認をチェックします。
 2. データベース TLS キーファイルへのパスを指定します。
 3. データベース TLSCA ファイルへのパスを指定します
 4. データベース TLS 証明書ファイルへのパスを指定します
 6. データベースホストの検証をチェックします
- PostgreSQL
で'verify_full'モードが有効

関連項目：[Encryption configuration examples for MySQL](#), [Encryption configuration examples for PostgreSQL](#).

Zabbix サーバー/プロキシの設定

データベースへの安全な接続は、Zabbixサーバーやプロキシの設定ファイルそれぞれのパラメータで設定できます。

設定

無し

1. DBTLSConnect=required をセット

結果

暗号化せずにデータベースに接続します。
サーバー/プロキシはデータベースへTLS 接続します。暗号化されていない接続は許可されていません。

設定	結果
1. DBTLSConnect=verify_ca をセット 2. DBTLSCAFile に TLS 認証局ファイルを指定	サーバー/プロキシは、データベース証明書を確認した後、データベースへ TLS 接続します。
1. DBTLSConnect=verify_full をセット 2. DBTLSCAFile に TLS 認証局ファイルを指定	サーバー/プロキシは、データベース証明書とデータベースホスト ID を確認した後、データベースへ TLS 接続します。
1. DBTLSCAFile に TLS 認証局ファイルを指定 2. DBTLSCertFile にクライアントの公開鍵証明書ファイルを指定 3. DBTLSKeyFile にクライアントの秘密鍵ファイルを指定 1.1. DBTLSCipher にクライアントが TLS1.2 までの TLS プロトコルを使用した接続を許可する暗号化暗号のリストをセット または	サーバー/プロキシは、データベースへの接続中にクライアント証明書を提供します。
DBTLSCipher13 にクライアントが TLS1.3 プロトコルを使用した接続に許可する暗号化暗号のリストをセット	(MySQL) TLS 接続は、提供されたリストの暗号を使用して行われます。 (PostgreSQL) このオプションを設定すると、エラーと見なされます。

1 MySQL の暗号化設定

概要

このセクションは CentOS8.2 および MySQL8.0.21 の暗号化設定の例をいくつか示し、データベースへの接続を暗号化するためのクイックスタートガイドとして使用できます。

Attention:

MySQL ホストが localhost に設定されている場合、暗号化オプションは使用できません。この場合、Zabbix フロントエンドとデータベース間の接続はソケットファイル (Unix の場合) または共有メモリ (Windows の場合) を使用し、暗号化できません。

Note:

暗号化の組み合わせのリストは、このページにリストされているものに限定されません。利用できる組み合わせはもっとたくさんあります。

前提条件

[公式リポジトリ](#)から MySQL データベースをインストールします。

MySQL リポジトリの使用方法詳細については[MySQL のドキュメント](#)を参照してください。

MySQL サーバーは、自己署名証明書を使用して安全な接続を受け入れる準備ができています。

暗号化された接続を使用しているユーザーを確認するには、次のクエリを実行します (パフォーマンススキーマをオンにする必要があります) :

```
mysql> SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, pr
FROM performance_schema.status_by_thread AS sbt
JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
ORDER BY tls_version;
```

Required モード

MySQL の設定

最新バージョンのデータベースは、'required'暗号化モード. ですぐに使用できます。サーバー側の証明書は、初期設定と起動後に作成されます。

主要コンポーネントのユーザーとロールを作成します :

```
mysql> CREATE USER
'zbx_srv'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>',
'zbx_web'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>'
REQUIRE SSL
```



```
PASSWORD HISTORY 5;
```

```
mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';
```

```
mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_srv_role';
```

```
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';
```

```
mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
```

```
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';
```

```
mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
```

```
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

X.509 プロトコルは ID のチェックには使用されませんが、ユーザーは暗号化された接続のみを使用するように構成されていることに注意してください。ユーザーの構成の詳細については、[MySQL のドキュメント](#)を参照してください。

接続確認のため次を実行（ソケット接続を使用して安全な接続をテストすることはできません）：

```
$ mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

現在のステータスと利用可能な暗号スイートを確認：

```
mysql> status
```

```
-----  
mysql Ver 8.0.21 for Linux on x86_64 (MySQL Community Server - GPL)
```

```
Connection id: 62
```

```
Current database:
```

```
Current user: zbx_srv@bfdb.local
```

```
SSL: Cipher in use is TLS_AES_256_GCM_SHA384
```

```
mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
```

```
***** 1. row *****
```

```
Variable_name: Ssl_cipher_list
```

```
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256
```

```
1 row in set (0.00 sec)
```

```
ERROR:
```

```
No query specified
```

フロントエンド

Zabbix フロントエンドとデータベース間の接続でトランスポートのみの暗号化を有効にするには：

- データベースの TLS 暗号化をチェック
- データベース証明書の確認をチェックしない

サーバ

サーバーとデータベース間の接続でトランスポートのみの暗号化を有効にするには、`/etc/zabbix/zabbix_server.conf` を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Verify CA モード

必要な `MySQLCA` を Zabbix フロントエンドサーバーにコピーし、Web サーバーがこのファイルを読み取れるように適切な権限を割り当てます。

Note:

古い MySQL ライブラリが原因で、Verify CA モードは SLES12 および RHEL7 で機能しません。

フロントエンド

Zabbix フロントエンドとデータベース間接続の証明書検証による暗号化を有効にするには：

- データベース TLS 暗号化とデータベース証明書の確認をチェック
- データベース TLSCA ファイルへのパスを指定

または/etc/zabbix/web/zabbix.conf.php で設定 :

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

コマンドラインツールを使用してユーザーのトラブルシューティングを行い、必要なユーザーが接続できるかどうかを確認します :

```
$ mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

サーバ

Zabbix サーバーとデータベース間の証明書検証による暗号化接続を有効にするには/etc/zabbix/zabbix_server.conf を設定します :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Verify Full モード

MySQL の設定

MySQL CE サーバー構成オプション (/etc/my.cnf.d/server-tls.cnf) を次のように設定します :

```
[mysqld]
...
#この例では、キーはMySQLCEdatadirディレクトリにあります
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
```

...

MySQL CE サーバーとクライアント (Zabbix フロントエンド) のキーは、MySQL CE のドキュメントに従って手動で作成する必要があります : [MySQL を使用した SSL および RSA 証明書とキーの作成](#)または[openssl を使用した SSL 証明書とキーの作成](#)

Attention:

Zabbix フロントエンドは DNS 名を使用してデータベースまたはデータベースホストの IP アドレスと通信するため、MySQL サーバー証明書には FQDN 名に設定された共通名フィールドが含まれている必要があります。

MySQL ユーザーの作成 :

```
mysql> CREATE USER
'zbx_srv'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>',
'zbx_web'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>'
REQUIRE X509
PASSWORD HISTORY 5;
```

作成したユーザーでログインできるかを確認します。:

```
$ mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=
フロントエンド
```

Zabbix フロントエンドとデータベース間の接続を完全に検証して暗号化を有効にするには :

- データベース TLS 暗号化とデータベース証明書の確認をチェック
- データベース TLS キーファイルへのパスを指定
- データベース TLSCA ファイルへのパスを指定
- データベース TLS 証明書ファイルへのパスを指定

データベースホストの検証がチェックされ、グレイアウトされていることに注意してください。MySQL ではこの手順をスキップできません。

Warning:

フロントエンドとサーバーが両方でサポートされているものから必要なものをネゴシエートできるように、暗号リストは空にする必要があります。

または/etc/zabbix/web/zabbix.conf.php で設定します :

```
...
// Used for TLS connection with strictly defined Cipher list.
```

```

$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AE
...
// or
...
// Used for TLS connection without Cipher list defined - selected by MySQL server
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...

```

サーバー

Zabbix サーバーとデータベース間の接続を完全に検証して暗号化を有効にするには、`/etc/zabbix/zabbix_server.conf` を設定します：

```

...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...

```

2 PostgreSQL の暗号化設定

概要

このセクションでは、CentOS8.2 および PostgreSQL13 の暗号化設定例をいくつか示します。

Note:

Database host フィールドの値がスラッシュで始まるか、空の場合、Zabbix フロントエンドと PostgreSQL 間の接続を暗号化できません (GUI のパラメーターが無効になっています)

前提条件

[公式リポジトリ](#) を使用して PostgreSQL データベースをインストールします。

PostgreSQL は、すぐに TLS 接続を受け入れるように構成されていません。[postgresql.conf](#) を使用した証明書の準備および `ph_hba.conf` を介した [ユーザーアクセス制御](#) については、PostgreSQL のドキュメントの指示に従ってください。

デフォルトでは、PostgreSQL ソケットはローカルホストにバインドされています。これは、ネットワークリモート接続が実際のネットワークインターフェイスでリッスンできるようにするためです。

すべての **モード** の PostgreSQL 設定は次のようになります。

`/var/lib/pgsql/13/data/postgresql.conf:`

```

...
ssl = on
ssl_ca_file = 'root.crt'
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'
ssl_prefer_server_ciphers = on
ssl_min_protocol_version = 'TLSv1.3'
...

```

アクセス制御については `/var/lib/pgsql/13/data/pg_hba.conf` を調整します。

```

...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...

```

Required モード

フロントエンド

Zabbix フロントエンドとデータベース間の接続でトランスポートのみの暗号化を有効にするには：

- データベースの TLS 暗号化をチェックします
- データベース証明書の確認をチェックしません

ZABBIX

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type: PostgreSQL

Database host: 10.211.55.9

Database port: 0 (0 - use default port)

Database name: zabbix

Database schema:

Store credentials in: Plain text (selected), HashiCorp Vault

User: zabbix

Password:

Database TLS encryption:

Verify database certificate:

Buttons: Back, Next step

サーバー

サーバーとデータベース間の接続に対してトランスポートのみの暗号化を有効にするには/etc/zabbix/zabbix_server.conf を設定します：

```

...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...


```

Verify CA モード

フロントエンド

Zabbix フロントエンドとデータベース間の接続の認証局検証による暗号化を有効にするには：

- データベース TLS 暗号化とデータベース証明書の確認をチェック
- データベース TLS キーファイルへのパスを指定
- データベース TLSCA ファイルへのパスを指定
- データベース TLS 証明書ファイルへのパスを指定



Configure DB connection

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Database name

Database schema

User

Password

Database TLS encryption

Verify database certificate

* Database TLS CA file

Database TLS key file

Database TLS certificate file

Database host verification

または/etc/zabbix/web/zabbix.conf.php: で設定 :

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続の証明書検証による暗号化を有効にするには、/etc/zabbix/zabbix_server.conf を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Verify Full モード

フロントエンド

Zabbix フロントエンドとデータベース間接続の証明書とデータベースホスト ID 検証による暗号化を有効にするには :

- データベース TLS 暗号化とデータベース証明書の確認をチェック
- データベース TLS キーファイルへのパスを指定
- データベース TLSCA ファイルへのパスを指定
- データベース TLS 証明書ファイルへのパスを指定
- データベースホストの検証をチェック

または/etc/zabbix/web/zabbix.conf.php で設定 :

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続の証明書とデータベースホスト ID 検証による暗号化を有効にするには/etc/zabbix/zabbix_server.conf を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

5 TimescaleDB のセットアップ

概要

Zabbix は TimescaleDB をサポートしています。これは、データを時間ベースのチャンクに自動的に分割する PostgreSQL ベースのデータベースソリューションであり、大規模なパフォーマンスの高速化をサポートします。

Warning:
現在 TimescaleDB は Zabbix プロキシではサポートされていません。

このページの手順は、TimescaleDB データベースの作成、または既存の PostgreSQL テーブルから TimescaleDB への移行に使用できません。

設定

TimescaleDB 拡張機能がデータベースサーバーに既にインストールされていることを前提としています ([インストール手順](#)を参照)

次のコマンドを実行して、特定の DB に対して TimescaleDB 拡張機能を有効にする必要があります。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

このコマンドを実行するには、データベース管理者権限が必要です。

Note:

'public' 以外のデータベーススキーマを使用する場合は、上記のコマンドに SCHEMA 句を追加する必要があります。例：
'echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix'

次に、database/postgresqlにある timescaledb.sql スクリプトを実行します。新規インストールの場合、通常の PostgreSQL データベースが初期スキーマ/データで作成された後にスクリプトを実行する必要があります ([データベースの作成](#) を参照)

```
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb.sql | sudo -u zabbix psql zabbix
```

既存の履歴およびトレンドデータの移行には、時間がかかる場合があります。移行中は、Zabbix サーバーとフロントエンドを停止する必要があります。

timescaledb.sql スクリプトは、次のハウスキーピングパラメータを設定します。

- Override item history period
- Override item trend period

履歴とトレンドに分割されたハウスキーピングを使用するには、これらのオプションを両方とも有効にする必要があります。履歴のみまたはトレンドのみのオーバーライドを個別に有効にすることもできます。

PostgreSQL バージョン 10.2 以降および TimescaleDB バージョン 1.5 以降の場合、timescaledb.sql スクリプトは 2 つの追加パラメータを設定します。

- Enable compression
- Compress records older than 7 days

圧縮は、アイテムの履歴の保存期間設定を上書きとアイテムのトレンドの保存期間設定を上書きのオプションの両方が有効になっている場合にのみ使用できます。上書き設定が無効で、テーブルに圧縮されたチャンクがある場合、housekeeper はこれらのテーブルからデータを削除せず、[データの保存期間](#) および [システム情報](#) セクションの管理画面に誤った構成に関する警告が表示されません。

これらのパラメータはすべて、インストール後に [管理](#) → [一般](#) → [ハウスキーピング](#) で変更できます。

Note:

TimescaleDB が提供する timescaledb-tune ツールを実行して、postgresql.conf の PostgreSQL 構成パラメータを最適化することができます。

TimescaleDB 圧縮

ネイティブの TimescaleDB 圧縮は、PostgreSQL バージョン 10.2 以降では Zabbix 5.0 以降、TimescaleDB によって管理されるすべての Zabbix テーブルでは TimescaleDB バージョン 1.5 以降でサポートされています。TimescaleDB へのアップグレードまたは移行では大きなテーブルの初期圧縮に時間がかかる場合があります。

圧縮は "timescale" Timescale Community ライセンスでサポートされており、"apache" Apache 2.0 ライセンスではサポートされていないことに注意してください。Zabbix 6.0.7 以降、Zabbix は圧縮がサポートされているかどうかを検出します。サポートされていない場合、警告メッセージが Zabbix サーバーログに書き込まれ、ユーザーはフロントエンドで圧縮を有効にできません。

Note:

ユーザーは、圧縮を使用する前に [TimescaleDB 圧縮ドキュメント](#) を読解することをお勧めします。

圧縮によって課せられる特定の制限があることに注意してください。具体的には次のとおりです。

- 圧縮されたチャンクの変更 (挿入、削除、更新) は許可されません
- 圧縮されたテーブルのスキーマの変更は許可されません。

圧縮設定は、Zabbix フロントエンドの [管理](#) → [一般設定](#) → [データの保存期間](#) セクションの履歴とトレンドの圧縮ブロックで変更できます。

パラメータ	デフォルト	コメント
圧縮を有効にする	有効	<p>チェックボックスをオンまたはオフにしても、圧縮はすぐにはアクティブ化/非アクティブ化されません。圧縮はハウスキーパーによって処理されるため、変更は最大 2 倍の「ハウスキーピング頻度」時間 (<code>zabbix_server.conf</code> で設定) で有効になります。</p> <p>後圧縮を無効にすると、圧縮期間に入る新しいチャンクは圧縮されません。ただし、以前に圧縮されたデータはすべて圧縮されたままになります。以前に圧縮されたチャンクを圧縮解除するには、TimescaleDB ドキュメントの手順に従ってください。</p> <p>TimescaleDB をサポートする古いバージョンの Zabbix からアップグレードする場合は、圧縮はデフォルトでは有効になりません。</p>
Compress records older than	7d	<p>このパラメーターは 7 日未満にすることはできません。</p> <p>圧縮されたチャンクの不変性により、この値よりも古いすべての遅延データ (プロキシによって遅延されたデータなど) は削除されます。</p>

6 Elasticsearch のセットアップ

Attention:

Elasticsearch のサポートは試験的です

Zabbix はデータベースの代わりに Elasticsearch を使用した履歴データの保存をサポートしています。ユーザーは、互換性のあるデータベースと Elasticsearch の間で履歴データの保存場所を選択できます。このセクションで説明するセットアップ手順は、Elasticsearch バージョン 7.X に適用できます。以前のバージョンまたは新しいバージョンの Elasticsearch を使用すると、一部の機能が意図したとおりに機能しない可能性があります。

Warning:

すべての履歴データが Elasticsearch に保存されている場合、トレンドは計算もデータベースにも保存されません。トレンドが計算および保存されていないため、履歴保存期間を延長する必要がある場合があります。

設定

関連するすべての要素間の適切な通信を確保するには、サーバー構成ファイルとフロントエンド構成ファイルのパラメーターが適切に構成されていることを確認してください。

Zabbix サーバーとフロントエンド

更新するパラメーターを含む Zabbix サーバー構成ファイルの草案：

```
### Option: HistoryStorageURL
# History storage HTTP[S] URL.
#
# Mandatory: no
# Default:
# HistoryStorageURL=
### Option: HistoryStorageTypes
# Comma separated list of value types to be sent to the history storage.
#
# Mandatory: no
# Default:
# HistoryStorageTypes=uint,dbl,str,log,text
```

Zabbix サーバー構成ファイルに入力するパラメーター値の例：

```
HistoryStorageURL=http://test.elasticsearch.lan:9200
HistoryStorageTypes=str,log,text
```

この設定により、Zabbix サーバーは数値タイプの履歴値を対応するデータベースに保存し、テキスト履歴データを Elasticsearch に保存します。

Elasticsearch は次のアイテムタイプをサポートしています：

```
uint,dbl,str,log,text
```

サポートされているアイテムタイプの説明：

アイテム値のタイプ	データベーステーブル	Elasticsearch でのタイプ
符号なし数値型	history_uint	uint
浮動小数点数型	history	dbl
固定長文字列	history_str	str
ログ	history_log	log
テキスト	history_text	text

更新するパラメータを含む Zabbix フロントエンド設定ファイル (conf/zabbix.conf.php) の草案：

```
// Elasticsearch url (can be string if same url is used for all types).
$HISTORY['url'] = [
    'uint' => 'http://localhost:9200',
    'text' => 'http://localhost:9200'
];
// Value types stored in Elasticsearch.
$HISTORY['types'] = ['uint', 'text'];
```

Zabbix フロントエンド設定ファイルに入力するパラメータ値の例：

```
$HISTORY['url'] = 'http://test.elasticsearch.lan:9200';
$HISTORY['types'] = ['str', 'text', 'log'];
```

この設定により、Elasticsearch に Text、Character、および Log の履歴値が強制的に保存されます。

また、すべてが正しく機能するように conf/zabbix.conf.php で \$HISTORY をグローバルにする必要があります。(方法については conf/zabbix.conf.php.example を参照してください)

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;
```

Elasticsearch のインストールとマッピングの作成

最後 2 つのステップは、Elasticsearch 自体のインストールとマッピングプロセスを作成することです。

Elasticsearch をインストールするには [Elasticsearch インストールガイド](#) を参照してください。

Note:

マッピングは Elasticsearch のデータ構造です (データベースのテーブルに似ています)。すべての履歴データ型のマッピングは、database/elasticsearch/elasticsearch.map で利用できます。

Warning:

マッピングの作成は必須です。指示に従ってマッピングを作成しないと、一部の機能が機能しなくなります。

text タイプのマッピングを作成するには、Elasticsearch に次のリクエストを送信します。

```
curl -X PUT \
  http://your-elasticsearch.here:9200/text \
  -H 'content-type:application/json' \
  -d '{
    "settings": {
      "index": {
        "number_of_replicas": 1,
        "number_of_shards": 5
      }
    },
    "mappings": {
      "properties": {
        "itemid": {
          "type": "long"
        },
        "clock": {
          "format": "epoch_second",
          "type": "date"
        }
      }
    }
  }'
```

```

    "value": {
      "fields": {
        "analyzed": {
          "index": true,
          "type": "text",
          "analyzer": "standard"
        }
      },
      "index": false,
      "type": "text"
    }
  }
}'

```

Character および Log 履歴値に対しても、対応するタイプを修正のうえで同様のリクエストを実行する必要があります。

Note:

Elasticsearch を使用するには、追加情報について [要件ページ](#) を参照してください。

Note:

ハウスキーパーは Elasticsearch からデータを削除しません

複数の日付ベースのインデックスに履歴データを保存する

このセクションでは、パイプラインと取り込みノードを操作するために必要な追加の手順について説明します。

まず、インデックスのテンプレートを作成する必要があります。

次の例は、uint テンプレートを作成するためのリクエストです：

```

curl -X PUT \
  http://your-elasticsearch.here:9200/_template/uint_template \
  -H 'content-type:application/json' \
  -d '{
    "index_patterns": [
      "uint*"
    ],
    "settings": {
      "index": {
        "number_of_replicas": 1,
        "number_of_shards": 5
      }
    },
    "mappings": {
      "properties": {
        "itemid": {
          "type": "long"
        },
        "clock": {
          "format": "epoch_second",
          "type": "date"
        },
        "value": {
          "type": "long"
        }
      }
    }
  }'

```

他のテンプレートを作成するには、URL を変更し（最後の部分はテンプレートの名前です）、「index_patterns」フィールドをインデックス名と一致するように変更して、データベース/elasticsearch/elasticsearch.map から取得できる有効なマッピングを設定する必要があります。

たとえば、次のコマンドを使用して、テキストインデックスのテンプレートを作成できます。

```

curl -X PUT \
  http://your-elasticsearch.here:9200/_template/text_template \
  -H 'content-type:application/json' \
  -d '{
    "index_patterns": [
      "text*"
    ],
    "settings": {
      "index": {
        "number_of_replicas": 1,
        "number_of_shards": 5
      }
    },
    "mappings": {
      "properties": {
        "itemid": {
          "type": "long"
        },
        "clock": {
          "format": "epoch_second",
          "type": "date"
        },
        "value": {
          "fields": {
            "analyzed": {
              "index": true,
              "type": "text",
              "analyzer": "standard"
            }
          },
          "index": false,
          "type": "text"
        }
      }
    }
  }'

```

これは、Elasticsearch が自動的に作成されたインデックスに有効なマッピングを設定できるようにするために必要です。次に、パイプライン定義を作成する必要があります。パイプラインは、データをインデックスに入れる前の、ある種のデータの前処理です。次のコマンドを使用して作成できます uint インデックスのパイプライン：

```

curl -X PUT \
  http://your-elasticsearch.here:9200/_ingest/pipeline/uint-pipeline \
  -H 'content-type:application/json' \
  -d '{
    "description": "daily uint index naming",
    "processors": [
      {
        "date_index_name": {
          "field": "clock",
          "date_formats": [
            "UNIX"
          ],
          "index_name_prefix": "uint-",
          "date_rounding": "d"
        }
      }
    ]
  }'

```

ユーザーは、丸めパラメーター ("date_rounding") を変更して、特定のインデックスローテーション期間を設定できます。他のパイプラインを作成するには、URL (最後の部分はパイプラインの名前) を変更し "index_name_prefix" フィールドをインデックス名と一致するように変更する必要があります。

[Elasticsearch ドキュメント](#) も参照してください

さらに、履歴データを複数の日付ベースのインデックスに保存することも、Zabbix サーバー構成の新しいパラメーターで有効にする必要があります。

```
### Option: HistoryStorageDateIndex
# Enable preprocessing of history values in history storage to store values in different indices based on
# 0 - disable
# 1 - enable
#
# Mandatory: no
# Default:
# HistoryStorageDateIndex=0
```

トラブルシューティング

次の手順は Elasticsearch のセットアップに関する問題のトラブルシューティングに役立つ場合があります。

1. マッピングが正しいかどうかを確認します (http://localhost:9000/_mapping などの必要なインデックス URL への GET リクエスト)。
2. シャードが失敗状態になっていないかどうかを確認します (Elasticsearch の再起動が役立つはずですが)。
3. Elasticsearch の設定を確認します。Zabbix フロントエンドホストと Zabbix サーバーホストからのアクセスを許可する必要があります。
4. Elasticsearch ログを確認します。

それでもインストールで問題が発生する場合は、このリストのすべての情報 (マッピング、エラーログ、構成、バージョンなど) を含むバグレポートを作成してください。

7 イベント、アイテム値、トレンドのリアルタイムエクスポート

概要

トリガーイベント、アイテム値、トレンドのリアルタイムエクスポートを改行で区切られた JSON 形式で構成することができます。

エクスポートはファイルに行われ、エクスポートファイルの各行は JSON オブジェクトです。値のマッピングは適用されません。

エラー (データをエクスポートファイルに書き込めない、エクスポートファイルの名前を変更できない、または名前を変更した後に新しいデータを作成できない) の場合、データアイテムは削除され、エクスポートファイルに書き込まれず、Zabbix データベースにのみ書き込まれます。エクスポートファイルへの書き込み問題が解決すると、データの書き込みが再開されます。

エクスポートされる情報の詳細については、[エクスポートプロトコルページ](#)を参照してください。

データの受信後、サーバーがデータをエクスポートする前にホスト/アイテムが削除された場合、ホスト/アイテムにメタデータ (ホストグループ、ホスト名、アイテム名) を含めることができないことに注意してください。

設定

トリガーイベント、アイテム値、トレンドのリアルタイムエクスポートは、エクスポートファイルのディレクトリを指定することで構成されます。サーバー設定の `ExportDir` パラメーターを参照してください。

他、2つのパラメータを使用できます。

- `ExportFileSize` 個々のエクスポートファイルの最大サイズを設定するために使用します。プロセスがファイルに書き込む必要がある場合、最初にファイルのサイズをチェックし、設定されたサイズ制限を超える場合は、名前に `.old` を追加してファイルの名前を変更し、元の名前の新しいファイルを作成します。

Attention:

データを書き込むプロセスごとにファイルが作成されます。(約 4~30 ファイル) エクスポートファイルあたりのデフォルトサイズは 1G であるため、ディスク容量が急速に消費される可能性があります。

- `ExportType` エクスポートするエンティティタイプ (イベント、履歴、トレンド) を指定できます。

8 Zabbix 用 Nginx のセットアップに関するディストリビューション固有の注意事項

RHEL

Nginx は EPEL でのみ利用可能です

```
# yum -y install epel-release
```

SLES 12

SUSE Linux Enterprise Server 12 では、Nginx をインストールする前に Nginx リポジトリを追加する必要があります。

```
zypper addrepo -G -t yum -c 'http://nginx.org/packages/sles/12' nginx
```

また php-fpm を設定する必要があります

```
cp /etc/php5/fpm/php-fpm.conf{.default,}
sed -i 's/user = nobody/user = wwwrun;/ s/group = nobody/group = www/' /etc/php5/fpm/php-fpm.conf
```

SLES 15

SUSE Linux Enterprise Server 15 では、php-fpm を設定する必要があります。

```
cp /etc/php7/fpm/php-fpm.conf{.default,}
cp /etc/php7/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun;/ s/group = nobody/group = www/' /etc/php7/fpm/php-fpm.d/www.conf
```

9 root ユーザーでのエージェントの実行

バージョン **5.0.0** 以降、[公式パッケージ](#)の Zabbix エージェントの systemd サービスファイルが更新され、ユーザーとグループのデイレクティブが明示的に含まれるようになりました。どちらも zabbix に設定されています。

これは、zabbix_agentd.conf ファイルを介してどのユーザー Zabbix エージェントを実行するかを構成する古い機能がバイパスされ、エージェントが常に systemd サービスファイルで指定されたユーザーとして実行されることを意味します。

この新しい動作を上書きして旧来のように root で実行するには、次の内容で /etc/systemd/system/zabbix-agent.service.d/override ファイルを作成します。

```
[Service]
User=root
Group=root
```

次にデーモンをリロードし、zabbix-agent サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Zabbix エージェント 2 の場合、これで実行されるユーザーを決定します。

古いエージェントの場合、上記だけでは zabbix_agentd.conf ファイルでのユーザー設定する機能のみを再度有効にするだけなので、zabbix エージェントを root として実行するには、エージェント [構成ファイル](#) を編集し、User=root と AllowRoot=1 オプションを追加で指定する必要があります。

Zabbix agent

To override the default user and group for Zabbix agent, run:

```
systemctl edit zabbix-agent
```

Then, add the following content:

```
[Service]
User=root
Group=root
```

Reload daemons and restart the zabbix-agent service:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

For **Zabbix agent** this re-enables the functionality of configuring user in the zabbix_agentd.conf file. Now you need to set User=root and AllowRoot=1 configuration parameters in the agent [configuration file](#).

Zabbix agent 2

To override the default user and group for Zabbix agent 2, run:

```
systemctl edit zabbix-agent2
```

Then, add the following content:

```
[Service]
User=root
Group=root
```

Reload daemons and restart the zabbix-agent service:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

For **Zabbix agent2** this completely determines the user that it runs as. No additional modifications are required.

10 MicrosoftWindows 上の Zabbix エージェント

エージェントの設定

Zabbix エージェントは両世代とも Windows サービスとして実行されます。Zabbix エージェント 2 の場合、以下の手順で agentd を agent2 に置き換えます。

MicrosoftWindows ホスト上で Zabbix エージェントの単一インスタンスまたは複数インスタンスを実行できます。単一インスタンスの場合は、デフォルトの設定ファイル C:\zabbix_agentd.conf またはコマンドラインで指定された設定ファイルを使用できます。複数インスタンスの場合は、各エージェントインスタンスに独自の構成ファイルが必要です。(1つのインスタンスだけデフォルト構成ファイルを使用できます)

設定ファイルの例は Zabbix ソースアーカイブで conf/zabbix_agentd.win.conf として入手できます。

Zabbix Windows エージェントの設定詳細については[設定ファイルオプション](#)を参照してください。

ホストネームパラメータ

ホストで[アクティブチェック](#)を実行するには、Zabbix エージェントでホスト名を定義する必要があります。また、エージェント側で設定するホスト名の値は、フロントエンドのホストに設定されている“ホスト名”と完全に一致している必要があります。

エージェント側のホスト名値は、エージェント構成ファイルの **Hostname** または **Hostnameltem** パラメーターのいずれかで定義できます。これらのパラメーターは指定されていない場合、デフォルト値が使用されます。

Hostnameltem パラメータのデフォルト値は“system.hostname” エージェントキーによって返される値です。Windows の場合、gethostname() 関数の結果を返します。この関数は、名前空間プロバイダーにクエリを実行して、ローカルホスト名を決定します。名前空間プロバイダーが応答しない場合は、NetBIOS 名が返されます。

Hostname のデフォルト値は **Hostnameltem** パラメーターによって返される値です。したがってこれらのパラメーターが両方とも設定されていない場合、ホスト名はホストの NetBIOS 名になります。Zabbix エージェントは NetBIOS ホスト名を使用して、Zabbix サーバーからアクティブなチェックのリストを取得し、結果をサーバーに送信します。

“system.hostname” キーは、type と transform の 2 つのオプションパラメーターをサポートします。

Type パラメーターは、アイテムが返す名前のタイプを決定します。サポートされている値：

- netbios (default) - 15 シンボルに制限された大文字のみの NetBIOS ホスト名を返します。
- host - 大文字と小文字が区別された、完全な実際の Windows ホスト名 (ドメインなし) を返します。
- shorthost (Zabbix 5.4.7 以降でサポート) - 最初のドットの前のホスト名の一部を返します。名前にドットが含まれていない場合は、完全な文字列が返されます。

Transform パラメータは Zabbix5.4.7 以降でサポートされており、ホスト名に追加の変換ルールを指定できます。サポートされている値：

- none (default) - 元の大大文字小文字を使用します。
- lower - テキストを小文字に変換します。

したがって、zabbix_agentd.conf ファイルの設定を簡素化し、統合するために、2 つの異なるアプローチができます。

1. **Hostname** または **Hostnameltem** パラメータを未定義のままにすると、Zabbix エージェントはホスト名として NetBIOS ホスト名を使用します。
2. ****Hostname** パラメータを未定義のままにして、**Hostnameltem**** を次のように定義します。****Hostnameltem=system.hostname[host]**** - Zabbix エージェントが完全な実際の (大文字と小文字を区別する) Windows ホスト名をホスト名として使用する場合 ****Hostnameltem=system.hostname[shorthost,lower]**** - Zabbix エージェントが最初のドットの前のホスト名の一部のみを使用し、小文字に変換する場合。

ホスト名は、Windows サービスのインストール、開始、停止、およびアンインストールに使用される Windows サービス名の一部としても使用されます。たとえば、Zabbix エージェント設定ファイルで **Hostname=Windows_db_server** が指定されている場合、エージェントは Windows サービス “ZabbixAgent [Windows_db_server]” としてインストールされます。したがって、Zabbix エージェントインスタンスごとに異なる Windows サービス名を使用するには、インスタンスごとに異なるホスト名を使用する必要があります。

Windows サービスとしてのエージェントのインストール

デフォルトの設定ファイル c:\zabbix_agentd.conf を使用して Zabbix エージェントの単一インスタンスをインストールするには：


```
zabbix_agentd.exe --install
```

Attention:

64 ビットシステムでは、64 ビットプロセスの実行に関連するすべてのチェックが正しく機能するために、64 ビット版の Zabbix エージェントが必要です。

c:\zabbix_agentd.conf 以外の設定ファイルを使用するには、サービスのインストールに次のコマンドを使用する必要があります。

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

構成ファイルへのフルパスを指定する必要があります。

Zabbix エージェントの複数のインスタンスは、次のようなサービスとしてインストールできます。

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

インストールされたサービスがコントロールパネルに表示されます。

エージェントの起動

エージェントサービスを開始するには、コントロールパネルを使用するか、コマンドラインから実行します。

デフォルトの設定ファイルで Zabbix エージェントの単一インスタンスを起動するには：

```
zabbix_agentd.exe --start
```

Zabbix エージェントの単一インスタンスを別の設定ファイルで起動するには：

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Zabbix エージェントの複数インスタンスのうち 1 つを開始するには：

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

エージェントの停止

エージェントサービスを停止するには、コントロールパネルを使用するか、コマンドラインからコマンドを実行します。

デフォルトの設定ファイルで開始された Zabbix エージェントの単一インスタンスを停止するには

```
zabbix_agentd.exe --stop
```

別の設定ファイルで開始された Zabbix エージェントの単一インスタンスを停止するには

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Zabbix エージェントの複数インスタンスのうち 1 つを停止するには

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Windows サービスとしてのエージェントのアンインストール

デフォルトの設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには

```
zabbix_agentd.exe --uninstall
```

デフォルト以外の設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Windows サービスから Zabbix エージェントの複数のインスタンスをアンインストールするには

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

制限

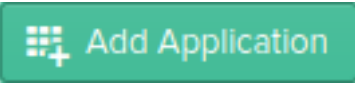
Windows 用 Zabbix エージェントは、CPU が NUMA ノード全体に不均一に分散されている非標準の Windows 構成をサポートしていません。論理 CPU が不均一に分散されている場合、一部の CPU では CPU パフォーマンスメトリックが使用できない場合があります。たとえば、2 つの NUMA ノードを持つ 72 個の論理 CPU がある場合、両方のノードにそれぞれ 36 個の CPU が必要です。

11 Okta を使用した SAML セットアップ

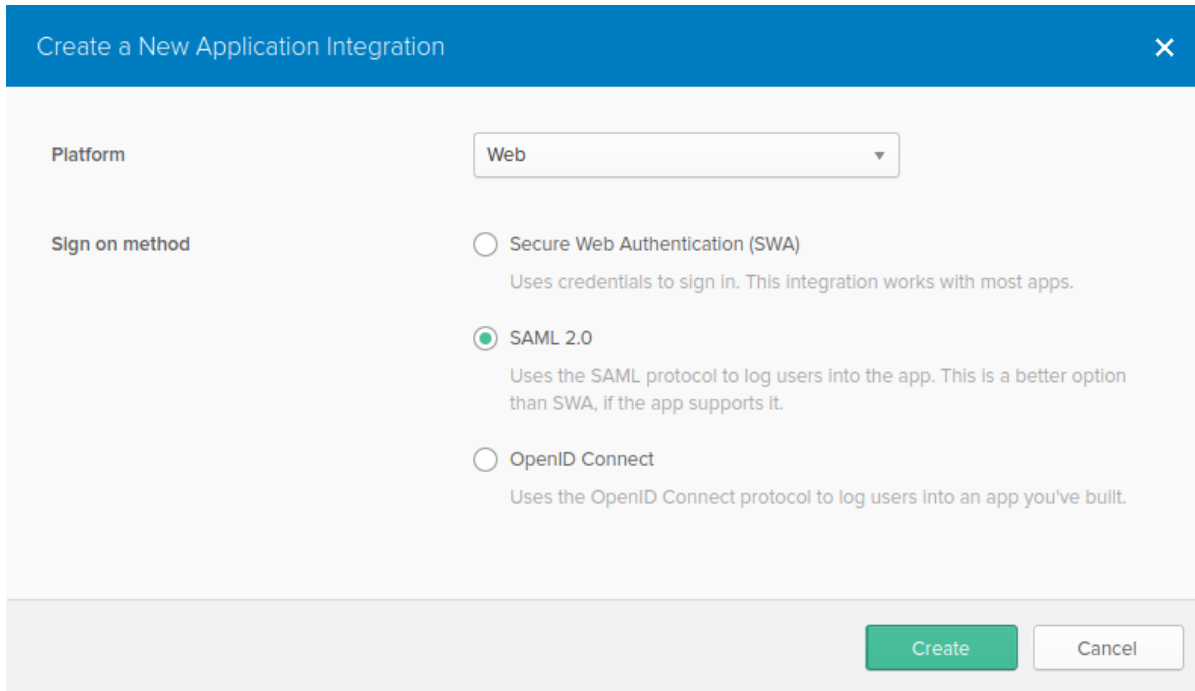
このセクションでは、Zabbix の SAML2.0 認証を有効にするように Okta を設定する方法について説明します

Okta の設定

1. <https://okta.com> にアクセスして、アカウントに登録またはサインインします。
2. OktaWeb インターフェースでアプリケーション → アプリケーションに移動し、[アプリケーションの追加] ボタン

() を押します。

3. 「新しいアプリの作成」ボタン () を押し、ポップアップウィンドウで Platform: Web, Sign on method:SAML2.0 と選択して「作成」ボタンを押します。



Create a New Application Integration

Platform: Web

Sign on method:

- Secure Web Authentication (SWA)
Uses credentials to sign in. This integration works with most apps.
- SAML 2.0
Uses the SAML protocol to log users into the app. This is a better option than SWA, if the app supports it.
- OpenID Connect
Uses the OpenID Connect protocol to log users into an app you've built.

Create Cancel

4. 必要であれば [一般設定] タブ (最初に表示されるタブ) のフィールドに入力し、[次へ] を押します。

5. Configure SAML タブで以下に示す値を入力し [次へ] を押します。

• **GENERAL** セクション:

- Single sign on URL: https://<your-zabbix-url>/ui/index_sso.php?acs
Use this for Recipient URL and Destination URL のチェックボックスにマーク
- Audience URI (SP Entity ID): zabbix
この値は、SAML アサーション内で一意のサービスプロバイダー識別子として使用されることに注意してください (一致しない場合、操作は拒否されます)。このフィールドには、URL または任意のデータ文字列を指定できます。
- Default RelayState:
このフィールドは空白のままにします。カスタムリダイレクトが必要な場合は、Zabbix の [管理] → **ユーザー** 設定で追加できます。
- 必要であれば、他のフィールドも入力します。

GENERAL

Single sign on URL ?

Use this for Recipient URL and Destination URL

Allow this app to request other SSO URLs

Audience URI (SP Entity ID) ?

Default RelayState ?

If no value is set, a blank RelayState is sent

Name ID format ?

Application username ?

Update application username
on

[Show Advanced Settings](#)

Note:

暗号化接続の使用を計画している場合は、プライベートおよびパブリック暗号化証明書を生成してから、パブリック証明書を Okta にアップロードします。Assertion Encryption が暗号化に設定されている場合、証明書のアップロードフォームが表示されます（このパラメーターを見つけるには Show Advanced Settings をクリックします）

• **ATTRIBUTE STATEMENTS(OPTIONAL)** セクションで、次の属性ステートメントを追加します。

- Name: usrEmail
- Name format: Unspecified
- Value: user.email

ATTRIBUTE STATEMENTS (OPTIONAL)

[LEARN MORE](#)

Name

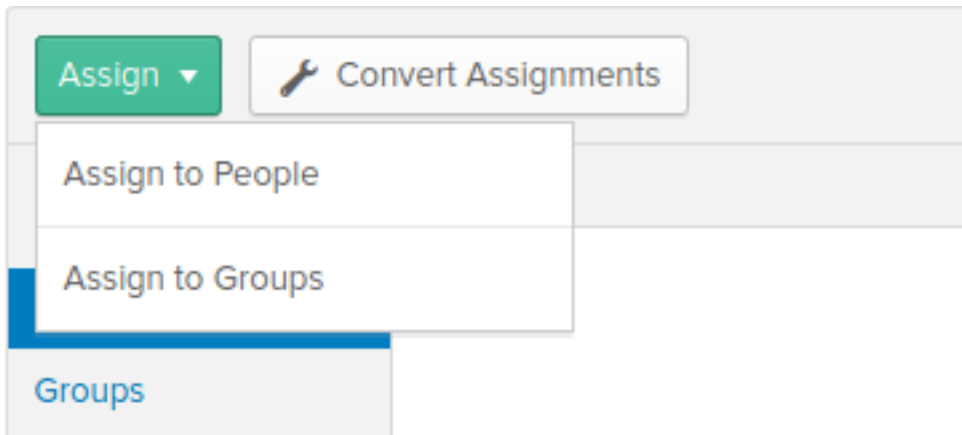
Name format
(optional)

Value

[Add Another](#)

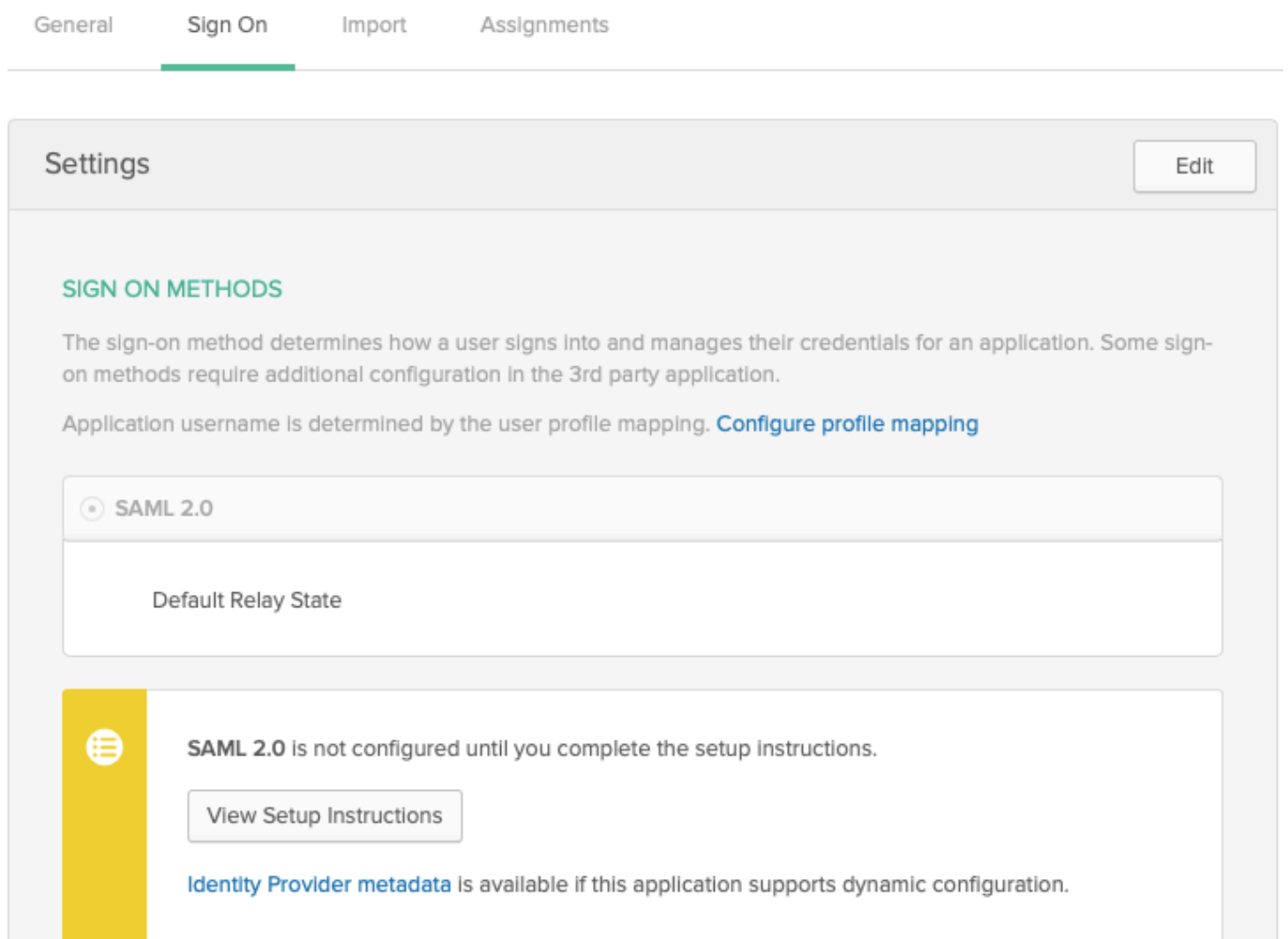
6. 次のタブで「I'm a software vendor. I'd like to integrate my app with Okta」を選択し「完了」を押します。

7. 次に Assignments タブに移動して "Assign" ボタンを押し、ドロップダウンから Assign to People を選択します。



8. 表示されるポップアップで、作成したアプリを SAML 2.0 を使用して Zabbix で認証するユーザーに割り当て、"Save and go back" を押します。

9. Sign On タブに移動し、"View Setup Instructions" ボタンを押します。セットアップ手順は新しいタブに表示されます。Zabbix を設定している間、このタブを開いたままにしてください。



Zabbix の設定

1. Zabbix で [管理]→**認証**セクションの SAML 設定に移動し、Okta のセットアップ手順から対応するフィールドに情報をコピーします。

- Identity Provider Single Sign-On URL → SSO service URL
- Identity Provider Issuer → IdP entity ID
- Username attribute → Attribute name (usrEmail)
- SP entity ID → Audience URI

2. Okta のセットアップ手順ページに記載されている証明書を idp.crt として ui/conf/certs フォルダーにダウンロードし、次のコマン

ドを実行してアクセス許可を 644 に設定します。

```
chmod 644 idp.crt
```

古いバージョンから Zabbix5.0 にアップグレードした場合は、以下を zabbix.conf.php ファイル (//ui//conf//ディレクトリにあります) に手動で追加する必要があることに注意してください。

```
// Used for SAML authentication.
$SSO['SP_KEY'] = 'conf/certs/sp.key'; // Path to your private key.
$SSO['SP_CERT'] = 'conf/certs/sp.crt'; // Path to your public key.
$SSO['IDP_CERT'] = 'conf/certs/idp.crt'; // Path to IdP public key.
$SSO['SETTINGS'] = []; // Additional settings
```

詳細については、一般的なSAML 認証の手順を参照してください。

3. Assertion Encryption が Okta で Encrypted に設定されている場合、Zabbix でも Encrypt パラメータのチェックボックス「Assertions」をマークする必要があります。

[Authentication](#) [HTTP settings](#) [LDAP settings](#) [SAML settings](#)

Enable SAML authentication

* IdP entity ID

* SSO service URL

SLO service URL

* Username attribute

* SP entity ID

SP name ID format

Sign Messages

Assertions

AuthN requests

Logout requests

Logout responses

Encrypt Name ID

Assertions

Case sensitive login

4. "更新" ボタンを押して設定を保存します。

Note:

SAML でサインインするには、Zabbix のユーザー名が Okta の電子メールと一致している必要があります。これらの設定は、ZabbixWeb インターフェースの管理 → ユーザーセクションで変更できます。

12 オラクルデータベースセットアップ

概要

このセクションには Oracle データベースを作成し、データベースと Zabbix サーバー、プロキシ、およびフロントエンド間の接続を構成する手順が含まれています。

データベースの作成

password というパスワードを持つ zabbix データベースユーザーが存在し、host の Oracle データベースサーバーにある ORCL サービスでデータベースオブジェクトを作成する権限を持っていることを前提としています。Zabbix には、Unicode データベースの文字セットと UTF8 国別文字セットが必要です。現在の設定を確認します。

```
sqlplus> select parameter,value from v$nls_parameters where parameter='NLS_CHARACTERSET' or parameter='NLS
```

次にデータベースを準備します。

```
shell> cd /path/to/zabbix-sources/database/oracle
shell> sqlplus zabbix/password@oracle_host/ORCL
sqlplus> @schema.sql
# Zabbixプロキシ用のデータベースを作成している場合はここまで
sqlplus> @images.sql
sqlplus> @data.sql
```

Note:

最高のパフォーマンスを得るには、初期化パラメータ CURSOR_SHARING=FORCE を設定してください。

接続設定

Zabbix は 2 種類の接続識別子（接続方法）をサポートしています。

- Easy Connect
- Net Service Name

Zabbix サーバーと Zabbix プロキシの接続設定パラメータは設定ファイルで設定できます。サーバーとプロキシの重要なパラメータは DBHost、DBUser、DBName、および DBPassword です。同じようにフロントエンドでも次のパラメータが重要です：
\$DB["SERVER"],\$DB["PORT"], \$DB["DATABASE"], \$DB["USER"],\$DB["PASSWORD"]

Zabbix は次の接続文字列構文を使用します。

```
{DBUser/DBPassword[@<connect_identifier>]}
```

<connect_identifier> は "Net Service Name" または "Easy Connect" のいずれかの形式で指定できます。

```
@[[/]]Host[:Port]/<service_name> | <net_service_name>
```

Easy Connect

Easy Connect は、次のパラメータを使用してデータベースに接続します。

- Host - データベースサーバーのホスト名または IP アドレス（構成ファイルの DBHost パラメータ）
- Port - データベースサーバーのリスニングポート（構成ファイルの DBPort パラメータ。設定されていない場合はデフォルトの 1521 ポートが使用されます）
- <service_name> - アクセスするデータベースのサービス名（構成ファイルの DBName パラメータ）

例

サーバーまたはプロキシ設定ファイル（zabbix_server.conf および zabbix_proxy.conf）で設定されたデータベースパラメータ：

```
DBHost=localhost
DBPort=1521
DBUser=myusername
DBName=ORCL
DBPassword=mypassword
```

Zabbix が接続を確立するために使用する接続文字列：

```
DBUser/DBPassword@DBHost:DBPort/DBName
```

Zabbix フロントエンドのインストール中に、セットアップウィザードのデータベース接続設定ステップで対応するパラメータを設定します。

- Database host: localhost
- Database port: 1521
- Database name: ORCL
- User: myusername
- Password: mypassword



Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database.
Press "Next step" button when done.

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

GUI settings

Pre-installation summary

Install

Database type	<input type="text" value="Oracle"/>
Database host	<input type="text" value="localhost"/>
Database port	<input type="text" value="1521"/> 0 - use default port
Database name	<input type="text" value="ORCL"/>
Store credentials in	<input checked="" type="radio"/> Plain text <input type="radio"/> HashiCorp Vault
User	<input type="text" value="myusername"/>
Password	<input type="password" value="*****"/>

またはこれらのパラメーターをフロントエンド構成ファイル (zabbix.conf.php) で設定することもできます。

```
$DB["TYPE"]           = 'ORACLE';  
$DB["SERVER"]        = 'localhost';  
$DB["PORT"]          = '1521';  
$DB["DATABASE"]     = 'ORCL';  
$DB["USER"]          = 'myusername';  
$DB["PASSWORD"]     = 'mypassword';
```

Net service name

Zabbix 5.4.0 以降では Net service name を使用して Oracle に接続できます。

<net_service_name> は接続記述子に解決されるサービスの単純な名前です。

接続の作成にサービス名を使用するには、このサービス名をデータベースサーバーとクライアントシステムの両方にある tnsnames.ora ファイルで定義する必要があります。接続が成功することを確認する最も簡単な方法は TNS_ADMIN 環境変数で tnsnames.ora ファイルの場所を定義することです。tnsnames.ora ファイルのデフォルトの場所は次のとおりです。

```
$ORACLE_HOME/network/admin/
```

簡単な tnsnames.ora ファイルの例：

```
ORCL =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = ORCL)  
    )  
  )
```

"Net Service Name" 接続のパラメータを設定するには、次のいずれかのオプションを使用します。To set configuration parameters for the "Net Service Name" connection method, use one of the following options:

- DBHost を空で設定し、DBName を通常どおり設定します

```
DBHost=  
DBName=ORCL
```

- パラメーターを設定せず両方を空のままにします。

```
DBHost=  
DBName=
```

2 番目のケースでは、TWO_TAKS 環境変数にデフォルトのリモート Oracle サービス (サービス名) を設定する必要があります。この変数が定義されている場合、コネクタは、接続要求を受け入れる Oracle リスナーを使用して、指定されたデータベースに接続します。この変数は、Linux および UNIX でのみ使用されます。Microsoft Windows では LOCAL 環境変数を使用します。

例

ORCL と設定されたネットサービス名とデフォルトのポートを使用してデータベースに接続します。サーバーまたはプロキシ設定ファイル (zabbix_server.conf and zabbix_proxy.conf) で設定されたデータベースパラメータは次の通りです。

```
DBHost=  
#DBPort=  
DBUser=myusername  
DBName=ORCL  
DBPassword=mypassword
```

Zabbix フロントエンドのインストール中に、セットアップウィザードのデータベース接続設定ステップで対応するパラメータを設定します。

- Database host:
- Database port: 0
- Database name: ORCL
- User: myusername
- Password: mypassword

または、これらのパラメーターをフロントエンド構成ファイル (zabbix.conf.php) で設定することもできます。

```
$DB["TYPE"]           = 'ORACLE';  
$DB["SERVER"]         = '';  
$DB["PORT"]           = '0';  
$DB["DATABASE"]       = 'ORCL';  
$DB["USER"]           = 'myusername';  
$DB["PASSWORD"]       = 'mypassword';
```

Zabbix が接続を確立するために使用する接続文字列は以下です。

```
DBUser/DBPassword@ORCL
```

Known issues

To improve performance, you can convert the field types from nlob to nvarchar2, see [known issues](#).

13 レポートスケジュールの設定

概要

このセクションでは、Zabbix Web サービスをインストールし、スケジュールにそってレポートを生成するように Zabbix を設定する手順について説明します。

Attention:

現在、レポートスケジュールのサポートは試験的です。

インストール

スケジュールされたレポートを生成できるようにするには、新しいZabbix Web サービスプロセスと GoogleChrome ブラウザをインストールする必要があります。Web サービスは、Zabbix サーバーがインストールされているのと同じマシンまたは別のマシンにインストールできます。Google Chrome ブラウザは、Web サービスがインストールされているのと同じマシンにインストールする必要があります。

ソースから ZabbixWeb サービスをコンパイルするにはZabbix Web サービスのインストールを参照してください。

インストール後、Web サービスがインストールされているマシンで `zabbix_web_service` を実行します。

```
shell> zabbix_web_service
```

設定

関連するすべての要素間の適切な通信を確保するには、サーバー構成ファイルとフロントエンド構成パラメーターが適切に構成されていることを確認してください。

Zabbix サーバー

Zabbix サーバー構成ファイルの次のパラメーターを更新する必要があります：WebServiceURL および StartReportWriters

WebServiceURL

このパラメーターは、Web サービスとの通信を有効にするために必要です。URL は `<host:port>/report` の形式である必要があります。

- デフォルトでは Web サービスはポート 10053 でリッスンします。Web サービス構成ファイルで別のポートを指定できます。
- `/report` パスの指定は必須です。(パスはハードコードされており、変更できません)

例:

```
WebServiceURL=http://localhost:10053/report
```

StartReportWriters

このパラメーターは、開始するレポートライタープロセスの数を決定します。未設定または 0 の場合、レポートの生成は無効になります。必要なレポート数と頻度に基づいて、1~100 のレポートライタープロセスを有効にすることができます。

例:

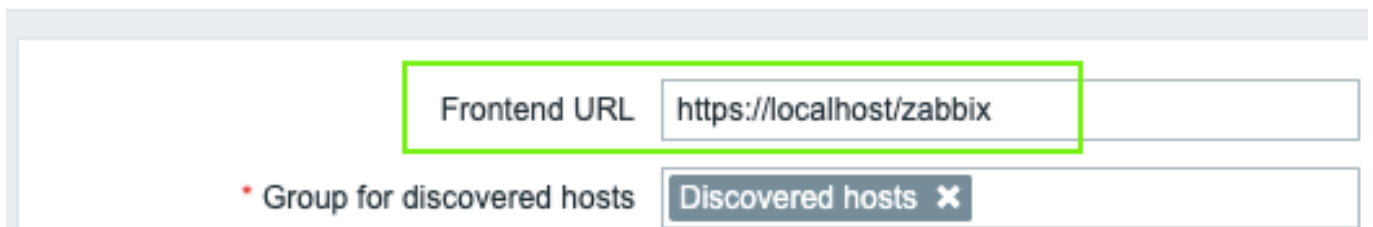
```
StartReportWriters=3
```

Zabbix フロントエンド

Zabbix フロントエンドと ZabbixWeb サービス間の通信を有効にするには、Frontend URL パラメーターを設定する必要があります。

- `*[管理]→一般→[その他のパラメータ]` からフロントエンドメニューセクションに進みます。
- Frontend URL パラメーターで ZabbixWeb インターフェースの完全な URL を指定します。

Other configuration parameters



Note:

セットアップ完了後に**テストレポート**を設定して送信し、すべて正しく機能することを確認することをお勧めします。

公式の `zabbix-web-service` パッケージは、[Zabbix リポジトリ](#) で入手できます。Google Chrome ブラウザはこれらのパッケージに含まれていないため、個別にインストールする必要があります。

14 フロントエンドへの言語追加

概要

Zabbix Web インターフェースで英語以外の言語を使用するには、そのロケールを Web サーバーにインストールする必要があります。さらに翻訳を機能させるには、PHP の `gettext` 拡張機能が必要です。

ロケールのインストール

インストールされているすべての言語を一覧表示するには、次のコマンドを実行します。

```
locale -a
```

必要な言語がリストされていない場合は/etc/locale.gen ファイルを開き、必要なロケールのコメントを解除します。Zabbix は UTF-8 エンコーディングを使用するため、UTF-8 文字セットを使用してロケールを選択する必要があります。

次に以下を実行します。

```
locale-gen
```

web サーバーを再起動してください。

これで、ロケールがインストールされます。新しい言語を表示するには、Ctrl+F5 でブラウザ上の Zabbix フロントエンドページをリロードする必要がある場合があります。

Zabbix のインストール

Zabbix git リポジトリから直接 Zabbix をインストールする場合は、翻訳ファイルを手動で生成する必要があります。翻訳ファイルを生成するには、次のコマンドを実行します。

```
make gettext
locale/make_mo.sh
```

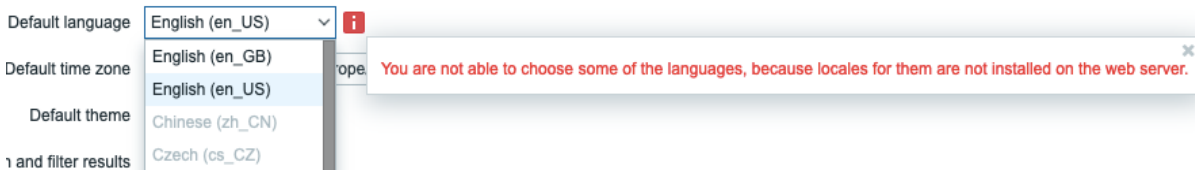
パッケージまたはソース tar.gz ファイルから Zabbix をインストールする場合、この手順は必要ありません。

言語設定

ZabbixWeb インターフェースで言語を選択する方法はいくつかあります。

- Web インターフェイスをインストールする場合 - フロントエンドインストールウィザードで選択した言語がシステムデフォルトとして設定されます。
- インストール後、システムのデフォルト言語は管理 → 一般 → GUI のメニューセクションで変更できます。
- 特定のユーザーの言語はユーザープロファイルで変更できます

対象の言語ロケールがマシンにインストールされていない場合、その言語は Zabbix 言語セクターでグレー表示されます。少なくとも 1 つのロケールが欠落している場合、言語セクターの横に赤いアイコンが表示されます。このアイコンを押すと、次のメッセージが表示されます。「Web サーバーにインストールされていない言語ロケールがあるため、一部の言語は選択できません。」



3 プロセスの設定

1 Zabbix サーバー

概要

このセクションでは、Zabbix サーバー設定ファイル (zabbix_server.conf) でサポートされているパラメーターを一覧表示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix はBOMなしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が" #" で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
AlertScriptsPathno			/usr/local/share/zabbix/alertscripts	AlertScripts スクリプトの場所 (コンパイル時のインストール変数 datadir によって異なります)
AllowRoot	no		0	サーバーを 'root' が実行できるようにします。無効化の時にサーバーを 'root' にて起動しようとする、サーバーは代わりに 'zabbix' ユーザーで起動しようとする。通常のユーザーで開始した場合は影響ありません。 0 - 許可しない 1 - 許可する

パラメータ	必須	範囲	デフォルト	説明
AllowUnsupportedDBVersions	no		0	サーバーがサポートされていないデータベースバージョンで動作することを許可します 0 - 許可しない 1 - 許可する
CacheSize	no	128K-64G	32M	構成キャッシュのサイズ (バイト単位) ホスト、アイテム、およびトリガーデータを格納するための共有メモリサイズ
CacheUpdateFrequency	no	1-3600	60	Zabbix が構成キャッシュの更新を実行する頻度を秒単位で決定します。 ランタイムコントロール オプションも参照してください
DBHost	no		localhost	データベースのホスト名 MySQL で localhost または空の文字列の場合、ソケットが使用されます。 PostgreSQL の場合、空の文字列のみがソケットを使用しようとします。 Oracle の場合、空の文字列はネットサービス名の接続方法を使用します。この場合、TNS_ADMIN 環境変数を使用して tnsnames.ora ファイルのディレクトリを指定することを検討してください。
DBName	yes			データベース名。 Oracle の場合、ネットサービス名接続方式を使用する場合は、tnsnames.ora からサービス名を指定するか、空文字列に設定されます。DBName が空の文字列に設定されている場合は、TWO_TASK 環境変数を設定します。
DBPassword	no			データベースのパスワード。 パスワードを使用しない場合は、この行をコメントにしてください。
DBPort	no	1024-65535		ローカルソケットを使用しない場合のデータベースポート。 Oracle でネットサービス名接続が使用される場合、このパラメータは無視されます。代わりに、tnsnames.ora ファイルのポート番号が使用されます。
DBSchema	no			スキーマ名。PostgreSQL で使用されます。
DBSocket	no			MySQL ソケットファイルへのパス。
DBUser	no			データベースユーザー。
DBTLSConnect	no			このオプションを設定するとデータベースへの TLS 接続の使用が強制されます: required - TLS を使用して接続します verify_ca - TLS を使用して接続し、証明書を検証します verify_full - TLS を使用して接続し、証明書を検証し、DBHost によって指定されたデータベース ID がその証明書と一致することを確認します
				5.7.11 以降の MySQL および PostgreSQL では "required", "verify_ca", "verify_full" の値がサポートされています。バージョン 10.2.6 以降の MariaDB では、"required" および "verify_full" の値がサポートされています。 デフォルトではどのオプションにも設定されておらず、動作はデータベースの構成によって異なります。
DBTLSCAFile	no (DBTLSConnect が次のいずれかに設定されている場合は yes: verify_ca, verify_full)			このパラメータは zabbix 5.0.0 移行でサポートされています。 データベース証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSCertFile	no			データベースに対して認証するための Zabbix サーバー証明書を含むファイルのフルパス名。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。

パラメータ	必須	範囲	デフォルト	説明
DBTLSKeyFile	no			データベースに対して認証するための秘密鍵を含むファイルのフルパス名。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSCipher	no			Zabbix サーバーが TLSv1.2 までの TLS プロトコルに対して許可する暗号化方式のリスト。 MySQL でのみサポートされています。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSCipher13	no			Zabbix サーバーが TLSv1.3 プロトコルで許可する暗号化方式のリスト。 バージョン 8.0.16 以降の MySQL でのみサポートされています。 このパラメータは Zabbix 5.0.0 以降でサポートされています。
DebugLevel	no	0-5	3	デバッグレベルを指定します。 0 - Zabbix プロセスの開始と停止に関する基本情報 1 - 重要な情報 2 - エラー情報 3 - 警告 4 - デバッグ用 (多くの情報を生成) 5 - 拡張デバッグ (さらに多くの情報を生成) ランタイムコントロール オプションも参照してください。 改行区切りの JSON 形式のイベント、履歴、トレンドのリアルタイムエクスポートのディレクトリ。設定すると、リアルタイムエクスポートが有効になります。 このパラメータは、Zabbix 4.0.0 以降でサポートされています。
ExportDir	no			エクスポートファイルあたりの最大サイズ (バイト単位)。 ExportDir が設定されている場合にのみローテーションに使用されます。 このパラメータは Zabbix 4.0.0 以降でサポートされています。
ExportFileSize	no	1M-1G	1G	リアルタイムエクスポートのコンマ区切りのエンティティタイプ (イベント、履歴、トレンド) リスト (デフォルトではすべてのタイプ)。ExportDir が設定されている場合にのみ有効です。 注意 ExportType が指定されているが、ExportDir が指定されていない場合、これは設定エラーであり、サーバーは起動しません。 例: ExportType=history,trends - 履歴と傾向のみをエクスポートします ExportType=events - イベントのみをエクスポートします
ExportType	no			外部にエクスポートされる場所 (コンパイル時のインストール変数 datadir によって異なります)
ExternalScripts	no		/usr/local/share/perl5/Net::SNMP	fping6 の場所 fping6 バイナリにルート所有権があり、SUID フラグが設定されていることを確認してください。 fping ユーティリティが IPv6 アドレスを処理できる場合は、空 ("Fping6Location=") にします。
Fping6Location	no		/usr/sbin/fping6	fping の場所。 fping バイナリにルート所有権と SUID フラグが設定されていることを確認してください。
FpingLocation	no		/usr/sbin/fping	高可用性クラスタノード名。 空の場合、サーバーはスタンドアロンモードで動作し、名前が空のノードが作成されます。
HANodeName	no			履歴キャッシュのサイズ (バイト単位) 履歴データを保存する共有メモリのサイズ。
HistoryCacheSize	no	128K-2G	16M	ヒストリーインデックスキャッシュのサイズ (バイト単位)。 ヒストリーキャッシュに格納されたヒストリーデータをインデックス化するための共有メモリサイズ。 インデックスキャッシュサイズは、1つのアイテムをキャッシュするのに約 100 バイト必要です。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
HistoryIndexCacheSize	no	128K-2G	4M	

パラメータ	必須	範囲	デフォルト	説明
HistoryStorageDateIndex			0	履歴ストレージの履歴値の前処理を有効にして、日付に基づいて異なるインデックスに値を保存します: 0 - 無効にする 1 - 有効にする
HistoryStorageURL				履歴ストレージ HTTP[S] URL。 このパラメータは、Elasticsearch セットアップに使用されま
HistoryStorageTypes			uint,dbl,str,log,t	履歴ストレージに送信される値の型のカンマ区切りリスト。 このパラメータはElasticsearch セットアップに使用されま
HousekeepingFrequency		0-24	1	Zabbix がハウスキーピング手順を実行する頻度を時間単位で決定します。 ハウスキーピングはデータベースから古い情報を削除します。 注: ハウスキーピングが過負荷になるのを防ぐため (たとえば、履歴とトレンドの期間が大幅に短縮された場合)、アイテムごとに、1 回のハウスキーピングサイクルで HousekeepingFrequency 時間の古い情報が 4 回以下削除されま
Include	no			す。したがって、HousekeepingFrequency が 1 の場合、1 サイクルあたり (最も古いエントリから開始して) 4 時間以内の古い情報が削除されます。したがって、HousekeepingFrequency が 1 の場合、サーバー起動後の最初のハウスキーピング手順は 30 分後に実行され、その後 1 時間遅れて繰り返されます。 Zabbix 3.0.0 以降、HousekeepingFrequency を 0 に設定することで自動ハウスキーピングを無効にすることができます。この場合、ハウスキーピング手順は housekeeper_execute ランタイム制御オプションによってのみ開始でき、1 回のハウスキーピングサイクルで削除される古い情報の期間は、最後のハウスキーピングサイクル以降の期間の 4 倍ですが、4 時間以上であり、4 日以上。 ランタイムコントロール オプションも参照してください。 個々のファイルまたはディレクトリ内のすべてのファイルを構成ファイルに含めることができます。 指定したディレクトリに関連ファイルのみを含めるために、パターンマッチングでアスタリスクワイルドカード文字がサポートされています。例: /absolute/path/to/config/files/*.conf 制限については、特記事項を参照してください。
JavaGateway	no			Zabbix Java ゲートウェイの IP アドレス (またはホスト名)。 Java ポーラーが開始されている場合にのみ必要です。
JavaGatewayPort		1024-32767	10052	Zabbix Java ゲートウェイがリスンするポート
ListenBacklog	no	0 - INT_MAX	SOMAXCONN	TCP キュー内の保留中の接続の最大数。 デフォルト値はハードコードされた定数で、システムによって異なります。 サポートされている最大値はシステムによって異なります。値が高すぎると '実装で指定された最大値' に切り捨てられる場合があります。
ListenIP	no		0.0.0.0	トラッパーがリスンするカンマ区切りの IP アドレスのリスト。 このパラメータが欠落している場合、トラッパーはすべてのネットワークインターフェイスでリスンします。
ListenPort	no	1024-32767	10051	トラッパーのリスンポート
LoadModule	no			サーバー起動時にロードするモジュール。モジュールは、サーバーの機能を拡張するために使用されます。 形式: LoadModule=<module.so> LoadModule=<path/module.so> LoadModule=</abs_path/module. so> モジュールが LoadModulePath で指定されたディレクトリに配置されているか、パスがモジュール名の前にある必要があります。前のパスが絶対パス (「/」で始まる) の場合、LoadModulePath は無視されます。 複数の LoadModule パラメータを含めることができます。

パラメータ	必須	範囲	デフォルト	説明
LoadModulePath				サーバーモジュールの場所へのフルパス。 デフォルトはコンパイルオプションによって異なります。
LogFile	LogType が file に設定されている場合は yes そうでない場合は no			ログファイル名
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB) 0 - ログの自動ローテーションを無効にします 注: ログファイルのサイズ制限に達し、ファイルのローテーションが何らかの理由で失敗した場合、既存のログファイルは切り捨てられ、新しいファイルで開始されます。
LogType	no		file	ログ出力タイプ: file - LogFile パラメータで指定されたファイルにログを書き込みます。 system - ログを syslog に書き込みます。 console - ログを標準出力に書き込みます。
LogSlowQueries	no	0-3600000	0	このパラメータは Zabbix 3.0.0 以降でサポートされています。 データベースクエリがログに記録されるまでの時間をミリ秒単位で決定します。 0 - スロークエリをログに記録しません。
MaxHousekeeperDelete		0-1000000	5000	このオプションは、DebugLevel=3 で有効になります。 1 回のハウスキーピングサイクルでは 1 つのタスクにつき 'MaxHousekeeperDelete' 行までしか削除されません。 このパラメータを 0 に設定すると制限がなくなります。この場合データベースが過負荷 ² にならないように、自分が何をしているのかを把握しておく必要があります。 このパラメータは、既に削除されたアイテムの履歴とトレンドを削除する場合にのみ適用されます。
NodeAddress	no			フロントエンドがサーバーに接続する方法をオーバーライドするためのオプションのポートを含む IP またはホスト名。 形式: <アドレス>[:ポート] サーバーアドレスを指定するためにフロントエンドが使用するアドレスの優先度: NodeAddress (1); ListenIP (0.0.0.0 または::でない場合) (2); localhost (デフォルト) (3) 参照: HANodeName パラメータ
PidFile	no		/tmp/zabbix_server.pid	PID ファイル名
ProxyConfigFrequency		1-604800	3600	Zabbix サーバーが設定データを Zabbix プロキシに送信する頻度を秒単位で決定します。パッシブモードのプロキシにのみ適用されます。
ProblemHousekeepingFrequency		1-3600	60	削除されたトリガーの障害を Zabbix が削除する頻度を秒単位で決定します。
ProxyDataFrequency		1-3600	1	Zabbix サーバーが Zabbix プロキシから履歴データを要求する頻度を秒単位で決定します。パッシブモードのプロキシにのみ使用されます。
ServiceManagerSyncFrequency		1-3600	60	Zabbix がサービスマネージャーの設定を同期する頻度を秒単位で決定します。
SNMPTrapperFile			/tmp/zabbix_trap_receiver.pl	SNMP トラップデーモンからサーバーにデータを渡すために使用される一時ファイル。 zabbix_trap_receiver.pl または SNMPTT 構成ファイルと同じである必要があります。
SocketDir	no		/tmp	内部 Zabbix サービスで使用される IPC ソケットを格納するディレクトリ。 このパラメータは Zabbix 3.4.0 以降でサポートされています。

パラメータ	必須	範囲	デフォルト	説明
SourceIP	no			ソース IP アドレス: - Zabbix プロキシおよび Zabbix エージェントへの発信接続; - エージェントレス接続 (VMware、SSH、JMX、SNMP、Telnet、および単純なチェック); - HTTP エージェント接続; - スクリプトアイテム JavaScript HTTP リクエスト; - JavaScript HTTP リクエストの前処理; - 通知メールの送信 (SMTP サーバーへの接続); - Webhook 通知 (JavaScript HTTP 接続); - Vault への接続
SSHKeyLocationno				SSH チェックおよびアクション用の公開鍵と秘密鍵の場所
SSLCertLocationno				クライアント認証用の SSL クライアント証明書ファイルの場所。 このパラメータは Web モニタリングでのみ使用されます。
SSLKeyLocationno				クライアント認証用の SSL 秘密鍵ファイルの場所。 このパラメータは Web モニタリングでのみ使用されます。
SSLCALocationno				SSL サーバー証明書検証用の認証局 (CA) ファイルの場所をオーバーライドします。設定しない場合、システム全体のディレクトリが使用されます。 このパラメータの値は libcurl オプション CURLOPT_CAPATH として設定されることに注意してください。7.42.0 より前のバージョンの libcurl では、これは libcurl が OpenSSL を使用するようにコンパイルされている場合にのみ効果があります。詳細については、 cURL Web ページ を参照してください。 このパラメータは、Zabbix 2.4.0 以降の Web 監視と、Zabbix 3.0.0 以降の SMTP 認証で使用されます。
StartAlerters	no	1-100	3	alerters の事前フォークされたインスタンスの数。 このパラメータは、Zabbix 3.4.0 以降でサポートされていません。
StartDBSyncers	no	1-100	4	history syncers の事前フォークされたインスタンスの数。おおよそ、デフォルト値は最大 4000 NVPS を処理するのに十分ではありません。
StartDiscoverers	no	0-250	1	discoverers の事前フォークされたインスタンスの数。
StartEscalators	no	1-100	1	escalators の事前フォークされたインスタンスの数。 このパラメータは、Zabbix 3.0.0 以降でサポートされていません。
StartHistoryPollers		0-1000	5	history pollers の事前にフォークされたインスタンスの数。 このパラメータは、Zabbix 5.4.0 以降でサポートされていません。
StartHTTTPollersno		0-1000	1	HTTP ポーラー¹ の事前フォークされたインスタンスの数。
StartIPMIPollersno		0-1000	0	IPMI ポーラー の事前フォークされたインスタンスの数。
StartJavaPollersno		0-1000	0	Java ポーラー¹ の事前フォークされたインスタンスの数。
StartLLDProcessors		1-100	2	ローレベルディスカバリ (LLD) ワーカー¹ の事前フォークされたインスタンスの数 LLD ワーカーが開始されると、LLD マネージャープロセスが自動的に開始されます。 このパラメータは、Zabbix 4.2.0 以降でサポートされていません。
StartODBCPollersno		0-1000	1	ODBC ポーラー¹ の事前フォークされたインスタンスの数。
StartPingers	no	0-1000	1	ICMP ピンガー¹ の事前フォークされたインスタンスの数。
StartPollersUnreachable		0-1000	1	到達不能ホストのポーラー (IPMI および Java を含む)¹ の事前分岐されたインスタンスの数 少なくとも 1 つ通常の IPMI または Java ポーラーが開始されている場合は、到達不能ホストのポーラーが実行されている必要があります。
StartPollers	no	0-1000	5	ポーラー¹ の事前フォークされたインスタンスの数。
StartPreprocessors		1-1000	3	前処理 ワーカー¹ の事前フォークされたインスタンスの数 プリプロセスワーカーが開始されると、プリプロセスマネージャープロセスが自動的に開始されます。 このパラメータは、Zabbix 3.4.0 以降でサポートされています。
StartProxyPollersno		0-250	1	パッシブプロキシのポーラー¹ の事前フォークされたインスタンスの数。

パラメータ	必須	範囲	デフォルト	説明
StartReportWriters		0-100	0	<p>レポートライター の事前フォークされたインスタンスの数。 0 に設定すると、スケジュールされたレポートの生成が無効になります。 レポートライターが開始されると、レポートマネージャプロセスが自動的に開始されます。 このパラメータは、Zabbix 5.4.0 以降でサポートされています。</p>
StartSNMPTrappers		0-1	0	<p>1 に設定すると SNMP トラッパー プロセスが開始されます。</p>
StartTimers	no	1-1000	1	<p>タイマーの事前フォークされたインスタンスの数。 タイマープロセスのメンテナンス期間。</p>
StartTrappers	no	0-1000	5	<p>trappers¹ の事前フォークされたインスタンスの数。 トラッパーは、Zabbix 送信者、アクティブエージェント、およびアクティブプロキシからの着信接続を受け入れます。</p>
StartVMwareCollectors		0-250	0	<p>事前にフォークされた VMware コレクター インスタンスの数。</p>
StatsAllowedIP	no			<p>カンマ区切りの IP アドレスのリスト (オプションで CIDR 表記、または外部 Zabbix インスタンスの DNS 名)。統計情報のリクエストは、ここに記載されているアドレスからのみ受け付けます。 このパラメータが設定されていない場合、統計リクエストは受け入れられません。 IPv6 サポートが有効になっている場合、 '127.0.0.1','::127.0.0.1','::ffff:127.0.0.1' は同等に扱われ、 ':::0' は任意の IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' を使用して任意の IPv4 アドレスを許可できます。 例: StatsAllowedIP=127.0.0.1,192.168.1.0/24,:::1,2001:db8::/32,zabbix.example.com このパラメータは、Zabbix 4.2.0 以降でサポートされています。</p>
Timeout	no	1-30	3	<p>エージェント、SNMP デバイス、または外部チェックを待機する時間を秒単位で指定します。</p>
TLSCAFile	no			<p>Zabbix コンポーネント間の暗号化された通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>
TLSCertFile	no			<p>Zabbix コンポーネント間の暗号化された通信に使用される、サーバー証明書または証明書チェーンを含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>
TLSCipherAll	no			<p>GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。 証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 例: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256 このパラメータは、Zabbix 4.4.7 以降でサポートされています。</p>
TLSCipherAll13	no			<p>TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NUL:::+SIGN-ALL:+CTYPE-X.509 OpenSSL の例: EECDH +aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128 このパラメータは、Zabbix 4.4.7 以降でサポートされています。</p>

パラメータ	必須	範囲	デフォルト	説明
TLSCipherCert	no			GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC: +AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NUL:+SIGN-ALL:+CTYPE-X.509 OpenSSL の例: EECDH+aRSA+AES128:RSA+aRSA+AES128 このパラメータは、Zabbix 4.4.7 以降でサポートされています。
TLSCipherCert1.3	no			TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 このパラメータは、Zabbix 4.4.7 以降でサポートされています。
TLSCipherPSK	no			GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC: +AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NUL:+SIGN-ALL OpenSSL の例: kECDHEPSK+AES128:kPSK+AES128 このパラメータは Zabbix 4.4.7 からサポートされています。
TLSCipherPSK1.3	no			TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 例: TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256 このパラメータは、Zabbix 4.4.7 以降でサポートされています。
TLSCRLFile	no			失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSKeyFile	no			Zabbix コンポーネント間の暗号化された通信に使用される、サーバーの秘密鍵を含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TmpDir	no		/tmp	一時ディレクトリ
TrapperTimeout	no	1-300	300	トラッパーが新しいデータの処理に費やすことができる秒数を指定します。
TrendCacheSize	no	128K-2G	4M	トレンドキャッシュのサイズ (バイト単位)。 トレンドデータを保存するための共有メモリサイズ。
TrendFunctionCacheSize	no	128K-2G	4M	トレンド関数キャッシュのサイズ (バイト単位)。 計算されたトレンド関数データをキャッシュするための共有メモリサイズ。
UnavailableDelay	no	1-3600	60	unavailability 期間中にホストが可用性をチェックする頻度を秒単位で決定します。
UnreachableDelay	no	1-3600	15	unreachability 期間中にホストが可用性をチェックする頻度を秒単位で決定します。
UnreachablePeriod	no	1-3600	45	到達不能 が何秒後にホストを使用不可として扱うかを決定します。
User	no		zabbix	システム上の特定の既存のユーザーに特権をドロップします。 'root' として実行し、AllowRoot が無効になっている場合にのみ効果があります。

パラメータ	必須	範囲	デフォルト	説明
ValueCacheSize	no	0,128K-64G	8M	履歴値キャッシュのサイズ (バイト単位)。アイテム履歴データリクエストをキャッシュするための共有メモリサイズ。 0 に設定すると、値キャッシュが無効になります (推奨されません)。 値キャッシュが共有メモリを使い果たすと、警告が表示されず。メッセージは 5 分ごとにサーバーログに書き込まれます。キー 'password' および 'username' によってデータベースの認証情報が取得されるポールドパス。 例: secret/zabbix/database このオプションは、DBUser および DBPassword が指定されていない場合にのみ使用できます。 このパラメータは、Zabbix 5.2.0 以降でサポートされています。
VaultDBPath	no			Vault マクロ で指定されたパスへの読み取り専用アクセス許可と、オプションで指定されたパスへの読み取り専用アクセス許可を持つ、Zabbix サーバー専用で生成されたはずの Vault 認証トークン VaultDBPath 構成パラメータ。 VaultToken と VAULT_TOKEN 環境変数が同時に定義されているとエラーになります。 このパラメータは、Zabbix 5.2.0 以降でサポートされています。
VaultURL	no		https://127.0.0.1:8200	サーバーの HTTP[S] URL。SSLCALocation が指定されていない場合、システム全体の CA 証明書ディレクトリが使用されます。 このパラメータは、Zabbix 5.2.0 以降でサポートされています。
VMwareCacheSize	no	256K-2G	8M	VMware データを格納するための共有メモリサイズ。 VMware 内部チェック zabbix[vmware,buffer,...] を使用して、VMware キャッシュの使用状況を監視できます (内部チェック を参照)。 起動するように構成された vmware コレクタインスタンスがない場合、共有メモリは割り当てられないことに注意してください。
VMwareFrequency	no	10-86400	60	単一の VMware サービスからデータを収集するまでの秒単位の遅延。 この遅延は、VMware 監視項目の最小の更新間隔に設定する必要があります。
VMwarePerfFrequency	no	10-86400	60	単一の VMware サービスからパフォーマンスカウンターの統計情報を取得するまでの秒単位の遅延。 この遅延は、任意の VMware 監視 アイテム の最小の更新間隔に設定する必要があります。VMware パフォーマンスカウンターを使用します。
VMwareTimeout	no	1-300	10	VMware コレクターが VMware サービス (vCenter または ESX ハイパーバイザー) からの応答を待機する最大秒数。
WebServiceURL	no			HTTP[S] <host:port>/report 形式の Zabbix Web サービスへの URL。例: http://localhost:10053/report このパラメータは、Zabbix 5.4.0 以降でサポートされています。

脚注

¹ IPMI マネージャー、SNMP トラッパー、および前処理ワーカーと一緒にデータ収集プロセス (pollers, unreachable pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) が多すぎると、前処理マネージャーのプロセスごとのファイル記述子の制限が使い果たされる可能性があることに注意してください。

Warning:

上記が発生すると Zabbix サーバーが停止します (通常は起動直後に停止しますが、時間がかかる場合もあります)。この状況を回避するには、設定ファイルを修正するか、制限を引き上げる必要があります。

² 多くのアイテムが削除されると、ハウスキーパーがこれらのアイテムが持っていたすべての履歴データを削除する必要があるため、

データベースへの負荷が増加します。たとえば、1つのアイテムのプロトタイプを削除するだけで、このプロトタイプが50のホストにリンクされ、ホストごとにプロトタイプが100の実際のアイテムに拡張される場合、合計5000のアイテムを削除する必要があります(1 * 50 * 100)。MaxHousekeeperDeleteに500が設定されている場合(MaxHousekeeperDelete = 500)、ハウスキーパープロセスは、1サイクルで履歴テーブルとトレンドテーブルから削除されたアイテムの最大2500000値(5000 * 500)を削除する必要があります。

2 Zabbix プロキシ

概要

このセクションでは、Zabbix プロキシ設定ファイル(zabbix_proxy.conf)でサポートされているパラメータの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- ZabbixはBOMなしのUTF-8エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が"#"で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
AllowRoot	no		0	プロキシが'root'で実行されることを許可します。許可しないになっていて、プロキシが'root'によって起動されようとしている場合、プロキシは'zabbix'ユーザーに切り替えようとしません。通常のユーザーで起動した場合は効果ありません。 0 - 許可しない 1 - 許可する
AllowUnsupportedDBVersions			0	サポートされていないデータベースバージョンでのプロキシ実行を許可します。 0 - 許可しない 1 - 許可する
CacheSize	no	128K-64G	32M	構成キャッシュのサイズ(バイト単位) ホストとアイテムのデータを保存するための共有メモリのサイズ
ConfigFrequency	no	1-604800	3600	プロキシがZabbixサーバーから構成データを取得する頻度(秒単位) アクティブプロキシのパラメータであり、パッシブプロキシの場合は無視されます(ProxyModeパラメータを参照)
DataSenderFrequency		1-3600	1	プロキシは収集したデータをN秒ごとにサーバーに送信します。アクティブプロキシは引き続き、リモートコマンドタスクのために毎秒Zabbixサーバーをポーリングすることに注意してください。 アクティブプロキシのパラメータであり、パッシブプロキシの場合は無視されます(ProxyModeパラメータを参照)
DBHost	no		localhost	データベースのホスト名 MySQL:localhost または空の文字列の場合、ソケットが使用されます。 PostgreSQL: 空文字列の場合、ソケットを使用しようとしています。 Oracle: 空文字列の場合、Net Service Name 接続方法を使用します。この場合、TNS_ADMIN 環境変数を使用して tnsnames.ora ファイルのディレクトリを指定することを検討してください。
DBName	yes			SQLite3のデータベース名またはデータベースファイルへのパス(Zabbixのマルチプロセスアーキテクチャでは、インメモリデータベースの使用は許可されません)。 file::memory:?cache=shared または file:memdb1?mode=memory&cache=shared
				警告: Zabbixサーバーが使用しているのと同じデータベースを使用しないでください。 データベースがOracleかつ、Net Service Name 接続方法を使用する場合は、tnsnames.ora からサービス名を指定するか、空の文字列に設定します。DBName が空の文字列に設定されている場合は、TWO_TASK 環境変数を設定します。

パラメータ	必須	範囲	デフォルト	説明
DBPassword	no			データベースのパスワード。SQLite では無視されます。パスワードを使用しない場合は、この行をコメントアウトしてください。
DBSchema	no			スキーマ名。PostgreSQL で使用されます。
DBSocket	no		3306	MySQL ソケットへのパス。 ローカルソケットを使用しない場合のデータベースポート。 SQLite では無視されます。
DBUser				データベースユーザー。SQLite では無視されます。
DBTLSConnect	no			このオプションを設定するとデータベースへの接続に TLS の使用が強制されます: required - TLS を使用して接続します verify_ca - TLS を使用して接続し、証明書を検証します verify_full - TLS を使用して接続し、証明書を検証して、DBHost によって指定されたデータベース ID がその証明書と一致することを確認します 5.7.11 以降の MySQL および PostgreSQL では、 "required","verify","verify_full" の値がサポートされています。 バージョン 10.2.6 以降の MariaDB では、"required" および "verify_full" の値がサポートされています。 デフォルトではどのオプションにも設定されておらず、動作はデータベースの構成によって異なります。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSCAFile	no			(DBTLSConnect が verify_ca,verify_full のいずれかに設定されている場合は yes)
DBTLSCertFile	no			データベースに対して認証するための Zabbix サーバー証明書を含むファイルのフルパス名。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSKeyFile	no			データベースに対して認証するための秘密鍵を含むファイルのフルパス名。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSCipher	no			Zabbix サーバーが TLSv1.2 までの TLS プロトコルに対して許可する暗号化方式のリスト。 MySQL でのみサポートされています。 このパラメータは、Zabbix 5.0.0 以降でサポートされています。
DBTLSCipher13	no			Zabbix サーバーが TLSv1.3 プロトコルで許可する暗号化方式のリスト。 バージョン 8.0.16 以降の MySQL でのみサポートされています。 このパラメータは Zabbix 5.0.0 以降でサポートされています。
DebugLevel	no	0-5	3	デバッグレベルを指定します。 0 - Zabbix プロセスの開始と停止に関する基本情報 1 - 重要な情報 2 - エラー情報 3 - 警告 4 - デバッグ用 (多くの情報を生成) 5 - 拡張デバッグ (さらに多くの情報を生成)

パラメータ	必須	範囲	デフォルト	説明
EnableRemoteCommands			0	Zabbix サーバーからのリモートコマンドを許可するかどうか。 0 - 許可しない 1 - 許可する このパラメータは、Zabbix 3.4.0 以降でサポートされています。
ExternalScripts	no		/usr/local/share	外部スクリプトの場所 (コンパイル時のインストール変数 datadir によって異なります)。
Fping6Location	no		/usr/sbin/fping6	fping6 の場所。 fping6 バイナリに root 所有権があり、SUID フラグが設定されていることを確認します。 fping ユーティリティが IPv6 アドレスを処理できる場合は、空 ("Fping6Location=") にします。
FpingLocation	no		/usr/sbin/fping	fping の場所。 fping バイナリに root 所有権があり、SUID フラグが設定されていることを確認してください。
HeartbeatFrequency		0-3600	60	ハートビートメッセージの頻度 (秒単位)。 サーバー側でプロキシの可用性を監視するために使用されます。 0 - ハートビートメッセージ無効 アクティブプロキシのパラメータであり、パッシブプロキシの場合は無視されます (ProxyMode パラメータを参照)。
HistoryCacheSize		128K-2G	16M	履歴キャッシュのサイズ (バイト単位)。
HistoryIndexCacheSize		128K-2G	4M	履歴データを保存する共有メモリのサイズ ヒストリーインデックスキャッシュのサイズ (バイト単位)。 ヒストリーキャッシュに格納されたヒストリーデータをインデックス化するための共有メモリサイズ。 インデックスキャッシュサイズは、1つのアイテムをキャッシュするのに約 100 バイト必要です。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
Hostname	no		Hostnameltem で設定	大文字と小文字が区別される一意のプロキシ名。プロキシ名がサーバーに認識されていることを確認してください! 使用できる文字: 英数字, '.', ':', '_', 'および'-'。 最大長: 128
Hostnameltem	no		system.hostname	hostname が未定義の場合に設定するために使用されるアイテム (これは、エージェントと同様にプロキシで実行されます)。 UserParameters、performance counters、aliases をサポートしていませんが、system.run[] をサポートしています。 Hostname が設定されている場合は無視されます。
HousekeepingFrequency		0-24	1	Zabbix がハウスキーピング手順を実行する頻度 (時間単位)。 ハウスキーピングにより、古い情報がデータベースから削除されます。 注: ハウスキーパーが過負荷になるのを防ぐため (たとえば、構成パラメーター ProxyLocalBuffer または ProxyOfflineBuffer が大幅に削減された場合)、1回のハウスキーピングサイクルで古い情報が HousekeepingFrequency 時間の 4 倍以下で削除されます。したがって、HousekeepingFrequency が 1 の場合、1 サイクルごとに (最も古いエントリから開始して) 4 時間以内の古い情報が削除されます。 注: プロキシ起動時の負荷を下げるため、ハウスキーピングはプロキシ起動後 30 分間延期されます。したがって、HousekeepingFrequency が 1 の場合、プロキシ開始後の最初のハウスキーピングプロシージャは 30 分後に実行され、その後は 1 時間ごとに繰り返されます。 Zabbix 3.0.0 以降、HousekeepingFrequency を 0 に設定することで、自動ハウスキーピングを無効にすることができます。この場合、ハウスキーピング手順は housekeeper_execute ランタイム制御オプションによってのみ開始でき、1回のハウスキーピングサイクルで削除される古い情報の期間は、最後のハウスキーピングサイクル以降の期間の 4 倍ですが、4 時間以上 4 日以下となります。

パラメータ	必須	範囲	デフォルト	説明
Include	no			個々のファイルまたはディレクトリ内のすべてのファイルを設定ファイルに含めることができます。 指定したディレクトリに関連ファイルのみを含めるために、パターンマッチングでアスタリスクワイルドカード文字がサポートされています。例: /absolute/path/to/config/files/*.conf. 制限については 特記事項 を参照してください。
JavaGateway	no			Zabbix Java ゲートウェイの IP アドレス (またはホスト名)。Java ポーラーが開始されている場合にのみ必要です。
JavaGatewayPort	no	1024-32767	10052	Zabbix Java ゲートウェイがリスンするポート
ListenBacklog	no	0 - INT_MAX	SOMAXCONN	TCP キュー内の保留中の接続の最大数。 デフォルト値はハードコードされた定数で、システムによって異なります。 サポートされている最大値はシステムによって異なります。値が高すぎると”実装で指定された最大値”に切り捨てられる場合があります。
ListenIP	no		0.0.0.0	トラッパーがリスンするカンマ区切りの IP アドレスのリスト。 このパラメータが未設定の場合、トラッパーはすべてのネットワークインターフェイスでリスンします。
ListenPort	no	1024-32767	10051	トラッパーのリスンポート。
LoadModule	no			プロキシの起動時にロードするモジュール。モジュールは、プロキシの機能を拡張するために使用されます。 フォーマット: LoadModule=<module.so> LoadModule=<path/module.so> LoadModule=</abs_path/module. so> モジュールが LoadModulePath で指定されたディレクトリに配置されているか、パスがモジュール名の前にある必要があります。前のパスが絶対パス (/ で始まる) の場合、LoadModulePath は無視されます。 複数の LoadModule パラメータを含めることができます。 プロキシモジュールの場所へのフルパス。 デフォルトはコンパイルオプションによって異なります。 ログファイル名
LoadModulePath	no			
LogFile	LogType が file に設定されている場合:yes それ以外の場合:no			
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB) 0 - 自動ログローテーションを無効にします。 注: ログファイルのサイズ制限に達し、ファイルのローテーションが何らかの理由で失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。
LogRemoteCommands			0	実行されたシェルコマンドの警告としてのロギングを有効にします。 0 - 無効 1 - 有効 このパラメータは、Zabbix 3.4.0 以降でサポートされています。
LogType	no		file	ログ出力タイプ: file - LogFile パラメータで指定されたファイルにログを書き込みます。 system - ログを syslog に書き込みます。 console - ログを標準出力に書き込みます。 このパラメータ Zabbix 3.0.0 以降でサポートされています。
LogSlowQueries	no	0-3600000	0	データベースクエリがログに記録されるまでにかかる時間 (ミリ秒単位)。 0 - スロークエリをログに記録しません。 このオプションは DebugLevel=3 で有効になります。

パラメータ	必須	範囲	デフォルト	説明
PidFile	no		/tmp/zabbix_proxy.pid	PIDファイル名
ProxyLocalBufferno	no	0-720	0	プロキシは、データがサーバーと既に同期されている場合でも、N 時間ローカルにデータを保持します。 このパラメータは、ローカルデータがサードパーティのアプリケーションによって使用される場合に使用できます。
ProxyMode	no	0-1	0	プロキシ動作モード 0 - アクティブモードのプロキシ 1 - パッシブモードのプロキシ 注意アクティブなプロキシを使用している場合、(機密性の高い) プロキシ設定データは、Zabbix サーバーのトラッパポートにアクセスできる関係者が利用可能になる可能性があります。これが可能なのは、だれでもアクティブなプロキシになりすまして設定データを要求する可能性があるためです。認証は行われません。
ProxyOfflineBufferno	no	1-720	1	Zabbix サーバーとの接続がない場合、プロキシはデータを N 時間保持します。 古いデータは失われます。
Server	yes			ProxyMode がアクティブモードに設定されている場合: Zabbix サーバーの IP アドレスまたは DNS 名 (アドレス: ポート) または クラスター (アドレス: ポート; アドレス 2: ポート) を取得します ポートが指定されていない場合は、デフォルトのポートが使用されます。 クラスターノードはセミコロンで区切る必要があります。 ProxyMode がパッシブモードに設定されている場合: カンマ区切りの IP アドレスのリスト (オプションで CIDR 表記、または Zabbix サーバーの DNS 名)。着信接続は、ここにリストされているアドレスからのみ受け入れられます。IPv6 サポートが有効になっている場合、'127.0.0.1'、 '::127.0.0.1'、 '::ffff:127.0.0.1' は同等に扱われます。 '::/0' は、任意の IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' を使用して、任意の IPv4 アドレスを許可できます。 例: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
SNMPTrapperFileno	no		/tmp/zabbix_trap_receiver.pl	SNMPトラップデーモンからプロキシにデータを渡すために使用される一時ファイル。 zabbix_trap_receiver.pl または SNMPPTT 構成ファイルと同じである必要があります。
SocketDir	no		/tmp	内部 Zabbix サービスで使用される IPC ソケットを格納するディレクトリ。
SourceIP	no			このパラメータは Zabbix 3.4.0 以降でサポートされています。 次の送信元 IP アドレス: - Zabbix サーバーへの発信接続。 - エージェントレス接続 (VMware、SSH、JMX、SNMP、Telnet、および簡単なチェック); - HTTP エージェント接続。 - スクリプトアイテム JavaScript HTTP リクエスト。 - JavaScript HTTP リクエストの前処理。 - Vault への接続
SSHKeyLocationno	no			SSH チェックおよびアクション用の公開鍵と秘密鍵の場所
SSLCertLocationno	no			クライアント認証用の SSL クライアント証明書ファイルの場所。 このパラメータは Web モニタリングでのみ使用されます。
SSLKeyLocationno	no			クライアント認証用の SSL 秘密鍵ファイルの場所。 このパラメータは Web モニタリングでのみ使用されます。

パラメータ	必須	範囲	デフォルト	説明
SSLCALocation	no			SSL サーバー証明書検証用の認証局 (CA) ファイルの場所。 このパラメーターの値は、libcurl オプション CURLOPT_CAPATH として設定されることに注意してください。7.42.0 より前のバージョンの libcurl では、これは libcurl が OpenSSL を使用するようコンパイルされている場合にのみ効果があります。詳細については、 cURL Web ページ を参照してください。 このパラメータは、Zabbix 2.4.0 以降では Web 監視で使用され、Zabbix 3.0.0 以降では SMTP 認証で使用されます。
StartDBSyncers	no	1-100	4	history syncers の事前フォークされたインスタンスの数。 注: この値を変更するときは注意してください。値を大きくすると、メリットよりもデメリットが大きくなる可能性があります。
StartDiscoverers	no	0-250	1	discoverers の事前フォークされたインスタンスの数。
StartHistoryPollers	no	0-1000	1	history pollers の事前分岐されたインスタンスの数。 このパラメータは、Zabbix 5.4.0 以降でサポートされています。
StartHTTTPollers	no	0-1000	1	HTTP ポーラー の事前フォークされたインスタンスの数。
StartIPMIPollers	no	0-1000	0	IPMI ポーラー の事前フォークされたインスタンスの数。
StartJavaPollers	no	0-1000	0	Java ポーラー の事前フォークされたインスタンスの数。
StartODBCPollers	no	0-1000	1	ODBC ポーラー の事前フォークされたインスタンスの数。
StartPingers	no	0-1000	1	ICMP ピンガー の事前フォークされたインスタンスの数。
StartPollersUnreachable	no	0-1000	1	到達不能のポーラー ホスト (IPMI および Java を含む) の事前フォークされたインスタンスの数。 通常の IPMI または Java ポーラーが開始されている場合は、到達不能ホスト用のポーラーが少なくとも 1 つ実行されている必要があります。
StartPollers	no	0-1000	5	ポーラー の事前フォークされたインスタンスの数。
StartPreprocessors	no	1-1000	3	前処理ワーカー ¹ の事前フォークされたインスタンスの数 プリプロセッサワーカーが開始されると、プリプロセスマネージャプロセスが自動的に開始されます。 このパラメータは、Zabbix 4.2.0 以降でサポートされています。
StartSNMPTrappers	no	0-1	0	1 に設定すると、 SNMP トラッパー プロセスが開始されます。
StartTrappers	no	0-1000	5	トラッパー の事前フォークされたインスタンスの数。 トラッパーは、Zabbix 送信者とアクティブなエージェントからの着信接続を受け入れます。
StartVMwareCollectors	no	0-250	0	事前にフォークされた VMware コレクター インスタンスの数。
StatsAllowedIP	no			カンマ区切りの IP アドレスのリスト (オプションで CIDR 表記、または外部 Zabbix インスタンスの DNS 名)。統計情報のリクエストは、ここに記載されているアドレスからのみ受け付けます。このパラメータが設定されていない場合、統計リクエストは受け入れられません。 IPv6 サポートが有効になっている場合、 '127.0.0.1',':::127.0.0.1','::ffff:127.0.0.1' は同等に扱われ、 ':::0' は、任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' を使用して、任意の IPv4 アドレスを許可できます。 例: StatsAllowedIP=127.0.0.1,192.168.1.0/24,:::1,2001:db8::/32,zabbix.example.com このパラメータは、Zabbix 4.2.0 以降でサポートされています。
Timeout	no	1-30	3	エージェント、SNMP デバイス、または外部チェックを待機する時間を指定します (秒単位)

パラメータ	必須	範囲	デフォルト	説明
TLSCipherPSK	no			GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC: +AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NUL:+SIGN-ALL OpenSSL の例: kECDHEPSK+AES128:kPSK+AES128 このパラメータは Zabbix 4.4.7 からサポートされています。
TLSCipherPSK13no				TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 例: TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256 このパラメータは、Zabbix 4.4.7 以降でサポートされています。
TLSConnect	TLS 証明書または PSK パラメータが定義されている場合 (暗号化されていない接続の場合でも) またはアクティブプロキシの場合は yes、それ以外の場合は no			プロキシが Zabbix サーバーに接続する方法。アクティブプロキシに使用され、パッシブプロキシでは無視されます。指定できる値は 1 つだけです。 unencrypted - 暗号化なしで接続 (デフォルト) psk - TLS と事前共有キー (PSK) を使用して接続 cert - TLS と証明書 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSCRLFile	no			失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化された通信に使用されません。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSKeyFile	no			Zabbix コンポーネント間の暗号化された通信に使用される、プロキシの秘密鍵を含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSPSKFile	no			プロキシ事前共有キーを含むファイルのフルパス名。Zabbix サーバーとの暗号化通信に使用されます。 このパラメータは Zabbix 3.0.0 以降でサポートされています。
TLSPSKIdentity	no			Zabbix サーバーとの暗号化された通信に使用される事前共有キー ID 文字列。 このパラメータは Zabbix 3.0.0 以降でサポートされています。
TLSServerCertificateIssuer	no			許可されたサーバー証明書の発行者。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSServerCertificateSubject	no			許可されたサーバー証明書サブジェクト。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TmpDir	no		/tmp	テンポラリディレクトリ
TrapperTimeout	no	1-300	300	トラッパーが新しいデータの処理に費やすことができる秒数を指定します。
User	no		zabbix	システム上の特定の既存のユーザーに特権をドロップします。'root' として実行し、AllowRoot が無効になっている場合にのみ有効です。
UnavailableDelay	no	1-3600	60	使用不可 期間中にホストの可用性がチェックされる頻度 (秒単位)。
UnreachableDelay	no	1-3600	15	到達不能 期間中にホストの可用性がチェックされる頻度 (秒単位)。

パラメータ	必須	範囲	デフォルト	説明
UnreachablePeriod		1-3600	45	到達不能が何秒経過した後、ホストを使用不可として扱うか。
VaultDBPath	no			キー'password'および'username'によってデータベースの認証情報が取得されるポルトパス。 例: secret/zabbix/database このオプションは、DBUser および DBPassword が指定されていない場合にのみ使用できます。 このパラメータは、Zabbix 5.2.0 以降でサポートされています。
VaultToken	no			オプションの VaultDBPath 構成パラメータで指定されたパスへの読み取り専用権限を持つ Zabbix プロキシ専用生成されたはずの Vault 認証トークン。 VaultToken と VAULT_TOKEN 環境変数が同時に定義されている場合はエラーになります。 このパラメータは、Zabbix 5.2.0 以降でサポートされています。
VaultURL	no		https://127.0.0.1:8200	Vaultサーバーの HTTP[S] URL。SSLCALocation が指定されていない場合、システム全体の CA 証明書ディレクトリが使用されます。 このパラメータは、Zabbix 5.2.0 以降でサポートされています。
VMwareCacheSize		256K-2G	8M	VMware データを格納するための共有メモリサイズ。 VMware 内部チェック zabbix[vmware,buffer,...] を使用して、VMware キャッシュの使用状況を監視できます (内部チェックを参照)。 起動するように構成された vmware コレクタインスタンスがない場合、共有メモリは割り当てられないことに注意してください。)
VMwareFrequency		10-86400	60	単一の VMware サービスからデータを収集するまでの秒単位の遅延。 この遅延は、VMware 監視項目の最小の更新間隔に設定する必要があります。
VMwarePerfFrequency		10-86400	60	単一の VMware サービスからパフォーマンスカウンターの統計を取得するまでの秒単位の遅延。 この遅延は、VMware パフォーマンスカウンターを使用する VMware 監視項目の最小の更新間隔に設定する必要があります。
VMwareTimeoutno		1-300	10	VMware コレクターが VMware サービス (vCenter または ESX ハイパーバイザー) からの応答を待機する最大秒数。

3 Zabbix エージェント (UNIX)

概要

このセクションでは、Zabbix エージェント設定ファイル (zabbix_agentd.conf) でサポートされているパラメータの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix はBOMなしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が" #" で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Alias	no			<p>項目キーのエイリアスを設定します。長くて複雑な項目キーを、より小さくて単純なものに置き換えるために使用できます。複数の Alias パラメータが存在する場合があります。同じ Alias キーを持つ複数のパラメータを使用できます。異なる Alias キーが同じ項目キーを参照する場合があります。HostMetadataItem ではエイリアスを使用できますが、HostnameItem パラメータでは使用できません。</p> <p>< br> 例:</p> <ol style="list-style-type: none"> 1. ユーザー'zabbix' の ID を取得しています。 Alias=zabbix.userid:vfs.file.regex[<code>/etc/passwd,"^zabbix:([0-9]+)"</code>] <ul style="list-style-type: none"> これで、短縮キー zabbix.userid を使用してデータを取得できます。 2. デフォルトおよびカスタムパラメータで CPU 使用率を取得します。 Alias=cpu.util:system.cpu.util Alias=cpu.util[*]:system.cpu.util[*] <ul style="list-style-type: none"> これにより、cpu.util キーを使用してデフォルトパラメータで CPU 使用率を取得したり、cpu.util[all, idle, avg15] を使用して CPU 使用率に関する特定のデータを取得したりできます。< br> 3. 同じディスクバリアーアイテムを処理する複数のローレベルディスクバリアー ルールを実行しています。 Alias=vfs.fs.discovery[*]:vfs.fs.discovery vfs.fs.discovery を使用して、ルールごとに異なるパラメータを使用して複数の検出ルールを設定できます (例: vfs.fs.discovery[foo]、vfs.fs.discovery)。[bar] などパターンに一致する項目キーの実行を許可します。キーパターンは "*" 文字をサポートするワイルドカード式で、任意の数の任意の文字に一致します。 DenyKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限
AllowKey	no			<p>パターンに一致する項目キーの実行を許可します。キーパターンは "*" 文字をサポートするワイルドカード式で、任意の数の任意の文字に一致します。 DenyKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限</p>
AllowRoot	no		0	<p>エージェントを 'root' として開始できるようにします。許可しない設定でエージェントが 'root' によって開始された場合、エージェントは代わりにユーザー 'zabbix' で開始しようとします。通常のユーザーで開始した場合は影響ありません</p> <p>0 - 許可しない 1 - 許可する</p>
BufferSend	no	1-3600	5	データを N 秒より長くバッファに保持しない
BufferSize	no	2-65535	100	メモリバッファ内の値の最大数。バッファがいっぱいの場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。
DebugLevel	no	0-5	3	<p>デバッグレベルを指定します。</p> <p>0 - Zabbix プロセスの開始と停止に関する基本情報 1 - 重要な情報 2 - エラー情報 3 - 警告 4 - デバッグ用 (多くの情報を生成) 5 - 拡張デバッグ (さらに多くの情報を生成)</p>
DenyKey	no			<p>パターンに一致する項目キーの実行を拒否します。キーパターンは、任意の数の任意の文字に一致する "*" 文字をサポートするワイルドカード式です。 AllowKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限。</p>

パラメータ	必須	範囲	デフォルト	説明
EnableRemoteCommands			0	Zabbix サーバーからのリモートコマンドを許可するかどうか。このパラメータは非推奨です。代わりに、AllowKey=system.run[*] または DenyKey=system.run[*] を使用してください これは、値に応じた AllowKey/DenyKey パラメータの内部エイリアスです: 0 - DenyKey=system.run[*] 1 - AllowKey=system.run[*]
HostInterface	no	0-255 characters		ホストインターフェイスを定義するオプションのパラメータ。ホストインターフェイスは、ホストの自動登録プロセスで使用されます。 値が 255 文字の制限を超えると、エージェントはエラーを発生し、起動しません。 未定義の場合、HostInterfaceItem から値を取得します。Zabbix 4.4.0 以降でサポートされています。
HostInterfaceItem				ホストインターフェイスの取得に使用される項目を定義するオプションのパラメータ。 ホストインターフェイスは、ホストのautoregistration プロセスで使用されます。 自動登録リクエスト中、エージェントは警告メッセージをログに記録します。 このオプションは、HostInterface が定義されていない場合にのみ使用されます。 Zabbix 4.4.0 以降でサポートされています。
HostMetadata	no	0-255 characters		ホストのメタデータを定義するオプションのパラメータ。ホストのメタデータは、ホストの自動登録プロセス (アクティブなエージェント) でのみ使用されます。 未定義の場合、値は HostMetadataItem から取得されます。指定された値が制限を超えているか、UTF-8 以外の文字列である場合、エージェントはエラーを発生して起動しません。
HostMetadataItem				ホストメタデータの取得に使用される Zabbix エージェントアイテムを定義するオプションのパラメータ。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。 UserParameters とエイリアスをサポートします。 AllowKey/DenyKey の値に関係なく、system.run[] をサポートします。 HostMetadataItem の値は、自動登録の試行ごとに取得され、ホストの自動登録プロセス (アクティブなエージェント) でのみ使用されます。 自動登録リクエスト中、エージェント指定されたアイテムによって返された値が 255 文字の制限を超えている場合、警告メッセージがログに記録されます。 アイテムによって返された値は UTF-8 文字列である必要があり、それ以外の場合は無視されます。
Hostname	no		Set by HostnameItem	コンマで区切られた一意のホスト名のリスト。大文字と小文字が区別されます。 アクティブチェックに必要であり、サーバーで構成されているホスト名と一致する必要があります。未定義の場合、値は HostnameItem から取得されます。 使用できる文字: 英数字、'、'、'、'、および'-。 最大長: ホスト名ごとに 128 文字、行全体で 2048 文字。
HostnameItem	no		system.hostname	ホスト名の取得に使用される Zabbix エージェントアイテムを定義するオプションのパラメータ。このオプションは、ホスト名が定義されていない場合にのみ使用されます。 UserParameters またはエイリアスはサポートされていませんが、AllowKey/DenyKey の値に関係なく、system.run[] はサポートされています。 出力の長さは 512 KB に制限されています

パラメータ	必須	範囲	デフォルト	説明
Include	no			個々のファイルまたはディレクトリ内のすべてのファイルを構成ファイルに含めることができます。 指定したディレクトリに関連ファイルのみを含めるために、パターンマッチングでアスタリスクワイルドカード文字がサポートされています。例: /absolute/path/to/config/files/*.conf. 制限については、 特記事項 を参照してください。
ListenBacklog	no	0 - INT_MAX	SOMAXCONN	TCP キュー内の保留中の接続最大数。 デフォルト値はハードコードされた定数で、システムによって異なります。 サポートされている最大値はシステムによって異なります。値が高すぎると'実装で指定された最大値'に切り捨てられる場合があります。
ListenIP	no		0.0.0.0	エージェントがリスンする IP アドレスのカンマ区切りのリスト。 バージョン 1.8.3 以降では複数の IP アドレスがサポートされています。
ListenPort	no	1024-32767	10050	エージェントは、このポートでサーバーからの接続をリスンします。
LoadModule	no			エージェントの起動時にロードするモジュール。モジュールは、エージェントの機能を拡張するために使用されます。 形式: LoadModule=<module.so> LoadModule=<path/module.so> LoadModule=</abs_path/module.so>
LoadModulePath	no			モジュールが LoadModulePath で指定されたディレクトリに配置されているか、パスがモジュール名の前にある必要があります。前のパスが絶対パス ('/' で始まる) の場合、LoadModulePath は無視されます。 複数の LoadModule パラメータを含めることができます。 エージェントモジュールの場所へのフルパス。 デフォルトはコンパイルオプションによって異なります。
LogFile	LogType が file に設定されている場合 yes そうでない場合 no			ログファイル名
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB) 0 - 自動ログローテーションを無効にします。 注: ログファイルのサイズ制限に達し、ファイルのローテーションが何らかの理由で失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。
LogType	no		file	ログ出力タイプ: file - LogFile パラメータで指定されたファイルにログを書き込みます。 system - ログを syslog に書き込みます。 console - ログを標準出力に書き込みます。
LogRemoteCommands			0	このパラメータは Zabbix 3.0.0 以降でサポートされています。 実行されたシェルコマンドの警告としてのログを有効にします。 0 - 無効 1 - 有効 コマンドは、リモートで実行された場合にのみログに記録されます。System.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメーターによってローカルで起動された場合、ログエントリは作成されません。

パラメータ	必須	範囲	デフォルト	説明
MaxLinesPerSecond		1-1000	20	'ログ' および 'イベントログ' のアクティブチェックを処理するときに、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。 指定された値は、パラメータ 'maxlines' によって上書きされません。 'log' または 'eventlog' アイテムキーで提供されます。 注: Zabbix は、MaxLinesPerSecond で設定されたよりも 10 倍多くの新しい行を処理して、ログ項目で必要な文字列を検索します。
PidFile	no		/tmp/zabbix_agentd.pid	PID ファイルの名前
RefreshActiveChecks		60-3600	120	アクティブなチェックのリストが更新される頻度 (秒単位)。 アクティブなチェックの更新に失敗すると、60 秒後に次の更新が試行されることに注意してください。
Server	yes, if StartAgents is not explicitly set to 0			カンマ区切りの IP アドレスのリスト。オプションで CIDR 表記、または Zabbix サーバーと Zabbix プロキシのホスト名。 着信接続は、ここにリストされているホストからのみ受け入れられます。 IPv6 サポートが有効になっている場合は、'127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' は同等に扱われ、 ':::0' は任意の IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' を使用して任意の IPv4 アドレスを許可します。 -2.5.5). 例: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.domain スペースを使用できます。
ServerActive	no			アクティブチェックを取得する Zabbix サーバー/プロキシアドレスまたはクラスタ構成。 サーバー/プロキシアドレスは、コロンで区切られた IP アドレスまたは DNS 名とオプションのポートです。 クラスタ構成は、セミコロンで区切られた 1 つまたは複数のサーバーアドレスです。 複数の Zabbix サーバー/クラスタと Zabbix プロキシをカンマで区切って指定できます。 各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定しないでください。 Zabbix プロキシが指定されている場合は、Zabbix サーバー/クラスタそのプロキシには指定しないでください。 複数の独立した Zabbix サーバーを並行して使用するために、複数のカンマ区切りのアドレスを指定できます。スペースを使用できます。 ポートが指定されていない場合、デフォルトのポートが使用されます。 そのホストのポートが指定されている場合、IPv6 アドレスは角括弧で囲む必要があります。 ポートが指定されていない場合、IPv6 の角括弧アドレスはオプションです。 このパラメータが指定されていない場合、アクティブチェックは無効になります。 Zabbix プロキシの例: ServerActive=127.0.0.1:10051 複数のサーバーの例: ServerActive= 127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1] 高可用性の例: ServerActive=zabbix.cluster.node1 ;zabbix.cluster.node2:20051;zabbix.cluster.node3 2 つのクラスターと 1 つのサーバーによる高可用性の例: ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster.node1;zabbix.cluster2.node2,zabbix.domain

パラメータ	必須	範囲	デフォルト	説明
SourceIP	no			次の送信元 IP アドレス: - Zabbix サーバーまたは Zabbix プロキシへの発信接続。 - 一部のアイテム (web.page.get、net.tcp.port など) の実行中に接続を確立する
StartAgents	no	0-100	3	パッシブチェックを処理する事前フォークされた zabbix_agentd インスタンスの数。 0 に設定すると、パッシブチェックが無効になり、エージェントはどの TCP ポートでもリッスンしません。
Timeout	no	1-30	3	Timeout 秒の設定
TLSAccept	TLS 証明書または PSK パラメータが定義されている場合 (暗号化されていない接続の場合でも) は yes、それ以外の場合 は no			受け入れる着信接続。パッシブチェックに使用されます。コマンドで区切って複数の値を指定できます。 unencrypted - 暗号化なしで接続を受け入れる (デフォルト) psk - TLS と事前共有キー (PSK) を使用した接続を受け入れる cert - TLS と証明書による接続を受け入れる このパラメータは Zabbix 3.0.0 以降でサポートされています。
TLSCAFile	no			Zabbix コンポーネント間の暗号化された通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSCertFile	no			Zabbix コンポーネントとの暗号化された通信に使用される、エージェント証明書または証明書チェーンを含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSCipherAll	no			GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 例: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256 このパラメータは、Zabbix 4.4.7 以降でサポートされています。
TLSCipherAll13	no			TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。証明書および PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NONE:+SIGN-ALL:+CTYPE-X.509 OpenSSL の例: EECDH +aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128 このパラメータは、Zabbix 4.4.7 以降でサポートされています。
TLSCipherCert	no			GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。 GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC: +AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NONE:+SIGN-ALL:+CTYPE-X.509 OpenSSL の例: EECDH+aRSA+AES128:RSA+aRSA+AES128 このパラメータは、Zabbix 4.4.7 以降でサポートされています。

パラメータ	必須	範囲	デフォルト	説明
TLSCipherCert13no				<p>TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。</p> <p>このパラメータは、Zabbix 4.4.7 以降でサポートされています。</p>
TLSCipherPSK no				<p>GnuTLS 優先度文字列または OpenSSL (TLS 1.2) 暗号文字列。PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。</p> <p>GnuTLS の例: NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC: +AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NUL:+SIGN-ALL</p> <p>OpenSSL の例: kECDHEPSK+AES128:kPSK+AES128</p> <p>このパラメータは Zabbix 4.4.7 からサポートされています。</p>
TLSCipherPSK13no				<p>TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化のデフォルトの暗号スイート選択基準をオーバーライドします。</p> <p>例: TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256</p> <p>このパラメータは、Zabbix 4.4.7 以降でサポートされています。</p>
TLSConnect	TLS 証明書または PSK パラメータが定義されている場合 (暗号化されていない接続の場合でも) は yes それ以外の場合は no			<p>エージェントが Zabbix サーバーまたはプロキシに接続する方法。アクティブチェックに使用されます。指定できる値は 1 つだけです。</p> <p>unencrypted - 暗号化なしで接続 (デフォルト) psk - TLS と事前共有キー (PSK) を使用して接続 cert - TLS と証明書</p> <p>このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>
TLSCRLFile	no			<p>失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネントとの暗号化通信に使用されます。</p> <p>このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>
TLSKeyFile	no			<p>Zabbix コンポーネントとの暗号化された通信に使用されるエージェントの秘密鍵を含むファイルのフルパス名。</p> <p>このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>
TLSPSKFile	no			<p>Zabbix コンポーネントとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名。</p> <p>このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>
TLSPSKIdentity	no			<p>Zabbix サーバーとの暗号化された通信に使用される事前共有キー ID 文字列。</p> <p>このパラメータは Zabbix 3.0.0 以降でサポートされています。</p>
TLSServerCertificateIssuer				<p>許可されたサーバー (プロキシ) 証明書の発行者。</p> <p>このパラメータは Zabbix 3.0.0 以降でサポートされています。</p>
TLSServerCertificateSubject				<p>許可されたサーバー (プロキシ) 証明書のサブジェクト。</p> <p>このパラメータは Zabbix 3.0.0 以降でサポートされています。</p>
UnsafeUserParameters		0,1	0	<p>ユーザー定義パラメータの引数にすべての文字を渡せるか。</p> <p>0 - 許可しない 1 - 許可する</p> <p>次の文字は使用できません: \'\"'*?[]{}~\$!&;()> #@</p> <p>さらに、改行文字は使用できません。</p>
User	no		zabbix	<p>システム上の特定の既存のユーザーに特権をドロップします。</p> <p>'root' として実行し、AllowRoot が無効になっている場合にのみ効果があります。</p>

パラメータ	必須	範囲	デフォルト	説明
UserParameter no				監視するユーザー定義のパラメータ。複数のユーザー定義パラメータを指定できます。 形式: UserParameter=<キー>,<シェルコマンド> シェルコマンドは、空の文字列または EOL のみを返してはならないことに注意してください。 シェルコマンドには、相対パスを指定できます。 UserParameterDir パラメータが指定されている場合。 例: UserParameter=system.test,who wc -l UserParameter=check_cpu,./custom_script.sh
UserParameterDir				UserParameter コマンドのデフォルトの検索パス。使用すると、エージェントはコマンドを実行する前に、ここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドは、フルパスの代わりに相対 ./ プレフィックスを持つことができます。 許可されるエンタリは 1 つだけです。 例: UserParameterDir=/opt/myscripts

参照

1. [バージョン 2.0.0 以降のアクティブチェックとパッシブチェックの Zabbix エージェント構成の違い](#)

4 Zabbix エージェント 2 (UNIX)

概要

Zabbix エージェント 2 は新世代の Zabbix エージェントであり、Zabbix エージェントの代わりに使用できます。

このセクションでは、Zabbix エージェント 2 設定ファイル (zabbix_agent2.conf) でサポートされているパラメータの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は BOM なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " #" で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Alias	no			<p>項目キーのエイリアスを設定します。長くて複雑な項目キーを、より短くて単純なものに置き換えるために使用できます。複数の Alias パラメータが存在する場合があります。同じ Alias キーを持つ複数のパラメータを使用できます。異なる Alias キーが同じ項目キーを参照する場合があります。HostMetadataItem ではエイリアスを使用できませんが、HostnameItem パラメータでは使用できません。</p> <p>< br> 例:</p> <ol style="list-style-type: none"> 1. ユーザー zabbix の ID を取得します。 Alias=zabbix.userid:vfs.file.regex[/etc/passwd,"^zabbix:::([0-9]+)"", "\1"] これで短縮キー zabbix.userid を使用してデータを取得できます。 2. デフォルトおよびカスタムパラメータで CPU 使用率を取得します。 Alias=cpu.util:system.cpu.util Alias=cpu.util[*]:system.cpu.util[*]< br> これにより、cpu.util キーを使用してデフォルトパラメータで CPU 使用率を取得したり、cpu.util[all, idle, avg15] を使用して CPU 使用率に関する特定のデータを取得したりできます。< br> 3. 同じディスクバリエーションアイテムを処理する複数のローレベルディスクバリエーションルールを実行します。 Alias=vfs.fs.discovery[*]:vfs.fs.discovery vfs.fs.discovery を使用して、ルールごとに異なるパラメータを使用して複数のディスクバリエーションルールを設定できます (例: vfs.fs.discovery[foo]、vfs.fs.discovery)。[bar] など
AllowKey	no			<p>パターンに一致する項目キーの実行を許可します。キーパターンは、任意の数の任意の文字に一致する "*" 文字をサポートするワイルドカード式です。DenyKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは、出現順序に従って 1 つずつ処理されます。このパラメータは、Zabbix 5.0.0 以降でサポートされています。</p> <p>参照: エージェントチェックの制限</p>
BufferSend	no	1-3600	5	<p>値がバッファから Zabbix サーバーに送信される時間間隔 (秒単位)。</p> <p>バッファがいっぱいの場合、データはより早く送信されることに注意してください。</p>
BufferSize	no	2-65535	100	<p>メモリバッファ内の値の最大数。エージェントは、バッファがいっぱいになると、収集されたすべてのデータを Zabbix サーバーまたはプロキシに送信します。このパラメータは、永続バッファが無効になっている場合 (EnablePersistentBuffer=0) にのみ使用してください。</p>
ControlSocket	no		/tmp/agent.sockR'	<p>オプションを使用して実行時コマンドを送信するために使用される制御ソケット。</p>
DebugLevel	no	0-5	3	<p>デバッグレベルを指定します。</p> <p>0 - Zabbix プロセスの開始と停止に関する基本情報</p> <p>1 - 重要な情報</p> <p>2 - エラー情報</p> <p>3 - 警告</p> <p>4 - デバッグ用 (多くの情報を生成)</p> <p>5 - 拡張デバッグ (さらに多くの情報を生成)</p>

パラメータ	必須	範囲	デフォルト	説明
DenyKey	no			<p>パターンに一致する項目キーの実行を拒否します。キーパターンは、任意の数の任意の文字に一致する "*" 文字をサポートするワイルドカード式です。</p> <p>AllowKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。参照: エージェントチェックの制限。</p>
EnablePersistentBuffer		0-1	0	<p>アクティブなアイテムのローカル永続ストレージの使用を有効にします。</p> <p>0 - 無効 1 - 有効</p> <p>永続ストレージが無効の場合、メモリバッファが使用されます。</p>
ForceActiveChecksOnStart		0-1	0	<p>最初に受信した構成の再起動直後にアクティブチェックを実行します。</p> <p>0 - 無効 1 - 有効</p> <p>プラグイン構成パラメーターごとにも利用可能です。例: Plugins.Uptime.System.ForceActiveChecksOnStart=1 br> Zabbix 6.0.2 以降でサポートされています。</p>
HostInterface	no	0~255 文字		<p>ホストインターフェースを定義するオプションパラメータ。ホストインターフェースは、ホストの自動登録プロセスで使用されます。</p> <p>値が 255 文字の制限を超えると、エージェントはエラーを発行し、起動しません。</p> <p>定義されていない場合、値 HostInterfaceItem から取得されます。</p> <p>Zabbix 4.4.0 以降でサポートされています。</p>
HostInterfaceItem	no			<p>ホストインターフェースの取得に使用されるアイテムを定義するオプションパラメータ。</p> <p>ホストインターフェースは、ホストの自動登録プロセスで使用されます。</p> <p>自動登録リクエスト中に、指定されたアイテムによって返される値が 255 の制限を超えている場合、エージェントは警告メッセージを記録します。</p> <p>このオプションは、HostInterface が定義されていない場合にのみ使用されます。</p> <p>Zabbix 4.4.0 以降でサポートされています。</p>
HostMetadata	no	0~255 文字		<p>ホストのメタデータを定義するオプションのパラメーター。</p> <p>ホストメタデータは、ホストの自動登録プロセスで使用されます。</p> <p>指定された値が制限を超えているか、UTF-8 以外の文字列である場合、エージェントはエラーを発行し、起動しません。</p> <p>未定義の場合、値は HostMetadataItem から取得されます。</p>
HostMetadataItem	no			<p>ホストメタデータの取得に使用される項目を定義するオプションのパラメーター。ホストメタデータアイテム値は、ホスト自動登録プロセスの各自動登録試行で取得されます。</p> <p>自動登録リクエスト中に、指定されたアイテムによって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。</p> <p>このオプション HostMetadata が定義されていない場合にのみ使用されます。</p> <p>UserParameters とエイリアスをサポートします。</p> <p>AllowKey/DenyKey の値に関係なく、system.run[] をサポートします。</p> <p>アイテムによって返される値は、UTF-8 文字列でなければなりません。それ以外の場合は無視されます。</p>

パラメータ	必須	範囲	デフォルト	説明
Hostname	no		Hostnameltem で設定	コマンドで区切られた一意のホスト名のリスト。大文字と小文字が区別されます。 アクティブチェックに必要であり、サーバーで構成されているホスト名と一致する必要があります。未定義の場合、値は Hostnameltem から取得されます。 使用できる文字: 英数字、'.'、'_'、および '-'。 最大長: ホスト名ごとに 128 文字、行全体で 2048 文字。
Hostnameltem	no		system.hostnames	ホスト名が定義されていない場合に生成するために使用されるアイテム。Hostname が定義されている場合は無視されます。 UserParameters またはエイリアスはサポートされませんが、AllowKey/DenyKey の値に関係なく、system.run[] はサポートされます。 出力の長さは 512 KB に制限されています。
Include	no			個々のファイルまたはディレクトリ内のすべてのファイルを構成ファイルに含めることができます。 コンパイル時に変更されない限り、Zabbix はインストール中に /usr/local/etc にインクルードディレクトリを作成します。関連するもののみを含めるには指定したディレクトリ内のファイルの場合、パターンマッチングでアスタリスクワイルドカード文字がサポートされます。例: /absolute/path/to/config/files/*.conf Zabbix 6.0.0 以降、パスは zabbix_agent2.conf ファイルの場所からの相対パスにすることができます。 制限事項については 特記事項 を参照してください。
ListenIP	no		0.0.0.0	エージェントがリスンするカンマ区切りの IP アドレスのリスト。 Zabbix サーバーに接続している場合、アクティブなチェックのリストを取得するために、最初の IP アドレスが Zabbix サーバーに送信されます。
ListenPort	no	1024-32767	10050	エージェントは、この番号のポートでサーバーからの接続をリスンします。
LogFile	LogType が file に設定されている場合:yes それ以外の場合:no		/tmp/zabbix_agentd.log	LogType が 'file' の場合のログファイル名。
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB)。 0 - ログの自動ローテーションを無効にします。 注: ログファイルのサイズ制限に達し、ファイルのローテーションが何らかの理由で失敗した場合、既存のログファイルは切り捨てられ、新たに作成されます。
LogType	no		file	ログメッセージが書き込まれる場所を指定します。 system - syslog file - LogFile パラメータで指定されたファイル console - 標準出力
PersistentBufferFileno				Zabbix Agent2 が SQLite データベースを保持するファイル。完全なファイル名である必要があります。 このパラメータは、永続バッファが有効な場合 (EnablePersistentBuffer=1) にのみ使用されます。
PersistentBufferPeriod	no	1m-365d	1h	サーバーまたはプロキシへの接続がない場合に、データを保存する期間。古いデータは失われます。ログデータは保持されます。 このパラメータは、永続バッファが有効な場合にのみ使用されます (EnablePersistentBuffer=1)
PidFile	no		/tmp/zabbix_agentd.pid	pidファイル名

パラメータ	必須	範囲	デフォルト	説明
Plugin	no			Zabbix 6.0.0 以降、ほとんどのプラグインには独自の 設定ファイル があります。エージェント設定ファイルには、以下に示すプラグインパラメータが含まれています。
Plugins.Log.MaxLinesPerSecond		1-1000	20	'ログ' および 'イベントログ' のアクティブチェックを処理するときに、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。指定された値は、パラメータ 'maxlines' によって上書きされます。 'log' または 'eventlog' アイテムキーで提供されます。 注: Zabbix は、MaxLinesPerSecond で設定されたよりも 10 倍多くの新しい行を処理して、ログ項目に必要な文字列を検索します。 このパラメータは 4.4.2 以降でサポートされており、MaxLinesPerSecond を置き換えます。
Plugins.SystemRun.LogRemoteCommands			0	実行されたシェルコマンドのログを警告として有効にします。 0 - 無効 1 - 有効 コマンドは、リモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合、ログエントリは作成されません。 このパラメータは 4.4.2 以降でサポートされ、LogRemoteCommands を置き換えます。
PluginSocket	no		/tmp/agent.plugins	可能なプラグイン通信用の Unix ソケットパス
PluginTimeout	no	1-30	グローバルタイムアウト	ロード可能なプラグインとの接続のタイムアウト。
RefreshActiveChecks	no	60-3600	120	アクティブチェックのリストを更新する頻度 (秒単位)。アクティブチェックの更新に失敗すると、60 秒後に次の更新が試行されることに注意してください。
Server	yes			カンマ区切りの IP アドレスのリスト (オプションで CIDR 表記、または Zabbix サーバーと Zabbix プロキシの DNS 名)。着信接続は、ここにリストされているホストからのみ受け入れられます。 IPv6 サポートが有効になっている場合は、'127.0. 0.1'、':::ffff:127.0.0.1' は同等に扱われ、':::/0' はすべての IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' はすべての IPv4 アドレスを許可するために使用できます。 例: Server=127.0.0.1,192.168.1.0/24,:::1,2001:db8::/32,zabbix.example.com スペースを使用できます。

	必須	範囲	デフォルト	説明
ServerActive	no			<p>アクティブなチェックを取得する Zabbix サーバー/プロキシアドレスまたはクラスター構成。</p> <p>サーバー/プロキシアドレスは、コロンで区切られた IP アドレスまたは DNS 名とオプションのポートです。</p> <p>クラスター構成は、セミコロンで区切られた 1 つ以上のサーバーアドレスです。</p> <p>複数の Zabbix サーバー/クラスターおよび Zabbix プロキシをカンマで区切って指定できます。</p> <p>各 Zabbix サーバー/クラスターから複数の Zabbix プロキシを指定しないでください。</p> <p>Zabbix プロキシが指定されている場合、そのプロキシの Zabbix サーバー/クラスターは指定しないでください。</p> <p>カンマ区切りの複数のアドレスを指定して、複数の独立した Zabbix サーバーを並行して使用できます。スペースは使用できません。</p> <p>ポートが指定されていない場合、デフォルトのポートが使用されます。</p> <p>そのホストのポートを指定する場合は、IPv6 アドレスを角括弧で囲む必要があります。</p> <p>ポートが指定されていない場合、IPv6 アドレスの角括弧はオプションです。</p> <p>このパラメータが指定されていない場合、アクティブチェックは無効になります。</p> <p>Zabbix プロキシの例: ServerActive=127.0.0.1:10051</p> <p>複数のサーバーの例: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]</p> <p>高可用性の例: ServerActive= zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3</p> <p>2 つのクラスターと 1 つのサーバーによる高可用性の例: ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.clu</p>
SourceIP	no			<p>次の送信元 IP アドレス:</p> <ul style="list-style-type: none"> - Zabbix サーバーまたは Zabbix プロキシへの発信接続。 - 一部のアイテム (web.page.get、net.tcp.port など) の実行中に接続を確立する
StatusPort	no	1024-32767		<p>設定されている場合、エージェントはこのポートで HTTP ステータスリクエストをリッスンします (http://localhost:<port>/status)</p>
Timeout	no	1-30	3	<p>処理にかかる時間は Timeout 秒以内です。</p>
TLSAccept	TLS 証明書 または PSK パラメータ が定義され ている場合 (暗号化され ていない接 続の場合で も):yes、そ れ以外の場 合:no			<p>受け入れる着信接続。パッシブチェックに使用されます。コンマで区切って複数の値を指定できます。</p> <p>unencrypted - 暗号化なしで接続を受け入れる (デフォルト)</p> <p>psk - TLS と事前共有キー (PSK) を使用した接続を受け入れる</p> <p>cert - TLS と証明書による接続を受け入れる</p>
TLSCAFile	no			<p>Zabbix コンポーネント間の暗号化された通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。</p>
TLSCertFile	no			<p>Zabbix コンポーネントとの暗号化通信に使用される、エージェント証明書または証明書チェーンを含むファイルのフルパス名。</p>

パラメータ	必須	範囲	デフォルト	説明
TLSConnect	TLS 証明書 または PSK パラメータ が定義され ている場合 (暗号化され ていない接 続の場合で も):yes、そ れ以外の場 合:no			エージェントが Zabbix サーバーまたはプロキシに接続する 方法。アクティブチェックに使用されます。指定できる値は 1 つだけです。 unencrypted - 暗号化なしで接続 (デフォルト) psk - TLS と事前共有キー (PSK) を使用して接続 cert - TLS と証明書
TLSCRLFile	no			失効した証明書を含むファイルのフルパス名。このパラメ ータは、Zabbix コンポーネントとの暗号化通信に使用されま す。
TLSCKeyFile	no			Zabbix コンポーネントとの暗号化された通信に使用される エージェントの秘密鍵を含むファイルのフルパス名。
TLSPSKFile	no			Zabbix コンポーネントとの暗号化された通信に使用される エージェントの事前共有キーを含むファイルのフルパス名。
TLSPSKIdentity	no			Zabbix サーバーとの暗号化された通信に使用される事前共有 キー ID 文字列。
TLSServerCertIssuer	no			許可されたサーバー (プロキシ) 証明書の発行者。
TLSServerCertSubject	no			許可されたサーバー (プロキシ) 証明書のサブジェクト。
UnsafeUserParameters	no	0,1	0	ユーザー定義パラメータの引数にすべての文字を渡すことが できます。 ただし次の文字は使用できません: `' " * ? [] { } ~ \$! & ; () > # @` さらに、改行文字は使用できません。 監視するユーザー定義のパラメーター。いくつかのユーザー 定義パラメーターが存在する場合があります。 書式: UserParameter=< キー >,< シェルコマンド > シェルコマンドは、空の文字列または EOL のみを返してはな らないことに注意してください。 UserParameterDir パラメータが指定されている場合、シェ ルコマンドは相対パスを持つことができます。 例: UserParameter=system.test,who wc -l UserParameter=check_cpu,./custom_script.sh
UserParameter	no			UserParameter コマンドのデフォルトの検索パス。使用する と、エージェントはコマンドを実行する前に、ここで指定さ れたディレクトリに作業ディレクトリを変更します。これに より、UserParameter コマンドは、フルパスの代わりに相対 ./プレフィックスを持つことができます。 許可されるエンタリは 1 つだけです。 例: UserParameterDir=/opt/myscripts
UserParameterDir	no			

5 Zabbix エージェント (Windows)

概要

このセクションでは、Zabbix エージェント (Windows) 構成ファイル (zabbix_agent.conf) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は BOM なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " #" で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Alias	no			<p>項目キーのエイリアスを設定します。長くて複雑な項目キーを、より短くて単純なものに置き換えるために使用できます。複数の Alias パラメータが存在する場合があります。同じ Alias キーを持つ複数のパラメータを使用できます。異なる Alias キーで同じアイテムキーを参照できます。HostMetadataltem ではエイリアスを使用できますが、Hostnameltem または PerfCounter パラメータでは使用できません。</p> <p>例:</p> <ol style="list-style-type: none"> 1. サーバーからページングファイルの使用率をパーセントで取得します。 Alias=pg_usage:perf_counter(\Paging File(_Total)\% Usage) これにより短縮キー pg_usage をデータの取得に使用できます。 2. デフォルトおよびカスタムパラメータで CPU 負荷を取得します。 Alias=cpu.load:system.cpu.load Alias=cpu.load[*]:system.cpu.load[*]
 これにより cpu.load キーを使用してデフォルトパラメータの CPU 使用率を取得したり、cpu.load[percpu,avg15] を使用して CPU 負荷に関する特定のデータを取得したりできます。 3. 同じディスクバリアーアイテムを処理する複数の ローレベルディスクバリアー ルールを実行しています。 Alias=vfs.fs.discovery[*]:vfs.fs.discovery vfs.fs.discovery を使用して、ルールごとに異なるパラメータを使用して複数の検出ルールを設定できます (例: vfs.fs.discovery[foo]、vfs.fs.discovery)。[bar] などパターンに一致する項目キーの実行を許可します。キーパターンは "*" 文字をサポートするワイルドカード式で、任意の数の任意の文字に一致します。 DenyKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限。
AllowKey	no			<p>パターンに一致する項目キーの実行を許可します。キーパターンは "*" 文字をサポートするワイルドカード式で、任意の数の任意の文字に一致します。 DenyKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限。</p>
BufferSend	no	1-3600	5	データを N 秒より長くバッファに保持しません
BufferSize	no	2-65535	100	メモリバッファ内の値の最大数。バッファがいっぱいの場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。
DebugLevel	no	0-5	3	<p>デバッグレベルを指定します。</p> <p>0 - Zabbix プロセスの開始と停止に関する基本情報</p> <p>1 - 重要な情報</p> <p>2 - エラー情報</p> <p>3 - 警告</p> <p>4 - デバッグ用 (多くの情報を生成)</p> <p>5 - 拡張デバッグ (さらに多くの情報を生成)</p>
DenyKey	no			<p>パターンに一致する項目キーの実行を拒否します。キーパターンは、任意の数の任意の文字に一致する "*" 文字をサポートするワイルドカード式です。 AllowKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順に従って 1 つずつ処理されます。このパラメータは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限。</p>

パラメータ	必須	範囲	デフォルト	説明
EnableRemoteCommands			0	Zabbix サーバーからのリモートコマンドを許可するかどうか。このパラメータは非推奨です。代わりに、AllowKey=system.run[*] または DenyKey=system.run[*] を使用してください これは、値に応じた AllowKey/DenyKey パラメータの内部エイリアスです: 0 - DenyKey=system.run[*] 1 - AllowKey=system.run[*]。
HostInterface	no	0-255 文字		ホストインターフェースを定義するオプションパラメータ。ホストインターフェースは、ホストの自動登録プロセスで使用されます。 値が 255 文字の制限を超えると、エージェントはエラーを発生し、起動しません。 定義されていない場合、値 HostInterfaceItem から取得されます。 Zabbix 4.4.0 以降でサポートされています。
HostInterfaceItem				ホストインターフェースの取得に使用されるアイテムを定義するオプションパラメータ。 ホストインターフェースは、ホストの自動登録プロセスで使用されます。 自動登録リクエスト中に、指定されたアイテムによって返される値が 255 の制限を超えている場合、エージェントは警告メッセージを記録します。 このオプションは、HostInterface が定義されていない場合にのみ使用されます。 Zabbix 4.4.0 以降でサポートされています。
HostMetadata	no	0-255 文字		ホストのメタデータを定義するオプションのパラメータ。ホストのメタデータは、ホストの自動登録プロセス (アクティブなエージェント) でのみ使用されます。 未定義の場合、値は HostMetadataItem から取得されます。 指定された値が制限を超えているか、UTF-8 以外の文字列である場合、エージェントはエラーを発生して起動しません。
HostMetadataItem				ホストメタデータの取得に使用される Zabbix エージェントアイテムを定義するオプションのパラメータ。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。 UserParameters、パフォーマンスカウンター、およびエイリアスをサポートします。EnableRemoteCommands 値に関係なく、system.run[] をサポートします。 HostMetadataItem 値は、自動登録の試行ごとに取得され、ホスト自動登録プロセス (アクティブなエージェント) でのみ使用されます。 自動登録リクエスト中、エージェント指定されたアイテムによって返された値が 255 文字の制限を超えている場合、警告メッセージがログに記録されます。 アイテムによって返された値は UTF-8 文字列である必要があり、それ以外の場合は無視されます。
Hostname	no		HostnameItem で設定	カンマ区切りの一意のホスト名のリスト。未定義の場合は HostnameItem から値を取得します。 使用できる文字: 英数字、'/'、'_' および '-' 最大長: ホスト名ごとに 128 文字、行全体で 2048 文字。
HostnameItem	no		system.hostname	ホスト名の取得に使用される Zabbix エージェントアイテムを定義するオプションのパラメータ。このオプションは、ホスト名が定義されていない場合にのみ使用されます。 UserParameters、パフォーマンスカウンター、またはエイリアスはサポートされませんが、EnableRemoteCommands 値に関係なく system.run[] はサポートされます。 出力の長さは次のとおりです。512KB に制限されています。 より詳細な説明も参照してください。

パラメータ	必須	範囲	デフォルト	説明
Include	no			個々のファイルまたはディレクトリ内のすべてのファイルを構成ファイルに含めることができます。 指定したディレクトリに関連ファイルのみを含めるために、パターンマッチングでアスタリスクワイルドカード文字がサポートされています。例: /absolute/path/to/config/files/*.conf. 制限については、 特記事項 を参照してください。
ListenBacklog	no	0 - INT_MAX	SOMAXCONN	TCP キュー内の保留中の接続の最大数。 デフォルト値はハードコードされた定数で、システムによって異なります。 サポートされている最大値もシステムによって異なります。値が高すぎると'実装で指定された最大値'に切り捨てられる場合があります。
ListenIP	no		0.0.0.0	エージェントがリスンする IP アドレスのカンマ区切りリスト
ListenPort	no	1024-32767	10050	エージェントは、このポートでサーバーからの接続をリスンします。
LogFile	LogType が file に設定されている場合:yes それ以外:no		C:\zabbix_agentd\log	エージェントログファイル名
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB)。 0 - ログの自動ローテーションを無効にします。 注: ログファイルのサイズ制限に達し、ファイルのローテーションが何らかの理由で失敗した場合、既存のログファイルは切り捨てられ、新たなファイルで開始されます。
LogType	no		file	ログ出力タイプ: file - LogFile パラメーターで指定されたファイルにログを書き込みます。 system - ログを書き込みます。Windows イベントログ、console - ログを標準出力に書き込みます。 これパラメータは Zabbix 3.0.0 以降でサポートされています。
LogRemoteCommands			0	実行されたシェルコマンドの警告ログを有効にします。 0 - 無効 1 - 有効
MaxLinesPerSecond		1-1000	20	エージェントが'log'、'logrt' および'eventlog' アクティブチェックを処理する Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。 指定された値は、パラメータ'maxlines'によって上書きされません。 'log','logrt' または'eventlog' 項目キーで提供されます。 注: Zabbix は、MaxLinesPerSecond で設定されたよりも 10 倍多くの新しい行を処理して、ログ項目に必要な文字列を検索します。
PerfCounter	no			指定された < 期間 >(秒単位) のシステムパフォーマンスカウンター <perf_counter_path> の平均値である新しいパラメーター <parameter_name> を定義します。 構文: <parameter_name>," <perf_counter_path>",<period> たとえば、過去 1 分間の 1 秒あたりの平均プロセッサ割り込み数を取得する場合は、新しいパラメータ"interrupts" を次のように定義できます。 PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60 パフォーマンスカウンターパスを二重引用符で囲むことに注意してください。 パラメーター名 (interrupts) は、作成時にアイテムキーとして使用されます。 平均値を計算するためのサンプルが毎秒取得されます。 "typeperf -qx" を実行すると、Windows で使用可能なすべてのパフォーマンスカウンターのリストを取得できます。

パラメータ	必須	範囲	デフォルト	説明
PerfCounterEn	no			<p>指定された期間 < 期間 > (秒単位) のシステムパフォーマンスカウンター <perf_counter_path> の平均値である新しいパラメータ <parameter_name> を定義します。</p> <p>構文: <parameter_name>," <perf_counter_path>",<period></p> <p>PerfCounter と比較して、perfcounter のパスは英語である必要があります。</p> <p>Windows Server 2008/Vista 以降でのみサポートされています。</p> <p>例: 過去 1 分間の 1 秒あたりのプロセッサ割り込みの平均数を取得する場合は、新しいパラメータ "interrupts" を次のように定義できます。</p> <p>PerfCounterEn = interrupts,"\\Processor(0)\\Interrupts/sec", 60</p> <p>パフォーマンスカウンターパスを二重引用符で囲んでください。パラメーター名 (interrupts) は、アイテムを作成するときにアイテムキーとして使用されます。</p> <p>平均値を計算するためのサンプルは、1 秒ごとに取得されます。</p> <p>< br> 次のレジストリキーを表示すると、英語の文字列のリストを見つけることができます:</p> <p>HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Perflib\\009.</p> <p>このパラメータは Zabbix 4.0.13 および 4.2.7 以降でサポートされています。</p>
RefreshActiveChecks		60-3600	120	<p>アクティブなチェックのリストが更新される頻度 (秒単位)。</p> <p>アクティブなチェックの更新に失敗すると、60 秒後に次の更新が試行されることに注意してください。</p>
Server	StartAgents が明示的に 0 に設定されて いない場合: yes			<p>カンマ区切りの IP アドレスのリスト (オプションで CIDR 表記、または Zabbix サーバーのホスト名)。</p> <p>着信接続は、ここにリストされているホストからのみ受け入れられます。</p> <p>IPv6 サポートが有効になっている場合は、'127.0.0.1',':::127.0.0.1','::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。</p> <p>'0.0.0.0/0' を使用して、任意の IPv4 アドレスを許可できます。</p> <p>"IPv4 互換の IPv6 アドレス"(0000::/96 プレフィックス) はサポートされていますが、RFC4291 によって非推奨になっていることに注意してください。</p> <p>例: Server=127.0.0.1,192.168.1.0/24,:::1,2001:db8::/32,zabbix.domain スペースを使用できます。</p>

パラメータ	必須	範囲	デフォルト	説明
ServerActive	no	(*)		<p>アクティブチェックを取得する Zabbix サーバー/プロキシアドレスまたはクラスタ構成。</p> <p>サーバー/プロキシアドレスは、コロンで区切られた IP アドレスまたは DNS 名とオプションのポートです。</p> <p>クラスタ構成は、セミコロンで区切られた 1 つまたは複数のサーバーアドレスです。</p> <p>br> 複数の Zabbix サーバー/クラスタと Zabbix プロキシをカンマで区切って指定できます。</p> <p>各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定しないでください。</p> <p>Zabbix プロキシが指定されている場合は、Zabbix サーバー/クラスタそのプロキシには指定しないでください。</p> <p>複数の独立した Zabbix サーバーを並行して使用するために、複数のカンマ区切りのアドレスを指定できます。スペースを使用できます。</p> <p>ポートが指定されていない場合、デフォルトのポートが使用されます。</p> <p>そのホストのポートが指定されている場合、IPv6 アドレスは角括弧で囲む必要があります。</p> <p>ポートが指定されていない場合、IPv6 の角括弧アドレスはオプションです。</p> <p>このパラメータが指定されていない場合、アクティブチェックは無効になります。</p> <p>Zabbix プロキシの例: ServerActive=127.0.0.1:10051</p> <p>複数のサーバーの例: ServerActive= 127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]</p> <p>高可用性の例: ServerActive=zabbix.cluster.node1 ;zabbix.cluster.node2:20051;zabbix.cluster.node3</p> <p>2 つのクラスターと 1 つのサーバーによる高可用性の例: ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster.node1;zabbix.cluster2.node2,zabbix.domain</p>
SourceIP	no			<p>ソース IP アドレス:</p> <ul style="list-style-type: none"> - Zabbix サーバーまたは Zabbix プロキシへの発信接続; - いくつかのアイテム (web.page.get、net.tcp.port など) の実行中に接続を確立する
StartAgents	no	0-63 (*)	3	<p>パッシブチェックを処理する事前フォークされた zabbix_agentd インスタンスの数。</p> <p>0 に設定すると、パッシブチェックが無効になり、エージェントはどの TCP ポートでもリッスンしません。</p>
Timeout	no	1-30	3	<p>処理に Timeout 秒を超えないようにします</p>
TLSAccept	TLS 証明書または PSK パラメータが定義されている場合 (暗号化されていない接続の場合でも):yes、それ以外の場合:no			<p>受け入れる着信接続。パッシブチェックに使用されます。コマンドで区切って複数の値を指定できます。</p> <p>unencrypted - 暗号化なしで接続を受け入れる (デフォルト)</p> <p>psk - TLS と事前共有キー (PSK) を使用した接続を受け入れる</p> <p>cert - TLS と証明書による接続を受け入れる</p> <p>このパラメータは Zabbix 3.0.0 以降でサポートされています。</p>
TLSCAFile	no			<p>Zabbix コンポーネント間の暗号化された通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。</p> <p>このパラメータは、Zabbix 3.0.0 以降でサポートされています。</p>

パラメータ	必須	範囲	デフォルト	説明
TLSCertFile	no			Zabbix コンポーネントとの暗号化された通信に使用される、エージェント証明書または証明書チェーンを含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSConnect	TLS 証明書または PSK パラメータが定義されている場合 (暗号化されていない接続の場合でも):yes、それ以外の場合:no			エージェントが Zabbix サーバーまたはプロキシに接続する方法。アクティブチェックに使用されます。指定できる値は 1 つだけです。 unencrypted - 暗号化なしで接続 (デフォルト) psk - TLS と事前共有キー (PSK) を使用して接続 cert - TLS と証明書 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSCRLFile	no			失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネントとの暗号化通信に使用されます。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSKeyFile	no			Zabbix コンポーネントとの暗号化された通信に使用されるエージェントの秘密鍵を含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSPSKFile	no			Zabbix コンポーネントとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名。 このパラメータは、Zabbix 3.0.0 以降でサポートされています。
TLSPSKIdentity	no			Zabbix サーバーとの暗号化された通信に使用される事前共有キー ID 文字列。 このパラメータは Zabbix 3.0.0 以降でサポートされています。
TLSIssuer				許可されたサーバー (プロキシ) 証明書の発行者。 このパラメータは Zabbix 3.0.0 以降でサポートされています。
TLSSubject				許可されたサーバー (プロキシ) 証明書のサブジェクト。 このパラメータは Zabbix 3.0.0 以降でサポートされています。
UnsafeUserParameters		0-1	0	ユーザー定義パラメータの引数にすべての文字を渡すことができるようにします。 0 - 許可しない 1 - 許可する ただし次の文字は使用できません: \'\"'*?[]{}~\$!&;()> #@ さらに、改行文字は使用できません。 監視するユーザー定義のパラメーター。複数のユーザー定義パラメータが存在する場合があります。 形式: UserParameter=< キー >,< シェルコマンド > シェルコマンドは、空の文字列または EOL のみを返してはならないことに注意してください。 UserParameterDir パラメータが指定されている場合、シェルコマンドには相対パスが含まれる場合があります。 例: UserParameter=system.test,who wc -l UserParameter=check_cpu,./custom_script .sh
UserParameter	no			UserParameter コマンドのデフォルトの検索パス。使用すると、エージェントはコマンドを実行する前に、ここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドは、フルパスの代わりに相対 ./ プレフィックスを持つことができます。 許可されるエンタリは 1 つだけです。
UserParameterDir	no			例: UserParameterDir=/opt/myscripts

Note:

(*) ServerActive にリストされているアクティブなサーバーの数と、StartAgents で指定されたパッシブチェック用の事前にフォークされたインスタンスの数を加えた数は、64 未満である必要があります。

参照

- バージョン 2.0.0 以降のアクティブチェックとパッシブチェックの Zabbix エージェント構成の違い。

6 Zabbix エージェント 2 (Windows)

概要

Zabbix エージェント 2 は新世代の Zabbix エージェントであり、Zabbix エージェントの代わりに使用できます。

このセクションでは Zabbix エージェント 2 設定ファイル (zabbix_agent2.win.conf) でサポートされているパラメータの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は BOM なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Alias	no			<p>項目キーのエイリアスを設定します。長くて複雑な項目キーを、より短くて単純なものに置き換えるために使用できます。複数の Alias パラメータが存在する場合があります。同じ Alias キーを持つ複数のパラメータを使用できます。異なる Alias キーが同じ項目キーを参照する場合があります。HostMetadataItem ではエイリアスを使用できますが、HostNameItem パラメータでは使用できません。</p> <p>< br> 例:</p> <ol style="list-style-type: none"> ユーザー zabbix の ID を取得します。 Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix::([0-9]+)"",\1] これで短縮キー zabbix.userid を使用してデータを取得できます。 デフォルトおよびカスタムパラメータで CPU 使用率を取得します。 Alias=cpu.util:system.cpu.util Alias=cpu.util[*]:system.cpu.util[*]< br> これにより、cpu.util キーを使用してデフォルトパラメータで CPU 使用率を取得したり、cpu.util[all, idle, avg15] を使用して CPU 使用率に関する特定のデータを取得したりできます。< br> 同じディスクバリエーションアイテムを処理する複数のローレベルディスクバリエーションルールを実行します。 Alias=vfs.fs.discovery[*]:vfs.fs.discovery vfs.fs.discovery を使用して、ルールごとに異なるパラメータを使用して複数のディスクバリエーションルールを設定できます (例: vfs.fs.discovery[foo]、vfs.fs.discovery)。[bar] など

パラメータ	必須	範囲	デフォルト	説明
AllowKey	no			パターンに一致する項目キーの実行を許可します。キーパターンは、任意の数の任意の文字に一致する "*" 文字をサポートするワイルドカード式です。 DenyKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメーターは、出現順序に従って 1 つずつ処理されます。 このパラメーターは、Zabbix 5.0.0 以降でサポートされています。 参照: エージェントチェックの制限
BufferSend	no	1-3600	5	値がバッファから Zabbix サーバーに送信される時間間隔 (秒単位)。 バッファがいっぱいの場合、データはより早く送信されることに注意してください。
BufferSize	no	2-65535	100	メモリバッファ内の値の最大数。エージェントは、バッファがいっぱいになると、収集されたすべてのデータを Zabbix サーバーまたはプロキシに送信します。 このパラメーターは、永続バッファが無効になっている場合 (EnablePersistentBuffer=0) にのみ使用してください。
ControlSocket	no		\\.\pipe\agent.sock	ソックオプションを使用して実行時コマンドを送信するために使用される制御ソケット。
DebugLevel	no	0-5	3	デバッグレベルを指定します。 0 - Zabbix プロセスの開始と停止に関する基本情報 1 - 重要な情報 2 - エラー情報 3 - 警告 4 - デバッグ用 (多くの情報を生成) 5 - 拡張デバッグ (さらに多くの情報を生成)
DenyKey	no			パターンに一致する項目キーの実行を拒否します。キーパターンは、任意の数の任意の文字に一致する "*" 文字をサポートするワイルドカード式です。 AllowKey と組み合わせて、複数のキー一致ルールを定義できます。パラメーターは出現順に従って 1 つずつ処理されます。 このパラメーターは Zabbix 5.0.0 からサポートされています。 参照: エージェントチェックの制限
EnablePersistentBuffer		0-1	0	アクティブなアイテムのローカル永続ストレージの使用を有効にします。 0 - 無効 1 - 有効 永続ストレージが無効の場合、メモリバッファが使用されます。
ForceActiveChecksOnStart		0-1	0	最初に受信した構成の再起動直後にアクティブチェックを実行します。 0 - 無効 1 - 有効 プラグイン構成パラメーターごとにも利用可能です。例: Plugins.Uptime.System.ForceActiveChecksOnStart=1 br> Zabbix 6.0.2 以降でサポートされています。
HostInterface	no	0~255 文字		ホストインターフェースを定義するオプションパラメーター。ホストインターフェースは、ホストの自動登録プロセスで使用されます。 値が 255 文字の制限を超えると、エージェントはエラーを発行し、起動しません。 定義されていない場合、値 HostInterfaceItem から取得されます。 Zabbix 4.4.0 以降でサポートされています。

パラメータ	必須	範囲	デフォルト	説明
HostInterfaceltem	no			<p>ホストインターフェースの取得に使用されるアイテムを定義するオプションパラメータ。</p> <p>ホストインターフェースは、ホストの自動登録プロセスで使用されます。</p> <p>自動登録リクエスト中に、指定されたアイテムによって返される値が 255 の制限を超えている場合、エージェントは警告メッセージを記録します。</p> <p>このオプションは、HostInterface が定義されていない場合にのみ使用されます。</p> <p>Zabbix 4.4.0 以降でサポートされています。</p>
HostMetadata	no	0~255 文字		<p>ホストのメタデータを定義するオプションのパラメーター。</p> <p>ホストメタデータは、ホストの自動登録プロセスで使用されます。</p> <p>指定された値が制限を超えているか、UTF-8 以外の文字列である場合、エージェントはエラーを発行し、起動しません。</p> <p>未定義の場合、値は HostMetadataltem から取得されます。</p>
HostMetadataltem	no			<p>ホストメタデータの取得に使用される項目を定義するオプションのパラメーター。ホストメタデータアイテム値は、ホスト自動登録プロセスの各自動登録試行で取得されます。</p> <p>自動登録リクエスト中に、指定されたアイテムによって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。</p> <p>このオプション HostMetadata が定義されていない場合にのみ使用されます。</p> <p>UserParameters とエイリアスをサポートします。</p> <p>EnableRemoteCommands の値に関係なく、system.run[] をサポートします。</p> <p>アイテムによって返される値は、UTF-8 文字列でなければなりません。それ以外の場合は無視されます。</p>
Hostname	no		Hostnameltem で設定	<p>コマンドで区切られた一意のホスト名のリスト。大文字と小文字が区別されます。</p> <p>アクティブチェックに必要であり、サーバーで構成されているホスト名と一致する必要があります。未定義の場合、値は Hostnameltem から取得されます。</p> <p>使用できる文字: 英数字、'.'、'_'、'-'、および'/'。</p> <p>最大長: ホスト名ごとに 128 文字、行全体で 2048 文字。</p>
Hostnameltem	no		system.hostname	<p>ホスト名が定義されていない場合に生成するために使用されるアイテム。Hostname が定義されている場合は無視されます。</p> <p>UserParameters またはエイリアスはサポートされませんが、EnableRemoteCommands の値に関係なく、system.run[] はサポートされます。</p> <p>出力の長さは 512 KB に制限されています。</p>
Include	no			<p>個々のファイルまたはディレクトリ内のすべてのファイルを構成ファイルに含めることができます。</p> <p>コンパイル時に変更されない限り、Zabbix はインストール中に /usr/local/etc にインクルードディレクトリを作成します。</p> <p>関連するもののみを含めるには指定したディレクトリ内のファイルの場合、パターンマッチングでアスタリスクワイルドカード文字がサポートされます。例:</p> <pre>/absolute/path/to/config/files/*.conf</pre> <p>Zabbix 6.0.0 以降、パスは zabbix_agent2.win.conf ファイルの場所からの相対パスにすることができます。</p> <p>制限事項については特記事項を参照してください。</p>

パラメータ	必須	範囲	デフォルト	説明
ListenIP	no		0.0.0.0	エージェントがリッスンするカンマ区切りの IP アドレスのリスト。 Zabbix サーバーに接続している場合、アクティブなチェックのリストを取得するために、最初の IP アドレスが Zabbix サーバーに送信されます。
ListenPort	no	1024-32767	10050	エージェントは、この番号のポートでサーバーからの接続をリッスンします。
LogFile	LogType が file に設定されている場合:yes それ以外の場合:no		c:\zabbix_agent2\log	LogType が 'file' の場合のログファイル名。
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB)。 0 - ログの自動ローテーションを無効にします。 注: ログファイルのサイズ制限に達し、ファイルのローテーションが何らかの理由で失敗した場合、既存のログファイルは切り捨てられ、新たに作成されます。
LogType	no		file	ログメッセージが書き込まれる場所を指定します。 file - LogFile パラメータで指定されたファイル console - 標準出力。
PersistentBufferFileno				Zabbix Agent2 が SQLite データベースを保持するファイル。完全なファイル名である必要があります。 このパラメータは、永続バッファが有効な場合 (EnablePersistentBuffer=1) にのみ使用されます。
PersistentBufferPeriod	no	1m-365d	1h	サーバーまたはプロキシへの接続がない場合に、データを保存する期間。古いデータは失われます。ログデータは保持されます。 このパラメータは、永続バッファが有効な場合にのみ使用されます (EnablePersistentBuffer=1)
Plugins	no			Zabbix 6.0.0 以降、ほとんどのプラグインには独自の 設定ファイル があります。エージェント設定ファイルには、以下に示すプラグインパラメータが含まれています。
Plugins.Log.MaxLinesPerSecond	no	1-1000	20	'ログ' および 'イベントログ' のアクティブチェックを処理するときに、エージェントが Zabbix サーバーまたはプロキシに 1 秒あたりに送信する新しい行の最大数。 指定された値は、パラメータ 'maxlines' によって上書きされます。 'log' または 'eventlog' アイテムキーで提供されます。 注: Zabbix は、MaxLinesPerSecond で設定されたよりも 10 倍多くの新しい行を処理して、ログ項目に必要な文字列を検索します。 このパラメータは 4.4.2 以降でサポートされており、MaxLinesPerSecond を置き換えます。
Plugins.SystemRun.LogRemoteCommands	no		0	実行されたシェルコマンドのログを警告として有効にします。 0 - 無効 1 - 有効 コマンドは、リモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合、ログエントリは作成されません。 このパラメータは 4.4.2 以降でサポートされ、LogRemoteCommands を置き換えます。
PluginSocket	no		\\.\pipe\agent.sock	ローカル可能なプラグイン通信の UNIX ソケットへのパス。
PluginTimeout	no	1-30	Global timeout	ロード可能なプラグインとの接続のタイムアウト。

パラメータ	必須	範囲	デフォルト	説明
RefreshActiveChecks		60-3600	120	アクティブチェックのリストを更新する頻度 (秒単位)。アクティブチェックの更新に失敗すると、60 秒後に次の更新が試行されることに注意してください。
Server	yes			カンマ区切りの IP アドレスのリスト (オプションで CIDR 表記、または Zabbix サーバーと Zabbix プロキシの DNS 名)。着信接続は、ここにリストされているホストからのみ受け入れられます。 IPv6 サポートが有効になっている場合は、'127.0. 0.1'、 '::ffff:127.0.0.1' は同等に扱われ、 ':::0' はすべての IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' はすべての IPv4 アドレスを許可するために使用できます。 例: Server=127.0.0.1,192.168.1.0/24,:::1,2001:db8::/32,zabbix.example.com スペースを使用できます。
ServerActive	no			アクティブチェックを取得する Zabbix サーバー/プロキシアドレスまたはクラスタ構成。 サーバー/プロキシアドレスは、コロンで区切られた IP アドレスまたは DNS 名とオプションのポートです。 クラスタ構成は、セミコロンで区切られた 1 つまたは複数のサーバーアドレスです。 br> 複数の Zabbix サーバー/クラスタと Zabbix プロキシをカンマで区切って指定できます。 各 Zabbix サーバー/クラスタから複数の Zabbix プロキシを指定しないでください。 Zabbix プロキシが指定されている場合は、Zabbix サーバー/クラスタそのプロキシは指定しないでください。 複数の独立した Zabbix サーバーを並行して使用するために、複数のアドレスを指定できます。スペースを使用できます。 ポートが指定されていない場合、デフォルトのポートが使用されます。 そのホストのポートが指定されている場合、IPv6 アドレスは角括弧で囲む必要があります。 ポートが指定されていない場合、IPv6 の角括弧アドレスはオプションです。 このパラメータが指定されていない場合、アクティブチェックは無効になります。 Zabbix プロキシの例: ServerActive=127.0.0.1:10051 複数のサーバーの例: ServerActive= 127.0.0.1:20051,zabbix.domain,[:::1]:30051,:::1,[12fc::1] 高可用性の例: ServerActive=zabbix.cluster.node1 ;zabbix.cluster.node2:20051;zabbix.cluster.node3 2 つのクラスターと 1 つのサーバーによる高可用性の例: ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster.node1;zabbix.cluster2.node2,zabbix.domain
SourceIP	no			次の送信元 IP アドレス: - Zabbix サーバーまたは Zabbix プロキシへの発信接続。 - 一部のアイテム (web.page.get、net.tcp.port など) の実行中に接続を確立する
StatusPort	no	1024-32767		設定されている場合、エージェントはこのポートで HTTP ステータスリクエストをリッスンします (http://localhost:<port>/status)
Timeout	no	1-30	3	処理にかかる時間は Timeout 秒以内です。

パラメータ	必須	範囲	デフォルト	説明
UserParameterDir	no			UserParameter コマンドのデフォルトの検索パス。使用すると、エージェントはコマンドを実行する前に、ここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドは、フルパスの代わりに相対 ./ プレフィックスを持つことができます。許可されるエントリは 1 つだけです。 例: UserParameterDir=/opt/myscripts

7 Zabbix エージェント 2 プラグイン

概要

このセクションには、Zabbix エージェント 2 プラグインの設定ファイルパラメータの説明が含まれています。特定のプラグインに関する情報にアクセスするには、サイドバーを使用してください。

1 Ceph プラグイン

概要

このセクションでは、Ceph Zabbix エージェント 2 プラグイン構成ファイル (ceph.conf) でサポートされているパラメータの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は BOM なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Ceph.InsecureSkipVerify	false	false / true	false	http クライアントがサーバーの証明書チェーンとホスト名を確認する必要があるかどうかを決定します。 true の場合、TLS はサーバーによって提示された証明書とその証明書内のホスト名を受け入れます。このモードでは、TLS は中間者攻撃の影響を受けやすくなります (テストにのみ使用してください)
Plugins.Ceph.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)
Plugins.Ceph.Sessions.<SessionName>.ApiKey				名前付きセッション API キー
Plugins.Ceph.Sessions.<SessionName>.User				<SessionName> - アイテムキーで使用するセッションの名前 名前付きセッションのユーザー名
Plugins.Ceph.Sessions.<SessionName>.Uri			https://localhost:8003	名前付きセッションの接続文字列 <SessionName> - アイテムキーで使用するセッション名
				ID やパスワードといった認証情報を含めないでください (無視されます) 必須 URI 形式と一致します サポートされているのは https スキームのみです。スキームは省略できます (バージョン 5.2.3 以降) ポートは省略できます (デフォルト = 8003) 例: https://127.0.0.1:8003 localhost
Plugins.Ceph.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#)の設定手順

2 Docker プラグイン

概要

このセクションでは Docker Zabbix エージェント 2 プラグイン設定ファイル (docker.conf) でサポートされているパラメーターを一覧表示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix はBOMなしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が" #" で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Dockerデーモンの unix ソケットの場所 スキームが含まれている必要があります (unix://のみがサポートされています)
Plugins.Docker.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#)の設定手順

3 Ember+ plugin

Overview

This section lists parameters supported in the Ember+ Zabbix agent 2 plugin configuration file (ember.conf).

The Ember+ plugin is a loadable plugin and is available and fully described in the [Ember+ plugin repository](#).

This plugin is supported since Zabbix 6.0.30 and currently only available to be built from the source (for both Unix and Windows).

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with " #" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.EmberPlus.Default.Uri			tcp://localhost:9999	9999 default URI to connect. The only supported schema is tcp://. A schema can be omitted. Embedded credentials will be ignored.
Plugins.EmberPlus.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9999	9999 URI to connect, for the named session. The only supported schema is tcp://. A schema can be omitted. Embedded credentials will be ignored. <SessionName> - define name of a session for using in item keys.
Plugins.EmberPlus.System.Path				Path to the Ember+ plugin executable. Example usage: Plugins.EmberPlus.System.Path=/usr/sbin/zabbix-agent2-pl

Parameter	Mandatory	Range	Default	Description
Plugins.EmberPlaaS.Timeout		1-30	global timeout	The amount of time to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: [Zabbix agent 2 \(UNIX\) / Zabbix agent 2 \(Windows\)](#)
- Instructions for configuring [plugins](#)

3 Memcached プラグイン

概要

このセクションでは、Memcached Zabbix エージェント 2 プラグイン構成ファイル (memcached.conf) でサポートされているパラメーターを一覧表示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は **BOM** なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Memcached.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)
Plugins.Memcached.Sessions.<SessionName>.Password				名前付きセッションパスワード <SessionName> - アイテムキーで使用するセッションの名前
Plugins.Memcached.Sessions.<SessionName>.Uri			tcp://localhost:11211	名前付きセッションの接続文字列 <SessionName> - アイテムキーで使用するセッション名 埋め込みクレデンシャルを含めないでください (無視されます) 必須 URI 形式に一致します サポートされているスキーム: tcp、unix スキームは省略できます (バージョン 5.2.3 以降) ポートは省略できます (デフォルト = 11211) 例: tcp://localhost:11211 localhost unix:/var/run/memcached.sock
Plugins.Memcached.Sessions.<SessionName>.User				名前付きセッションのユーザー名 <SessionName> - アイテムキーで使用するセッション名
Plugins.Memcached.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#) の設定手順

4 Modbus プラグイン

概要

このセクションでは Modbus Zabbix エージェント 2 プラグイン設定ファイル (modbus.conf) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は **BOM** なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Modbus.Sessions.<SessionName>.Endpoint				エンドポイントは、プロトコルスキーム、ホストアドレス、ポートまたはシリアルポートの名前と属性で構成される接続文字列です
Plugins.Modbus.Sessions.<SessionName>.SlaveID				<SessionName> - アイテムキーで使用するセッション名 名前付きセッションのスレーブ ID <SessionName> - アイテムキーで使用するセッション名 例: Plugins.Modbus.Sessions.MB1.SlaveID=20 注この名前付きセッションパラメータはアイテムキースレーブ ID パラメータで指定された値が空の場合にのみチェックされます。
Plugins.Modbus.Sessions.<SessionName>.Timeout				名前付きセッションのタイムアウト <SessionName> - アイテムキーで使用するセッション名 例: Plugins.Modbus.Sessions.MB1.Timeout=2
Plugins.Modbus.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#) の設定手順

5 MongoDB プラグイン

概要

このページでは、MongoDB Zabbix エージェント 2 プラグインの設定オプションに関する情報を提供します。

Zabbix 6.0.6 以降、MongoDB は読み込み可能なプラグインであり、[MongoDB プラグインリポジトリ](#) で利用可能で完全に説明されています。

注意:

- デフォルト値は出荷された設定ファイルの値ではなくプロセスのデフォルトを反映しています
- Zabbix は BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします
- "#" で始まるコメントは行頭でのみサポートされます

オプション

パラメータ	説明
-V --version	プラグインのバージョンとライセンス情報を出力します
-h --help	省略形のヘルプを表示します)

設定ファイル

MongoDB プラグインの設定パラメーター

Note:

バージョン 6.0.6 以前の Zabbix では、パラメータ名は Plugins.MongoDB.<Parameter> ではなく Plugins.Mongo.<Parameter> で始まります。例: Plugins.Mongo.KeepAlive

パラメータ	必須	範囲	デフォルト	説明
Plugins.MongoDB.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)。
Plugins.Mongo.Sessions.<SessionName>.Password				名前付きセッションパスワード <SessionName> - アイテムキーで使用するセッションの名前。

パラメータ	必須	範囲	デフォルト	説明
Plugins.MongoDBSessions.<SessionName>.Uri				名前付きセッションの接続文字列 <SessionName> - アイテムキーで使用するセッションの名前 埋め込み資格情報を含めないでください (無視されます) URI 形式と一致する必要があります tcp スキームのみがサポートされています。スキームは省略できます ポートは省略可能です (デフォルト = 27017) 例: tcp://127.0.0.1:27017、tcp:localhost、localhost
Plugins.MongoDBSessions.<SessionName>.User				名前付きセッションユーザー名 <SessionName> - アイテムキーで使用するセッションの名前。
Plugins.MongoDBSystem.Path				外部プラグイン実行可能ファイルへのパス。Zabbix 6.0.6 以降でサポート
Plugins.MongoDBTimeout		1-30	global timeout	リクエスト実行タイムアウト (リクエストをシャットダウンする前に完了するまで待機する時間)

参照:

- 一般的な Zabbix エージェント 2 設定パラメーターの説明: [Zabbix エージェント 2 \(UNIX\)](#) [Zabbix エージェント 2 \(Windows\)](#)
- [プラグインの設定手順](#)

Footnotes

¹ - Since Zabbix 6.0.13, loadable plugins started using the same versioning system as Zabbix itself. As a result, MongoDB plugin version has changed from 1.2.1 to 6.0.13.

6 MQTT プラグイン

概要

このセクションでは、MQTT Zabbix エージェント 2 プラグイン構成ファイル (mqtt.conf) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は **BOM** なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.MQTT.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: [Zabbix agent 2 \(UNIX\)](#)/[Zabbix agent 2 \(Windows\)](#)
- [プラグインの設定手順](#)

7 MSSQL plugin

Overview

This section lists parameters supported in the MSSQL Zabbix agent 2 plugin configuration file (mssql.conf).

This plugin is supported since Zabbix 6.0.27. For more information see the [MSSQL plugin](#) readme.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Zabbix supports configuration files only in UTF-8 encoding without **BOM**;

- Comments starting with “#” are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.CustomQueriesDir			empty	Specifies the file path to a directory containing user-defined .sql files with custom queries that the plugin can execute. The plugin loads all available .sql files in the configured directory at startup. This means that any changes to the custom query files will not be reflected until the plugin is restarted. The plugin is started and stopped together with Zabbix agent 2.
Plugins.MSSQL.Default.CACertPath				The default file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server. The certificate must be in PEM format.
Plugins.MSSQL.Default.Database				The default database name to connect to.
Plugins.MSSQL.Default.Encrypt				Specifies the default connection encryption type. Possible values are: true - data sending between plugin and server is encrypted; false - data sending between plugin and server is not encrypted beyond the login packet; strict - data sending between plugin and server is encrypted E2E using TDS8 ; disable - data sending between plugin and server is not encrypted.
Plugins.MSSQL.Default.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server by default.
Plugins.MSSQL.Default.Password				The password to be sent to a protected MSSQL server by default.
Plugins.MSSQL.Default.TLSMinVersion				The minimum TLS version to use by default. Possible values are: 1.0, 1.1, 1.2, 1.3.
Plugins.MSSQL.Default.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it by default. Possible values: true, false.
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	The default URI to connect. The only supported schema is sqlserver://. A schema can be omitted. Embedded credentials will be ignored.
Plugins.MSSQL.Default.User				The default username to be sent to a protected MSSQL server.
Plugins.MSSQL.KeyPAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				The file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server for the named session. The certificate must be in PEM format. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Database				The database name to connect to for the named session. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				Specifies the connection encryption type for the named session. Possible values are: true - data sending between plugin and server is encrypted; false - data sending between plugin and server is not encrypted beyond the login packet; strict - data sending between plugin and server is encrypted E2E using TDS8 ; disable - data sending between plugin and server is not encrypted. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server for the named session. <SessionName> - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Sessions.<SessionName>.Password				The password to be sent to a protected MSSQL server for the named session. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				The minimum TLS version to use for the named session. Possible values are: 1.0, 1.1, 1.2, 1.3. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it for the named session. Possible values: true, false. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Uri			sqlserver://localhost	URI to connect, for the named session. The only supported schema is sqlserver://. A schema can be omitted. Embedded credentials will be ignored. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.User				The username to be sent to a protected MSSQL server for the named session. <SessionName> - define name of a session for using in item keys.
Plugins.MSSQL.System.Path				Path to the MSSQL plugin executable. Global setting for the MSSQL plugin. Applied to all connections. Example usage: Plugins.MSSQL.System.Path=/usr/sbin/zabbix-agent2-plugin.
Plugins.MSSQL.Timeout		1-30	global timeout	The amount of time to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: [Zabbix agent 2 \(UNIX\) / Zabbix agent 2 \(Windows\)](#)
- Instructions for configuring [plugins](#)

7 MySQL プラグイン

概要

このセクションでは、MySQL Zabbix エージェント 2 プラグイン構成ファイル (mysql.conf) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は **BOM** なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.MySql.ConnectTimeout		1-30	global timeout	要求が実行されるのを待機する最大時間 (秒単位)
Plugins.MySql.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)
Plugins.MySql.Sessions.<SessionName>.Password				名前付きセッションパスワード <SessionName> - アイテムキーで使用するセッションの名前
Plugins.MySql.Sessions.<SessionName>.TLSCAFile				Zabbix エージェント 2 と監視対象データベース間の暗号化された通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名 <SessionName> - アイテムキーで使用するセッションの名前

パラメータ	必須	範囲	デフォルト	説明
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile				Zabbix エージェント 2 と監視対象データベース間の暗号化された通信に使用される、エージェント証明書または証明書チェーンを含むファイルのフルパス名 <SessionName> - アイテムキーで使用するセッションの名前
Plugins.Mysql.Sessions.<SessionName>.TLSConnect				Zabbix エージェント 2 と監視対象データベース間の通信の暗号化タイプ <SessionName> - アイテムキーで使用するセッション名 使用可能な値: required - require TLS 接続; verify_ca - 証明書を検証します verify_full - 証明書と IP アドレスを検証します
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile				Zabbix エージェント 2 と監視対象データベース間の暗号化された通信に使用されるデータベース秘密鍵を含むファイルの絶対パス名 <SessionName> - アイテムキーで使用するセッション名
Plugins.Mysql.Sessions.<SessionName>.Uri			tcp://localhost:3306	名前付きセッションの接続文字列 <SessionName> - アイテムキーで使用するセッション名 埋め込みクレデンシャルを含めないでください (無視されます) 必須 URI 形式に一致します サポートされているスキーム: tcp,unix スキームは省略できます (バージョン 5.2.3 以降) ポートは省略できます (デフォルト = 3306) 例: tcp://localhost:3306 localhost unix:/var/run/mysql.sock
Plugins.Mysql.Sessions.<SessionName>.User				名前付きセッションパスワード <SessionName> - アイテムキーで使用するセッションの名前
Plugins.Mysql.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#) の設定手順

8 オラクルプラグイン

概要

このセクションでは、Oracle Zabbix エージェント 2 プラグイン構成ファイル (ceph.conf) でサポートされているパラメータの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は **BOM** なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.Oracle.ConnTimeout		1-30	global timeout	リクエストが完了するまでの最大待機時間 (秒単位)
Plugins.Oracle.ConnectTimeout		1-30	global timeout	接続が確立されるまでの最大待機時間 (秒単位)
Plugins.Oracle.CustomQueriesPath				カスタムクエリを含む.sql ファイルを含むディレクトリのフルパス名 デフォルトでは無効になっています 例: /etc/zabbix/oracle/sql

パラメータ	必須	範囲	デフォルト	説明
Plugins.Oracle.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)
Plugins.Oracle.Sessions.<SessionName>.Password				名前付きセッションパスワード <SessionName> - アイテムキーで使用するセッション名
Plugins.Oracle.Sessions.<SessionName>.Service				接続に使用される名前付きセッションサービス名 (SID はサポートされていません) サポート対象: Oracle <PluginName> - プラグインの名前 <SessionName> - アイテムキーで使用するセッション名
Plugins.Oracle.Sessions.<SessionName>.Uri			tcp://localhost:1521	Oracle の名前付きセッション接続文字列 <SessionName> - アイテムキーで使用するセッション名 埋め込みクレデンシャルを含めないでください (無視されます) 必須 URI 形式に一致します tcp スキームのみがサポートされています。スキームは省略できます (バージョン 5.2.3 以降) ポートは省略できます (デフォルト = 1521) 例: tcp://127.0.0.1:1521 localhost
Plugins.Oracle.Sessions.<SessionName>.User				名前付きセッションのユーザー名 <SessionName> - アイテムキーで使用するセッションの名前

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグインの設定手順](#)

9 PostgreSQL plugin

Overview

This section lists parameters supported in the PostgreSQL Zabbix agent 2 plugin configuration file (postgresql.conf).

Since Zabbix 6.0.10, PostgreSQL is a loadable plugin, which is available and fully described in the [PostgreSQL plugin repository](#)

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).

Parameters

Note:

In Zabbix versions before 6.0.10, parameter names start with Plugins.Postgres.<Parameter> instead of Plugins.PostgreSQL.<Parameter>. For example, Plugins.Postgres.KeepAlive.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.CallTimeout		1-30	global timeout	Maximum wait time (in seconds) for a request to be completed.
Plugins.PostgreSQL.CustomQueriesPath			disabled	Full pathname of the directory containing .sql files with custom queries.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.Database				Default database for connecting to PostgreSQL; used if no value is specified in an item key or named session. Supported since version 6.0.18.
Plugins.PostgreSQL.Default.Password				Default password for connecting to PostgreSQL; used if no value is specified in an item key or named session. Supported since version 6.0.18.
Plugins.PostgreSQL.Default.TLSCAFile (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to one of: verify_ca, verify_full)				Full pathname of a file containing the top-level CA(s) certificate for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported since version 6.0.18.
Plugins.PostgreSQL.Default.TLSCertFile (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to one of: verify_ca, verify_full)				Full pathname of a file containing the PostgreSQL certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported since version 6.0.18.
Plugins.PostgreSQL.Default.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: required - connect using TLS as transport mode without identity checks; verify_ca - connect using TLS and verify certificate; verify_full - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection. Supported since version 6.0.18.
Plugins.PostgreSQL.Default.TLSKeyFile (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to one of: verify_ca, verify_full)				Full pathname of a file containing the PostgreSQL private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported since version 6.0.18.
Plugins.PostgreSQL.Default.Uri				Default URI for connecting to PostgreSQL; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Supported schemes: tcp, unix. Examples: tcp://127.0.0.1:5432 tcp://localhost unix:/var/run/postgresql/.s.PGSQL.5432 Supported since version 6.0.18.
Plugins.PostgreSQL.Default.User				Default username for connecting to PostgreSQL; used if no value is specified in an item key or named session. Supported since version 6.0.18.
Plugins.PostgreSQL.KeepAlive		60-900	300	Maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.PostgreSQL.Sessions.<SessionName>.Database				Database for session connection. <SessionName> - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Yes	the password format.		Password for session connection. <SessionName> - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	(yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile is set to one of: verify_ca, verify_full)			Full pathname of a file containing the top-level CA(s) certificate peer certificate verification. <SessionName> - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	(yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile is set to one of: verify_ca, verify_full)			Full pathname of a file containing the PostgreSQL certificate or certificate chain. <SessionName> - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Yes			Encryption type for PostgreSQL connection. <SessionName> - define name of a session for using in item keys. Supported values: required - connect using TLS as transport mode without identity checks; verify_ca - connect using TLS and verify certificate; verify_full - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	(yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full)			Full pathname of a file containing the PostgreSQL private key. <SessionName> - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.Uri	Yes			Connection string of a named session. <SessionName> - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Supported schemes: tcp, unix. Examples: tcp://127.0.0.1:5432 tcp://localhost unix:/var/run/postgresql/.s.PGSQL.5432
Plugins.PostgreSQL.Sessions.<SessionName>.User	Yes			Named session username. <SessionName> - define name of a session for using in item keys.
Plugins.PostgreSQL.System.Path	Yes			Path to external plugin executable. Supported since Zabbix 6.0.10.
Plugins.PostgreSQL.Timeout	Yes	1-30	global timeout	Request execution timeout (how long to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: [Zabbix agent 2 \(UNIX\) / Zabbix agent 2 \(Windows\)](#)
- Instructions for configuring [plugins](#)

10 Redis プラグイン

概要

このセクションでは Redis Zabbix エージェント 2 プラグイン設定ファイル (redis.conf) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は BOM なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメーター

パラメータ	必須	範囲	デフォルト	説明
Plugins.Redis.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒単位)
Plugins.Redis.Sessions.<SessionName>.Password				名前付きセッションパスワード <SessionName> - アイテムキーで使用するセッションの名前
Plugins.Redis.Sessions.<SessionName>.Uri			tcp://localhost:6379	名前付きセッションの接続文字列 <SessionName> - アイテムキーで使用するセッションの名前 埋め込み資格情報を含めることはできません (URI 形式と一致する必要があります) サポートされているスキーム: tcp,unix; スキームは省略できます (バージョン 5.2.3 以降) ポートは省略できます (デフォルト = 6379) 例 tcp://localhost:6379 localhost unix:/var/run/redis.sock
Plugins.Redis.Sessions.<SessionName>.User				名前付きセッションのユーザー名 <SessionName> - アイテムキーで使用するセッションの名前
Plugins.Redis.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#) の設定手順

11 スマートプラグイン

概要

このセクションでは Smart Zabbix エージェント 2 プラグイン設定ファイル (smart.conf) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix は BOM なしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が " # " で始まった場合のみコメントとして認識されます。

パラメーター

パラメータ	必須	範囲	デフォルト	説明
Plugins.Smart.Path			smartctl	smartctl 実行可能ファイルへのパス。
Plugins.Smart.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの時間)

参考情報:

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: [Zabbix agent 2 \(UNIX\)/Zabbix agent 2 \(Windows\)](#)
- [プラグイン](#) の設定手順

8 Zabbix Java ゲートウェイ

Zabbix Java ゲートウェイの起動に `startup.sh` および `shutdown.sh` スクリプトを使用する場合は、`settings.sh` ファイルで必要な設定パラメータを指定できます。起動スクリプトとシャットダウンスクリプトは設定ファイルを取得し、シェル変数（最初の列にリストされている）を Java プロパティ（2 番目の列にリストされている）に変換します。

java を直接実行して ZabbixJava ゲートウェイを手動で起動する場合は、コマンドラインで対応する Java プロパティを指定します。

```
|Variable|Property|Mandatory|Range|Default|Description| | 変数 | プロパティ | 必須 | 範囲 | デフォルト | 説明 | |-----|-----|-----|-----|-----|-----| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|LISTEN_IP|zabbix.listenIP|no|<|0.0.0.0| 待ち受け IP アドレス | |LISTEN_PORT|zabbix.listenPort|no|1024-32767|10052| 待ち受けポート番号 | |PID_FILE|zabbix.pidFile|no|<|/tmp/zabbix_java.pid|PID ファイル名。省略した場合、ZabbixJava ゲートウェイはコンソールアプリケーションとして起動されます。| |PROPERTIES_FILE|zabbix.propertiesFile|no|<| プロパティファイルの名前。コマンドラインに表示されないようにキー値形式を使用して追加のプロパティを設定したり、既存のプロパティを上書きしたりするために使用できます。<br>例:"javax.net.ssl.trustStorePassword=<password>" | |START_POLLERS|zabbix.startPollers|no|1-1000|5| 開始するワーカースレッドの数 | |TIMEOUT|zabbix.timeout|no|1-30|3| ネットワーク操作を待機する時間 |
```

Warning:

ポート 10052 はIANA 登録済みではありません

9 Zabbix Web サービス

概要

Zabbix Web サービスは、外部 Web サービスとの通信に使用されるプロセスです。

このセクションでは Zabbix Web サービス設定ファイル (`zabbix_web_service.conf`) でサポートされているパラメーターの一覧を提示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- Zabbix はBOMなしの UTF-8 エンコーディングでのみ構成ファイルをサポートします。
- 行の先頭が" #" で始まった場合のみコメントとして認識されます。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
AllowedIP	yes			カンマ区切りの IP アドレスのリスト（オプションで CIDR 表記、または Zabbix サーバーと Zabbix プロキシの DNS 名）着信接続は、ここにリストされているホストからのみ受け入れられます。 IPv6 サポートが有効になっている場合、127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 は同等に扱われ、::/0 は任意の IPv4 または IPv6 アドレスを許可します。 0.0.0.0/0 表記を使用して任意の IPv4 アドレスを許可できます。 例： 127.0.0.1,192.168.1.0/24,:::1,2001

パラメーター	必須	範囲	デフォルト	説明
DebugLevel	no	0-5	3	デバッグレベルを指定します 0 - Zabbix プロセスの開始と停止に関する基本情報 1 - 重要な情報 2 - エラー情報 3 - 警告 4 - デバッグ用 (多くの情報を生成します) 5 - 拡張デバッグ (さらに多くの情報を生成します)
ListenPort	no	1024-32767	10053	接続待機ポート番号
LogFile	LogType が file に設定されている場合は yes。それ以外の場合は no。			LogType 'file' パラメーターのログファイル名。 例:/tmp/zabbix_web_service.log
LogFileSize	no	0-1024	1	ログファイルの最大サイズ (MB 単位) 0 - 自動ログローテーションを無効にします。
LogType	no	system / file / console	file	ログメッセージの書き込み先を指定します system - syslog file - LogFile パラメーターで指定されたファイル console - 標準出力
Timeout	no	1-30	3	タイムアウトするまでの処理時間。秒指定
TLSAccept	no	unencrypted / cert	unencrypted	使用する接続のタイプを指定します。 unencrypted - 暗号化せずに接続を受け入れる (デフォルト) cert - TLS と証明書を使用した接続を受け入れる
TLSCAFile	no			Zabbix コンポーネント間の暗号化された通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。
TLSCertFile	no			Zabbix コンポーネントとの暗号化された通信に使用される、サービス証明書または証明書チェーンを含むファイルのフルパス名。
TLSKeyFile	no			Zabbix コンポーネントとの暗号化された通信に使用されるサービス秘密鍵を含むファイルのフルパス名。

10 インクルード

概要

Include パラメータを使用して、追加のファイルまたはディレクトリをサーバー/プロキシ/エージェント設定に含めることができます。

インクルード時の注意

ファイルを含めるために Include パラメータを使用する場合、ファイルは読み取り可能である必要があります。

ディレクトリを含めるために Include パラメータが使用される場合：

- ディレクトリ内のすべてのファイルが読み取り可能である必要があります。
- インクルードの順番を想定する必要はありません。(たとえばファイルはアルファベット順で読み込まれません)
- ディレクトリ内のすべてのファイルが設定に含まれます。

- 一部のテキストエディタによって自動的に作成されるファイルバックアップコピーに注意してください。たとえパターンを使用してファイルをインクルードするために Include パラメーターが使用される場合:

- パターンに一致するすべてのファイルは読み取り可能である必要があります。
- インクルードの順番を想定する必要はありません。(たとえばファイルはアルファベット順で読み込まれません)

4 プロトコル

1 サーバープロキシデータ交換プロトコル

概要

サーバー - プロキシ間のデータ交換は JSON 形式に基づいています。

要求メッセージと応答メッセージはヘッダーとデータの長さに則って始まる必要があります。

パッシブプロキシ

プロキシ設定のリクエスト

proxy config リクエストは、プロキシ設定データを提供するためにサーバーによって送信されます。このリクエストは ProxyConfigFrequency (サーバー設定パラメーター) 秒ごとに送信されます。

名前	値の型	説明
サーバーからプロキシ:		
request	string	'proxy config'
<table>	object	テーブルのデータを持つ 1 つ以上のオブジェクト
fields	array	フィールド名の配列
-	string	フィールド名
data	array	配列の行
-	array	配列の列
-	string,number	データベーススキーマの列タイプに応じたタイプの列値
プロキシからサーバー:		
response	string	リクエストの成功情報 ('success' または 'failed')
version	string	プロキシバージョン (<メジャー>.<マイナー>.<ビルド>)

例:

サーバーからプロキシ:

```
{
  "request": "proxy config",
  "globalmacro": {
    "fields": [
      "globalmacroid",
      "macro",
      "value"
    ],
    "data": [
      [
        2,
        "{$SNMP_COMMUNITY}",
        "public"
      ]
    ]
  },
  "hosts": {
    "fields": [
      "hostid",
      "host",
      "status",
      "ipmi_authtype",
```

```

    "ipmi_privilege",
    "ipmi_username",
    "ipmi_password",
    "name",
    "tls_connect",
    "tls_accept",
    "tls_issuer",
    "tls_subject",
    "tls_psk_identity",
    "tls_psk"
  ],
  "data": [
    [
      10001,
      "Linux",
      3,
      -1,
      2,
      "",
      "",
      "Linux",
      1,
      1,
      "",
      "",
      "",
      ""
    ],
    [
      10050,
      "Zabbix Agent",
      3,
      -1,
      2,
      "",
      "",
      "Zabbix Agent",
      1,
      1,
      "",
      "",
      "",
      ""
    ],
    [
      10105,
      "Logger",
      0,
      -1,
      2,
      "",
      "",
      "Logger",
      1,
      1,
      "",
      "",
      "",
      ""
    ]
  ]
},

```

```

"interface":{
  "fields":[
    "interfaceid",
    "hostid",
    "main",
    "type",
    "useip",
    "ip",
    "dns",
    "port",
    "bulk"
  ],
  "data":[
    [
      2,
      10105,
      1,
      1,
      1,
      "127.0.0.1",
      "",
      "10050",
      1
    ]
  ]
},
...
}

```

プロキシからサーバー:

```

{
  "response": "success",
  "version": "5.4.0"
}

```

プロキシリクエスト

proxy data リクエストはプロキシからホストインターフェースの可用性、ヒストリ、検出、および自動登録のデータを取得するために使用されます。このリクエストは ProxyDataFrequency (サーバー設定パラメーター) 秒ごとに送信されます。

名前	値の型	説明
サーバーからプロキシ: request	string	'proxy data'

名前	値の型	説明
session	string	データセッショントークン
interface	array	(オプション) インターフェイス可用性データオブジェクトの配列
availability		
interfaceid	number	インターフェイス ID
available	number	インターフェイスの可用性
		0 , INTERFACE_AVAILABLE_UNKNOWN - 不明 1 , INTERFACE_AVAILABLE_TRUE - 利用可能 2 , INTERFACE_AVAILABLE_FALSE - 利用不可
error	string	インターフェイスエラーメッセージまたは空の文字列
history data	array	(オプション) ヒストリデータオブジェクトの配列
itemid	number	アイテム ID
clock	number	アイテム値のタイムスタンプ (秒)
ns	number	アイテム値のタイムスタンプ (ナノ秒)
value	string	(オプション) アイテム値
id	number	ID (1 つのデータセッション内で一意の昇順カウンター)
timestamp	number	(オプション) ログタイプアイテムのタイムスタンプ
source	string	(オプション) イベントログアイテムのソース値
severity	number	(オプション) イベントログアイテムの深刻度の値
eventid	number	(オプション) イベントログアイテムのイベント ID 値
state	string	(オプション) アイテムの状態
		0 , ITEM_STATE_NORMAL 1 , ITEM_STATE_NOTSUPPORTED
lastlogsize	number	(オプション) ログタイプアイテムの最後のログサイズ
mtime	number	(オプション) ログタイプの項目の時刻を変更
discovery data	array	(オプション) ディスカバリデータオブジェクトの配列
clock	number	ディスカバリデータのタイムスタンプ
druleid	number	ディスカバリルール ID
dcheckid	number	ディスカバリチェック ID またはヌルのディスカバリ・ルール・データ
type	number	ディスカバリチェックタイプ:
		-1 ディスカバリルールデータ 0 , SVC_SSH - SSH サービスチェック 1 , SVC_LDAP - LDAP サービスチェック 2 , SVC_SMTP - SMTP サービスチェック 3 , SVC_FTP - FTP サービスチェック 4 , SVC_HTTP - HTTP サービスチェック 5 , SVC_POP - POP サービスチェック 6 , SVC_NNTP - NNTP サービスチェック 7 , SVC_IMAP - IMAP サービスチェック 8 , SVC_TCP - TCP ポートの可用性チェック 9 , SVC_AGENT - Zabbix エージェント 10 , SVC_SNMPv1 - SNMPv1 エージェント 11 , SVC_SNMPv2 - SNMPv2 エージェント 12 , SVC_ICMPPING - ICMP ピング 13 , SVC_SNMPv3 - SNMPv3 エージェント 14 , SVC_HTTPS - HTTPS サービスチェック 15 , SVC_TELNET - Telnet の可用性チェック
ip	string	ホストの IP アドレス
dns	string	ホストの DNS 名
port	number	(オプション) サービスポート番号
key_	string	(オプション) タイプのディスカバリチェック用アイテムキー 9
value	string	SVC_AGENT (オプション) サービスから受け取った値は、ほとんどのサービスで空になります

名前	値の型	説明
status	number	(オプション) サービスステータス: 0 , DOBJECT_STATUS_UP - サービスアップ 1 , DOBJECT_STATUS_DOWN - サービスダウン
auto registration	array	(オプション) オートレジストレーションデータオブジェクトの配列
clock	number	オートレジストレーションデータのタイムスタンプ
host	string	ホスト名
ip	string	(オプション) IP アドレス
dns	string	(オプション) IP アドレスから解決された DNS 名
port	string	(オプション) ホストのポート
host_metadata	string	(オプション) エージェントによって送信されたホストメタデータ (HostMetadata または HostMetadataItem エージェント構成パラメーターに基づく)
tasks	array	(オプション) タスクの配列
type	number	タスクのタイプ: 0 , ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT - リモートコマンドの結果
status	number	リモートコマンドの実行ステータス: 0 , ZBX_TM_REMOTE_COMMAND_COMPLETED - リモートコマンドは正常に完了 1 , ZBX_TM_REMOTE_COMMAND_FAILED - リモートコマンド失敗
error	string	(オプション) エラーメッセージ
parent_taskid	number	親タスク ID
more	number	(オプション) 1 - 追加の送信履歴データ
clock	number	(オプション) データ転送のタイムスタンプ (秒)
ns	number	(オプション) データ転送のタイムスタンプ (ナノ秒)
version	string	プロキシバージョン (< メジャー >.< マイナー >.< ビルド >)
サーバーからプロキシ:		
response	string	リクエストの成否 ('success' または 'failed')
tasks	array	(オプション) タスクの配列
type	number	タスクのタイプ: 1 , ZBX_TM_TASK_PROCESS_REMOTE_COMMAND - リモートコマンドタスクの作成時間
clock	number	タスクの作成時間
ttd	number	タスクが期限切れになるまでの秒単位の時間
commandtype	number	リモートコマンドのタイプ: 0 , ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT - カスタムスクリプト使用 1 , ZBX_SCRIPT_TYPE_IPMI - IPMI 使用 2 , ZBX_SCRIPT_TYPE_SSH - SSH 使用 3 , ZBX_SCRIPT_TYPE_TELNET - Telnet 使用 4 , ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT - グローバルスクリプト使用 (現在、機能的にはカスタムスクリプトと同等)
command	string	実行するリモートコマンド

名前	値の型	説明
execute_on	number	カスタムスクリプトの実行ターゲット: 0 , ZBX_SCRIPT_EXECUTE_ON_AGENT - エージェントでスクリプト実行 1 , ZBX_SCRIPT_EXECUTE_ON_SERVER - サーバーでスクリプト実行 2 , ZBX_SCRIPT_EXECUTE_ON_PROXY - プロキシでスクリプト実行
port	number	(オプション) telnet および ssh コマンドのポート
authtype	number	(オプション) ssh コマンドの認証タイプ
username	string	(オプション) telnet および ssh コマンドのユーザー名
password	string	(オプション) telnet および ssh コマンドのパスワード
publickey	string	(オプション) ssh コマンドの公開鍵
privatekey	string	(オプション) ssh コマンドの秘密鍵
parent_taskid	number	親タスク ID
hostid	number	ターゲットホスト ID

例:

サーバーからプロキシ:

```
{
  "request": "proxy data"
}
```

プロキシからサーバー:

```
{
  "session": "12345678901234567890123456789012"
  "interface availability": [
    {
      "interfaceid": 1,
      "available": 1,
      "error": ""
    },
    {
      "interfaceid": 2,
      "available": 2,
      "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection refused"
    },
    {
      "interfaceid": 3,
      "available": 1,
      "error": ""
    },
    {
      "interfaceid": 4,
      "available": 1,
      "error": ""
    }
  ],
  "history data": [
    {
      "itemid": "12345",
      "clock": 1478609647,
      "ns": 332510044,
      "value": "52956612",
      "id": 1
    },
    {
      "itemid": "12346",
      "clock": 1478609647,
      "ns": 330690279,
      "state": 1,
      "value": "Cannot find information for this network interface in /proc/net/dev."
    }
  ]
}
```



```

    "id": 2
  }
],
"discovery data": [
  {
    "clock": 1478608764,
    "drule": 2,
    "dcheck": 3,
    "type": 12,
    "ip": "10.3.0.10",
    "dns": "vdebian",
    "status": 1
  },
  {
    "clock": 1478608764,
    "drule": 2,
    "dcheck": null,
    "type": -1,
    "ip": "10.3.0.10",
    "dns": "vdebian",
    "status": 1
  }
],
"auto registration": [
  {
    "clock": 1478608371,
    "host": "Logger1",
    "ip": "10.3.0.1",
    "dns": "localhost",
    "port": "10050"
  },
  {
    "clock": 1478608381,
    "host": "Logger2",
    "ip": "10.3.0.2",
    "dns": "localhost",
    "port": "10050"
  }
],
"tasks": [
  {
    "type": 0,
    "status": 0,
    "parent_taskid": 10
  },
  {
    "type": 0,
    "status": 1,
    "error": "No permissions to execute task.",
    "parent_taskid": 20
  }
],
"version": "5.4.0"
}

```

サーバーからプロキシ:

```

{
  "response": "success",
  "tasks": [
    {
      "type": 1,
      "clock": 1478608371,

```

```

    "ttl": 600,
    "commandtype": 2,
    "command": "restart_service1.sh",
    "execute_on": 2,
    "port": 80,
    "authtype": 0,
    "username": "userA",
    "password": "password1",
    "publickey": "MIGfMAOGCSqGSIB3DQEBAQUAA4GNADCBiQKBgQCqGKuk01De7zhZj6+H0qtjTkVxwTCpvKe",
    "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u091HNsj6tQ5QCqGKuk01De7zhd",
    "parent_taskid": 10,
    "hostid": 10070
  },
  {
    "type": 1,
    "clock": 1478608381,
    "ttl": 600,
    "commandtype": 1,
    "command": "restart_service2.sh",
    "execute_on": 0,
    "authtype": 0,
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "parent_taskid": 20,
    "hostid": 10084
  }
]
}

```

アクティブプロキシ

プロキシハートビートリクエスト

proxy heartbeat リクエストはプロキシが実行中であることを報告するためにプロキシによって送信されます。このリクエストは HeartbeatFrequency (プロキシ設定パラメータ) 秒ごとに送信されます。

名前	値のタイプ	説明
プロキシからサーバー:		
request	string	'proxy heartbeat'
host	string	プロキシ名
version	string	プロキシバージョン (<major>.<minor>.<build>)
サーバーからプロキシ:		
response	string	リクエストの成否 ('success' または 'failed')

プロキシからサーバー:

```

{
  "request": "proxy heartbeat",
  "host": "Proxy #12",
  "version": "5.4.0"
}

```

サーバーからプロキシ:

```

{
  "response": "success"
}

```

プロキシコンフィグリクエスト

proxy config リクエストはプロキシ設定データを取得するためにプロキシによって送信されます。このリクエストは、ConfigFrequency (プロキシ設定パラメータ) 秒ごとに送信されます。

名前	値の型	説明
プロキシからサーバー:		
request	string	'proxy config'
host	string	プロキシ名
version	string	プロキシバージョン (<メジャー>.<マイナー>.<ビルド>)
サーバーからプロキシ:		
request	string	'proxy config'
<table>	object	<table> データを持つ 1 つ以上のオブジェクト
fields	array	フィールド名の配列
-	string	フィールド名
data	array	行の配列
-	array	列の配列
-	string,number	データベーススキーマの列タイプに応じたタイプの列値
プロキシからサーバー:		
response	string	リクエストの成否 ('success' または 'failed')

例:

プロキシからサーバー:

```
{
  "request": "proxy config",
  "host": "Proxy #12",
  "version": "5.4.0"
}
```

サーバーからプロキシ:

```
{
  "globalmacro": {
    "fields": [
      "globalmacroid",
      "macro",
      "value"
    ],
    "data": [
      [
        2,
        "${SNMP_COMMUNITY}",
        "public"
      ]
    ]
  },
  "hosts": {
    "fields": [
      "hostid",
      "host",
      "status",
      "ipmi_authtype",
      "ipmi_privilege",
      "ipmi_username",
      "ipmi_password",
      "name",
      "tls_connect",
      "tls_accept",
      "tls_issuer",
      "tls_subject",
      "tls_psk_identity",
      "tls_psk"
    ],
    "data": [
      [
        10001,

```

```

        "Linux",
        3,
        -1,
        2,
        "",
        "",
        "Linux",
        1,
        1,
        "",
        "",
        "",
        ""
    ],
    [
        10050,
        "Zabbix Agent",
        3,
        -1,
        2,
        "",
        "",
        "Zabbix Agent",
        1,
        1,
        "",
        "",
        "",
        ""
    ],
    [
        10105,
        "Logger",
        0,
        -1,
        2,
        "",
        "",
        "Logger",
        1,
        1,
        "",
        "",
        "",
        ""
    ]
]
],
"interface":{
    "fields":[
        "interfaceid",
        "hostid",
        "main",
        "type",
        "useip",
        "ip",
        "dns",
        "port",
        "bulk"
    ],
    "data":[

```

```

        2,
        10105,
        1,
        1,
        1,
        "127.0.0.1",
        "",
        "10050",
        1
    ]
}
},
...
}

```

プロキシからサーバー:

```

{
  "response": "success"
}

```

プロキシデータリクエスト

proxy data リクエストはプロキシによって送信され、ホストインターフェースの可用性、ヒストリ、ディスカバリ、および自動登録データを提供します。このリクエストは DataSenderFrequency (プロキシ設定パラメータ) 秒ごとに送信されます。

名前	値の型	説明
プロキシからサーバー:		
request	string	'proxy data'
host	string	プロキシ名
session	string	データセッショントークン
interface	array	(オプション) インターフェイス可用性データオブジェクトの配列
availability		
interfaceid	number	インターフェース ID
available	number	インターフェースの可用性
		0 , INTERFACE_AVAILABLE_UNKNOWN - 不明 1 , INTERFACE_AVAILABLE_TRUE - 使用可能 2 , INTERFACE_AVAILABLE_FALSE - 使用不可
error	string	インターフェースエラーメッセージまたは空白
history	array	(オプション) ヒストリデータオブジェクトの配列
data		
itemid	number	アイテム ID
clock	number	アイテム値のタイムスタンプ (秒)
ns	number	アイテム値のタイムスタンプ (ナノ秒)
value	string	(オプション) アイテム値
id	number	ID (1 つのデータセッション内で一意の昇順カウンター)
timestamp	number	(オプション) ログタイプアイテムのタイムスタンプ
source	string	(オプション) イベントログアイテムのソース値
severity	number	(オプション) イベントログアイテムの深刻度
eventid	number	(オプション) イベントログアイテムのイベントログ ID

名前	値の型	説明
state	string	(オプション) アイテムの状態 0 , ITEM_STATE_NORMAL 1 , ITEM_STATE_NOTSUPPORTED
lastlogsize	number	(オプション) ログタイプアイテムの最後のログサイズ
mtime	number	(オプション) ログタイプの項目の時刻を変更
discovery data	array	(オプション) ディスカバリデータオブジェクトの配列
clock	number	ディスカバリデータのタイムスタンプ
druleid	number	ディスカバリルール ID
dcheckid	number	ディスカバリチェック ID またはヌルのディスカバリ・ルール・データ
type	number	ディスカバリチェックタイプ: -1 ディスカバリルールデータ 0 , SVC_SSH - SSH サービスチェック 1 , SVC_LDAP - LDAP サービスチェック 2 , SVC_SMTP - SMTP サービスチェック 3 , SVC_FTP - FTP サービスチェック 4 , SVC_HTTP - HTTP サービスチェック 5 , SVC_POP - POP サービスチェック 6 , SVC_NNTP - NNTP サービスチェック 7 , SVC_IMAP - IMAP サービスチェック 8 , SVC_TCP - TCP ポートの可用性チェック 9 , SVC_AGENT - Zabbix エージェント 10 , SVC_SNMPv1 - SNMPv1 エージェント 11 , SVC_SNMPv2 - SNMPv2 エージェント 12 , SVC_ICMPPING - ICMP ピング 13 , SVC_SNMPv3 - SNMPv3 エージェント 14 , SVC_HTTPS - HTTPS サービスチェック 15 , SVC_TELNET - Telnet の可用性チェック
ip	string	ホストの IP アドレス
dns	string	ホストの DNS 名
port	number	(オプション) サービスポート番号
key_	string	(オプション) タイプのディスカバリチェック用アイテムキー 9 SVC_AGENT
value	string	(オプション) サービスから受け取った値は、ほとんどのサービスで空になります
status	number	(オプション) サービスステータス: 0 , DOBJECT_STATUS_UP - サービスアップ 1 , DOBJECT_STATUS_DOWN - サービスダウン
auto registration	array	(オプション I) オートレジストレーションデータオブジェクトの配列
clock	number	オートレジストレーションデータのタイムスタンプ
host	string	ホスト名
ip	string	(オプション) IP アドレス
dns	string	(オプション) IP アドレスから解決された DNS 名
port	string	(オプション) ホストのポート
host_metadata	string	(オプション) エージェントによって送信されたホストメタデータ (HostMetadata または HostMetadataItem エージェント構成パラメータに基づく)
tasks	array	(オプション) タスクの配列
type	number	タスクのタイプ: 0 , ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT - リモートコマンドの結果


```

"interface availability": [
  {
    "interfaceid": 1,
    "available": 1,
    "error": ""
  },
  {
    "interfaceid": 2,
    "available": 2,
    "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection
  },
  {
    "interfaceid": 3,
    "available": 1,
    "error": ""
  },
  {
    "interfaceid": 4,
    "available": 1,
    "error": ""
  }
],
"history data":[
  {
    "itemid":"12345",
    "clock":1478609647,
    "ns":332510044,
    "value":"52956612",
    "id": 1
  },
  {
    "itemid":"12346",
    "clock":1478609647,
    "ns":330690279,
    "state":1,
    "value":"Cannot find information for this network interface in /proc/net/dev.",
    "id": 2
  }
],
"discovery data":[
  {
    "clock":1478608764,
    "drule":2,
    "dcheck":3,
    "type":12,
    "ip":"10.3.0.10",
    "dns":"vdebian",
    "status":1
  },
  {
    "clock":1478608764,
    "drule":2,
    "dcheck":null,
    "type":-1,
    "ip":"10.3.0.10",
    "dns":"vdebian",
    "status":1
  }
],
"auto registration":[
  {
    "clock":1478608371,

```



```

    "host": "Logger1",
    "ip": "10.3.0.1",
    "dns": "localhost",
    "port": "10050"
  },
  {
    "clock": 1478608381,
    "host": "Logger2",
    "ip": "10.3.0.2",
    "dns": "localhost",
    "port": "10050"
  }
],
"tasks": [
  {
    "type": 2,
    "clock": 1478608371,
    "ttl": 600,
    "commandtype": 2,
    "command": "restart_service1.sh",
    "execute_on": 2,
    "port": 80,
    "authtype": 0,
    "username": "userA",
    "password": "password1",
    "publickey": "MIGfMA0GCsqGSIB3DQEBAQUAA4GNADCBiQKBgQCqGKuk01De7zhZj6+H0qtjTkVxwTCpvKe",
    "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u091HNsj6tQ5QCqGKuk01De7zhd",
    "parent_taskid": 10,
    "hostid": 10070
  },
  {
    "type": 2,
    "clock": 1478608381,
    "ttl": 600,
    "commandtype": 1,
    "command": "restart_service2.sh",
    "execute_on": 0,
    "authtype": 0,
    "username": "",
    "password": "",
    "publickey": "",
    "privatekey": "",
    "parent_taskid": 20,
    "hostid": 10084
  }
],
"tasks": [
  {
    "type": 0,
    "status": 0,
    "parent_taskid": 10
  },
  {
    "type": 0,
    "status": 1,
    "error": "No permissions to execute task.",
    "parent_taskid": 20
  }
],
"version": "5.4.0"
}

```

サーバーからプロキシ:

```

{
  "response": "success",
  "upload": "enabled",
  "tasks": [
    {
      "type": 1,
      "clock": 1478608371,
      "ttl": 600,
      "commandtype": 2,
      "command": "restart_service1.sh",
      "execute_on": 2,
      "port": 80,
      "authtype": 0,
      "username": "userA",
      "password": "password1",
      "publickey": "MIGfMAOGCSqGSIB3DQEBAQUAA4GNADCBiQKBgQCqGKuk01De7zhZj6+H0qtjTkVxwTCpvKe",
      "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u091HNsj6tQ5QcGKuk01De7zhd",
      "parent_taskid": 10,
      "hostid": 10070
    },
    {
      "type": 1,
      "clock": 1478608381,
      "ttl": 600,
      "commandtype": 1,
      "command": "restart_service2.sh",
      "execute_on": 0,
      "authtype": 0,
      "username": "",
      "password": "",
      "publickey": "",
      "privatekey": "",
      "parent_taskid": 20,
      "hostid": 10084
    }
  ]
}

```

2 Zabbix エージェントのプロトコル

詳細については[パッシブおよびアクティブエージェントチェックページ](#)を参照してください。

3 Zabbix エージェント 2 プロトコル

概要

このセクションでは、次の情報を提供します

- Agent2 -> Server : アクティブチェックリクエスト
- Server -> Agent2 : アクティブチェックレスポンス
- Agent2 -> Server : エージェントデータリクエスト
- Server -> Agent2 : エージェントデータレスポンス

アクティブチェックリクエスト

アクティブチェックリクエストは、エージェントによって処理されるアクティブチェックを取得するために使用されます。このリクエストは開始時にエージェントによって送信され、その後は RefreshActiveChecks で設定されている間隔で送信されます。

フィールド	タイプ	必須	値
request	string	はい	active checks

フィールド	タイプ	必須	値
host	string	はい	ホスト名
version	string	はい	エージェントのバージョン: <メジャー>.<マイナー>.
host_metadata	string	いいえ	設定パラメータ HostMetadata または HostMetadataItem メトリック値
interface	string	いいえ	設定パラメータ HostInterface または HostInterfaceItem メトリック値
ip	string	いいえ	設定パラメータ ListenIP が設定されている場合は先頭の IP
port	number	いいえ	設定パラメータ ListenPort 値 (設定されている場合) であり、デフォルトのエージェント・リスニング・ポートではありません。

例:

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "version": "6.0",
  "host_metadata": "mysql,nginx",
  "hostinterface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050
}
```

アクティブチェックレスポンス

アクティブチェックレスポンスは、アクティブチェックリクエストの処理後にサーバーからエージェントに返されます。

フィールド	タイプ	必須	値
response	string	yes	success failed
info	string	no	失敗した場合のエラー情報
data	array of objects	no	アクティブチェックアイテム
key	string	no	マクロが展開されたアイテムキー
itemid	number	no	アイテム ID
delay	string	no	アイテムの更新間隔
lastlogsize	number	no	アイテムの最終ログサイズ
mtime	number	no	アイテム修正時間
regexp	array of objects	no	グローバル正規表現
name	string	no	Global 正規表現名
expression	string	no	グローバル正規表現
expression_type	number	no	グローバル正規表現タイプ
exp_delimiter	string	no	グローバル正規表現の区切り文字
case_sensitive	number	no	グローバル正規表現の大文字小文字の区別設定

例:

```
{
  "response": "success",
  "data": [
    {
      "key": "log[/home/zabbix/logs/zabbix_agentd.log]",
      "itemid": 1234,
      "delay": "30s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0
    }
  ]
}
```

エージェントデータリクエスト

エージェントデータリクエストには、収集されたアイテムの値が含まれます。

フィールド	タイプ	必須	値
request	string	はい	agent data
host	string	はい	ホスト名
version	string	はい	エージェントバージョン: <メジャー>.<マイナー>.
session	string	はい	エージェントが開始されるたびに生成される固有のセッション ID
data	array of objects	はい	アイテムデータオブジェクト
id	number	はい	値の識別子 (ネットワークの問題が発生した場合に、重複した値をチェックするために)
itemid	number	はい	アイテム ID
value	string	いいえ	アイテム値
lastlogsize	number	いいえ	アイテムの最終ログサイズ
mtime	number	いいえ	アイテム修正日時
state	number	いいえ	アイテムステータス
source	string	いいえ	イベントログソース
eventid	number	いいえ	イベントログイベント ID
severity	number	いいえ	イベントログの深刻度
timestamp	number	いいえ	イベントログのタイムスタンプ
clock	number	はい	タイムスタンプ (エポックからの秒数)
ns	number	はい	タイムスタンプナノ秒

例:

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "2.4.0",
      "clock": 1400675595,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "lastlogsize": 112,
      "value": " 19845:20140621:141708.521 Starting Zabbix Agent [<hostname>]. Zabbix 2.4.0 (revision 5000)",
      "clock": 1400675595,
      "ns": 77053975
    }
  ],
  "host": "Zabbix server",
  "version": "6.0",
  "session": "1234456akdsjhfoi"
}
```

エージェントデータレスポンス

エージェントデータレスポンス、エージェントデータリクエストを処理した後、サーバーによってエージェントに送り返されます。| フィールド | タイプ | 必須 | 値 | |---| | response | string | はい | success または failed | | info | string | はい | アイテム処理結果 |

例:

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

5 Zabbix sender プロトコル

詳細については[トラッパーパーアイテム](#)ページを参照してください。

6 ヘッダー

概要

ヘッダーは、Zabbix コンポーネント間のすべての要求および応答メッセージに存在します。ヘッダーの情報からメッセージの長さ、圧縮されているかどうか、大きなパケットであるかどうかを判断できます。

Zabbix 通信プロトコルには、接続ごとに 1GB のパケットサイズ制限があります。受信パケットのデータ長、圧縮前のデータ長ともに 1GB の制限が適用されます。

構成を Zabbix プロキシに送信する場合、大きな構成を同期できるように、パケットサイズの制限が 4 GB に増加します。圧縮前のデータ長が 4GB を超えると、Zabbix サーバーは自動的に大きなパケット形式 (0x04 フラグ) を使用して起動し、パケットサイズの制限を 16GB に増やします。

任意のデータの送信に大きなパケット形式を使用できますが、現在、1GB を超えるパケットを処理できるのは Zabbix プロキシ構成シンカーのみであることを注意してください。

構造

ヘッダーは 4 つのフィールドで構成されます。ヘッダー内のすべての数値は、リトルエンディアンとしてフォーマットされています。

フィールド	サイズ	サイズ (ラージパケット)	説明
<PROTOCOL>	4	4	"ZBXD" または 5A 42 58 44
<FLAGS>	1	1	Protocol フラグ: 0x01 - Zabbix 通信プロトコル 0x02 - 圧縮 0x04 - ラージパケット
<DATALEN>	4	8	データ長
<RESERVED>	4	8	圧縮を使用する場合 (0x02 フラグ) - 非圧縮データの長さ 圧縮を使用しない場合 - 00 00 00 00

例

以下は、Zabbix に送信する必要があるパケットを取得して正しく解釈されるようにするために、送信するデータに Zabbix プロトコルヘッダーを追加する方法を示すコードスニペットです。これらのコードスニペットは、データが 1GB を超えないことを前提としているため、ラージパケットフォーマットは使用されません。

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

または

```
def zbx_create_header(plain_data_size, compressed_data_size=None):
    protocol = b"ZBXD"
    flags = 0x01
    if compressed_data_size is None:
        datalen = plain_data_size
        reserved = 0
    else:
        flags |= 0x02
        datalen = compressed_data_size
        reserved = plain_data_size
    return protocol + struct.pack("<BIII", flags, datalen, reserved)
```

```
packet = zbx_create_header(len(data)) + data
```

Perl

```
my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;
```

または

```

sub zbx_create_header($;$)
{
    my $plain_data_size = shift;
    my $compressed_data_size = shift;

    my $protocol = "ZBXD";
    my $flags = 0x01;
    my $datalen;
    my $reserved;

    if (!defined($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }

    return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);
}

my $packet = zbx_create_header(length($data)) . $data;

```

PHP

```
$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;
```

または

```

function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;

```

Bash

```

datalen=$(printf "%08x" ${#data})
datalen="\x${datalen:6:2}\x${datalen:4:2}\x${datalen:2:2}\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"

```

7 リアルタイムエクスポートプロトコル

このセクションではリアルタイムエクスポートプロトコルの詳細を改行区切りのJSON形式で示します。

- トリガーイベント

- アイテム値
- トレンド

すべてのファイルの拡張子は.ndjson です。エクスポートファイルの各行は JSON オブジェクトです。

トリガーイベント

障害イベントについて、次の情報がエクスポートされます。

フィールド	タイプ	説明
clock	number	エポックから障害が検出されるまでの秒数 (整数部)
ns	number	正確な障害検出時間を取得するために clock に追加されるナノ秒数。
value	number	1 (常に)
eventid	number	障害イベント ID
name	string	障害イベント名
severity	number	障害イベントの深刻度 (0 - 未分類,1 - 情報,2 - 警告,3 - 軽度の障害,4 - 重度の障害,5 - 致命的な障害)
hosts	array	トリガー式に関係するホストのリスト。配列には少なくとも 1 つの要素が必要です。
-	object	
host	string	ホスト名
name	string	表示用ホスト名
groups	array	トリガー式に関係するすべてのホストのホストグループのリスト。配列には少なくとも 1 つの要素が必要です。
-	string	ホストグループ名
tags	array	障害タグのリスト (空白可)
-	object	
tag	string	タグ名
value	string	タグ値 (空白可)。

次の情報は、リカバリエvent用にエクスポートされます。

フィールド	タイプ	説明
clock	number	エポックから障害が解決するまでの秒数 (整数部)
ns	number	正確な障害解決時間を取得するために clock に追加されるナノ秒数。
value	number	0 (常に)
eventid	number	リカバリエvent ID
p_eventid	number	障害イベント ID

例

障害:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

リカバリ:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

障害 (多重障害イベントの生成):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

リカバリ:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}
```

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

アイテム値

収集されたアイテム値について次の情報がエクスポートされます。

フィールド	タイプ	説明
host	object	アイテムホストのホスト名
	host string	ホスト名
	name string	表示ホスト名
groups	array	アイテムホストのホストグループのリスト。配列には少なくとも 1 つの要素が必要です。
	- string	ホストグループ名
itemid	number	アイテム ID
name	string	表示アイテム名
clock	number	エポックから値が収集されるまでの秒数 (整数部)
ns	number	正確な値の収集時間を取得するために clock に追加されるナノ秒数
timestamp (Log only)	number	利用できない場合は 0
source (Log only)	string	使用できない場合は空の文字列
severity (Log only)	number	利用できない場合は 0
eventid (Log only)	number	利用できない場合は 0
value	number (for numeric items) or string (for text items)	収集したアイテム値
type	number	収集された値のタイプ: 0 - 浮動小数点, 1 - 文字, 2 - ログ, 3 - 符号なしの数値, 4 - テキスト

例

数字 (符号なし) 値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"itemid":3,"name":"Host B visible"}
```

数字 (浮動小数) 値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"itemid":4,"name":"Host B visible"}
```

文字, テキスト値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"itemid":2,"name":"Host B visible"}
```

ログ値:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"itemid":1,"name":"Host A visible"}
```

トレンド

計算されたトレンド値について次の情報がエクスポートされます。

フィールド	タイプ	説明
host	object	アイテムホストのホスト名
	host string	ホスト名
	name string	表示ホスト名
groups	array	アイテムホストのホストグループのリスト。配列には少なくとも 1 つの要素が必要です。
	- string	ホストグループ名
itemid	number	アイテム ID
name	string	表示アイテム名
clock	number	エポックから値が収集されるまでの秒数 (整数部)
count	number	特定の時間に収集された値の数
min	number	特定の時間の最小アイテム値
avg	number	特定の時間の平均アイテム値
max	number	特定の時間の最大アイテム値
type	number	値のタイプ: 0 - 浮動小数, 3 - 符号なしの整数

例

数字 (符号なし) 値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"itemid":3,"name":"Host B visible"}
```

数字 (浮動小数点) 値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"itemid":4,"name":"Host B visible"}
```

Code

Type	Size	Comments
Byte	4	Payload type, currently only JSON is supported.

Size

Type	Size	Comments
Byte	4	Size of the current payload in bytes.

Payload data

Type	Size	Comments
Byte	Defined by the Size field	JSON formatted data.

Payload data definition

Common data

These parameters are present in all requests/responses:

Name	Type	Comments
id	uint32	For requests - the incrementing identifier used to link requests with responses. Unique within a request direction (i.e. from agent to plugin or from plugin to agent).
type	uint32	For responses - ID of the corresponding request. The request type.

Log request

A request sent by a plugin to write a log message into the agent log file.

direction	plugin → agent
response	no

Parameters specific to log requests:

Name	Type	Comments
severity	uint32	The message severity (log level).
message	string	The message to log.

Example:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Register request

A request sent by the agent during the agent startup phase to obtain provided metrics to register a plugin.

direction	agent → plugin
response	yes

Parameters specific to register requests:

Name	Type	Comments
version	string	The protocol version <major>.<minor>

Example:

```
{"id":1,"type":2,"version":"1.0"}
```

Register response

Plugin's response to the register request.

direction	plugin → agent
response	n/a

Parameters specific to register responses:

Name	Type	Comments
name	string	The plugin name.
metrics	array of strings (optional)	The metrics with descriptions as used in the plugin. Returns RegisterMetrics(). Absent if error is returned.
interfaces	uint32 (optional)	The bit mask of plugin's supported interfaces. Absent if error is returned.
error	string (optional)	An error message returned if a plugin cannot be started. Absent, if metrics are returned.

Examples:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

or

```
{"id":2,"type":3,"error":"error message"}
```

Start request

A request to execute the Start function of the Runner interface.

direction	agent → plugin
response	no

The request doesn't have specific parameters, it only contains **common data** parameters.

Example:

```
{"id":3,"type":4}
```

Terminate request

A request sent by the agent to shutdown a plugin.

direction	agent → plugin
-----------	----------------

response	no
----------	----

The request doesn't have specific parameters, it only contains **common data** parameters.

Example:

```
{"id":3,"type":5}
```

Export request

A request to execute the Export function of the Exporter interface.

direction	agent → plugin
response	no

Parameters specific to export requests:

Name	Type	Comments
key	string	The plugin key.
parameters	array of strings (optional)	The parameters for Export function.

Example:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Export response

Response from the Export function of the Exporter interface.

direction	plugin → agent
response	n/a

Parameters specific to export responses:

Name	Type	Comments
value	string (optional)	Response value from the Export function. Absent, if error is returned.
error	string (optional)	Error message if the Export function has not been executed successfully. Absent, if value is returned.

Examples:

```
{"id":5,"type":7,"value":"response"}
```

or

```
{"id":5,"type":7,"error":"error message"}
```

Configure request

A request to execute the Configure function of the Configurator interface.

direction	agent → plugin
response	n/a

Parameters specific to Configure requests:

Name	Type	Comments
global_options	JSON object	JSON object containing global agent configuration options.
private_options	JSON object (optional)	JSON object containing private plugin configuration options, if provided.

Example:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Validate request

A request to execute Validate function of the Configurator interface.

direction	agent → plugin
response	yes

Parameters specific to Validate requests:

Name	Type	Comments
private_options	JSON object (optional)	JSON object containing private plugin configuration options, if provided.

Example:

```
{"id":7,"type":9,"private_options":{...}}
```

Validate response

Response from Validate function of Configurator interface.

direction	plugin → agent
response	n/a

Parameters specific to Validate responses:

Name	Type	Comments
error	string (optional)	An error message returned if the Validate function is not executed successfully. Absent if executed successfully.

Example:

```
{"id":8,"type":10}
```

or

```
{"id":8,"type":10,"error":"error message"}
```

4 Zabbix agent 2 plugin protocol Zabbix agent 2 protocol is based on code, size and data model.

5 アイテム

1 各プラットフォームでサポートされているアイテム

この表は各プラットフォームでの Zabbix エージェントアイテムのサポートを示しています。

- X のマークが付いているアイテムはサポートされていますが、- のマークが付いているアイテムはサポートされていません。
- ? のマークが付いている場合、サポートされているかどうか不明です。
- r のマークが付いている場合は、root 権限が必要であることを意味します。
- 山かっこ < このように > に含まれるパラメータはオプションです。

Note:

Windows のみの Zabbix エージェントアイテムはこの表に含まれていません。

	1	2	3	4	5	6	7	8	9	10	11
NetBSD											▼▼
OpenBSD											▼▼
Mac										▼▼	
OS X											
Tru64											▼▼
AIX											▼▼
HP-UX											▼▼
Solaris											▼▼
FreeBSD											▼▼
Linux											▼▼
2.6 (and later)											▼▼
Linux											▼▼
2.4											▼▼
Windows	▼▼										
▼ Item											
▼											
agent.hostmetadata		X	X	X	X	X	X	X	X	X	X
agent.hostname	X	X	X	X	X	X	X	X	X	X	X
agent.ping	X	X	X	X	X	X	X	X	X	X	X
agent.variant	X	X	X	X	X	X	X	X	X	X	X
agent.version	X	X	X	X	X	X	X	X	X	X	X
kernel.maxfiles	-	X	X	X	-	-	-	?	X	X	X
kernel.maxproc	-	-	X	X	X	-	-	?	X	X	X
kernel.openfiles	-	X	X	?	?	?	?	?	?	?	?
log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<persistent_dir>]											X
persistent_dir	-	X	X	X	X	X	X	X	X	X	X
▲											
log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<persistent_dir>]											X
persistent_dir	-	X	X	X	X	X	X	X	X	X	X
▲											
logrt[file_regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent_dir>]											
persistent_dir	-	X	X	X	X	X	X	X	X	X	X
▲											
logrt.count[file_regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent_dir>]											
persistent_dir	-	X	X	X	X	X	X	X	X	X	X
▲											
modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]		X	-	-	-	-	-	-	-	-	-
net.dns[<ip>,<zone>,<type>,<timeout>,<count>]	X	X	X	X	X	X	X	X	X	X	X
net.dns.record[<ip>,<zone>,<type>,<timeout>,<count>]	X	X	X	X	X	X	X	X	X	X	X
net.if.collisions[if]	X	X	X	X	-	X	-	X	X	X	r
net.if.discovery	X	X	X	X	X	X	-	-	X	X	X
net.if.in[if,<mode>]	X	X	X	X	X ¹	X	-	X	X	X	r
mode	bytes	X	X	X	X ²	X	-	X	X	X	r
▲											
(de-fault)											
packets	X	X	X	X	X	X	X	-	X	X	r

errors	X	X	X	X	X ²	X	X	-	X	X	r		
dropped	X	X	X	X	-	X	-	-	X	X	r		
overruns-		X	X	-	-	-	-	-	-	-	-		
frame	-	X	X	-	-	-	-	-	-	-	-		
compressed		X	X	-	-	-	-	-	-	-	-		
multicast-		X	X	-	-	-	-	-	-	-	-		
net.if.out[if,<mode>]		X	X	X	X	X ¹	X	-	X	X	r		
mode	bytes	X	X	X	X ²	X	X	-	X	X	r		
▲	(de-												
	fault)												
packets	X	X	X	X	X	X	X	-	X	X	r		
errors	X	X	X	X	X ²	X	X	-	X	X	r		
dropped	X	X	X	-	-	X	-	-	-	-	-		
overruns-		X	X	-	-	-	-	-	-	-	-		
collision	-	X	X	-	-	-	-	-	-	-	-		
carrier	-	X	X	-	-	-	-	-	-	-	-		
compressed		X	X	-	-	-	-	-	-	-	-		
net.if.total[if,<mode>]		X	X	X	X	X ¹	X	-	X	X	r		
mode	bytes	X	X	X	X ²	X	X	-	X	X	r		
▲	(de-												
	fault)												
packets	X	X	X	X	X	X	X	-	X	X	r		
errors	X	X	X	X	X ²	X	X	-	X	X	r		
dropped	X	X	X	-	-	X	-	-	-	-	-		
overruns-		X	X	-	-	-	-	-	-	-	-		
collision	-	X	X	-	-	-	-	-	-	-	-		
carrier	-	X	X	-	-	-	-	-	-	-	-		
compressed		X	X	-	-	-	-	-	-	-	-		
net.tcp.listen[port]		X	X	X	X	-	-	-	X	-	-		
net.tcp.port[<ip>,<port>]		X	X	X	X	X	X	X	X	X	X		
net.tcp.service[service,<ip>,<port>]				X	X	X	X	X	X	X	X		
net.tcp.service.perf[service,<ip>,<port>]				X	X	X	X	X	X	X	X		
net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]													
net.udp.listen[port]		X	X	X	X	-	-	-	X	-	-		
net.udp.service[service,<ip>,<port>]				X	X	X	X	X	X	X	X		
net.udp.service.perf[service,<ip>,<port>]				X	X	X	X	X	X	X	X		
net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]													
	1	2	3	4	5	6	7	8	9	10	11		
proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]													
type	total	-	X	X	-	X	-	-	-	-	-		
▲	(de-												
	fault)												
user	-	X	X	-	X	-	-	-	-	-	-		
system	-	X	X	-	X	-	-	-	-	-	-		
mode	avg1	-	X	X	-	X	-	-	-	-	-		
▲	(de-												
	fault)												
avg5	-	X	X	-	X	-	-	-	-	-	-		
avg15	-	X	X	-	X	-	-	-	-	-	-		
zone	current	-	-	-	X	-	-	-	-	-	-		
▲	(de-												
	fault)												
all	-	-	-	-	X	-	-	-	-	-	-		
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]								X	X	-	X	X	
mode	sum	-	X	X	X	X	-	X	X	-	X	X	
▲	(de-												
	fault)												
avg	-	X	X	X	X	-	X	X	-	X	X		
max	-	X	X	X	X	-	X	X	-	X	X		
min	-	X	X	X	X	-	X	X	-	X	X		
memtype	-	X	X	X	X	-	X	-	-	-	-		
▲													
proc.num[<name>,<user>,<state>,<cmdline>,<zone>]								X	X	X	-	X	X

state	all	-	X	X	X	X	X	X	X	-	X	X
▲	(de-fault)											
	disk	-	X	X	X	-	-	-	-	-	X	X
	sleep	-	X	X	X	X	X	X	X	-	X	X
	zomb	-	X	X	X	X	X	X	X	-	X	X
	run	-	X	X	X	X	X	X	X	-	X	X
	trace	-	X	X	X	-	-	-	-	-	X	X
cmdline		-	X	X	X	X	X	X	X	-	X	X
▲												
zone	current	-	-	-	-	X	-	-	-	-	-	-
▲	(de-fault)											
	all	-	-	-	-	X	-	-	-	-	-	-
	sensor[device,sensor,<mode>]		X	-	-	-	-	-	-	-	X	-
	system.boottime		X	X	X	X	-	-	-	X	X	X
	system.cpu.discovery		X	X	X	X	X	X	X	X	X	X
	system.cpu.intr	-	X	X	X	X	-	X	-	-	X	X
	system.cpu.load[<cpu>,<mode>]		X	X	X	X	X	X	X	X	X	X
cpu ▲	all	X	X	X	X	X	X	X	X	X	X	X
	(de-fault)											
	percpu	X	X	X	X	X	X	X	-	X	X	X
mode	avg1	X	X	X	X	X	X	X	X	X	X	X
▲	(de-fault)											
	avg5	X	X	X	X	X	X	X	X	X	X	X
	avg15	X	X	X	X	X	X	X	X	X	X	X
	system.cpu.num[<type>]		X	X	X	X	X	X	-	X	X	X
type	online	X	X	X	X	X	X	X	-	X	X	X
▲	(de-fault)											
	max	-	X	X	X	X	-	-	-	X	-	-
	system.cpu.switches		X	X	X	X	-	X	-	-	X	X
	system.cpu.util[<cpu>,<type>,<mode>,<logical_or_physical>]		X	X	X	X	X	X	X	-	X	X
type	user	-	X	X	X	X	X	X	X	-	X	X
▲	(de-fault)											
	nice	-	X	X	X	-	X	-	X	-	X	X
	idle	-	X	X	X	X	X	X	X	-	X	X
	system	X	X	X	X	X	X	X	X	-	X	X
	(de-fault for Windows)											
	iowait	-	-	X	-	X	-	X	-	-	-	-
	interrupt-softirq	-	-	X	X	-	-	-	-	-	X	-
	steal	-	-	X	-	-	-	-	-	-	-	-
	guest	-	-	X	-	-	-	-	-	-	-	-
	guest_nice	-	-	X	-	-	-	-	-	-	-	-
mode	avg1	X	X	X	X	X	X	X	X	-	X	X
▲	(de-fault)											
	avg5	X	X	X	X	X	X	X	-	-	X	X
	avg15	X	X	X	X	X	X	X	-	-	X	X
logical_or_physical												
▲	(de-fault)											
	physical	-	-	-	-	-	-	X	-	-	-	-
		1	2	3	4	5	6	7	8	9	10	11

system.hostname [<type>,<transform>]	X	X	X	X	X	X	X	X	X	X	X	X	
system.hw.chassis [<info>]	X	-	-	-	-	-	-	-	-	-	-	-	
system.hw.cpu [<cpu>,<info>]	X	-	-	-	-	-	-	-	-	-	-	-	
system.hw.devices [<type>]	X	-	-	-	-	-	-	-	-	-	-	-	
system.hw.macaddr [<interface>,<format>]	-	-	-	-	-	-	-	-	-	-	-	-	
system.localtime [<type>]	X	X	X	X	X	X	X	X	X	X	X	X	
type	utc	X	X	X	X	X	X	X	X	X	X	X	
▲	(de- fault)												
	local	X	X	X	X	X	X	X	X	X	X	X	
system.run [command,<mode>]	X	X	X	X	X	X	X	X	X	X	X	X	
mode	wait	X	X	X	X	X	X	X	X	X	X	X	
▲	(de- fault)												
	nowait	X	X	X	X	X	X	X	X	X	X	X	
system.stat [resource,<type>]	-	-	-	-	X	-	-	-	-	-	-	-	
system.sw.arch	X	X	X	X	X	X	X	X	X	X	X	X	
system.sw.os [<info>]	X	X	-	-	-	-	-	-	-	-	-	-	
system.sw.packages [<package>,<manager>,<format>]	-	-	-	-	-	-	-	-	-	-	-	-	
system.swap.in [<device>,<type>]	-	-	X	-	-	-	-	-	-	X	-	-	
(device の指 定は Linux での みサ ポー トさ れて いま す)	type	count	-	X	X	-	X	-	-	-	-	X	-
▲	(ペー ジは デバ イス が指 定さ れて いな い場 合に のみ 機能 しま す)	(Linux 以外 での デフ オル ト)											
	sectors	-	X	X	-	-	-	-	-	-	-	-	-
	pages	-	X	X	-	X	-	-	-	-	X	-	-
	(Linux での デフ オル ト)												

system.swap.out[<device>,<type>]												
(device の指 定は Linux での みサ ポー トさ れて いま す)												
type	count	-	X	X	-	X	-	-	-	-	X	-
▲ (ペー ジは デバ イス が指 定さ れて いな い場 合に のみ 機能 しま す)												
	sectors	-	X	X	-	-	-	-	-	-	-	-
	pages	-	X	X	-	X	-	-	-	-	X	-
(Linux での デフ オル ト)												
system.swap.size[<device>,<type>]												
(デバ イス の指 定は FreeBSD での みサ ポー トさ れて いま す。他 のプ ラッ トフ オー ムで は空 また は"all" であ る必 要が あり ます)												
		X	X	-	X	X	-	X	X	-	X	-

type	free	X	X	X	X	X	-	X	X	-	X	-
▲	(de-fault)											
	total	X	X	X	X	X	-	X	X	-	X	-
	used	X	X	X	X	X	-	X	X	-	X	-
	pfree	X	X	X	X	X	-	X	X	-	X	-
	pusd	X ⁶	X	X	X	X	-	X	X	-	X	-
	system.uptime	X	X	X	X	X	X	X	X	X	X	X
	system.users.num	X	X	X	X	X	-	X	?	X	X	X
	systemd.unit.discovery	X	X	-	-	-	-	-	-	-	-	-
	systemd.unit.get	X	X	-	-	-	-	-	-	-	-	-
	systemd.unit.info	X	X	-	-	-	-	-	-	-	-	-
		1	2	3	4	5	6	7	8	9	10	11
	vfs.dev.discovery	X	X	-	-	-	-	-	-	-	-	-
	vfs.dev.read[<device>,<type>,<mode>]	X	X	-	-	-	-	X	-	-	X	-
type	sectors	-	X	X	-	-	-	-	-	-	-	-
▲												
	operations (OpenBSD と AIX のデフォルト)	X	X	X	X	-	X	-	-	-	X	-
	bytes (Solaris のデフォルト)	-	-	-	X	X	-	X	-	-	X	-
	sps (Linux のデフォルト)	-	X	X	-	-	-	-	-	-	-	-
	ops	-	X	X	X	-	-	-	-	-	-	-
	bps (FreeBSD のデフォルト)	-	-	-	X	-	-	-	-	-	-	-
mode	avg1 (de-fault)	-	X	X	X	-	-	-	-	-	-	-
▲	(タイプライントのみ互換性があります: sps,ops,bps)											
	avg5	-	X	X	X	-	-	-	-	-	-	-
	avg15	-	X	X	X	-	-	-	-	-	-	-
	vfs.dev.write[<device>,<type>,<mode>]	X	X	-	-	-	-	X	-	-	X	-
type	sectors	-	X	X	-	-	-	-	-	-	-	-
▲												

operations (OpenBSD と AIX のデ フォ ルト)	X	X	X	X	-	X	-	-	X	-	
bytes (So- laris のデ フォ ルト)	-	-	-	X	X	-	X	-	-	X	-
sps (Linux のデ フォ ルト)	-	X	X	-	-	-	-	-	-	-	-
ops	-	X	X	X	-	-	-	-	-	-	-
bps (FreeBSD のデ フォ ルト)	-	-	-	X	-	-	-	-	-	-	-
mode ▲ (de- fault)	-	X	X	X	-	-	-	-	-	-	-
▲ (タイ プイ ンと のみ 互換性 があ りま す： sps,ops,bps)											
avg5	-	X	X	X	-	-	-	-	-	-	-
avg15	-	X	X	X	-	-	-	-	-	-	-
vfs.dir.count [dir,<regex_incl>,<regex_excl>,<types_incl>,<types_excl>,<max_depth>,<min_size>,<max_size>,<min_age>]											
vfs.dir.get [dir,<regex_incl>,<regex_excl>,<types_incl>,<types_excl>,<max_depth>,<min_size>,<max_size>,<min_age>]											
vfs.dir.size [dir,<regex_incl>,<regex_excl>,<mode>,<max_depth>,<regex_excl_dir>]?									?	?	
vfs.file.cksum [file,<mode>]			X	X	X	X	X	X	X	X	X
vfs.file.contents [file,<encoding>]			X	X	X	X	X	X	X	X	X
vfs.file.exists [file,<types_incl>,<types_excl>]			X	X	X	X	X	X	X	X	X
vfs.file.get [file]	X	X	X	X	X	X	X	X	X	X	X
vfs.file.md5sum [file]		X	X	X	X	X	X	X	X	X	X
vfs.file.owner [file,<owner>,<resulttype>]			X	X	X	X	X	X	X	X	X
vfs.file.permissions [file]	X	X	?	?	?	?	?	?	?	?	?
vfs.file.regexp [file,regexp,<encoding>,<output>]	X	X	X	X	X	X	X	X	X	X	X
vfs.file.regmatch [file,regexp,<encoding>]	X	X	X	X	X	X	X	X	X	X	X
vfs.file.size [file,<mode>]	X	X	X	X	X	X	X	X	X	X	X
	1	2	3	4	5	6	7	8	9	10	11
vfs.file.time [file,<mode>]	X	X	X	X	X	X	X	X	X	X	X
mode modify	X	X	X	X	X	X	X	X	X	X	X
▲ (de- fault)											
access	X	X	X	X	X	X	X	X	X	X	X
change	X ⁵	X	X	X	X	X	X	X	X	X	X
vfs.fs.discovery	X	X	X	X	X	X	X	-	X	X	X
vfs.fs.get	X	X	X	X	X	X	X	-	X	X	X
vfs.fs.inode [fs,<mode>]	X	X	X	X	X	X	X	X	X	X	X

mode	total	-	X	X	X	X	X	X	X	X	X	X
▲	(de-fault)											
	free	-	X	X	X	X	X	X	X	X	X	X
	used	-	X	X	X	X	X	X	X	X	X	X
	pfree	-	X	X	X	X	X	X	X	X	X	X
	pusued	-	X	X	X	X	X	X	X	X	X	X
	vfs.fs.size[fs,<mode>]		X	X	X	X	X	X	X	X	X	X
mode	total	X	X	X	X	X	X	X	X	X	X	X
▲	(de-fault)											
	free	X	X	X	X	X	X	X	X	X	X	X
	used	X	X	X	X	X	X	X	X	X	X	X
	pfree	X	X	X	X	X	X	X	X	X	X	X
	pusued	X	X	X	X	X	X	X	X	X	X	X
	vm.memory.size[*mode>]		X	X	X	X	X	X	X	X	X	X
mode	total	X	X	X	X	X	X	X	X	X	X	X
▲	(de-fault)											
	active	-	-	-	X	-	X	-	-	X	X	X
	anon	-	-	-	-	-	-	-	-	-	-	X
	buffers	-	X	X	X	-	-	-	-	-	X	X
	cached	X	X	X	X	-	-	X	-	-	X	X
	exec	-	-	-	-	-	-	-	-	-	-	X
	file	-	-	-	-	-	-	-	-	-	-	X
	free	X	X	X	X	X	X	X	X	X	X	X
	inactive	-	-	-	X	-	-	-	-	X	X	X
	pinned	-	-	-	-	-	-	X	-	-	-	-
	shared	-	X	-	X	-	-	-	-	-	X	X
	wired	-	-	-	X	-	-	-	-	X	X	X
	used	X	X	X	X	X	X	X	X	X	X	X
	pusued	X	X	X	X	X	X	X	X	X	X	X
	available	X	X	X	X	X	X	X	X	X	X	X
	pavailable	X	X	X	X	X	X	X	X	X	X	X
	web.page.get[host,<path>,<port>]				X	X	X	X	X	X	X	X
	web.page.perf[host,<path>,<port>]				X	X	X	X	X	X	X	X
	web.page.regex[host,<path>,<port>,<regexp>,<length>,<output>]				X	X	X	X	X	X	X	X
		1	2	3	4	5	6	7	8	9	10	11

Note:

vm.memory.size パラメーターの説明も参照してください。

脚注

- ¹ net.if.in、net.if.out、および net.if.total アイテムは、ループバックインターフェイス（例：lo0）の統計を提供しません。
- ² これらの項目の値は Solaris 10 6/06 までの Solaris システムのループバックインターフェイスではサポートされていません。バイト、エラー、および使用率の統計は、カーネルによって保存および/または報告されません。ただし net-snmp を介して Solaris システムを監視している場合、net-snmp が 1997 年の cmu-snmp からのレガシーコードを伝送するため、値が返される場合があります。このコードは、インターフェイス統計からのバイト値の読み取りに失敗すると、パケットカウンタ（ループバックインターフェイスに存在します）に任意の値 308 を掛けます。これにより、パケットの平均長が 308 オクテットであると想定されます。ループバックインターフェイスの Solaris システムの MTU 制限は 8892 バイトであるため、これは非常に大まかな見積もりです。
- ³ /proc/pid/psinfo から取得した Solaris のコマンドラインは 80 バイトに制限されており、プロセスが開始されたときのコマンドラインが含まれています。
- ⁴ Windows イベントログではサポートされていません。
- ⁵ Windows XP では [file,change] は vfs.file.time[file,access] と同じである可能性があります。
- ⁶ Zabbix エージェント 2 でのみサポートされます。Zabbix エージェントではサポートされていません。
- ⁷ 64 ビット Windows 上の Zabbix エージェント 2 でのみサポートされます。Zabbix エージェントではサポートされていません。

2 vm.memory.size パラメーター

概要

このセクションでは `vm.memory.size[<mode>]` エージェントアイテムのパラメーターの詳細を説明します。

パラメーター

このアイテムでは、次のパラメーターを使用できます。

- **active** - 現在使用されている、または最近使用されたメモリで、RAM にあるもの
- **anon** - ファイルと関係ないメモリ (ファイルから再読み込みできないメモリ)
- **available** - プラットフォーム毎に異なる方法で計算された使用可能なメモリ (下の表を参照)
- **buffers** - ファイルシステムのメタデータのようなもののキャッシュ
- **cached** - ささまざまなもののキャッシュ
- **exec** - 実行コード。通常、(プログラム) ファイルから実行されます。
- **file** - 最近アクセスされたファイルコンテンツのキャッシュ
- **free** - メモリを必要とするどんな項目に対しても使用できるメモリ
- **inactive** - 使用されていないメモリ
- **pavailable** - 'total' に対する 'available' メモリのパーセンテージ ($\text{available}/\text{total} \times 100$ として計算)
- **pinned** - "wired" と同じ。
- **pusd** - "total" に対する "used" メモリのパーセンテージ ($\text{"used"}/\text{"total"} \times 100$ として計算)
- **shared** - 複数のプロセスによって同時にアクセスできるメモリ
- **slab** - カーネルが自身のデータ構造をキャッシュするために使用するメモリの総量
- **total** - 使用可能な物理メモリの合計
- **used** - プラットフォーム毎に異なる方法で計算された使用済みメモリ (下の表を参照)
- **wired** - 常に RAM にあると示されているメモリ。このメモリはディスクには移動されません。

Warning:

これらのパラメータの一部はプラットフォーム固有であり、ご使用のプラットフォームでは使用できない場合があります。詳細については、[プラットフォームでサポートされているアイテム](#)を参照してください。

available および **used** のプラットフォーム別計算方法:

プラットフォーム	"available"	"used"
AIX	free + cached	使用中の実メモリ
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+	/proc/meminfo の MemAvailable 値、MemAvailable	total - free
(RHEL7 では 3.10 にもバックポートされています)	の詳細については Linux カーネル ドキュメント を参照してください。 free + buffers + cached はすべてのページキャッシュを解放できず、計算に低ウォーターマークが使用されているため 'available' と等しくなくなりました。	
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:

`vm.memory.size[used]` と `vm.memory.size[available]` の合計は total と必ずしも同じではありません。たとえば FreeBSD の場合、

- * active、inactive、wired、cached のメモリは、いくつかの有用な情報を格納しているため、使用済みと見なされます。
- * 同時に、inactive、cached、free のメモリは、もっとメモリを必要とするプロセスにすぐに使用されるので、使用可能であるとみなされます。

このように、inactive メモリは used と同時に available でもあります。このため、`vm.memory.size[available]` はトリガーで使用されるために設計されていますが、`vm.memory.size[used]` アイテムは、情報提供の目的のみに設計されています。

以下も参照

1. 異なる OS でのメモリ計算に関する追加の詳細

3 パッシブおよびアクティブエージェントチェック

概要

このセクションでは、**Zabbix agent**によって実行されるパッシブおよびアクティブチェックの詳細について説明します。

Zabbix は、Zabbix エージェントとの通信に JSON ベースの通信プロトコルを使用します。

参照: **Zabbix agent 2** プロトコルの詳細。

パッシブチェック

パッシブチェックは単純なデータリクエストです。Zabbix サーバーまたはプロキシがデータ (CPU 負荷など) を要求し、Zabbix エージェントがデータをサーバーに送り返します。

サーバーリクエスト

ヘッダーとデータ長の定義については、[プロトコルの詳細](#)を参照してください。

<item key>

エージェントの応答

<DATA>[\0<ERROR>]

上記の角括弧内の部分はオプションであり、サポートされていないアイテムに対してのみ送信されます。

サポートされているアイテムの場合：

1. サーバーが TCP 接続を開きます
2. サーバーが **<HEADER><DATALEN>agent.ping** を送信します
3. エージェントはリクエストを読み取り、**<HEADER><DATALEN>1** で応答します
4. サーバーはデータを処理して値を取得します。今回の場合は"1"です。
5. TCP 接続が閉じられます

サポートされていないアイテムの場合：

1. サーバーが TCP 接続を開きます
2. サーバーは **<HEADER><DATALEN>vfs.fs.size[/nono]** を送信します
3. エージェントはリクエストを読み取り、次のレスポンスを返します。**<HEADER><DATALEN>ZBX_NOTSUPPORTED\0Cannot obtain filesystem information: [2] No such file or directory**
4. サーバーはデータを処理し、アイテムの状態をサポートされていない状態に変更し、指定されたエラーメッセージを表示します
5. TCP 接続が閉じられます

アクティブチェック

アクティブチェックには、より複雑な処理が必要です。エージェントは、最初にサーバーから独立した処理でアイテムのリストを取得します。

アクティブチェックを取得するサーバーは、エージェント構成ファイルの"ServerActive" パラメーターにリストされています。これらのチェックを要求する頻度は、同じ構成ファイルの"RefreshActiveChecks" パラメーターによって設定されます。ただしアクティブチェックの更新に失敗した場合は、ハードコードされた 60 秒後に再試行されます。

その後、エージェントは定期的に新しい値をサーバーに送信します。

Note:

エージェントがファイアウォールの背後にいる場合は、アクティブチェックのみを使用することを検討してください。そうすることで最初の着信接続を許可するためにファイアウォールを変更する必要がなくなります。

アイテムのリスト取得

エージェントリクエスト

アクティブチェック要求は、エージェントによって処理されるアクティブチェックを取得するために使用されます。この要求は、開始時にエージェントによって送信され、その後 RefreshActiveChecks 間隔で送信されます。

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
```

```

"hostinterface": "zabbix.server.lan",
"ip": "159.168.1.1",
"port": 12050
}

```

フィールド	タイプ	必須	値
request	string	yes	active checks
host	string	yes	ホスト名
host_metadata	string	no	設定パラメーター HostMetadata または HostMetadataItem メトリック値
hostinterface	string	no	設定パラメーター HostMetadata または HostMetadataItem メトリック値
ip	string	no	設定パラメーター ListenIP が設定されている場合は最初の IP
port	number	no	設定パラメーター ListenPort(設定されている場合) に設定されている値であり、デフォルトのエージェント・リスニング・ポートではありません。

サーバーレスポンス

アクティブチェック応答は、アクティブチェック要求を処理した後、サーバーによってエージェントに返されます。

```

{
  "response": "success",
  "data": [
    {
      "key": "log[/home/zabbix/logs/zabbix_agentd.log]",
      "key_orig": "log[/home/zabbix/logs/zabbix_agentd.log]",
      "itemid": 1234,
      "delay": "30s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "key_orig": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0
    }
  ]
}

```

フィールド	タイプ	必須	値	
response	string	yes	success failed	
info	string	no	失敗した場合のエラー情報	
data	array of objects	no	アクティブなチェック項目	
	key	string	no	マクロが展開された項目キー
	key_orig	string	no	展開されたマクロのない項目キー
	itemid	number	no	アイテム ID
	delay	string	no	アイテムの更新間隔
	lastlogsize	number	no	アイテムの最終ログサイズ
	mtime	number	no	アイテムの最終修正日時
refresh_unsupported	number	no	サポートされていないアイテムの更新間隔	
regex	array of objects	no	グローバル正規表現項目	
	name	string	no	グローバル正規表現名
	expression	string	no	グローバル正規表現
	expression_type	number	no	グローバル正規表現タイプ
	exp_delimiter	string	no	グローバル正規表現区切り文字
	case_sensitive	number	no	グローバル正規表現の大文字と小文字の区別設定

サーバーは成功で応答する必要があります。

例: 1. エージェントが TCP 接続を開きます 2. エージェントがチェックのリストを要求します 3. サーバーはアイテムのリスト (アイテムキー、遅延) で応答します 4. エージェントが応答を解析します 5. TCP 接続が閉じられます 6. エージェントが定期的なデータ収集を開始します

Attention:

アクティブチェックを使用する際、Zabbix サーバのトラッパーポートへアクセスは、(慎重に扱うべき) 設定データを利用できるようにしますので、注意してください。これができるのは、誰もがアクティブエージェントに成りすまし、アイテム設定データをリクエストできるためです。暗号化オプションを使用しない限り、認証は行われません。

収集データの送信

エージェント送信

エージェントデータリクエストには、収集されたアイテムの値が含まれます。

```
{
  "request": "agent data",
  "data": [
    {
      "host": "Zabbix server",
      "key": "agent.version",
      "value": "2.4.0",
      "clock": 1400675595,
      "ns": 76808644
    },
    {
      "host": "Zabbix server",
      "key": "log[/home/zabbix/logs/zabbix_agentd.log]",
      "lastlogsize": 112,
      "value": " 19845:20140621:141708.521 Starting Zabbix Agent [<hostname>]. Zabbix 2.4.0 (revision 5000)",
      "clock": 1400675595,
      "ns": 77053975
    }
  ],
  "session": "1234456akdsjhfoui"
}
```

フィールド	タイプ	必須	値
request	string	yes	agent data
session	string	yes	エージェントが開始されるたびに生成される固有のセッション ID
data	array of objects	yes	アイテム値
id	number	yes	値の識別子 (ネットワークの問題が発生した場合に、重複した値をチェックするために使用)
host	string	yes	ホスト名
key	string	yes	アイテムキー
value	string	no	アイテム値
lastlogsize	number	no	アイテムの最終ログサイズ
mtime	number	no	アイテム最終更新日時
state	number	no	アイテムステータス
source	string	no	値のイベントログソース。
eventid	number	no	値のイベントログ ID
severity	number	no	値のイベントログ深刻度
timestamp	number	no	値のイベントログタイムスタンプ
clock	number	yes	値のタイムスタンプ (エポックからの秒数)
ns	number	yes	値のタイムスタンプ (ナノ秒)

各値には仮想 ID が割り当てられます。値 ID は、1 つのデータセッション (セッショントークンで識別される) 内で一意の単純な昇順カウンターです。この ID は、接続性の低い環境で送信される可能性のある重複値を破棄するために使用されます。

サーバー応答

エージェントデータ応答は、エージェントデータ要求を処理した後、サーバーによってエージェントに送り返されます。

```
{
  "response": "success",
}
```



```
"info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

フィールド	タイプ	必須	値
response	string	yes	success failed
info	string	yes	アイテム処理結果

Attention:

サーバーで一部の値の送信が失敗した場合（たとえば、ホストまたはアイテムが無効化または削除されたため）、エージェントはそれらの値の送信を再試行しません。

例:

1. エージェントが TCP 接続を開きます。
2. エージェントが値のリストを送信します。
3. サーバがそのデータを処理し、ステータスを返します。
4. TCP 接続を閉じます。

上記の例では、`vfs.fs.size[/nono]` がサポートされていないステータスであることを、“state” の値が 1 および“value” プロパティのエラーメッセージによって指摘されています。

Attention:

エラーメッセージは、サーバー側で 2048 シンボルにトリミングされます。

古い XML プロトコル

Note:

Zabbix は、Base64 にエンコードされた XML のデータを 16MB まで取得しますが、デコードされる値は 1 つが 64KB 以下でなければなりません。そうでないと、デコード中にそれが 64KB に切り捨てられます。

4 トラッパーアイテム

概要

Zabbix サーバーは、JSON ベースの通信プロトコルで Zabbix sender から **トラッパーアイテム** を使用してデータを受信します。

リクエストメッセージとレスポンスメッセージは、**ヘッダーとデータの長さ**で始まる必要があります。

Zabbix sender のリクエスト

```
{
  "request": "sender data",
  "data": [
    {
      "host": "<hostname>",
      "key": "trap",
      "value": "test value"
    }
  ]
}
```

Zabbix server のレスポンス

```
{
  "response": "success",
  "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

タイムスタンプ付き Zabbix sender のリクエスト

Zabbix sender タイムスタンプとナノ秒を使用してリクエストを送信できます。

```

{
  "request": "sender data",
  "data": [
    {
      "host": "<hostname>",
      "key": "trap",
      "value": "test value",
      "clock": 1516710794,
      "ns": 592397170
    },
    {
      "host": "<hostname>",
      "key": "trap",
      "value": "test value",
      "clock": 1516710795,
      "ns": 192399456
    }
  ],
  "clock": 1516712029,
  "ns": 873386094
}

```

Zabbix server レスポンス

```

{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}

```

5 Windows エージェントアイテムの最小アクセス許可レベル

概要

エージェントを使用してシステムを監視する場合は、エージェントがインストールされているホストからメトリックを取得することをお勧めします。最小特権の原則を使用するには、エージェントから取得するメトリックを判別する必要があります。

このドキュメントの表では、Zabbix エージェントの正しい動作を保証するための最小権限を選択できます。

エージェントの実行ユーザーを「LocalSystem」ではなく別のユーザーにした場合、エージェントを Windows サービスとして実行するために対象ユーザーは「ローカルポリシー」→「ユーザー権限の割り当て」から「サービスとしてログオン」の権限と、Zabbix エージェントのログファイルを作成、書き込み、削除する権限を持っている必要があります。対象ユーザーを ActiveDirectory の PerformanceMonitorUsers グループに追加する必要があります。

Note:

「技術的に許容できる最小限の」グループに基づくエージェントの権利を使用する場合は、監視対象のオブジェクトに対する権利を事前に提供する必要があります。

Windows でサポートされている一般的なエージェントアイテム

アイテムキー	ユーザーグループ	
	推奨	技術的に許容できる最小値 (機能は制限されています)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests

アイテムキー	ユーザーグループ	
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Windows 固有のアイテムキー

アイテムキー	ユーザーグループ	
	推奨	技術的に許容できる最小値 (機能は制限されています)
eventlog	Event Log Readers	Guests
net.if.list	Guests	Guests
perf_counter	Performance Monitor Users	Performance Monitor Users
proc_info	Administrators	Guests
service.discovery	Guests	Guests
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Administrators	Guests
vm.vmemory.size	Guests	Guests

6 戻り値のエンコード

Zabbix サーバーは、UTF8 エンコーディングで返されるすべてのテキスト値を想定しています。これは、zabbix エージェント、ssh、telnet などのあらゆるタイプのチェックに関連しています。

監視対象のさまざまなシステム/デバイスおよびチェックで、値に非 ASCII 文字が返される場合があります。このような場合、ほとんどすべての zabbix キーには、追加のアイテムキーパラメータ - **<encoding>** が含まれています。このキーパラメータはオプション

ですが、戻り値が UTF8 エンコーディングではなく、非 ASCII 文字が含まれている場合は指定する必要があります。指定していないと結果が予期せぬ値になる可能性があります。

このような場合のさまざまなデータベースバックエンドでの動作の説明は次のとおりです。

MySQL

少なくとも MySQL バージョン 5.1.61 では、値に非 UTF8 エンコーディングの非 ASCII 文字が含まれている場合、この文字と次の文字は、データベースが値を格納するときに破棄されます。警告メッセージは `zabbix_server.log` に書き込まれません。

PostgreSQL

少なくとも PostgreSQL バージョン 9.1.3 では、値に非 UTF8 エンコーディングの非 ASCII 文字が含まれている場合、SQL クエリが失敗し (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) データは保存されません。適切な警告メッセージが `zabbix_server.log` に書き込まれます。

7 ラージファイルサポート

ラージファイルサポートとは、多くの場合 LFS と略され、32 ビットオペレーティングシステムで 2GB を超えるファイルを処理する機能に適用される用語です。Zabbix 2.0 以降ラージファイルサポートが追加されました。この変更は、少なくとも **ログファイルの監視** とすべての `vfs.file.*items` に影響します。ラージファイルサポートは、Zabbix コンパイル時のシステムの機能に依存しますが、`procfs` および `swapctl` との互換性がないため、32 ビット Solaris では完全に無効になっています。

8 センサー

各センサーチップは、`sysfs / sys/devices` ツリーで独自のディレクトリを作成します。すべてのセンサーチップを見つけるには `/sys/class/hwmon/hwmon*` からデバイスのシンボリックリンクをたどる方が簡単です。ここでの * は実数 (0,1,2...) です。

センサーの読み取り値は、仮想デバイスの場合は `/sys/class/hwmon/hwmon*/ディレクトリ` に、非仮想デバイスの場合は `/sys/class/hwmon/hwmon*/device` ディレクトリにあります。hwmon* または hwmon*/device ディレクトリ内にある name という名前のファイルには、センサーチップが使用するカーネルドライバーの名前に対応するチップの名前が含まれています。

センサー読み取り値はファイルごとに 1 つだけです。上記のディレクトリ内のセンサー読み取り値を含むファイルに名前を付けるための一般的なスキームは次のとおりです。 `<type><number>_<item>`, where

-タイプ-センサーチップの場合、「in」(電圧)、「temp」(温度)、「fan」(ファン) などです。-項目-「入力」(測定値)、「最大」(高しきい値)、「最小」(低しきい値) など、-数値-複数回存在する可能性のある要素に常に使用されます (0 から始まる電圧を除いて、通常は 1 から始まります)。ファイルが特定の要素を参照していない場合、ファイルには番号のない単純な名前が付けられます。- **type** - センサーチップの場合「in」(電圧)、「temp」(温度)、「fan」(ファン) など - **item** - "input" (測定値)、「max」(高しきい値)、「min」(低しきい値) など - **number** - 複数存在する可能性のある要素に常に使用されます。(0 から始まる電圧を除いて、通常は 1 から始まります) ファイルが特定の要素を参照していない場合、ファイルには番号のない単純な名前が付けられます。

ホストで利用可能なセンサーに関する情報は、**Sensor-detect** および **sensors** ツール (lm-sensors パッケージ : <http://lm-sensors.org/>) を使用して取得できます。** Sensors-detect は、使用可能なセンサーに必要なモジュールを判別するのに役立ちます。モジュールがロードされると、センサー ** プログラムを使用してすべてのセンサーチップの読み取り値を表示できます。このプログラムで使用されるセンサー読み取り値のラベル付けは、一般的な命名スキーム (`<type><number>_<item>`) とは異なる場合があります。

- `<type><number>_label` というファイルがある場合、このファイル内のラベルが `<type><number><item>name` の代わりに使用されます。
- `<type><number>_label` ファイルがない場合、プログラムは `/etc/sensors.conf` 内 (`/etc/sensors3.conf` 等、別の名前へ置換も可能) を検索します。

このラベル付けにより、ユーザーは使用されているハードウェアの種類を判別できます。構成ファイル内に `<type><number>_label` ファイルも label もない場合、ハードウェアのタイプは name 属性 (`hwmon*/device/name`) によって判別します。zabbix_agent が受け入れるセンサーの実際の名前は、-u パラメーターを指定して **sensors** プログラムを実行 (**sensors -u**) することで取得できます。

sensors プログラムでは、使用可能なセンサーはバスタイプ (ISA アダプター、PCI アダプター、SPI アダプター、仮想デバイス、ACPI インターフェース、HID アダプター) によって分けられています。

Linux 2.4 の場合 :

(センサーの読み取り値は `/proc/sys/dev/sensors` ディレクトリから取得されます)

- **device** - デバイス名 (<mode> が使用されている場合は正規表現です)
- **sensor** - センサー名 (<mode> が使用されている場合は正規表現です)
- **mode** - 使用可能な値 : avg,max,min (このパラメーターが省略されている場合、デバイスとセンサーは逐語的に扱われます)

キーの例 : `sensor[w83781d-i2c-0-2d,temp1]`

Zabbix 1.8.4 以前は `sensor [temp1]` 形式が使用されていました。

Linux 2.6 以降の場合：

(センサーの読み取り値は/sys/class/hwmon ディレクトリから取得されます)

- **device** - デバイス名 (非正規表現) デバイス名はデバイス実際の名前 (0000:00:18.3 など) またはセンサープログラムを使用して取得した名前 (k8temp-pci-00c3 など) の場合があります。使用する名前はユーザーが選択してください。
- **sensor** - センサー名 (非正規表現)
- **mode** - 使用可能な値: avg, max, min (このパラメーターを省略した場合、デバイスとセンサーは逐語的に扱われます)。

キーの例：

```
sensor[k8temp-pci-00c3,temp,max] or sensor[0000:00:18.3,temp1]
```

```
sensor[smc47b397-isa-0880,in,avg] or sensor[smc47b397.2176,in1]
```

センサー名の取得

ラベルの名前はセンサーチップベンダーごとに異なる可能性があるため、sensors コマンドで出力されるセンサーラベルは常に使用できるとは限りません。たとえば sensors の出力には次のような行が含まれる場合があります。

```
$ sensors
in0:          +2.24 V (min = +0.00 V, max = +3.32 V)
Vcore:        +1.15 V (min = +0.00 V, max = +2.99 V)
+3.3V:        +3.30 V (min = +2.97 V, max = +3.63 V)
+12V:         +13.00 V (min = +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C (low = -127.0°C, high = +127.0°C)
```

これらのうち直接使用できるラベルは 1 つだけです。

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

他のラベル (Vcore や +12V など) を使用しようとしても機能しません。

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Zabbix がセンサーの読み取り値を取得するために使用できる実際のセンサー名を確認するには sensors -u を実行します。出力では次のことが確認できます。

```
$ sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

上記から Vcore は in1 として、* +12V は in4* として照会する必要があります。⁵

```
$ zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix は電圧 (in) だけでなく電流 (curr) 温度 (temp) ファン速度 (fan) の読み取り値も取得できます。

[^ 1]: 仕様によると、これらはチップピンの電圧であり、スケールリングが必要な場合があります。

9 proc.mem アイテムの memtype パラメーターに関する注記

概要

memtype パラメーターは、Linux,AIX,FreeBSD, および Solaris でサポートされています。

⁵HttpOnly が true の場合、Cookie は HTTP プロトコルを介してのみアクセス可能になります。これは、JavaScript などのスクリプト言語から Cookie にアクセスできないことを意味します。この設定は、XSS 攻撃による個人情報の盗難を効果的に減らすのに役立ちます (ただし、すべてのブラウザでサポートされているわけではありません)

'memtype' の3つの一般的な値である pmem、rss、vsize は、これらすべてのプラットフォームでサポートされています。さらに一部のプラットフォームでは固有の'memtype' 値がサポートされています。

AIX

以下の表が AIX の'memtype' パラメーターでサポートされている値です。

サポート値	説明	procentry64 構造体のソース	互換
vsize ¹	仮想メモリサイズ	pi_size	
pmem	実メモリのパーセンテージ	pi_prm	ps -o pmem
rss	物理メモリ消費量	pi_trss + pi_drss	ps -o rssize
size	プロセスのサイズ (コード + データ)	pi_dvm	"ps gvW" SIZE column
dsize	データサイズ	pi_dsize	
tsize	テキスト (コード) サイズ	pi_tsize	"ps gvW" TSIZ column
sdsiz	シェアードライブラリのデータサイズ	pi_sdsiz	
drss	データの物理メモリ消費量	pi_drss	
trss	テキストの物理メモリ消費量	pi_trss	

Notes for AIX:

1. When choosing parameters for proc.mem[] item key on AIX, try to specify narrow process selection criteria. Otherwise there is a risk of getting unwanted processes counted into proc.mem[] result.

Example:

```
\$ zabbix_agentd -t proc.mem[, , NonExistingProcess, rss]
proc.mem[, , NonExistingProcess, rss] [u|2879488]
```

This example shows how specifying only command line (regular expression to match) parameter results in Zabbix agent self-accounting - probably not what you want.

2. Do not use "ps -ef" to browse processes - it shows only non-kernel processes. Use "ps -Af" to see all processes which will be seen by Zabbix agent.
3. Let's go through example of 'topasrec' how Zabbix agent proc.mem[] selects processes.

```
\$ ps -Af | grep topasrec
root 10747984      1  0   Mar 16      -   0:00 /usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf daily
```

proc.mem[] has arguments:

```
proc.mem[<name>, <user>, <mode>, <cmdline>, <memtype>]
```

The 1st criterion is a process name (argument <name>). In our example Zabbix agent will see it as 'topasrec'. In order to match, you need to either specify 'topasrec' or to leave it empty. The 2nd criterion is a user name (argument <user>). To match, you need to either specify 'root' or to leave it empty. The 3rd criterion used in process selection is an argument <cmdline>. Zabbix agent will see its value as '/usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -ystart_time=04:08:54,Mar16,2023'. To match, you need to either specify a regular expression which matches this string or to leave it empty.

Arguments <mode> and <memtype> are applied after using the three criteria mentioned above.

FreeBSD

以下の表が FreeBSD の'memtype' パラメーターでサポートされている値です。

サポート値	説明	kinfo_proc 構造体のソース	互換
vsize	仮想メモリサイズ	kp_eproc.e_vm.vm_map.size or ki_size	ps -o vsz
pmem	実メモリのパーセンテージ	calculated from rss	ps -o pmem
rss	物理メモリ消費量	kp_eproc.e_vm.vm_rssize or ki_rssize	ps -o rss
size ¹	プロセスのサイズ (コード + データ + スタック)	tsize + dsize + ssize	
tsize	テキスト (コード) サイズ	kp_eproc.e_vm.vm_tsize or ki_tsize	ps -o tsiz
dsize	データサイズ	kp_eproc.e_vm.vm_dsize or ki_dsize	ps -o dsiz
ssize	スタックサイズ	kp_eproc.e_vm.vm_ssize or ki_ssize	ps -o ssiz

Linux

以下の表が Linux の'memtype' パラメーターでサポートされている値です。

サポート値	説明	/proc/<pid>/status ファイルのソース
vsiz ¹	仮想メモリサイズ	VmSize
pmem	実メモリのパーセンテージ	(VmRSS/total_memory) * 100
rss	物理メモリ消費量	VmRSS
data	データセグメントサイズ	VmData
exe	コードセグメントサイズ	VmExe
hwm	物理メモリ消費のピーク	VmHWM
lck	ロックメモリのサイズ	VmLck
lib	シェアードライブラリのサイズ	VmLib
peak	仮想メモリサイズのピーク	VmPeak
pin	固定されたページのサイズ	VmPin
pte	ページテーブルエントリのサイズ	VmPTE
size	プロセスコードとデータとスタックセグメントの合計値	VmExe + VmData + VmStk
stk	スタックセグメントのサイズ	VmStk
swap	スワップスペース使用量	VmSwap

Linux に関する注意：

- 古い Linux カーネルではサポートされていない memtype 値があります。たとえば Linux 2.4 カーネルは、hwm、pin、peak、pte、および swap の値をサポートしていません。
- proc.mem[...,...,...,data] を使用した Zabbix エージェントのアクティブチェックプロセス自己監視は、エージェントの/proc/<pid>/status ファイルの VmData 行で報告される値よりも 4kB 大きい値が示されていることがわかりました。自己測定時にエージェントのデータセグメントは 4 kB 増加し、その後前のサイズに戻ります。

Solaris

以下の表が Solaris の'memtype' パラメーターでサポートされている値です。

サポート値	説明	psinfo 構造のソース	互換
vsiz ¹	プロセスイメージのサイズ	pr_size	ps -o vsz
pmem	実メモリのパーセンテージ	pr_pctmem	ps -o pmem
rss	物理メモリ消費量 過小評価されている可能性があります。"man ps" の rss の説明を参照してください	pr_rssize	ps -o rss

脚注

¹ デフォルト値

10 proc.mem および proc.num アイテムでのプロセスの選択に関する注記

コマンドラインを変更するプロセス

一部のプログラムでは、現在のアクティビティを表示する方法としてコマンドラインの変更を使用しています。ユーザーは、ps および top コマンドを実行することでアクティビティを確認できます。このようなプログラムとしては PostgreSQL, Sendmail, Zabbix が含まれます。

Linux での例を見てみましょう。いくつかの Zabbix エージェントプロセスを監視したいと思います。

ps コマンドは、対象のプロセスを次のように表示します

```
$ ps -fu zabbix
UID          PID  PPID  C  STIME TTY          TIME CMD
...
zabbix      6318     1   0  12:01 ?           00:00:00 sbin/zabbix_agendd -c /home/zabbix/ZBXNEXT-1078/zabbix_agendd
zabbix      6319   6318   0  12:01 ?           00:00:01 sbin/zabbix_agendd: collector [idle 1 sec]
zabbix      6320   6318   0  12:01 ?           00:00:00 sbin/zabbix_agendd: listener #1 [waiting for connection]
zabbix      6321   6318   0  12:01 ?           00:00:00 sbin/zabbix_agendd: listener #2 [waiting for connection]
zabbix      6322   6318   0  12:01 ?           00:00:00 sbin/zabbix_agendd: listener #3 [waiting for connection]
zabbix      6323   6318   0  12:01 ?           00:00:00 sbin/zabbix_agendd: active checks #1 [idle 1 sec]
...
```

名前とユーザーでプロセスを選択すると次のようになります。

次のプロセスをチェックするとき、エージェントは cmdline ファイルから zabbix_agentd_30: collector [idle 1 sec] を取得しようとしますが、name パラメータが zabbix_agentd_30 を満たすことができません。したがって、コマンドラインを変更しないメインプロセスのみがカウントされます。他のエージェントプロセスはコマンドラインを変更し、無視されます。

このことから今回の例では、プロセスを選択するために proc.mem[] と proc.num[] で name パラメータを使用できないことを示しています。

適切な正規表現で cmdline パラメータを使用すると、正しい結果が得られます。

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

コマンドラインを変更するプログラムを監視するために proc.mem[] と proc.num[] の項目を使用するときは注意してください。

name および cmdline パラメータを proc.mem[] および proc.num[] アイテムに入れる前に、proc.num[] アイテムおよび ps コマンドを使用してパラメータをテストすることをお勧めします。

Linux カーネルスレッド

proc.mem[] および proc.num[] 項目の cmdline パラメータでスレッドを選択することはできません

例としてカーネルスレッドの1つを上げましょう。

```
$ ps -ef | grep kthreadd
root      2      0  0 09:33 ?          00:00:00 [kthreadd]
```

プロセス name パラメータで選択できます。

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

しかしプロセス cmdline パラメータによる選択は機能しません。

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

その理由は、Zabbix エージェントが cmdline パラメータで指定された正規表現を受け取り、それをプロセス/proc/<pid>/cmdline のコンテンツに適用するためです。カーネルスレッドの場合、それら/proc/<pid>/cmdline ファイルは空です。よって cmdline パラメータが一致することはありません。

proc.mem[] および proc.num[] アイテムのスレッドカウント

Linux カーネルスレッドは proc.num[] アイテムによってカウントされますが、proc.mem[] アイテムのメモリは報告されません。

例:

```
$ ps -ef | grep kthreadd
root      2      0  0 09:51 ?          00:00:00 [kthreadd]
```

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1
```

```
$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

しかし、カーネルスレッドと同じ名前のユーザープロセスがある場合は次のようになります。

```
$ ps -ef | grep kthreadd
root      2      0  0 09:51 ?          00:00:00 [kthreadd]
zabbix   9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd
```

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2
```

```
$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

proc.num [] はカーネルスレッドとユーザープロセスの両方をカウントしました。proc.mem [] はユーザープロセスのメモリのみを報告し、カーネルスレッドメモリを0であるかのようにカウントします。これは ZBX_NOTSUPPORTED が報告された上記の場合とは異なります。

プログラム名がスレッドの1つと一致する場合は、proc.mem[] および proc.num[] アイテムを使用するときに注意してください。

パラメータを `proc.mem[]` および `proc.num[]` アイテムに入れる前に、`proc.num[]` アイテムと `ps` コマンドを使用してパラメータをテストすることをお勧めします。

11 net.tcp.service および net.udp.service チェックの実装の詳細

このページでは `net.tcp.service` および `net.udp.service` チェック実装における、`service` パラメーターで指定されたさまざまなサービスについて詳しく説明しています。

`net.tcp.service` アイテムのパラメータ

ftp

TCP 接続を作成し、応答の最初の 4 文字が "220" であることを検証して "QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 21 が使用されます。

http

特に何もせず TCP 接続を作成します。ポートが指定されていない場合はデフォルトのポート 80 が使用されます。

https

`libcurl` を使用し（そしてそれとともにのみ機能し）、証明書の信頼性を検証せず、SSL 証明書のホスト名を検証せず、応答ヘッダー (HEAD 要求) のみを読み出します。ポートが指定されていない場合はデフォルトポートの 443 が使用されます。

imap

TCP 接続を作成し、応答の最初の 4 文字が "* OK" であることを検証して "a1 LOGOUT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 143 が使用されます。

ldap

LDAP サーバーへの接続を開き、フィルターを (`objectClass=*`) に設定して LDAP 検索操作を実行して、最初のエントリの最初の属性が正常に取得できることを検証します。ポートが指定されていない場合はデフォルトのポート 389 が使用されます。

nntp

TCP 接続を作成し、応答の最初の 3 文字が "200" または "201" であることを検証して "QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 119 が使用されます。

pop

TCP 接続を作成し、応答の最初の 3 文字が "+OK" であることを検証して "QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 110 が使用されます。

smtp

TCP 接続を作成し、応答の最初の 3 文字が "220" であり、その後スペース、行末、またはダッシュが続くことを検証します。ダッシュを含む行は複数行の応答で、ダッシュのない行が受信されるまで応答が再読み取りされます。次に "QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 25 が使用されます。

ssh

TCP 接続を作成します。接続が確立されている場合、両側で識別文字列 (SSH-major.minor-XXXX) が交換されます。ここでのメジャーとマイナーはプロトコルバージョンであり、XXXX は文字列です。Zabbix は仕様に一致する文字列が見つかったかどうかを確認し、不一致の場合は文字列 "SSH-major.minor-zabbix_agent\r\n" または "0\r\n" を送り返します。ポートが指定されていない場合は、デフォルトのポート 22 が使用されます。

tcp

特に何もせず TCP 接続を作成します。他のチェックとは異なり、ポート指定必須です。

telnet

TCP 接続を作成し、ログインプロンプト (の最後に ':') が返ってくることを検証します。ポートが指定されていない場合はデフォルトのポート 23 が使用されます。

`net.udp.service` アイテムのパラメータ

nntp

UDP を介して SNMP パケットを送信し、RFC 4330、セクション 5 に従って応答を検証します。ポートが指定されていない場合は、デフォルトのポート 123 が使用されます。

12 到達不能または使用不可のホストインターフェイス設定

概要

いくつかの設定パラメータは、エージェントチェック (Zabbix、SNMP、IPMI、JMX) が失敗し、ホストインターフェイスに到達できなくなった場合の Zabbix サーバーの動作を定義します。

到達不能なインターフェイス

Zabbix、SNMP、IPMI、または JMX エージェントによるチェックが失敗 (ネットワークエラー、タイムアウト) すると、その後、当該のホストインターフェイスは到達不能として扱われます。Zabbix エージェントのアクティブチェックは、インターフェイスの可用性にまったく影響を与えないことに注意してください。

その瞬間から **UnreachableDelay** は、この到達不能な状態のアイテムの 1 つ (LLD ルールを含む) を使用してインターフェイスが再チェックされる頻度を定義します。再チェックは到達不能なポーラー (または IPMI チェックの場合は IPMI ポーラー) によって実行されます。デフォルトでは次のチェックの 15 秒前です。

Zabbix サーバーでは、到達できないことを次のようなメッセージでログに書き込みます。

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, ait f
```

失敗した正確なアイテムとアイテムタイプ (Zabbix エージェント) が示されていることに注意してください。

Note:

Timeout パラメーターは、到達不能時にインターフェイスが再チェックされるタイミングにも影響します。Timeout が 20 秒、UnreachableDelay が 30 秒の場合、次のチェックは最初の試行から 50 秒後になります。

UnreachablePeriod パラメーターは、到達不能期間の合計の長さを定義します。デフォルトでは、UnreachablePeriod は 45 秒です。UnreachablePeriod は、UnreachableDelay の数倍の大きさである必要があります。これにより、インターフェイスが使用できなくなる前に、インターフェイスが複数回再チェックされます。

インターフェイスを到達可能に戻す

到達不能期間が終了すると、インターフェイスが再度ポーリングされ、インターフェイスを到達不能状態にしたアイテムの優先度が低下します。到達不能なインターフェイスが再び到達可能になると、監視は自動的に通常に戻ります。

```
resuming Zabbix agent checks on host "New host": connection restored
```

Note:

インターフェイスが使用可能になっても、ホストは次の 2 つの理由ですべてのアイテムをすぐにポーリングしません。

- ホストが過負荷になる可能性があります。
- インターフェイスの復元時間は、計画されたアイテムのポーリングスケジュール時間と常に一致するとは限りません。

そのため、インターフェイスが使用可能になった後、アイテムはすぐにはポーリングされず、次のポーリングラウンドに再スケジュールされます。

利用できないインターフェイス

UnreachablePeriod の終了までにインターフェイスが到達可能に戻らなかった場合、インターフェイスは使用不可として扱われます。

サーバーログには次のようなメッセージが出力されます。

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

フロントエンドでは、ホストの可用性アイコンが緑/灰色から黄色/赤に変わります (使用不能なインターフェイスの詳細は、ホストの可用性アイコンにオンマウスしたときに表示されるヒントボックスに表示されます。)

Interface	Status	Error
127.0.0.1:10050	Available	
192.0.0.1:10050	Not available	Get value from agent failed: cannot connect to [[192.0.0.1]:10050]: [4] system call

UnavailableDelay パラメーターは、インターフェースが使用不可のときにインターフェースがチェックされる頻度を定義します。

デフォルトでは 60 秒です（したがって、この場合、上記のログメッセージから「一時的に無効にする」とは、1 分間チェックを無効にすることを意味します）。

インターフェースへの接続が復元されると、監視も自動的に通常に戻ります。

```
enabling Zabbix agent checks on host "New host": interface became available
```

13 Zabbix 統計のリモートモニタリング

概要

Zabbix サーバーとプロキシの一部の内部メトリックを別の Zabbix インスタンスまたはサードパーティツールからリモートでアクセスできるようにすることができます。これはサポーター/サービスプロバイダーがクライアントの Zabbix サーバー/プロキシをリモートで監視できるようにする、または Zabbix がメインの監視ツールではない組織で、Zabbix の内部メトリックを包括的な監視用セットアップでサードパーティのシステムによって監視できるようにするために役立ちます。

Zabbix の内部統計は、新しい 'StatsAllowedIP' **server/proxy** パラメーターにリストされた構成可能なアドレスのセットに公開されます。リクエストはこれらのアドレスからのみ受け付けられます。

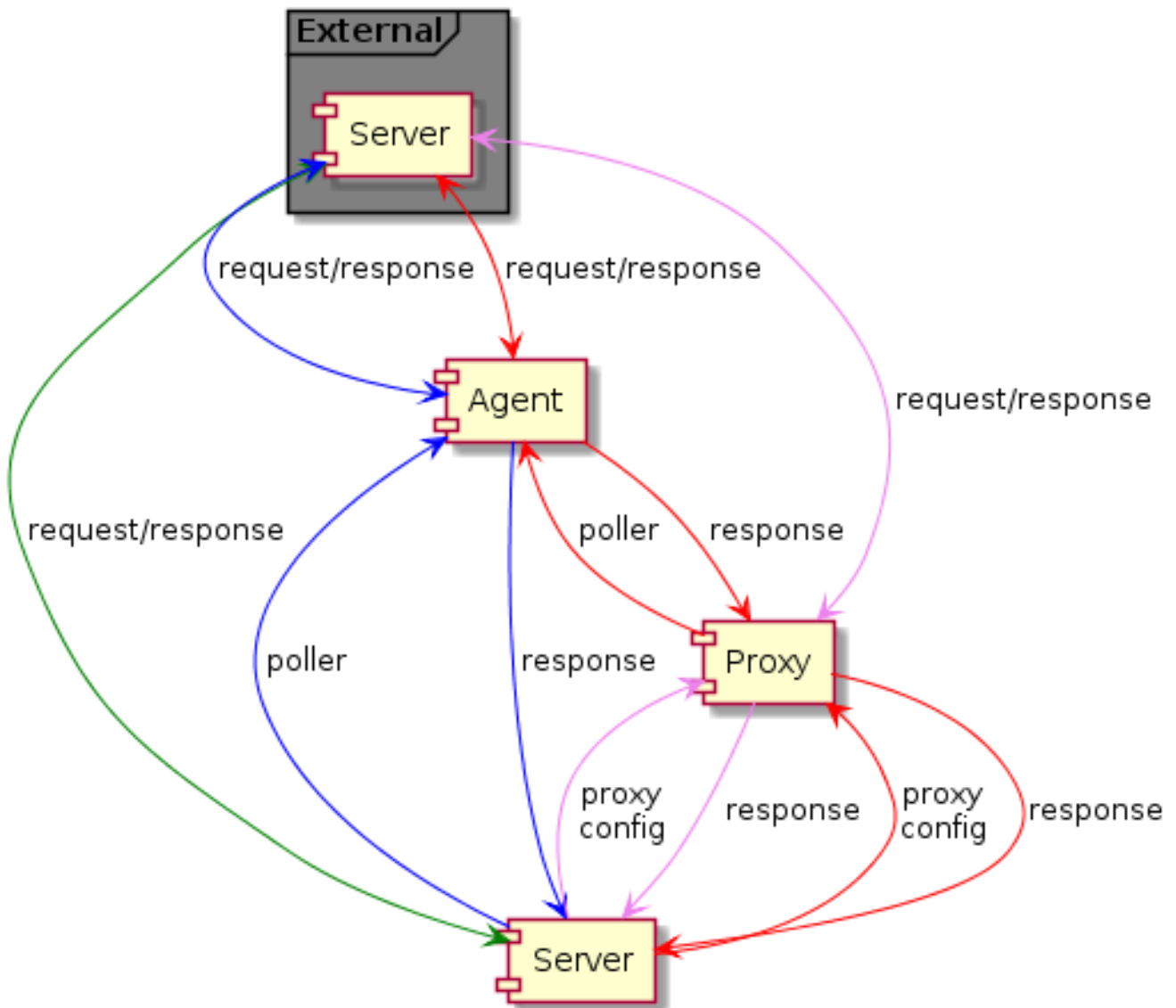
アイテム

別の Zabbix インスタンスで内部統計のクエリを設定するには、次の 2 つの項目を使用します。

- `zabbix[stats,<ip>,<port>]` internal item - Zabbix サーバー/プロキシの直接リモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用します
- `zabbix.stats[<ip>,<port>]` agent item - Zabbix サーバー/プロキシのエージェントベースのリモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用されます。

参考情報: [内部項目](#), [Zabbix エージェントアイテム](#)

次の図はコンテキストに応じた、いずれかのアイテムの使用法を示しています。



- ■ - Server → external Zabbix instance (zabbix[stats,<ip>,<port>])
- ■ - Server → proxy → external Zabbix instance (zabbix[stats,<ip>,<port>])
- ■ - Server → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])
- ■ - Server → proxy → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])

ターゲットインスタンスが外部インスタンスによるクエリを許可するには、ターゲットインスタンスの'StatsAllowedIP'パラメータに外部インスタンスのアドレスをリストします。

公開メトリック

統計アイテムは統計をまとめて収集し、JSONを返します。これは依存アイテムがデータを取得するための基礎となります。次の**内部メトリック**は、2つのアイテムのいずれかによって返されます。

- zabbix[boottime]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[preprocessing_queue] (server only)
- zabbix[process,<type>,<mode>,<state>] (only process type based statistics)
- zabbix[r-cache,<cache>,<mode>]
- zabbix[requiredperformance]
- zabbix[triggers] (server only)
- zabbix[uptime]
- zabbix[v-cache,buffer,<mode>] (server only)
- zabbix[v-cache,cache,<parameter>]

- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] ('trends' cache type server only)

テンプレート

テンプレートは、Zabbix サーバーの **リモートモニタリング** または外部インスタンスからのプロキシ内部メトリックに使用できます。

- Remote Zabbix server
- Remote Zabbix proxy

複数の外部インスタンスのリモート監視にテンプレートを使用するには、外部インスタンスの監視ごとに個別のホストが必要であることに注意してください。

トラッパープロセス

外部 Zabbix インスタンスからの内部メトリックリクエストの受信は、リクエストを検証し、メトリックを収集し、JSON データバッファを作成し、準備された JSON をサーバーなどから送り返すトラッパープロセスによって処理されます。

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
      ...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "requiredperformance": N,
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,

```


このセクションでは、ユーザー 'zabbix' で 'www.example.com' の Web 監視を実行する Zabbix サーバーで、Kerberos を構成する例について説明します。

手順

Step 1

Kerberos パッケージをインストールします。

Debian/Ubuntu の場合:

```
apt install krb5-user
```

RHEL の場合:

```
dnf install krb5-workstation
```

Step 2

Kerberos 設定ファイルを設定します (詳細については、MIT のドキュメントを参照してください)。

```
cat /etc/krb5.conf
[libdefaults]
    default_realm = EXAMPLE.COM

#### The following krb5.conf variables are only for MIT Kerberos.
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true

[realms]
    EXAMPLE.COM = {
    }

[domain_realm]
    .example.com=EXAMPLE.COM
    example.com=EXAMPLE.COM
```

Step 3

ユーザー zabbix の Kerberos チケットを作成します。ユーザー zabbix として次のコマンドを実行します。

```
kinit zabbix
```

Attention:

上記のコマンドをユーザー zabbix として実行することが重要です。root として実行すると認証は機能しません。

Step 4

Kerberos 認証タイプを使用して Web シナリオまたは HTTP エージェントアイテムを作成します。

次の curl コマンドおよびオプションを使用してテストできます。

```
curl -v --negotiate -u : http://example.com
```

長時間の Web 監視では、Kerberos チケットの更新に注意してください。チケットの有効期限のデフォルト時間は 10 時間です。

15 modbus.get パラメーター

概要

以下の表に modbus.get[] item パラメーターの詳細を示します。

パラメータ

パラメータ	説明	デフォルト	例
endpoint	<p>protocol://connection_stringとして定義されたエンドポイントのプロトコルとアドレス</p> <p>可能なプロトコル値 : rtu,ascii (エージェント 2 のみ) * tcp

 接続文字列形式 :

with tcp* - address : port シリアルライン付き : rtu,ascii - port_name: speed: params where 'speed' - 1200, 9600 など 'params' - データビット (5、6、7、または 8)、パリティ (n、e または o、なし/偶数/奇数)、ストップビット (1 または 2)</p>		<p>protocol: none</p> <p>rtu/ascii protocol: port_name: none speed: 115200 params: 8n1</p> <p>tcp protocol: address: none port: 502</p>
slave id	<p>対象となるデバイスの Modbus アドレス (1~247) MODBUS メッセージング実装ガイド (23 ページ)</p> <p>tcp デバイス (GW ではありません) フィールドを無視します</p>	<p>serial:1</p> <p>tcp: 255 (0xFF)</p>	2
function	<p>サポートされている関数の値または空白 :</p> <p>1 - コイルの読み取り 2 - ディスクリート入力の読み取り 3 - 保持レジスタの読み取り 4 - 入力レジスタの読み取り</p>	empty	3
address	<p>最初のレジストリ、コイル、または入力アドレス</p> <p>'function' が空の場合、'address' は次の範囲内である必要があります。 Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999</p> <p>'function' が空でない場合、'address' フィールドは 0 から 65535 までで、そのまま使用されます (PDU)</p>	<p>function が空白: 00001</p> <p>function が空白以外: 0</p>	9999
count	<p>デバイスから読み取られるシーケンスされた'type'の数。</p> <p>コイルまたはディスクリート入力の場合は'type'=1bit その他の場合 : (count * type) / 2 = real 読み取るレジスタの数 'offset' が 0 でない場合、値は'実際の数'に追加されます。 '実際の数'の許容範囲は 1:65535 です。</p>	1	2
type	<p>データ型 :</p> <p>読み取りコイルおよび読み取りディスクリート入力の場合 - bit</p> <p>読み取り保持レジスタおよび読み取り入力レジスタの場合 :</p> <p>int8 - 8bit uint8 - 8bit (符号なし) int16 - 16bit uint16 - 16bit (符号なし) int32 - 32bit uint32 - 32bit (符号なし) float - 32bit uint64 - 64bit (符号なし) double - 64bit</p>	<p>bit</p> <p>uint16</p>	uint64

パラメータ	説明	デフォルト	例
endianness	<p>エンディアンタイプ： be - ビッグエンディアン le - リトルエンディアン mbe - ミッドビッグエンディアン mle - ミッドリトルエンディアン</p> <p>制限： 1 ビットの場合 - be 8 ビットの場合 - be,le 16 ビットの場合 - be,le</p>	be	le
offset	<p>'address' から始まるレジスタの数。その結果は破棄されます。</p> <p>各レジスタのサイズは 16 ビットです (ランダム読み取りアクセスをサポートしない機器をサポートするため)</p>	0	4

16 VMware のカスタムパフォーマンスカウンター名作成

概要

VMware パフォーマンスカウンターパスのフォーマットは `group/counter[rollup]` です。

- group - cpu などのパフォーマンスカウンターグループ
- counter - usagemhz などのパフォーマンスカウンター名
- rollup - average などのパフォーマンスカウンターロールアップの種類

上記の例では次のカウンターパスが得られます: `cpu/usagemhz[average]`

パフォーマンスカウンターグループの説明、カウンター名、およびロールアップの種類については、[VMware のドキュメント](#)を参照してください。

Zabbix のスクリプトアイテムを使用して、内部名を取得し、カスタムパフォーマンスカウンター名を作成することもできます。

設定

1. 次のパラメーターを使用して、メインの VMware ホスト (`eventlog[]` アイテムが存在する場所) で無効化されたスクリプトアイテムを作成します。

Item Tags Preprocessing

* Name

Type

* Key

Type of information

Parameters

Name	Value
<input type="text"/>	<input type="text"/>

[Add](#)

* Script

* Timeout

* Update interval

Custom intervals

Type	Interval	Period
<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24"/>

[Add](#)

* History storage period

Populates host inventory field

Description

Enabled

- 名前: VMware metrics
- タイプ: スクリプト
- キー: vmware.metrics
- データ型: テキスト
- スクリプト: 以下のスクリプト をコピーして貼り付けます
- タイムアウト: 10
- ヒストリの保存期間: ヒストリを保持しない
- 有効: チェックしない

スクリプト

```
try {
  Zabbix.log(4, 'vmware metrics script');

  var result, resp,
  req = new HttpRequest();
  req.addHeader('Content-Type: application/xml');
```

```

req.addHeader('SOAPAction: "urn:vim25/6.0"');

login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vim25/6.0">
<soapenv:Header/>
<soapenv:Body>
  <urn:Login>
    <urn:_this type="SessionManager">SessionManager</urn:_this>
    <urn:userName>{$VMWARE.USERNAME}</urn:userName>
    <urn:password>{$VMWARE.PASSWORD}</urn:password>
  </urn:Login>
</soapenv:Body>
</soapenv:Envelope>'
resp = req.post("${VMWARE.URL}", login);
if (req.getStatus() != 200) {
  throw 'Response code: '+req.getStatus();
}

query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vim25/6.0">
<soapenv:Header/>
<soapenv:Body>
  <urn:RetrieveProperties>
    <urn:_this type="PropertyCollector">propertyCollector</urn:_this>
    <urn:specSet>
      <urn:propSet>
        <urn:type>PerformanceManager</urn:type>
        <urn:pathSet>perfCounter</urn:pathSet>
      </urn:propSet>
      <urn:objectSet>
        <urn:obj type="PerformanceManager">PerfMgr</urn:obj>
      </urn:objectSet>
    </urn:specSet>
  </urn:RetrieveProperties>
</soapenv:Body>
</soapenv:Envelope>'
resp = req.post("${VMWARE.URL}", query);
if (req.getStatus() != 200) {
  throw 'Response code: '+req.getStatus();
}
Zabbix.log(4, 'vmware metrics=' + resp);
result = resp;

logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vim25/6.0">
<soapenv:Header/>
<soapenv:Body>
  <urn:Logout>
    <urn:_this type="SessionManager">SessionManager</urn:_this>
  </urn:Logout>
</soapenv:Body>
</soapenv:Envelope>'

resp = req.post("${VMWARE.URL}",logout);
if (req.getStatus() != 200) {
  throw 'Response code: '+req.getStatus();
}

} catch (error) {
  Zabbix.log(4, 'vmware call failed : '+error);
  result = {};
}

return result;

```

項目を設定後、テストボタンを押して、値の取得を押します。

Get value from host

Host address

Port

Proxy (no proxy)

Get value

受信した XML を任意の XML フォーマッターにコピーし、目的のメトリックを見つけます。

1 つのメトリックの XML の例:

```
<PerfCounterInfo xsi:type="PerfCounterInfo">
  <key>6</key>
  <nameInfo>
    <label>Usage in MHz</label>
    <summary>CPU usage in megahertz during the interval</summary>
    <key>usagemhz</key>
  </nameInfo>
  <groupInfo>
    <label>CPU</label>
    <summary>CPU</summary>
    <key>cpu</key>
  </groupInfo>
  <unitInfo>
    <label>MHz</label>
    <summary>Megahertz</summary>
    <key>megaHertz</key>
  </unitInfo>
  <rollupType>average</rollupType>
  <statsType>rate</statsType>
  <level>1</level>
  <perDeviceLevel>3</perDeviceLevel>
</PerfCounterInfo>
```

XPath を使用して、受信した XML からカウンターパスを抽出します。上記の例では、XPath は次のようになります:

フィールド	xPath	値
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

この結果の場合のパフォーマンスカウンターパスは次のとおりです: `cpu/usagemhz [average]`

6 サポートされている関数

詳細を表示するには、それぞれの関数グループをクリックします。

関数群

関数

集計関数

avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp

Foreach 関数

avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum

関数群	関数
ビット関数	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
日時関数	date, dayofmonth, dayofweek, now, time
ヒストリ関数	change, changecount, count, countunique, find, first, fuzzytime, last, logeventid, logseverity, logsource, monodec, monoinc, nodata, percentile, rate
トレンド関数	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
数学関数	abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate
オペレーター関数	between, in
予測関数	forecast, timeleft
文字列関数	ascii, bitlength, bytelength, char, concat, insert, left, length, ltrim, mid, repeat, replace, right, rtrim, trim

これらの関数はトリガー式および計算項目でサポートされています。

Foreach 関数は集計計算のみサポートされています。

1 集計関数

特に明記されていない限り、ここにリストされているすべての機能は次の言語でサポートされています。

- トリガー式
- 計算項目

集計関数は次のいずれかで機能します。

- アイテムの履歴。例:min(/host/key,1h)
- foreach 関数のパラメータ。例:min(last_foreach(/*/key))

関数パラメーターに関する一般的な注意事項。

- 関数パラメータはコンマで区切られます。
- オプションの関数パラメータ (またはパラメータ一部分) は <> で示されます
- 機能固有のパラメータは、各機能で説明されています
- /host/key と (sec|#num)<:time shift> はパラメータを引用符で囲んではいけません

共通パラメータ

- /host/key はホストアイテムの履歴を参照する関数の一般的なパラメータの一つ目です。必須です。
- (sec|#num)<:time shift> は、ホストのアイテム履歴を参照する関数の一般的なパラメータの二つ目です。
 - **sec** - 秒単位の最大**評価期間** (タイム**サフィックス** を使用できます) または
 - **#num** - 最新の収集値の最大**評価範囲** (先頭にハッシュマークがある場合)
 - **time shift** (オプション) 評価ポイントを時間内に戻すことができます。時間シフトの指定については**詳細**を参照してください。

集計関数

関数	関数固有のパラメータ	コメント
avg (/host/key,(sec #num)<:time shift>) 定義された期間内のアイテムの平均値	共通パラメータ を参照	サポートされている値のタイプ: float,int 例: => avg (/host/key, 1h) → 現在 から過去 1 時間の平均値 => avg (/host/key, 1h:now-1d) → 現在 の 25 時間前から 24 時間前までの 1 時間の平均値 => avg (/host/key, #5) → 最新の 5 つの値の平均値 => avg (/host/key, #5:now-1d) → 過去 24 時間に受信した値を除く、最新の 5 つの値の平均値 時間シフトは、現在の平均値を少し前の平均値と比較する必要がある場合に便利です。
bucket_percentile (item filter,time period,percentage) ヒストグラムのバケットからパーセンタイルを計算します	item filter - item filter 参照 time period - time period 参照 percentage - パーセンテージ (0-100)	計算項目でのみサポートされます。 この関数は histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1)) のエイリアスです
count (func_foreach(item filter,<time period>)) foreach 関数によって返される配列内の値の数	func_foreach - 戻り値の数をカウントする foreach 関数 (サポートされている引数付き)。詳細については foreach 関数 を参照してください。	サポートされている値の型: int 例: => count (max_foreach(/*/net.if.in[*],1h)) → 現在 から 1 時間までにデータを受信した net.if.in アイテムの数 履歴関連の foreach 関数 (max_foreach, avg_foreach など) で count() を使用すると、パフォーマンスに影響を与える可能性があります。設定データではそのような影響はありません。
histogram_quantile (quantile,bucket1,value1,bucket2,value2,...)		

関数

ヒストグラムのバケットから ϕ 変位値を計算します

quantile - $0 \leq \phi \leq 1$
bucketN, valueN -
bucket_rate_foreachのパラメータまたは応答の手動で入力されたペア (≥ 2)

計算項目のみ対応。

機能的には PromQL の "histogram_quantile" に対応。

最後の 'Infinity' バケット ("+inf") の値が 0 の場合、-1 を返します。

例:

```
=> histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]))
=> histogram_quantile(0.5,bucket_rate_foreach(/item_key,
```

item_count (item filter)

構成内のフィルター条件に一致する既存アイテムの数

item filter - ホストグループ、ホスト、アイテムキー、タグによる参照が可能。ワイルドカードがサポートされています。詳細については、[アイテムフィルター](#)を参照してください。

サポートされている値のタイプ: int

count(exists_foreach(item_filter)) 関数のエイリアスとして機能します。

例:

```
=> item_count(/*/agent.ping?[group="Host group 1"])
-> "ホストグループ 1" に agent.ping 項目があるホストの数
```

kurtosis (/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の確率分布 "テーリング"

共通パラメータ参照

サポートされている値のタイプ: float, int

[尖度参照](#)

例:

```
=> kurtosis(/host/key,1h)
-> 現在から過去 1 時間の尖度
```

mad (/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の中央絶対偏差

共通パラメータ参照

サポートされている値のタイプ: float, int

[絶対偏差中央値参照](#)

例:

```
=> mad(/host/key,1h)
-> 現在から過去 1 時間の絶対偏差中央値
```

max (/host/key,(sec|#num)<:time shift>)

定義された期間内のアイテムの最高値

共通パラメータ参照

サポートされている値のタイプ: float, int

例:

```
=> max(/host/key,1h) - min(/host/key,1h)
-> 現在から過去 1 時間以内の最大値と最小値の差 (値のデルタ) を計算します。
```

min (/host/key,(sec|#num)<:time shift>)

定義された期間内のアイテムの最低値

共通パラメータ参照

サポートされている値のタイプ: float, int

例:

```
=> max(/host/key,1h) - min(/host/key,1h)
-> 現在から過去 1 時間以内の最大値と最小値の差 (値のデルタ) を計算します。
```

skewness

(/host/key,(sec|#num)<:time shift>)

関数

定義された期間内に収集された値の確率分布の非対称性

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

[歪度参照](#)

例:

=> **skewness**(/host/key,1h) → 現在から過去 1 時間の歪度

stddevpop

(/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の母標準偏差

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

[標準偏差参照](#)

例:

=> **stddevpop**(/host/key,1h) → 現在から過去 1 時間の標準偏差

stddevsamp

(/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値のサンプル標準偏差

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

[標準偏差参照](#)

この関数が機能するには少なくとも 2 つのデータ値が必要です。

例:

=> **stddevsamp**(/host/key,1h) → 現在から過去 1 時間のサンプル標準偏差

sum (/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の合計

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

例:

=> **sum**(/host/key,1h) → 現在から過去 1 時間の値の合計

sumofsquares

(/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の二乗和

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

例:

=> **sumofsquares**(/host/key,1h) → [現在] から過去 1 時間の平方和 (/manual/config/triggers#evaluation_period)

varpop (/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の母分散。

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

[分散参照](#)

例:

=> **varpop**(/host/key,1h) → 現在から過去 1 時間の母集団分散

varsamp

(/host/key,(sec|#num)<:time shift>)

定義された期間内に収集された値の標本分散

[共通パラメータ参照](#)

サポートされている値のタイプ: float, int

[分散参照](#)

この関数が機能するには少なくとも 2 つのデータ値が必要です

例:

=> **varsamp**(/host/key,1h) → 現在から過去 1 時間の標本分散

1 Foreach 関数

概要

foreach 関数は、複数のアイテム履歴から集計値を返します。Foreach 関数は、集計計算で使用されます。

次の foreach 関数がサポートされています。

関数	説明
avg_foreach	平均値を返します
bucket_rate_foreach	histogram_quantile() 関数での使用に適したペア (バケットの上限、レート値) を返します。ここでの "バケットの上限" は、<parameter number> パラメーター で定義された項目キーパラメーターの値です。
count_foreach	値の個数を返します
exists_foreach	現在有効になっているアイテムの数を返します
last_foreach	最後の値を返します
max_foreach	最大値を返します
min_foreach	最小値を返します
sum_foreach	合計値を返します

Foreach 関数は、項目ごとに値の配列を 1 つ返します。アイテムは最初のパラメーターでフィルターを使用して選択されます。

foreach 関数の使用の詳細については、**集計計算** を参照してください。

サポートされている関数

関数	説明
avg_foreach	各項目の平均値を返します
bucket_rate_foreach	histogram_quantile() 関数での使用に適したペア (バケットの上限、レート値) を返します。ここでの "バケットの上限" は、<パラメーター番号> パラメーター で定義されたアイテムキーパラメーターの値です。
count_foreach	各アイテムの値の数を返します
exists_foreach	現在有効になっているアイテムの数を返します
last_foreach	各アイテムの最後の値を返します
max_foreach	各アイテムの最大値を返します
min_foreach	各アイテムの最小値を返します
sum_foreach	各アイテムの値の合計を返します

パラメータ

Foreach 関数は、アイテムフィルターと期間という 2 つの共通パラメーターをサポートします。

foreach_function(item filter,time period)

例:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

一部の関数は、追加のパラメーターをサポートしています。

アイテムフィルター

一つ目のパラメーターを使用すると、関心のある項目のみにフィルターできます。例に示すように項目キー、ホストグループ、およびタグを参照する複雑なフィルターを使用できます。

構文例	説明
/host/key[abc,*]	このホストの類似アイテム
*/key	任意のホストの同じアイテム
*/key?[group="ABC" and tag="tagname:value"]	'tagname:value' タグを持つ ABC グループの任意のホストの同じアイテム
*/key[a,*c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]	それぞれのタグを持つ ABC または DEF グループの任意のホストの類似アイテム

データを収集するには、すべての参照項目が存在する必要があります。有効なホスト上の有効なアイテムのみが計算に含まれます。

Attention:

参照アイテムのアイテムキーが変更された場合、フィルターを手動で更新する必要があります。

親ホストグループを指定すると、親グループと、ネストされたすべてのホストグループとその項目が含まれます。

期間

2つ目のパラメータを使用すると集計の期間を指定できます。期間は時間としてのみ表現できます。値の量 (# で始まる) はサポートされていません。

サポートされている単位記号 はこのパラメーターで使用できます。たとえば '300s' (300 秒) の代わりに '5m' (5 分) または '86400' (86400 秒) 等。

last_foreach 関数で渡された場合はサーバーによって無視されるため、以下のように省略できます。

```
last_foreach( /*/key?[group="host group"] )
```

exists_foreach 関数ではサポートされていません。

追加のパラメータ

3 番目のオプションパラメータは、bucket_rate_foreach 関数でサポートされています。

```
bucket_rate_foreach( item filter, time period, <parameter number> )
```

ここで <parameter number> は項目キー内の " バケット " 値の位置です。たとえば myItem[aaa,0.2] の "bucket" 値が '0.2' の場合、その位置は 2 です。

<parameter number> のデフォルト値は '1' です。

Behavior depending on availability

The following table illustrates how each function behaves in cases of limited availability of host/item and history data.

Function	Disabled host	Unavailable host with data	Unavailable host without data	Disabled item	Unsupported item	Data retrieval error (SQL)
avg_foreach	ignore	return avg	ignore	ignore	ignore	ignore
bucket_rate_foreach	ignore	return bucket rate	ignore	ignore	ignore	ignore
count_foreach	ignore	return count	0	ignore	ignore	ignore
exists_foreach	ignore	1	1	ignore	1	n/a
last_foreach	ignore	return last	ignore	ignore	ignore	ignore
max_foreach	ignore	return max	ignore	ignore	ignore	ignore
min_foreach	ignore	return min	ignore	ignore	ignore	ignore
sum_foreach	ignore	return sum	ignore	ignore	ignore	ignore

If the item is ignored, nothing is added to the aggregation.

2 Bitwise functions

ここにリストされているすべての機能は、次でサポートされています。

- トリガー式
- 計算項目

関数パラメーターに関する一般的な注意事項:

- 関数パラメータはコマンドで区切られます
- 式はパラメーターとして受け入れられます
- オプションの関数パラメーター (またはパラメーター部分) は <> で示されます

関数**説明**

bitand (value,mask)

関数固有のパラメーター**コメント**

関数

項目値とマスクの”ビット論理積”の値	value - チェックする値 mask (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)	サポートされている値の型:int 比較はビットごとにすべての値を指定する必要があり、10 進数で返されます。たとえば 3 番目のビットのチェックは、100 ではなく 4 と比較して行われます。 例: => bitand (last(/host/key), 12)=8 または bitand (last(/host/key), 12)=4 → 3 番目または 4 番目のビットが設定されますが、両方が同時に設定されることはありません => bitand (last(/host/key), 20)=16 → 3 番目のビットが設定されておらず、5 番目のビットが設定されています。
bitlshift (value,bits to shift) アイテム値のビット単位左シフト	value - チェックする値 bits to shift (必須) - シフトするビット数	サポートされている値の型: int 比較はビットごとに行われますが、すべての値を指定する必要があり、10 進数で返されます。たとえば、3 番目のビットのチェックは、100 ではなく 4 と比較して行われます。
bitnot (value) 項目値の”ビット否定”の値	value - チェックする値	サポートされている値の型: int 比較はビットごとに行われますが、すべての値を指定する必要があり、10 進数で返されます。たとえば、3 番目のビットのチェックは、100 ではなく 4 と比較して行われます。
bitor (value,mask) 項目値とマスクの”ビット論理和”の値	value - チェックする値 mask (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)	サポートされている値の型: int 比較はビットごとに行われますが、すべての値を指定する必要があり、10 進数で返されます。たとえば、3 番目のビットのチェックは、100 ではなく 4 と比較して行われます。
bitrshift (value,bits to shift) アイテム値のビット単位右シフト	value - チェックする値 bits to shift (必須) - シフトするビット数	サポートされている値の型: int 比較はビットごとに行われますが、すべての値を指定する必要があり、10 進数で返されます。たとえば、3 番目のビットのチェックは、100 ではなく 4 と比較して行われます。
bitxor (value,mask) 項目値とマスクの”ビット排他的論理和”の値	value - チェックする値 mask (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)	サポートされている値の型: int 比較はビットごとに行われますが、すべての値を指定する必要があり、10 進数で返されます。たとえば、3 番目のビットのチェックは、100 ではなく 4 と比較して行われます。

3 日時関数

ここにリストされているすべての機能は、次でサポートされています。

- トリガー式
- 計算項目

Attention:

日時関数は、式だけでは使用できません。ホスト項目を参照する少なくとも 1 つの非時間ベースの関数が式に存在する必要があります。

関数	関数固有のパラメータ	コメント
date YYYYMMDD 形式の現在日付		例: => date() <20220101
dayofmonth 1 から 31 の範囲の日		例: => dayofmonth() =1
dayofweek 1 から 7 の範囲の曜日 (月 - 1, 日 - 7).		例: => dayofweek() <6
now エポックからの秒数 (00:00:00 UTC, January 1, 1970).		例: => now() <1640998800
time HHMMSS 形式の現在時刻		例: => time() >000000 and time() <060000

4 ヒストリ関数

ここにリストされているすべての機能は、次でサポートされています。

- トリガー式
- 計算項目

関数パラメーターに関する一般的な注意事項:

- 関数パラメータはコンマで区切られます。
- オプションの関数パラメーター (またはパラメーター部分) は <> で示されます
- 機能固有のパラメーターは、各機能で説明されています
- /host/key と (sec|#num)<:time shift> はパラメータを引用符で囲んではいけません

共通パラメータ

- /host/key はホストアイテムの履歴を参照する関数の一般的なパラメーターの一つ目です。必須です。
- (sec|#num)<:time shift> は、ホストのアイテム履歴を参照する関数の一般的なパラメーターの二つ目です。
 - **sec** - 秒単位の最大評価期間 (タイムサフィックスを使用できます) または
 - **#num** - 最新の収集値の最大評価範囲 (先頭にハッシュマークがある場合)
 - **time shift** (オプション) 評価ポイントを時間内に戻すことができます。時間シフトの指定については詳細を参照してください。

ヒストリ関数

関数	関数固有のパラメーター	コメント
change (/host/key)		

関数

前の値と最新値の差量

サポートされている値の型:
float,int,str,text,log

文字列の戻り値:
0 - 値が等しい
1 - 値が異なる

例:

= > **change**(/host/key)>10

これらの入力例の値で見られるように、数値の差が計算されます ('前' と '最新' の値 = 差):

'1' および '5' = +4

'3' および '1' = -2

'0' および '-2.5' = -2.5

比較のために **abs** も参照してください。

changecount

(/host/key,(sec|#num)<:time
shift>,<mode>)

評価期間内の隣接する値間の変更数

共通パラメータを参照してください。

mode (省略可能。二重引用符で囲む必要があります)

サポートされている modes:

all - すべての変更をカウントします (デフォルト)

dec - 減少回数をカウントします

inc - 増加回数をカウントします

サポートされている値の型:
float,int,str,text,log

非数値型の場合 mode パラメータは無視されます。

例:

=> ****changecount*** (/host/key, 1w) →

先週から今までの値の変更数

=> **changecount**(/host/key,#

10,"inc") → 最後の 10 個の値の中で (隣接する値と比較して) 値が増加する数

=>

changecount(/host/key,24h,"dec") →

24時間前から今まで (隣接する値に相対的) 値が減少する回数

count (/host/key,(sec|#num)<:time
shift>,<operator>,<pattern>)

定義された評価期間内の値の数

共通パラメーター を参照してください。

サポートされている値の型:
float,integer,string,text,log

operator (オプション。二重引用符で囲む必要があります)

サポートされている operator:

eq - 等しい (デフォルト)

ne - 等しくない

gt - 大きい

ge - 以上

lt - 小さい

le - 以下

like - パターンが含まれている場合に一致 (大文字と小文字を区別)

bitand - ビットごとの AND

regexp - pattern で指定された正規表現の大文字と小文字を区別する一致

iregexp - pattern で指定された正規表現の大文字と小文字を区別しない一致

pattern (オプション) - 必要なパターン (文字列引数は二重引用符で囲む必要があります)

float 項目は 2.22e-16 の精度に一致します。データベースが **アップグレードされていない** 場合、精度は 0.000001 です。

3 番目のパラメータとして bitand を使用すると、4 番目の pattern パラメータは '/' で区切られた 2 つの数字として指定できます:

number_to_compare_with/mask. count() は値と マスクから " ビットごとの AND " を計算し、結果を number_to_compare_with と比較します。" ビット単位の AND " の結果が number_to_compare_with と等しい場合、値がカウントされます。number_to_compare_with と mask が等しい場合、mask を指定する必要があります ('/' なし)。

global_regular_expressions) ('@' で始まる) 正規表現。グローバル正規表現の場合、大文字と小文字の区別はグローバル正規表現設定から継承されます。

regexp マッチングの目的で、float 値は常に '.' の後に 4 桁の 10 進数で表されます。また 10 進数 (データベースに保存) と 2 進数 (Zabbix サーバーで使用) の表現の違いが大きい場合、10 進数の 4 桁目に影響する可能性があることに注意してください。

例:

=> **count(/ host/key,10m)** → [今] までの過去 10 分間の値の数 (/manual/config/triggers#evaluation_period)
=> **count(/host/key,***

10m,"like","error")

→ **現在** までの過去 10 分間に「エラー」を含む値の数

=> **count (/host/key,10m,,12)** → **現在** までの過去 10 分間の値のうち、'12' に等しい値の数
=>

***count(/host/key,10m,"gt",12)** → **現在** までの過去 10 分間の'12' を超える値の数'
=>

count(/host/key,#10,"gt",12) → **今** が'12' を超えている
=>

count(/host/key,10m:now-1d,"gt",12) → **現在** から 24 時間前と 10 分前の値のうち、'12' を超えた値の数
=> **count(/host**

/key,10m,"bitand","6/7") → **現在** が「110」になるまでの過去 10 分間の値の数 (バイナリ)
=>

count(/host/key,10m:now-1d)** → **今** から 24 時間前と 10 分前の値の数

countunique

(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

評価期間内の一意の値の数

共通パラメーター を参照してください。

operator (オプション。二重引用符で囲む必要があります)

サポートされている operator:

eq - 等しい (デフォルト)

ne - 等しくない

gt - 大きい

ge - 以上

lt - 小さい

le - 以下

like - パターンが含まれている場合に一致 (大文字と小文字を区別)

bitand - ビットごとの AND

regexp - pattern で指定された正規表現の大文字と小文字を区別する一致

iregexp - pattern で指定された正規表現の大文字と小文字を区別しない一致

pattern (オプション) - 必要なパターン (文字列引数は二重引用符で囲む必要があります)

サポートされている値の型:

float,integer,string,text,log

float 項目は 2.22e-16 の精度に一致します。データベースが **アップグレードされていない** 場合、精度は 0.000001 です。

3 番目のパラメータとして bitand を使用すると、4 番目の pattern パラメータは '/' で区切られた 2 つの数字として指定できます:

number_to_compare_with/mask count() は値と マスクから " ビットごとの AND " を計算し、結果を number_to_compare_with と比較します。" ビット単位の AND " の結果が number_to_compare_with と等しい場合、値がカウントされます。number_to_compare_with と mask が等しい場合、mask を指定する必要があります ('/' なし)。

global_regular_expressions) ('@' で始まる) 正規表現。グローバル正規表現の場合、大文字と小文字の区別はグローバル正規表現設定から継承されます。

regexp マッチングの目的で、float 値は常に 「.」 の後に 4 桁の 10 進数で表されます。また、大きな数値の場合、10 進数 (データベースに保存) と 2 進数 (Zabbix サーバーで使用) の表現の違いが、10 進数の 4 桁目に影響する可能性があることに注意してください。

例:

```
=> countunique(/ host/key,10m) →
今までの過去 10 分間の一意の値の数
=> countunique(/host/key,
10m,"like","error") → 現在 までの過去
10 分間に 「エラー」 を含む一意の値の
数
=> ** countu-
nique(/host/key,10m,"gt",12) → 現
在 までの過去 10 分間で '12 を超える
一意の値の数' <br>=>
countunique(/host/key,#10,"gt",12)
→ 現在 までの最後の 10 個の値内の一
意の値の
数/triggers#evaluation_period)
that are over '12' <br>=>
countunique(/host/key,10m:now-
1d,"gt",12) → 24 の間の一意の
値の数 [今] から 24 時間前 10 分 (/man
ual/config/triggers#evaluation_period)
that were over '12' <br>=> countu-
nique(/host/key,10m,"bitand","6/7")
→ ユニーク数現在 までの最後の 10 分
間の値は、最下位 3 ビットが (バイナ
リで) '110' になります。 <br>=>
countunique(/host/key
,10m:now-1d**) → 今から 24 時間前と
10 分と 24 時間前の一意の値の数
```


find (/host/key,<(sec|#num)<:time shift>,<operator>,<pattern>)
一致する値を見つける

共通パラメータ を参照してください。

sec または **#num** (オプション) - 指定しない場合、デフォルトで最新の値になります

operator (オプション。二重引用符で囲む必要があります)

サポートされている「演算子」:

eq - 等しい (デフォルト)

ne - 等しくない

gt - 大きい

ge - 以上

lt - 少ない

le - 以下

like - 値には **pattern** で指定された文字列が含まれます (大文字と小文字が区別されます)

bitand - ビットごとの AND

regexp - **pattern** で指定された正規表現の大文字と小文字を区別する一致

iregexp - **pattern** で指定された正規表現の大文字と小文字を区別しない一致

pattern - 必要なパターン (文字列引数は二重引用符で囲む必要があります)。
[Perl 互換正規表現 \(PCRE\)](#) operator が `regexp`, `iregexp` の場合の正規表現。

共通パラメータ 参照

サポートされている値の型:
float,int,str,text,log

戻り値:

1 - 見つかりました

0 - それ以外

複数の値が処理された場合、一致する値が少なくとも 1 つあれば '1' が返されま

す。
3 番目のパラメータとして `regexp` または `iregexp` を使用すると、4 番目の `pattern` パラメータは通常の正規表現または **グローバル** ('@' で始まる) 正規表現にすることができます。グローバル正規表現の場合、大文字と小文字の区別はグローバル正規表現設定から継承されます。

例:

=> **find**(/host/key,**10m**,"like","error")

→ **now**

first (/host/key,sec<:time shift>)
評価期間内の最初の (最も古い) 値

サポートされている値の型:
float,int,str,text,log

例:

=> **first**(/host/key,**1h**) → 1 時間から **今** までで最も古い値を取得します。

fuzzytime (/host/key,sec)

`last()` も参照してください。

関数

パッシブエージェントの時間と Zabbix
サーバー/プロキシの時間との差を確認
します

共通パラメータ参照

サポートされている値の型: float,int

戻り値:

1 - パッシブアイテムの値 (タイムスタンプとして) と Zabbix サーバー/プロキシのタイムスタンプ (値収集のクロック) の差は T 秒以下です
0 - それ以外

通常では 'system.localtime' アイテムと一緒に使用して、現地時間が Zabbix サーバーの現地時間と同期していることを確認します。注 'system.localtime' は **パッシブチェック** として設定する必要があります。

vfs.file.time[/path/file,modify] キーと一緒に使用して、ファイルが長時間更新されていないことを確認することもできます。

例:

=> **fuzzytime**(/host/key,60s)=0 → 時差が 60 秒以上あれば問題を検出

この関数を複雑なトリガー式 (複数の項目が関係する) で使用することはお勧めしません。

例: **fuzzytime**(/Host/system.localtime,60s) or **last**(/Host/trap)<>0

last (/host/key,<#num<:time shift>)
最新の値

共通パラメータ参照

サポートされている値の型:
float,int,str,text,log

#num (オプション) - N 番目に新しい値

 ハッシュタグ付きの期間 (#N) は、ここでは他の多くの機能とは異なる働きをすることに注意してください。
例:
last() は常に **last**(#1)
last(#3) - 3 番目に新しい値 (最新の 3 つの値ではありません)

 履歴の 1 秒間に 2 つ以上の値が存在する場合、Zabbix は値の正確な順序を保証しません。

 例:
=> **last**(/host/key) → 最後の値を取得
=> **last**(/host/key,#2) → 前の値を取得
=> **last**(/host/key,#1) <> **last**(/host/key,#2) → 最後と前の値が異なる

first() も参照 | **logeventid** (/host/key,<#num<:time shift>,<pattern>)|<|<| 最後のログエントリのイベント ID が正規表現と一致するか | **共通パラメータ** を参照してください。

#num (オプション) - N 番目に新しい値

pattern (オプション) - 必要なパターンを記述する正規表現、**Perl 互換正規表現** (PCRE) スタイル (文字列引数) 二重引用符で囲む必要があります | サポートされている値の種類: log

 戻り値:
0 - 一致しません
1 - マッチ | **logseverity** (/host/key,<#num<:time shift>)|<|<| 最後のログエントリの重大度 |
0 - デフォルトの重大度
N - 重大度 (整数、Windows イベントログに役立ちます: 1 - 情報, 2 - 警告, 4 - エラー, 7 - 監査失敗, 8 - 監査成功, 9 - クリティカル, 10 - 詳細)
Zabbix は、Windows イベントログの **Information** フィールドからログの重大度を取得します。| **logsource** (/host/key,<#num<:time shift>,<pattern>)|<|<| 最後のログエントリのログソースが正規表現と一致するか | **共通パラメータ** を参照してください。

#num (オプション) - N 番目に新しい値

pattern (オプション) - 必要なパターンを記述する正規表現、**Perl 互換正規表現** (PCRE) スタイル (文字列引数) 二重引用符で囲む必要があります | サポートされている値の種類: log

 戻り値:
0 - 一致しない
1 - 一致する

 通常、Windows イベントログに使用されます。例: **logsource**("VMware Server") | **monodec** (/host/key,(sec)#num)<:time shift>,<mode>)|<|<| 値が単調減少しているか | **共通パラメータ** 参照

mode (二重引用符で囲む必要があります) - weak* (すべての値が前の値より小さいか同じ。デフォルト) または strict (すべての値が減少) | サポートされている値の型: int

 期間内のすべての要素が継続的に減少する場合は 1 を返し、それ以外の場合は 0 を返します。

 例:
=> **monodec**(/Host1/system.swap.size[all,free],60s) + **monodec**(/Host2/system.swap.size[all,free],60s) + **monodec**(/Host3/system.swap.size[all,free],60s) - 空きスワップサイズが減少したホストの数を計算します | **monoinc** (/host/key,(sec)#num)<:time shift>,<mode>)|<|<| 値が単調増加しているか | **共通パラメータ** 参照

mode (二重引用符で囲む必要があります) - weak (すべての値が前のものよりも大きいか同じ。デフォルト) または strict (すべての値が増加) | サポートされている値の型: int

 期間内のすべての要素が継続的に増加する場合は 1 を返し、それ以外の場合は 0 を返します。

 例:
=> **monoinc**(/Host1/system.localtime,#3,"strict")=0 - システムのローカル時間が一貫して増加しているかどうかを確認 | **nodata** (/host/key,sec,<mode>)|<|<| データが受信されていないか | **共通パラメータ** 参照

 秒間隔は 30 秒未満にしないでください。これは、履歴同期プロセスがこの関数を 30 秒ごとに計算するためです。

nodata(/host/key,0) は許可されていません。

mode - strict (二重引用符) に設定すると、この関数はプロキシの可用性に影響されなくなります (詳

細についてはコメントを参照してください| すべての値の型がサポートされています。

 戻り値:
1 - 定義された期間中にデータが受信されなかった場合
0 - それ以外

Zabbix 5.0 以降、プロキシによって監視される 'nodata' トリガーは、デフォルトで、プロキシの可用性に敏感です。プロキシが利用できなくなった場合、'nodata' トリガーは、接続が復元された直後に起動せず、スキップされます遅延期間のデータ。パッシブプロキシの場合、接続が 15 秒以上、少なくとも 2 & ProxyUpdateFrequency 秒後に復元されると、抑制がアクティブになることに注意してください。アクティブなプロキシの場合、接続が 15 秒以上後に復元されると、抑制が有効になります。

 プロキシの可用性に対する機密性をオフにするには、3 番目のパラメーターを使用します。例: **nodata(/host/key,5m,"strict")**; この場合、関数は 5.0.0 以前と同じように機能し、データの無い評価期間 (5 分間) が過ぎるとすぐに起動します。

 この関数は、第 1 パラメーターの期間内に次の場合にエラーを表示することに注意してください:
- データがなく、Zabbix サーバーが再起動された
- データがなく、メンテナンスが完了した
- データがなく、項目が追加または再有効化された
 エラーは、トリガー構成の Info 列に表示されます。

 Zabbix サーバー、プロキシ、エージェントの時差がある場合、本機能が正常に動作しない場合があります。参照: [時刻同期の要件](#)。

All value types are supported.

Returns:
1 - if no data received during the defined period of time
0 - otherwise

Since Zabbix 5.0, the 'nodata' triggers monitored by proxy are, by default, sensitive to proxy availability - if proxy becomes unavailable, the 'nodata' triggers will not fire immediately after a restored connection, but will skip the data for the delayed period. Note that for passive proxies suppression is activated if connection is restored more than 15 seconds and no less than 2 & ProxyUpdateFrequency seconds later. For active proxies suppression is activated if connection is restored more than 15 seconds later.

To turn off sensitiveness to proxy availability, use the third parameter, e.g.: **nodata(/host/key,5m,"strict")**; in this case the function will work the same as before 5.0.0 and fire as soon as the evaluation period (five minutes) without data has past.

Note that this function will display an error if, within the period of the 1st parameter:
- there's no data and Zabbix server was restarted
- there's no data and maintenance was completed
- there's no data and the item was added or re-enabled
Errors are displayed in the Info column in trigger **configuration**.

This function may not work properly if there are time differences between Zabbix server, proxy and agent. See also: [Time synchronization requirement](#).| **percentile** (/host/key,(sec|#num)<:time shift>,<percentage>)|<| 期間の P パーセンタイル。ここで P (パーセンテージ) は 3 番目のパラメーターで指定されます | **共通パラメータ参照**

percentage - 小数点以下 4 桁までの 0 から 100 までの浮動小数点数 | サポートされている値の型: float,int| **rate** (/host/key,sec<:time shift>)|<| 期間内で単調増加するカウンターの 1 秒あたりの平均増加率 | **共通パラメータ参照** | サポートされる値の型: float,int

 機能的には PromQL の 'rate' に対応します。

例:
=> **rate(/host/key,30s)** → 30 秒間の単調増加が 20 の場合、この関数は 0.67 を返します。|

5 トレンド関数

トレンド関数は、[履歴関数](#) とは対照的に、[トレンドデータ](#) を計算に使用します。

トレンドは時間ごとの集計値を保存します。トレンド関数はこれらの時間平均を使用するため、長期的な分析に役立ちます。

トレンド関数の結果はキャッシュされるため、同じパラメーターを使用して同じ関数を複数呼び出しても、データベースから情報が取得されるのは 1 回だけです。トレンド機能のキャッシュは **TrendCacheSize** サーバーパラメーターによって制御されます。

トレンド関数 のみを参照するトリガーは、式の最小期間ごとに 1 回評価されます。たとえば次のようなトリガーは **trendavg(/host/key,1d:now/d) > 1** or **trendavg(/host/key2,1w:now/w) > 2**

1 日 1 回評価されます。トリガーにトレンドと履歴 (または時間ベース) の両方の関数が含まれている場合は [通常原則](#) に従って計算されます。

ここにリストされているすべての機能は、次でサポートされています。

- [トリガー式](#)
- [計算項目](#)

関数パラメーターに関する一般的な注意事項:

- 関数パラメーターはコンマで区切られます
- オプションの関数パラメーター (またはパラメーター部分) は <> で示されます
- 機能固有のパラメーターは、各機能で説明されています
- /host/key および time period:time shift パラメーターは引用符で囲んではいけません

共通パラメータ

- /host/key は一般的な必須の最初のパラメータです
- time period:time shift は一般的な 2 番目のパラメーターです。
 - **time period** - 期間 (最小 '1h')、<N>< 時間単位 > として定義、'N' - 時間単位の数、'時間単位' - h (時間) d (日)、w (週)、M (月)、または y (年)。
 - **time shift** - 期間のオフセット (関数の例を参照)

トレンド関数

関数

説明

関数固有のパラメータ

コメント

baselinedev (/host/key,data

period:time

shift,season_unit,num_seasons)

最後のデータ期間と前のシーズンの同じデータ期間との間の偏差の数を (stddevpop アルゴリズムによって) 返します

data period - <N>< 時間単位 > として定義される、シーズン内のデータ収集期間

N - 時間単位数

time unit - h (時), d (日), w (週), M (月), y (年), シーズン以下である必要があります

Time shift - 期間オフセット (例を参照)

season_unit - 1 シーズンの期間 (h, d, w, M, y)。データ期間より小さくすることはできません

num_seasons - 評価するシーズン数

例:

=> **base-**

linedev(/host/key,1d:now/d,"M",6) → 前日と過去 6 か月間の同じ日の標準偏差 (母集団) の数を計算します。日付が前の月に存在しない場合は、その月の最終日が使用されます (7 月 31 日は、1 月 31 日、2 月 28 日、... 6 月 30 日に対して分析されます)。

=> **base-**

linedev(/host/key,1h:now/h,"d",10) → 昨日から 10 日前までの 1 時間前と同じ時間との間の標準偏差 (母集団) の数を計算します。

baselinewma (/host/key,data

period:time

shift,season_unit,num_seasons)

加重移動平均アルゴリズムを使用して、複数の等しい期間 ('シーズン') の同じ時間枠からのデータを平均することにより、ベースラインを計算します。

data period - <N>< 時間単位 > として定義される、シーズン内のデータ収集期間

N - 時間単位数

time unit - h (時), d (日), w (週), M (月), y (年), シーズン以下である必要があります

Time shift - 期間オフセットは、シーズンのデータ収集時間枠の終了を定義します (例を参照)

season_unit - 1 シーズンの期間 (h, d, w, M, y)、データ期間より小さくすることはできません。

num_seasons - 評価するシーズン数

例:

=> **base-**

linewma(/host/key,1h:now/h,"d",3) → 昨日までの 3 日間の最後の 1 時間に基づいてベースラインを計算します。"今" が月曜日の 13:30 の場合、金、土、日の 12:00 から 12:59 までのデータが分析対象になります。

=> **base-**

linewma(/host/key,2h:now/h,"d",3) → 昨日までの 3 日間の過去 2 時間に基づいてベースラインを計算します。"今" が月曜日の 13:30 の場合、金、土、日の 10:00 から 11:59 までのデータが分析対象になります。

=> **base-**

linewma(/host/key,1d:now/d,"M",4) → 最後の丸 1 か月間の '昨日' と過去 4 か月の同じ日に基づいてベースラインを計算します。必要な日付が存在しない場合は、月の最終日が取得されます。今日が 9 月 1 日の場合、7 月 31 日、6 月 30 日、5 月 31 日、4 月 30 日のデータが分析されます。

trendavg (/host/key,time period:time

shift)

定義された期間内のトレンド値の平均

共通パラメータ参照

例:

=> **trendavg**(/host/key,1h:now/h) → 1 時間前の平均 (例:12:00-13:00)

=>

trendavg(/host/key,1h:now/h-1h) → 2 時間前の平均 (11:00-12:00)

=>

trendavg(/host/key,1h:now/h-2h) → 3 時間前の平均 (10:00-11:00)

=>

trendavg(/host/key,1M:now/M-1y) → 1 年前の前月の平均

trendcount (/host/key,time

period:time shift)

関数

定義された期間内に正常に取得された
トレンド値の数

共通パラメータ参照

例:

=> **trendcount(/host/key,1h:now/h)**

→ 1 時間前のカウント (例:

12:00-13:00)

=>

trendcount(/host/key,1h:now/h-1h)

→ 2 時間前のカウント (11:00-12:00)

=>

trendcount(/host/key,1h:now/h-2h)

→ 3 時間前のカウント (10:00-11:00)

=>

trendcount(/host/key,1M:now/M-1y)

→ 1 年前の前月のカウント

trendmax (/host/key,time period:time
shift)

定義された期間内のトレンド値の最大
値

共通パラメータ参照

例:

=> **trendmax(/host/key,1h:now/h)** →

1 時間前の最大値 (例: 12:00-13:00)

=> **trendmax(/host/key,1h:now/h) -**

trendmin(/host/key,1h:now/h) → 前

の 1 時間 (12:00 ~ 13:00) の最大値と最

小値の差 (トレンドデルタ) を計算しま

す。

=>

trendmax(/host/key,1h:now/h-1h) →

2 時間前の最大値 (11:00-12:00)

=>

trendmax(/host/key,1h:now/h-2h) →

3 時間前の最大値 (10:00-11:00)

=>

trendmax(/host/key,1M:now/M-1y)

→ 1 年前の前月の最大値

trendmin (/host/key,time period:time
shift)

定義された期間内のトレンド値の最小
値

共通パラメータ参照

例:

=> **trendmin(/host/key,1h:now/h)** →

1 時間前の最小値 (例: 12:00-13:00)

=> **trendmin(/host/key,1h:now/h) -**

trendmin(/host/key,1h:now/h) → 前

の 1 時間の最大値と最小値の差 (トレン

ドデルタ) を計算する。(12:00-13:00)

=>

trendmin(/host/key,1h:now/h-1h) →

2 時間前の最小値 (11:00-12:00)

=>

trendmin(/host/key,1h:now/h-2h) →

3 時間前の最小値 (10:00-11:00)

=>

trendmin(/host/key,1M:now/M-1y)

→ 1 年前の前月の最小値

trendstl (/host/key,eval period:time
shift,detection pe-

riod,season,<deviations>,<devalg>,<s_window>)

関数

検出期間中の異常の割合を返します - 0 から 1 までの 10 進数値つまり ((異常値の数)/(値の総数))

eval period - <N><time unit> として定義された、分解する必要がある期間 (最小'1h')

N - 時間単位数

time unit - h (時), d (日), w (週), M (月) or y (年).

Time shift - 期間オフセット (例を参照)

detection period - 異常が計算される評価期間の終了前の期間 (最小'1h'、評価期間より長くすることはできません)。<N><time unit> として定義されます。

N - 時間単位数

time unit - h (時), d (日), w (週).

season - 繰り返しパターン ("シーズン") が予想される最短期間 (最小'2h'、評価期間よりも長くすることはできません。評価期間のエントリ数は、結果の頻度 (シーズン/時間) の 2 倍よりも大きくする必要があります)、<N>< 時間単位数 > として定義

N - 時間単位数

time unit - h (時), d (日), w (週).

deviations - 異常としてカウントする偏差の数 (devalg で計算) (10 進数の場合もあります) (1 以上である必要があります、デフォルトは 3 です)

devalg (二重引用符で囲む必要があります) - 偏差アルゴリズム。stddevpop、stddevsamp、mad (デフォルト) のいずれか。

s_window - 季節抽出の黄土ウィンドウのスパン (ラグ) (デフォルトは 10 * 評価期間のエントリ数 + 1)

trendsum (/host/key,time period:time shift)

定義された期間内のトレンド値の合計

[共通パラメータ参照](#)

例:

=> **trend-**

stl(/host/key,**100h:now/h**,10h,2h) → 過去 100 時間の傾向データを分析し、その期間の最後の 10 時間の異常率を見つめます。

周期が 2 時間であると予想します。

評価期間の残りの系列値は、次の場合に異常と見なされます。それらは、残りの系列の MAD の 3 偏差の値に達します。

=> **trendstl**(/host/key,**100h:now/h-10h**,100h,2h,2.1,"mad") → 10 時間前までの 100 時間の傾向データの期間を分析し、

その期間全体の異常率を見つめます

周期性が 2 時間であると予想し、

評価期間の残りの系列値が考慮されずその残りの系列の MAD の 2,1 偏差の値に達した場合の異常

=> **trendstl**(/host/key,**100d:now/d-1d**,10d,1d,4,,10) → 1 日前までの 100 日間の傾向データを分析し、

その期間の最後の 10 日間の異常率を見つめます。

周期性が 1 日間であると予想します。

評価期間の残りの系列値その残りの系列の MAD の 4 偏差の値に達した場合、異常と見なされます。

「10 * 評価期間のエントリ数 + 1」の季節抽出用の黄土ウィンドウのデフォルトのスパンを次のスパンでオーバーライドします。10 ラグ

=> **trendstl**(/host/key,**1M:now/M-1y**,1d,2h,,,"stddevsamp") → 1 年前の前月を分析し、

その期間の最後の日の異常率を見つめます

周期性が 2 時間であると予想し、

評価期間の残りの系列値は、それらがその剰余系列のサンプル標準偏差の 3 偏差の値

例:

=> **trendsum**(/host/key,**1h:now/h**) → 一時間前の合計 (例: 12:00-13:00)

=>

trendsum(/host/key,**1h:now/h-1h**) → 二時間前の合計 (11:00-12:00)

=>

trendsum(/host/key,**1h:now/h-2h**) → 3 時間前の合計 (10:00-11:00)

=>

trendsum(/host/key,**1M:now/M-1y**)

→ 1 年前の前月の合計

6 数学関数

ここにリストされているすべての機能は、次でサポートされています。

- [トリガー式](#)

- 計算項目

特に明記されていない限り、数学関数は float および integer 値型でサポートされています。

関数パラメーターに関する一般的な注意事項:

- 関数パラメータはコンマで区切られます。
- オプションの関数パラメーター (またはパラメーター部分) は <> で示されます
- 式はパラメーターとして受け入れられます

関数	関数固有のパラメーター	コメント
abs (value) 値の絶対値	value - チェックする値	サポートされている値の型: float,int,str,text,log 文字列の場合: 0 - 値は等しい br>1 - 値が異なる 例: => abs (last(/host/key))>10 絶対数値の差が計算されます。入力例の 値 ('前' と '最新' の値 = 絶対差): '1' と '5' = 4 '3' と '1' = 2 '0' と '-2.5' = 2.5
acos (value) 値の逆余弦。値はラジアンで表した角 度です。	value - 確認する値	値は -1 から 1 の間でなければなりません。 たとえば、値 '0.5' の逆余弦' は '2.0943951' になります。 例: => acos (last(/host/key))
asin (value) 値の逆正弦。値はラジアンで表した角 度です。	value - 確認する値	値は -1 から 1 の間でなければなりません。 たとえば、値 '0.5' の逆正弦 は '-0.523598776' になります。 例: => asin (last(/host/key))
atan (value) 値の逆正接。値はラジアンで表した角 度です。	value - 確認する値	たとえば、値 '1' の逆正接 は '0.785398163' になります 例: => atan (last(/host/key))
atan2 (value,abscissa) 角度として指定された縦座標 (exprue) と横座標の逆正接 (ラジアン単位)	value - 確認する値 abscissa - 横座標値	たとえば、値 '1' の縦座標と横座標の逆 正接は '2.21429744' になります。 例: => atan2 (last(/host/key),2)
avg (<value1>,<value2>,...) 参照アイテムの平均値	valueX - いずれかの履歴関数によって 返される値	例: => avg (avg(/host/key),avg(/host2/key2))
cbrt (value)		

関数

値の立方根	value - 確認する値	たとえば'64'の立方根は'4'、'63'の立方根は'3.97905721'になります。 例: => cbrt (last(/host/key))
ceil (value) 値を最も近い整数に切り上げます	value - 確認する値	たとえば'2.4'は'3'に切り上げられます。 例: => ceil (last(/host/key)) floor() も参照してください。
cos (value) 値の余弦。値はラジアンで表した角度です。	value - 確認する値	たとえば、値'1'の余弦は'0.54030230586'になります。 例: => cos (last(/host/key))
cosh (value) 値の双曲線余弦	value - 確認する値	たとえば値'1'の双曲線余弦は'1.54308063482'になります。 科学表記法ではなく、実数として値を返します。 例: => cosh (last(/host/key))
cot (value) 値の余接。値はラジアンで表した角度です。	value - 確認する値	たとえば値'1'の余接は'0.54030230586'になります。 例: => cot (last(/host/key))
degrees (value) 値をラジアンから度に変換します	value - 確認する値	たとえば値'1'を度に変換すると'57.2957795'になります。 例: => degrees (last(/host/key))
e オイラー数 (2.718281828459045)		例: => e ()
exp (value) 値の累乗でのオイラー数	value - 確認する値	たとえば値'2'の累乗におけるオイラー数は'7.38905609893065'になります。 例: => exp (last(/host/key))
expm1 (value) 値から 1 を引いた値のオイラー数	value - 確認する値	たとえば値'2'から 1 を引いた値のオイラー数は'6.38905609893065'になります。 例: => expm1 (last(/host/key))
floor (value)		

関数

値を切り捨ててそれよりも小さいか等しい整数にします	value - 確認する値	たとえば'2.6' は'2' に切り捨てられます。 例: => floor (last(/host/key)) ceil() も参照してください
log (value) 自然対数	value - 確認する値	たとえば値'2' の自然対数は'0.69314718055994529' になります。 例: => log (last(/host/key))
log10 (value) 十進対数	value - 確認する値	たとえば、値'5' の十進対数は'0.69897000433' になります。 例: => log10 (last(/host/key))
max (<value1>,<value2>,...) 参照された項目の最高値	valueX - いずれかの履歴関数によって返される値	例: => max (avg(/host/key),avg(/host2/key2))
min (<value1>,<value2>,...) 参照された項目の最低値	valueX - いずれかの履歴関数によって返される値	例: => min (avg(/host/key),avg(/host2/key2))
mod (value,denominator) 除算剰余	value - 確認する値 denominator - 除算分母	たとえば除算分母が'2' の値'5' の除算剰余は'1' になります。 例: => mod (last(/host/key),2)
pi パイ定数 (3.14159265358979).		例: => pi ()
power (value,power value) べき乗	value - 確認する値 power value - 使用する N 乗	たとえば値'2' の 3 乗は'8' になります。 例: => power (last(/host/key),3)
radians (value) 値を度からラジアンに変換	value - 確認する値	たとえば値'1' をラジアンに変換すると'0.0174532925' になります。 例: => radians (last(/host/key))
rand ランダムな整数値を返します		時間をシードとして生成された疑似乱数 (数学的目的には十分ですが、暗号化には不十分)。 例: => rand ()
round (value,decimal places) 値を小数点以下の桁数で丸める	value - 確認する値 decimal places - 丸め的小数点以下の桁数を指定します (0 も可能)	
signum (value) 値が負の場合は'-1'、値がゼロの場合は'0'、値が正の場合は'1' を返します。	value - 確認する値	例: => signum (last(/host/key))

関数		
sin (value) 値の正弦。値はラジアンで表した角度です。	value - 確認する値	たとえば値'1'の正弦は'0.8414709848'になります。 例: => sin (last(/host/key))
sinh (value) 値の双曲線正弦	value - 確認する値	たとえば値'1'の双曲線正弦は'1.17520119364'になります。 例: => sinh (last(/host/key))
sqrt (value) 値の平方根	value - 確認する値	この関数は負の値では失敗します。 たとえば値'3.5'の平方根は'1.87082869339'になります。 例: => sqrt (last(/host/key))
sum (<value1>,<value2>,...) 参照項目値の合計	valueX - いずれかの履歴関数によって返される値	例: => sum (avg(/host/key),avg(/host2/key2))
tan (value) 値の正接	value - 確認する値	たとえば、値'1'の正接は'1.55740772465'になります。 例: => tan (last(/host/key))
truncate (value,decimal places) 値を小数点以下に切り捨て	value - 確認する値 decimal places - 切り捨てる小数点以下の桁数を指定 (0 も可能)	例: => truncate (last(/host/key),2)

7 オペレーター関数

ここにリストされているすべての機能は、次でサポートされています。

- [トリガー式](#)
- [計算項目](#)

関数パラメーターに関する一般的な注意事項:

- 関数パラメーターはコンマで区切られます
- 式はパラメーターとして受け入れられます

関数		
説明	関数固有のパラメーター	コメント
between (value,min,max) 値が指定された範囲内かどうかを確認します。	value - チェックする値 min - 最小値 max - 最大値	サポートされている型: integer, float 戻り値: 1 - 範囲内 0 - 範囲外 例: => between (last(/host/key), 1 , 10)=1 - 値が 1 ~ 10 の場合にトリガーします
in (value,value1,value2,...valueN)		

関数

値がリストされた値の少なくとも 1 つと等しいかどうかを確認します。

value - チェックする値
value1,value2,...valueN - リストされた値 (文字列値は二重引用符で囲む必要があります)

サポートされている型: すべて

戻り値:
1 - 等しい値がある
0 - 等しい値が無い

これらの値がすべて数値に変換できる場合、値は数値としてリストされた値と比較されます。それ以外の場合は文字列として比較されます。

例:

=> **in**(last(/host/key),**5,10**)=1 - 最後の値が 5 または 10 の場合にトリガーする
=> **in**("text",
last(/host/key),last(/host/key,#2))=1 - "text" が最後の 2 つの値のいずれかと等しい場合にトリガーします

8 予測関数

ここにリストされているすべての機能は、次でサポートされています。

- [トリガー式](#)
- [計算項目](#)

関数パラメーターに関する一般的な注意事項:

- 関数パラメーターはコマンドで区切られます。
- オプションの関数パラメーター (またはパラメーター部分) は <> で示されます
- 機能固有のパラメーターは、各機能で説明されています
- /host/key と (sec|#num)<:time shift> はパラメーターを引用符で囲んではいけません

共通パラメーター

- /host/key はホストアイテムの履歴を参照する関数の一般的なパラメーターの一つ目です。必須です。
- (sec|#num)<:time shift> は、ホストのアイテム履歴を参照する関数の一般的なパラメーターの二つ目です。
 - **sec** - 秒単位の最大**評価期間** (タイム**サフィックス** を使用できます) または
 - **#num** - 最新の収集値の最大**評価範囲** (先頭にハッシュマークがある場合)
 - **time shift** (オプション) 評価ポイントを時間内に戻すことができます。時間シフトの指定については[詳細](#)を参照してください。

予測関数

関数

説明

関数固有のパラメーター

コメント

forecast

(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

関数

アイテムの将来の値、最大、最小、デルタ、平均

共通パラメーターを参照

サポートされている値の型: float, int

time - 秒単位の予測期間 (タイムサフィックスを使用できます)。負の値がサポートされています

返される値が
1.7976931348623157E+308 より大きい場合、または
-1.7976931348623157E+308 より小さい場合、戻り値は
1.7976931348623157E+308 または
-1.7976931348623157E+308 にトリミングされます。

fit (オプション。二重引用符で囲む必要があります) - 履歴データの適合に使用される関数

式が誤用された場合 (間違っアイテムタイプや無効なパラメーター等) にのみサポートされなくなります。それ以外の場合でエラーの場合は -1 が返されま

サポートされている 'fits':

linear - 線形関数

polynomialN - 次数 N の多項式 (1 <= N <= 6)

exponential - 指数関数

logarithmic - 対数関数

power - パワー関数

例:

=> **forecast**(/host/key,#10,1h) → 過去 10 個の値に基づいて 1 時間のアイテム値を予測

=> **forecast**(/host/key,1h,30m) → 過去 1 時間のデータに基づいて 30 分間のアイテム値を予測

=>

forecast(/host/key,1h:now-1d,12h)

→ 1 日前の 1 時間に基づいて 12 時間後のアイテム値を予測

=> **fore-**

cast(/host/key,1h,10m,"exponential")

→ 過去 1 時間のデータと指数関数に基づいて 10 分間のアイテム値を予測

=> **fore-**

cast(/host/key,1h,2h,"polynomial3","max")

→ 過去 1 時間のデータと 3 次 (3 次) 多項式に基づいてアイテムが次の 2 時間で到達できる最大値を予測

=> **forecast**(/host/key,#2,-20m) →

最後の 2 つの値に基づいて 20 分前の項目の値を予測 (これは last() を使用するよりも正確です。特に 1 時間に 1 回などアイテムがめったに更新されない場合等)

注意:

linear はデフォルトです。polynomial1 は linear と同等です。

mode (オプション。二重引用符で囲む必要があります) - 要求された出力

サポートされている 'mode':

value - 値 (デフォルト)

max - 最大

min - 最小

delta - 最大-最小

avg - 平均

注意:

value は現時点でのアイテムの推定値
now + time

max, min, delta, avg は 'now' と 'now' + 'time' の間隔でアイテム値の見積もりを調べる

予測トリガー機能に関する追加情報も参照してください。

timeleft (/host/key,(sec|#num)<:time shift>,threshold,<fit>)

関数

アイテムが指定されたしきい値に達するまでに必要な時間 (秒)

共通パラメーターを参照

threshold - しきい値 (単位サフィックスを使用できます)

fit (オプション。二重引用符で囲む必要があります) - Forecast() を参照してください

サポートされている値の型: float,int

返される値が

1.7976931348623157E+308 より大きい場合、戻り値は

1.7976931348623157E+308 にトリミングされます。

しきい値に到達できない場合、1.7976931348623157E+308 を返します。

式で誤用された場合 (間違ったアイテムタイプや無効なパラメーター等) にのみサポートされなくなります。それ以外のエラーは -1 が返されます。

例:

=> **timeleft(/host/key, #10,0)** → 最後の 10 個の値に基づいて、アイテム値がゼロになるまでの時間

=> **timeleft(/host/key,1h,100)** → 過去 1 時間のデータに基づいてアイテム値が 100 になるまでの時間

=>

timeleft(/host/key,1h:now-1d,100)

→ 1 日前の 1 時間に基づいてアイテム値が 100 になるまでの時間

=>

timeleft(/host/key,1h,200,"polynomial2")

→ 過去 1 時間のデータと、アイテムが 2 次多項式のように動作するという仮定に基づいて、アイテムの値が 200 に達するまでの時間

予測トリガー関数 に関する追加情報も参照してください。

9 文字列関数

ここにリストされているすべての機能は、次でサポートされています。

- **トリガー式**
- **計算項目**

関数パラメーターに関する一般的な注意事項:

- 関数のパラメーターはコンマで区切られます
- 式はパラメーターとして受け入れられます
- 文字列のパラメーターは二重引用符で囲む必要があります。そうしないと、他の型と認識される可能性があります
- オプションの関数パラメーター (またはパラメーター部分) は <> で示されます

関数

説明

ascii (value)

関数固有のパラメーター

コメント

関数

値の左端文字の ASCII コード	value - チェックする値	サポートされている値のタイプ: 文字列、テキスト、ログ たとえば「Abc」のような値の場合、'A' の ASCII コードである '65' を返します。 例: => ascii (last(/host/key))
bitlength (value) 値の長さ (ビット単位)	value - チェックする値	サポートされている型: string, text, log, integer 例: => bitlength (last(/host/key))
bytelength (value) 値の長さ (バイト単位)	value - チェックする値	サポートされている型: string, text, log, integer 例: => bytelength (last(/host/key))
char (value) 値を ASCII コードと解釈して文字を返します	value - チェックする値	サポートされている型: integer 値は 0 ~ 255 の範囲内である必要があります。たとえば '65' のような値は (ASCII コードとして解釈された結果)、'A' を返します。 例: => char (last(/host/key))
concat (<value1>,<value2>,...) 参照された項目値または定数値を連結した結果の文字列	value - いずれかの履歴関数によって返される値、または定数値 (文字列、整数、または浮動小数点数)	サポートされている型: string, text, log, float, integer たとえば、'Zab' に 'bix' (定数文字列) を連結した場合は 'Zabbix' を返します。 少なくとも 2 つのパラメータを含める必要があります。 例: => concat (last(/host/key),"bix") => concat ("1 min:",last(/host/system.cpu.load[all,avg1]),",",15 min:",last(/host/system.cpu.load[all,avg15]))
insert (value,start,length,replacement) 文字列内の指定された位置から始まる文字列に、指定された文字またはスペースを挿入します。	value - チェックする値 start - 開始位置 length - 置換位置 replacement - 変換後文字	サポートされている型: string, text, log たとえば 'Zabbix' の 'bb' (開始位置 3、置換位置 2) が 'b' に置き換えられた場合、'Zabbix' となります。 例: => insert (last(/host/key),3,2,"b")
left (value,count)		

関数

値の左端の文字	value - チェックする値 count - 返す文字数	サポートされている型: string, text, log たとえば、'Zabbix' の左端 3 文字を返すように指定すると 'Zab' を返します。 例: => left (last(/host/key), 3) - 左端の 3 文字を返す 参照:right()
length (value) value の長さ (文字数)	value - チェックする値	サポートされている型: str, text, log 例: => length (last(/host/key)) → 最新値の長さ => length (last(/host/key,#3)) → 3 番目に新しい値の長さ => length (last(/host/key,#1:now-1d)) → 1 日前の最新値の長さ
ltrim (value,<chars>) 文字列の先頭から指定された文字を削除します。	value - チェックする値 chars - (オプション) 削除する文字を指定 デフォルトでは (オプションの文字が指定されていない場合)、空白が左詰めされます。	サポートされている型: string, text, log 例: => ltrim (last(/host/key)) - 文字列の先頭から空白を削除します => ltrim (last(/host/key),"Z") - 文字列の先頭から 'Z' を削除します => ltrim (last(/host/key)," Z") - 文字列の先頭からスペースと 'Z' を削除します 参照: rtrim(), trim()
mid (value,start,length) 'start' で指定された文字位置から始まる length 文字の部分文字列を返します。	value - チェックする値 start - 部分文字列の開始位置 length - 部分文字列からの文字数	サポートされている型: string, text, log たとえば値が 'Zabbix'、start が 2、length が 4 の場合、'abbi' が返されません。 例: => mid (last(/host/key), 2,4)="abbi"
repeat (value,count) 文字列を繰り返す	value - チェックする値 count - 繰り返す回数	サポートされている型: string, text, log 例: => repeat (last(/host/key), 2) - 値を 2 回繰り返す
replace (value,pattern,replacement) 値から特定のパターンを見つけて置換します。マッチしたパターンすべてが置き換えられます。	value - チェックする値 pattern - 探すパターン replacement - パターンを置き換える文字列	サポートされている型: string, text, log 例: => replace (last(/host/key),"ibb","abb") - 全ての 'ibb' を 'abb' に置換します

関数

right (value,count)

値の右端の文字

value - チェックする値

count - 返す文字数

サポートされている型: string, text, log

たとえば 'Zabbix' から右端の 3 文字を返すように指定すると、'bix' を返すことができます。

例:

=> **right**(last(/host/key),**3**) - 右端の 3 文字を返す

参照 left().

rtrim (value,<chars>)

文字列の末尾から指定された文字を削除します。

value - チェックする値

chars - (オプション) 削除する文字を指定します

デフォルトでは (オプションの文字が指定されていない場合)、空白が右詰めされます。

サポートされている型: string, text, log

例:

=> **rtrim**(last(/host/key)) - 文字列の末尾から空白を削除します

=> **rtrim**(last(/host/key),"x") - 文字列の末尾から 'x' を削除します

=> **rtrim**(last(/host/key),"x ") - 文字列の末尾から 'x' またはスペースを削除します

参照: ltrim(), trim()

trim (value,<chars>)

文字列の先頭と末尾から指定された文字を削除します。

value - チェックする値

chars - (オプション) 削除する文字を指定

デフォルトでは (オプションの文字が指定されていない場合)、空白が両側から削除されます。

サポートされている型: string, text, log

例:

=> **trim**(last(/host/key)) - 文字列の先頭と末尾から空白を削除します

=> **trim**(last(/host/key),"_") - r 文字列の先頭と末尾から "_" を削除します

参照: ltrim(), rtrim()

7 マクロ

1 サポートされているマクロ

概要

この表には、すぐに使用できる Zabbix でサポートされているマクロの完全なリストが含まれています。

Note:

ある場所 (" マップ URL" など) でサポートされているすべてのマクロを表示するには、ブラウザウィンドウの下部にある検索ボックス (CTRL+F でアクセス可能) に場所の名前を貼り付けて、次を検索します。

マクロ	サポート対象	説明
{ACTION.ID}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知	トリガーアクションの数値 ID
{ACTION.NAME}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知	トリガーアクションの名前
{ALERT.MESSAGE}	→ アラートスクリプトのパラメーター	アクション設定の'デフォルトメッセージ' 値 3.0.0 以降でサポートされています
{ALERT.SENDTO}	→ アラートスクリプトのパラメーター	ユーザーのメディア設定の'送信先' の値 3.0.0 以降でサポートされています
{ALERT.SUBJECT}	→ アラートスクリプトのパラメーター	アクション設定の'デフォルトの件名' 値 3.0.0 以降でサポートされています
{DATE}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	yyyy.mm.dd 形式の現在日付
{DISCOVERY.DEVICE.IPADDRESS}	→ 検出通知とコマンド	検出されたデバイスの IP アドレス 追加されるホストに依存せず、常に使用可能
{DISCOVERY.DEVICE.DNS}	→ 検出通知とコマンド	検出されたデバイスの DNS 名 追加されるホストに依存せず、常に使用可能
{DISCOVERY.DEVICE.STATUS}	→ 検出通知とコマンド	検出されたデバイスのステータス: UP または DOWN のいずれか
{DISCOVERY.DEVICE.UPTIME}	→ 検出通知とコマンド	特定のデバイスの検出ステータスが最後に変更されてからの時間。精度は 1 秒までです。 例:1 時間 29 分 01 秒 ステータスが DOWN のデバイスの場合、これはダウンタイムの期間です。
{DISCOVERY.RULE.NAME}	→ 検出通知とコマンド	デバイスまたはサービスの有無を検出した検出ルール名
{DISCOVERY.SERVICE.NAME}	→ 検出通知とコマンド	検出されたサービス名 例:HTTP
{DISCOVERY.SERVICE.PORT}	→ 検出通知とコマンド	検出されたサービスポート 例:80
{DISCOVERY.SERVICE.STATUS}	→ 検出通知とコマンド	検出された service:// のステータスは UP または DOWN のいずれかです。 {DISCOVERY.SERVICE.UPTIME} → ディスカバリ通知とコマンド 特定のサービスのディスカバリステータスが最後に変更されてからの時間。精度は 1 秒まで。 例: 1 時間 29 分 01 秒。 ステータスが DOWN のサービスの場合、これはダウンタイムの期間です。 {ESC.HISTORY} → トリガーベースの通知とコマンド → 問題の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → 内部通知 エスカレーション履歴。以前に送信されたメッセージのログ。 以前に送信された通知、送信されたエスカレーションステップ、およびそのステータス (送信済み、進行中、または失敗 *) を表示します。

マクロ	サポート対象	説明
{EVENT.ACK.STATUS}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → 手動イベントアクションスクリプト	イベント確認ステータス (はい/いいえ)
{EVENT.AGE}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス回復の通知とコマンド → 検出通知およびコマンド → 自動登録の通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	アクショントリガーのイベント経過時間。精度は 1 秒まで。 エスカレーションされたメッセージで役立ちます
{EVENT.DATE}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス回復の通知とコマンド → 検出通知およびコマンド → 自動登録の通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	アクショントリガーのイベント日付
{EVENT.DURATION}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	イベントの継続時間 (障害イベントと回復イベントの時間差)。精度は 1 秒まで。 障害回復メッセージで役立ちます。 5.0.0 以降でサポートされています。
{EVENT.ID}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知 → トリガー URL → 手動イベントアクションスクリプト	アクショントリガーイベントの ID(数値)
{EVENT.NAME}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 内部通知 < br>→ 手動イベントアクションスクリプト	アクショントリガーの障害イベント名 4.0.0 以降でサポートされています
{EVENT.NSEVERITY}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス回復の通知とコマンド → 手動イベントアクションスクリプト	イベント深刻度の数値使用可能な値: 0 - 未分類, 1 - 情報, 2 - 警告, 3 - 軽度の障害, 4 - 重度の障害, 5 - 致命的な障害 4.0.0 以降でサポートされています
{EVENT.OBJECT}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	イベントオブジェクトの数値使用可能な値: 0 - トリガー, 1 - ホストの検出, 2 - サービスの検出, 3 - 自動登録, 4 - アイテム, 5 - ローレベルディスカバリルール 4.4.0 以降でサポートされています

マクロ	サポート対象	説明
{EVENT.OPDATA}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → 手動イベントアクションスクリプト	障害の根本的なトリガーの操作データ 4.4.0以降でサポートされています
{EVENT.RECOVERY.DAT}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントの日付
{EVENT.RECOVERY.ID}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントの数値 ID
{EVENT.RECOVERY.NAME}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントの名前 4.4.1以降でサポートされています
{EVENT.RECOVERY.STAT}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントのステータステキスト
{EVENT.RECOVERY.TAG}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントタグのコンマ区切りリスト。タグが存在しない場合は空の文字列に展開されます。 3.2.0以降でサポートされています。
{EVENT.RECOVERY.TAGS}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	イベントタグを含む JSON 配列 objects 。タグが存在しない場合は空の配列に展開されます。 5.0.0以降でサポートされています。
{EVENT.RECOVERY.TIME}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントの時間
{EVENT.RECOVERY.VALUE}	障害復旧通知とコマンド → 障害更新の通知とコマンド (復旧が行われた場合) → サービス回復通知とコマンド → 手動イベントアクションスクリプト (復旧が行われた場合)	リカバリイベントの数値
{EVENT.SEVERITY}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 手動イベントアクションスクリプト	イベント深刻度の名前 4.0.0以降でサポートされています
{EVENT.SOURCE}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	イベントソースの数値。使用可能な値: 0 - トリガー、1 - 検出、2 - 自動登録、3 - 内部 4.4.0以降でサポートされています。

マクロ	サポート対象	説明
{EVENT.STATUS}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	アクションをトリガーしたイベントの言語値
{EVENT.TAGS}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 手動イベントアクションスクリプト	イベントタグのコンマ区切りリストタグが存在しない場合は空の文字列に展開されます。 3.2.0以降でサポートされています。
{EVENT.TAGSJSON}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 手動イベントアクションスクリプト	イベントタグを含む JSON 配列オブジェクト タグが存在しない場合空の配列に展開されます 5.0.0以降でサポートされています
{EVENT.TAGS.<tag name>}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → Webhook メディアタイプの URL 名と URL → 手動イベントアクションスクリプト	タグ名で参照されるイベントタグ値 英数字以外の文字 (英語以外のマルチバイト UTF 文字を含む) を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。 4.4.2以降でサポートされています。
{EVENT.TIME}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 検出通知とコマンド → 自動登録通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	アクションをトリガーしたイベントの時刻
{EVENT.UPDATE.ACTION}	→ 障害の更新通知とコマンド	障害の更新 中に実行される アクションの人間が判読できる名前 次の値に解決されます: 確認済み、コメント済み、(元の深刻度) から (更新された深刻度) への深刻度の変更、および クローズド (1 回の更新で実行されるアクションの数による)。 4.0.0以降でサポートされています
{EVENT.UPDATE.DATE}	→ 障害の更新通知とコマンド → サービス更新通知とコマンド	イベントの日付 更新 (確認など) 非推奨の名前:{ACK.DATE}
{EVENT.UPDATE.HISTORY}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → 手動イベントアクションスクリプト	障害の更新のログ (確認など) 非推奨の名前:{EVENT.ACK.HISTORY}
{EVENT.UPDATE.MESSAGE}	→ 障害の更新通知とコマンド	障害の更新メッセージ 非推奨の名前:{ACK.MESSAGE}
{EVENT.UPDATE.STATUS}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → 手動イベントアクションスクリプト	障害の更新ステータスの数値使用可能な値: 0 - 障害/回復イベントのために Webhook が呼び出されました。1 - 更新操作 4.4.0以降でサポートされています。
{EVENT.UPDATE.TIME}	→ 障害の更新の通知とコマンド → サービスの更新の通知とコマンド	イベントの時間 更新 (確認など) 非推奨の名前:{ACK.TIME}
{EVENT.VALUE}	→ トリガーベースの通知とコマンド → 障害の更新通知とコマンド → サービスベースの通知とコマンド → サービス更新の通知とコマンド → サービス復旧の通知とコマンド → 内部通知 → 手動イベントアクションスクリプト	アクションをトリガーしたイベントの数値 (障害の場合は 1、復旧の場合は 0)

マクロ	サポート対象	説明
{FUNCTION.VALUE<1-9>}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害の更新通知とコマンド → 手動イベントアクションスクリプト → イベント名 	<p>N 番目の項目の結果イベント時のトリガー式に基づく関数</p> <p>最初のパラメーターとして /host/key を持つ関数のみがカウントされます。インデックス付きマクロを参照してください。</p>
{FUNCTION.RECOVER<1-9>}	<ul style="list-style-type: none"> → 障害復旧通知とコマンド → 障害更新通知とコマンド → 手動イベントアクションスクリプト 	<p>イベント時のリカバリー式 N 番目の項目ベースの関数結果</p> <p>最初のパラメーターとして /host/key を持つ関数のみがカウントされます。インデックス付きマクロを参照してください。</p>
{HOST.CONN}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → マップエレメントラベル, URL の名前と値をマップ → アイテムの主要なパラメーター¹ → ホストインターフェイスの IP/DNS → トラッパーアイテムの"許可されたホスト"フィールド → データベース監視の追加パラメーター → SSH および Telnet スクリプト
→ JMX アイテムエンドポイントフィールド → Web モニタリング⁴ → ローレベルディスカバリールールフィルターの正規表現 → 動的 URL ダッシュボードウィジェットの URL フィールド → トリガー名、イベント名、運用データと説明 → トリガー URL → タグ名と値 → スクリプト型アイテム、アイテムプロトタイプ、検出ルールパラメータ名前と値 → HTTP エージェントタイプアイテム、アイテムプロトタイプ、検出ルールフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーファイル、許可されたホスト。 → マニュアルホストアクションスクリプト (確認テキストを含む) → 手動イベントアクションスクリプト (確認テキストを含む) → 項目値ウィジェットの説明 	<p>ホストの設定に応じて、ホストの IP アドレスまたは DNS 名².</p> <p>{HOST.CONN<1-9>} のように数値インデックスとともに使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロを参照してください。</p>
{HOST.DESCRPTION}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → マップエレメントラベル → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ホストの説明.</p> <p>このマクロは、数値インデックスと一緒に使用できます。{HOST.DESCRPTION<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>

マクロ	サポート対象	説明
{HOST.DNS}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → マップエレメントラベル, URL の名前と値をマップ → アイテムの主要なパラメーター¹ → ホストインターフェイスの IP/DNS → トラッパアイテムの"許可されたホスト" フィールド → データベース監視の追加パラメーター → SSH および Telnet スクリプト
→ JMX アイテムエンドポイントフィールド → Web モニタリング⁴ → ローレベルディスカバリルールフィルターの正規表現 → 動的 URL ダッシュボードウィジェットの URL フィールド → トリガー名、イベント名、運用データと説明 → トリガー URL → タグ名と値 → スクリプト型アイテム、アイテムプロトタイプ、検出ルールパラメータ名前と値 → HTTP エージェントタイプアイテム、アイテムプロトタイプ、検出ルールフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーファイル、許可されたホスト。 → マニュアルホストアクションスクリプト (確認テキストを含む) → 手動イベントアクションスクリプト (確認テキストを含む) → 項目値ウィジェットの説明 	<p>ホスト DNS 名².</p> <p>このマクロは、数値で使用できます。インデックス {HOST.DNS<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>

マクロ	サポート対象	説明
{HOST.HOST}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 自動登録の通知とコマンド → 内部通知 → アイテムキーパラメータ → マップ要素ラベル、URL の名前と値をマップ → ホストインターフェース IP/DNS → トラッパアイテム”許可されたホスト”フィールド → データベース監視追加パラメーター → SSH および Telnet スクリプト → JMX アイテムエンドポイントフィールド → Web モニタリング⁴ → ローレベルディスカバリルールフィルターの正規表現 → 動的 URL ダッシュボードウィジェットの URL フィールド → トリガー名、イベント名、運用データと説明 → トリガー URL → タグ名と値 → スクリプトタイプアイテム、アイテムプロトタイプ、検出ルールパラメータの名前と値 → HTTP エージェントタイプアイテム、アイテムプロトタイプ、検出ルールフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーファイル、許可されたホスト。 → 手動ホストアクションスクリプト (確認テキストを含む) → 手動イベントアクションスクリプト (確認テキストを含む) 	<p>ホスト名.</p> <p>このマクロは数値インデックスと共に使用できません。{HOST.HOST<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p> <p>{HOSTNAME<1-9>} は非推奨です。</p>
{HOST.ID}	<ul style="list-style-type: none"> → 項目値ウィジェットの説明 → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → マップエレメントラベル、URL の名前と値をマップ → 動的 URL ダッシュボードウィジェットの URL フィールド → トリガー URL → タグ名と値 → 手動イベントアクションスクリプト → 項目値ウィジェットの説明 	<p>ホスト ID.</p> <p>{HOST.ID<1-9>} のように数値インデックスで使用できません。トリガー式で最初、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>

マクロ	サポート対象	説明
{HOST.IP}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 自動登録の通知とコマンド → 内部通知 → マップエレメントラベル、URL の名前と値をマップ → アイテムキーパラメータ¹ → ホストインターフェイス IP/DNS → 「トラッパアイテム」許可されたホスト」フィールド → データベース監視の追加パラメーター → SSH および Telnet スクリプト → JMX アイテムエンドポイントフィールド → Web 監視⁴ → Low-レベル検出ルールフィルタの正規表現 → 動的 URL ダッシュボードウィジェットの URL フィールド → トリガー名、イベント名、運用データと説明 → トリガー URL → タグ名と値 → スクリプトタイプのアイテム、アイテムのプロトタイプ、検出ルールのパラメータ名と値 → HTTP エージェントタイプのアイテム、アイテムのプロトタイプ、検出ルールのフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーファイル、許可されたホスト → 手動ホストアクションスクリプト (確認テキストを含む) → 手動イベントアクションスクリプト (確認文含む) → 項目値の説明 widget 	<p>ホスト IP アドレス².
 このマクロは、数値インデックスと一緒に使用できます。{HOST.IP<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p> <p>{IPADDRESS<1-9>} は非推奨です。</p>
{HOST.METADATA}	<ul style="list-style-type: none"> → 自動登録の通知とコマンド 	<p>ホストメタデータ。 アクティブなエージェントの自動登録にのみ使用されます。 表示可能なホスト名。</p>
{HOST.NAME}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → マップエレメントラベル、URL の名前と値をマップ → 項目キーパラメータ → ホストインターフェイス IP/DNS → 「トラッパ項目」許可されたホスト」フィールド → データベース監視追加パラメータ → SSH および Telnet スクリプト → Web 監視⁴ → 低レベル検出ルールフィルタの正規表現 → 動的 URL ダッシュボードウィジェットの URL フィールド → トリガー名、イベント名、運用データおよび説明 → トリガー URL
→ タグの名前と値 → スクリプトタイプのアイテム、アイテムのプロトタイプ、および検出ルールのパラメータ名と値 → HTTP エージェントタイプのアイテム、アイテムのプロトタイプ、および検出ルールのフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーファイル、許可されたホスト。 → 手動ホストアクションスクリプト (確認テキストを含む) → 手動イベントアクションスクリプト (確認テキストを含む) → 説明アイテム値ウィジェットの設定 	<p>このマクロは、数値インデックスと一緒に使用できます。{HOST.NAME<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>

マクロ	サポート対象	説明
{HOST.PORT}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 自動登録の通知とコマンド → 内部通知 → トリガー名、イベント名、運用データと説明 → トリガー URL → JMX アイテムエンドポイントフィールド → タグ名と値 → 手動イベントアクションスクリプト → 項目値ウィジェットの説明 	<p>ホスト (エージェント) ポート²。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{HOST.PORT<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{HOST.TARGET.CONN}	<ul style="list-style-type: none"> → トリガーベースのコマンド → 障害の更新コマンド → ディスカバリコマンド → 自動登録コマンド 	<p>ホストの設定に応じて、ターゲットホストの IP アドレスまたは DNS 名。</p> <p>5.4.0 以降でサポートされています。</p>
{HOST.TARGET.DNS}	<ul style="list-style-type: none"> → トリガーベースのコマンド → 障害の更新コマンド → ディスカバリコマンド → 自動登録コマンド 	<p>ターゲットホストの DNS 名。</p> <p>5.4.0 以降でサポートされています。</p>
{HOST.TARGET.HOST}	<ul style="list-style-type: none"> → トリガーベースのコマンド → 障害の更新コマンド → ディスカバリコマンド → 自動登録コマンド 	<p>ターゲットホストの技術名。</p> <p>5.4.0 以降でサポートされています。</p>
{HOST.TARGET.IP}	<ul style="list-style-type: none"> → トリガーベースのコマンド → 障害の更新コマンド → ディスカバリコマンド → 自動登録コマンド 	<p>ターゲットホストの IP アドレス。</p> <p>5.4.0 以降でサポートされています。</p>
{HOST.TARGET.NAME}	<ul style="list-style-type: none"> → トリガーベースのコマンド → 障害の更新コマンド → ディスカバリコマンド → 自動登録コマンド 	<p>ターゲットホストの表示名。</p> <p>5.4.0 以降でサポートされています。</p>
{HOSTGROUP.ID}	<ul style="list-style-type: none"> → マップ要素のラベル, URL の名前と値をマップ 	<p>ホストグループ ID</p>
{INVENTORY.ALIAS}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ホストインベントリのエイリアスフィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.ALIAS<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{INVENTORY.ASSET.TAG}	<ul style="list-style-type: none"> → 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ホストインベントリのアセットタグフィールド。</p> <p>このマクロは、数値インデックスなどで使用できます。{INVENTORY.ASSET.TAG<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{INVENTORY.CHASSIS}	<ul style="list-style-type: none"> → 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ホストインベントリのシャーシフィールド。</p> <p>このマクロは、数値インデックスなどで使用できます。{INVENTORY.CHASSIS<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{INVENTORY.CONTACT}	<ul style="list-style-type: none"> → 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ホストインベントリの連絡先フィールド。</p> <p>このマクロは、数値インデックスなどで使用できます。{INVENTORY.CONTACT<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。indexed macros を参照してください。</p>
		<p>{PROFILE.CONTACT<1-9>} は非推奨です。</p>

マクロ	サポート対象	説明
{INVENTORY.CONTRACT.NUMBER}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのコントラクト番号フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.CONTRACT.NUMBER<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.DEPLOYMENT.STATUS}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの展開ステータスフィールド。 このマクロは、数値インデックスなどで使用できます。{INVENTORY.DEPLOYMENT.STATUS<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HARDWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのハードウェアフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HARDWARE<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。indexed macros. を参照してください。
{INVENTORY.HARDWARE.FULL}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.HARDWARE<1-9>} は非推奨です。 ホストインベントリのハードウェア (完全な詳細) フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HARDWARE.FULL<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HOST.NETMASK}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのホストサブネットマスクフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HOST.NETMASK<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HOST.NETWORKS}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのホストネットワークフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HOST.NETWORKS<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HOST.ROUTER}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのホストルータフィールド。 このマクロは、数値インデックスなどで使用できます。{INVENTORY.HOST.ROUTER<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HW.ARCH}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのハードウェアアーキテクチャフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HW.ARCH<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{INVENTORY.HW.DATE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのハードウェア廃止日フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HW.DATE.DECOMM<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HW.DATE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのハードウェアメンテナンスの有効期限フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HW.DATE.EXPIRY<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HW.DATE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのハードウェアインストールフィールドの日付。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HW.DATE.INSTALL<1-9>} は、トリガー式で最初、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.HW.DATE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのハードウェア購入日フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.HW.DATE.PURCHASE<1-9>} を使用して、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.INSTALLER}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのインストーラ名フィールド。 このマクロは、数値インデックスとともに使用できます。{INVENTORY.INSTALLER.NAME<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.LOCATION}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのロケーションフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.LOCATION<1-9>} を使用して、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。indexed macros. を参照してください。
{INVENTORY.LOCATION}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.LOCATION<1-9>} は非推奨です。 ホストインベントリの場所の緯度フィールド。 このマクロは、数値インデックスなどで使用できます。{INVENTORY.LOCATION.LAT<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.LOCATION}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの場所の経度フィールド。 このマクロは数値インデックスと一緒に使用できません。{INVENTORY.LOCATION.LON<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{INVENTORY.MACADDRESS.A}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	MAC アドレスホストインベントリのフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.MACADDRESS.A<1-9>} は、トリガー式で最初、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.MACADDRESS.B}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.MACADDRESS<1-9>} は非推奨です。 ホストインベントリの MAC アドレス B フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.MACADDRESS.B<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.MODEL}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのモデルフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.MODEL<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.NAME}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの名前フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.NAME<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.NOTES}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.NAME<1-9>} は非推奨です。 ホストインベントリのメモフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.NOTES<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.OOB.IP}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.NOTES<1-9>} は非推奨です。 ホストインベントリの OOB IP アドレスフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.OOB.IP<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.OOB.NETMASK}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの OOB サブネットマスクフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.OOB.NETMASK<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.OOB.ROUTER}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの OOB ルーターフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.OOB.ROUTER<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{INVENTORY.OS}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの OS フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.OS<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.OS.FULL}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.OS<1-9>} は非推奨です。 ホストインベントリの OS (完全な詳細) フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.OS.FULL<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.OS.SHORT}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの OS (Short) フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.OS.SHORT<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.POC.PRIMARY.CELL}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのプライマリ POC セルフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.CELL<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.POC.PRIMARY.EMAIL}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのプライマリ POC メールフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.EMAIL<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.POC.PRIMARY.NAME}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのプライマリ POC 名フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.NAME<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.POC.PRIMARY.NOTES}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのプライマリ POC メモフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.NOTES<1-9>} を使用して、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.POC.PRIMARY.PHONE.A}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのプライマリ POC phone A フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.PHONE.A<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{INVENTORY.POC.PRIMARY.PHONE.B}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのプライマリ POC phone B フィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.PHONE.B<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{INVENTORY.POC.PRIMARY.SCREEN}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのプライマリ POC スクリーン名フィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.PRIMARY.SCREEN<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{INVENTORY.POC.SECONDARY.CELL}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのセカンダリ POC セルフィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.SECONDARY.CELL<1-9>} を使用して、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{INVENTORY.POC.SECONDARY.EMAIL}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのセカンダリ POC メールフィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.SECONDARY.EMAIL<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{INVENTORY.POC.SECONDARY.NAME}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのセカンダリ POC 名フィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.SECONDARY.NAME<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{INVENTORY.POC.SECONDARY.NOTES}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのセカンダリ POC メモフィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.SECONDARY.NOTES<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{INVENTORY.POC.SECONDARY.PHONE.A}	<p>内部通知</p> <p>→ タグ名と値</p> <p>→ マップ要素のラベル, URL の名前と値をマップ</p> <p>→ 手動イベントアクションスクリプト</p> <p>→ アイテム値ウィジェットの説明</p>	<p>ホストインベントリのセカンダリ POC phone A フィールド。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.POC.SECONDARY.PHONE.A<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>

マクロ	サポート対象	説明
{INVENTORY.POC.SECONDARY.PHONE.B}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのセカンダリ POC phone B フィールド。 このマクロは、数値インデックスと一緒に使用できません。 {INVENTORY.POC.SECONDARY.PHONE.B<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.POC.SECONDARY.SCREEN}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのセカンダリ POC スクリーン名フィールド。 このマクロは、数値インデックスと一緒に使用できません。 {INVENTORY.POC.SECONDARY.SCREEN<1-9>} を使用して、トリガー式の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SERIALNO.A}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのシリアル番号 A フィールド。 このマクロは、数値インデックスと一緒に使用できません。{INVENTORY.SERIALNO.A<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SERIALNO.B}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.SERIALNO<1-9>} は非推奨です。 ホストインベントリのシリアル番号 B フィールド。 このマクロは、数値インデックスと一緒に使用できません。{INVENTORY.SERIALNO.B<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.ADDRESS.A}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイトアドレス A フィールド。 このマクロは、数値インデックスと一緒に使用できません。{INVENTORY.SITE.ADDRESS.A<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.ADDRESS.B}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイトアドレス B フィールド。 このマクロは、数値インデックスと一緒に使用できません。{INVENTORY.SITE.ADDRESS.B<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.ADDRESS.C}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイトアドレス C フィールド。 このマクロは、数値インデックスと一緒に使用できません。{INVENTORY.SITE.ADDRESS.C<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{INVENTORY.SITE.CITY}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイト都市フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SITE.CITY<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.COUNTRY}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイト国フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SITE.COUNTRY<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.NOTES}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイトノートフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SITE.NOTES<1-9>} を使用して、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.RACK}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイトロケーションフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SITE.RACK<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.STATE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイト都道府県フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SITE.STATE<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SITE.ZIP}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのサイト郵便番号フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SITE.ZIP<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SOFTWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのソフトウェアフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。indexed macros. を参照してください。
{INVENTORY.SOFTWARE.APP.A}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.SOFTWARE<1-9>} は非推奨です。 ホストインベントリのソフトウェアアプリケーション A フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE.APP.A<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{INVENTORY.SOFTWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのソフトウェアアプリケーション B フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE.APP.B<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SOFTWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのソフトウェアアプリケーション C フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE.APP.C<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SOFTWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのソフトウェアアプリケーション D フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE.APP.D<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SOFTWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのソフトウェアアプリケーション E フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE.APP.E<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.SOFTWARE}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのソフトウェア (完全な詳細) フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.SOFTWARE.FULL<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.TAG}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのタグフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.TAG<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.TYPE}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	{PROFILE.TAG<1-9>} は非推奨です。 ホストインベントリのタイプフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.TYPE<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。 {PROFILE.DEVICETYPE<1-9>} は非推奨です。

マクロ	サポート対象	説明
{INVENTORY.TYPE.FULL}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのタイプ (完全な詳細) フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.TYPE.FULL<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.URL.A}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの URL A フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.URL.A<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.URL.B}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの URL B フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.URL.B<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.URL.C}	→ 内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリの URL C フィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.URL.C<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{INVENTORY.VENDOR}	内部通知 → タグ名と値 → マップ要素のラベル, URL の名前と値をマップ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	ホストインベントリのベンダーフィールド。 このマクロは、数値インデックスと一緒に使用できます。{INVENTORY.VENDOR<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{ITEM.DESCRPTION}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	通知を引き起こしたトリガー式の N 番目の項目の説明。 このマクロは、数値インデックスと一緒に使用できます。{ITEM.DESCRPTION<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{ITEM.DESCRPTION.ORIG}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	通知の原因となったトリガー式の N 番目の項目の説明 (未解決のマクロを含む)。 このマクロは、数値インデックス (例: {ITEM.DESCRPTION.ORIG<1-9>}) は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。
{ITEM.ID}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → スクリプトタイプのアイテム、アイテムのプロトタイプ、および検出ルールのパラメーター名と値 ⁶ → HTTP エージェントタイプのアイテム、アイテムのプロトタイプ、および検出ルールのフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーフ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	5.2.0 以降でサポートされています。 通知を引き起こしたトリガー式の N 番目の項目の数値 ID。 このマクロは、数値インデックスと共に使用できます。{ITEM.ID<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

マクロ	サポート対象	説明
{ITEM.KEY}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → スクリプトタイプのアイテム、アイテムのプロトタイプ、および検出ルールのパラメーター名と値⁶ → HTTP エージェントタイプのアイテム、アイテムのプロトタイプ、および検出ルールのフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーフファイル → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>通知を引き起こしたトリガー式の N 番目の項目のキー。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.KEY<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p> <p>{TRIGGER.KEY} は非推奨です。</p>
{ITEM.KEY.ORIG}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → スクリプトタイプのアイテム、アイテムのプロトタイプ、および検出ルールのパラメーター名と値⁶ → HTTP エージェントタイプのアイテム、アイテムのプロトタイプ、および検出ルールのフィールド: URL、クエリフィールド、リクエストボディ、ヘッダー、プロキシ、SSL 証明書ファイル、SSL キーフファイル、許可されたホスト。⁶ → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>通知の原因となったトリガー式の N 番目の項目の元のキー (展開されていないマクロを含む)⁴。</p> <p>このマクロは、数値インデックスなどで使用できます。{ITEM.KEY.ORIG<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{ITEM.LASTVALUE}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、イベント名、運用データと説明 → タグ名と値 → トリガー URL → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>通知の原因となったトリガー式の N 番目の項目の最新の値。</p> <p>最新の履歴値が最大履歴表示期間を超えて収集された場合、フロントエンドで *UNKNOWN* に解決されます。(管理 → 全般 メニューセクションで設定)。問題イベントを表示するときの最新のアイテム値ではなく、問題が発生した時点からのアイテム値を保持します。</p> <p>last(/{HOST.HOST}/{ITEM.KEY}) のエイリアスです。</p> <p>解決された値は、トリガー URL などで使用できるように 20 文字に切り捨てられます。完全な値に解決するには、マクロ関数を使用できます。</p> <p>カスタマイズ マクロ値はこのマクロでサポートされています。Zabbix 3.2.0 以降。</p> <p>このマクロは、数値インデックスで使用できます。{ITEM.LASTVALUE<1-9>} は、トリガー式の最初、2 番目、3 番目などの項目を指します。インデックス付きマクロを参照してください。</p>
{ITEM.LOG.AGE}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ログアイテムイベントの経過時間。精度は 1 秒までです。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.LOG.AGE<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>
{ITEM.LOG.DATE}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ログアイテムイベントの日付。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.LOG.DATE<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。</p>

マクロ	サポート対象	説明
{ITEM.LOG.EVENTID}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>イベントログ内のイベントの ID。 Windows イベントログの監視のみ。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.LOG.EVENTID<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.LOG.NSEVERITY}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>イベントログ内のイベントの深刻度の数値。 Windows イベントログの監視のみ。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.LOG.NSEVERITY<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.LOG.SEVERITY}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>イベントログ内のイベントの口頭による重大度。 Windows イベントログ監視のみ。</p> <p>{ITEM.LOG.SEVERITY<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.LOG.SOURCE}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>イベントログ内のイベントのソース。 Windows イベントログの監視のみ。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.LOG.SOURCE<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.LOG.TIME}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、運用データ、説明 → トリガー URL → イベントタグと値 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>ログアイテムイベントの時間。</p> <p>このマクロは、数値インデックスなどで使用できません。{ITEM.LOG.TIME<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.NAME}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>通知を引き起こしたトリガー式の N 番目の項目の名前。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.NAME<1-9>} は、トリガー式で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.NAME.ORIG}	<ul style="list-style-type: none"> → トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明 	<p>このマクロは、Zabbix 6.0 以降では非推奨です。ユーザーマクロと位置マクロがアイテム名でサポートされていた 6.0 より前のバージョンの Zabbix では、アイテムの元の名前（つまり、マクロが解決されていない）に解決されていました。</p> <p>このマクロは、数値インデックス。 {ITEM.NAME.ORIG<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>
{ITEM.STATE}	<ul style="list-style-type: none"> → アイテムベースの内部通知 → アイテム値ウィジェットの説明 	<p>通知の原因となったトリガー式の N 番目の項目の最新状態。使用可能な値: Not supported および Normal。</p> <p>このマクロは、数値インデックスと一緒に使用できます。{ITEM.STATE<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。</p>

マクロ	サポート対象	説明
{ITEM.STATE.ERROR}	アイテムベースの内部通知	アイテムがサポートされなくなった理由の詳細を含むエラーメッセージ。 アイテムがサポートされていない状態になり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。 次のいずれかに解決されます: 1) トリガー式の N 番目のアイテムの履歴 (イベント時の) 値。たとえば、イベントの表示や送信など、トリガーステータス変更のコンテキストで使用される場合 notifications。 2) トリガー式の N 番目の項目の最新の値。たとえば、ポップアップ選択ウィンドウにトリガーのリストを表示するときなど、トリガーステータス変更のコンテキストなしで使用された場合。この場合、{ITEM.LASTVALUE} と同じように機能します。 最初のケースでは、履歴値が既に削除されているか、保存されていない場合、*UNKNOWN* に解決されます。 2 番目のケースでは、また、フロントエンドのみで、最新の履歴値が最大履歴表示期間よりも前に収集された場合 (管理 → 一般 に解決されます。 /administration/general#gui) menu section)。 解決された値は、トリガー URL などで使用できるように 20 文字に切り捨てられます。完全な値に解決するには、マクロ関数を使用できます。
{ITEM.VALUE}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー名、イベント名、運用データと説明 → タグ名と値 → トリガー URL → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	カスタマイズ マクロ値は、Zabbix 3.2.0 以降、このマクロでサポートされています。 このマクロは、数値インデックスなどで使用できます。{ITEM.VALUE<1-9>} は、トリガー式の最初、2 番目、3 番目などの項目を指します。 インデックス付きマクロ を参照してください。 通知の原因となったトリガー式の N 番目の項目の値の型使用可能な値: 0 - 浮動小数点数、1 - 文字、2 - ログ、3 - 符号なし数値、4 - テキスト このマクロは数値インデックスで使用できます。例えば {ITEM.VALUE<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。 インデックス付きマクロ を参照してください。
{ITEM.VALUETYPE}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → 内部通知 → 手動イベントアクションスクリプト → アイテム値ウィジェットの説明	5.4.0 以降でサポートされています。 通知の原因となったローレベルディスカバリルールの説明 通知の原因となったローレベルディスカバリルールの説明 (未解決のマクロを含む)。 5.2.0 以降でサポートされています。 通知の原因となったローレベルディスカバリルールの数値 ID 通知の原因となったローレベルディスカバリルールのキー。 通知の原因となったローレベルディスカバリルールの元のキー (展開されていないマクロを含む) 通知の原因となったローレベルディスカバリルールの名前 (マクロが解決されたもの) 通知の原因となったローレベルディスカバリルールの元の名前 (つまり、マクロが解決されていないもの)。
{LLDRULE.DESCRPTION}	LLD ルールに基づく内部通知	
{LLDRULE.DESCRPTION.ORIG}	LLD ルールに基づく内部通知	
{LLDRULE.ID}	LLD ルールに基づく内部通知	
{LLDRULE.KEY}	LLD ルールに基づく内部通知	
{LLDRULE.KEY.ORIG}	LLD ルールに基づく内部通知	
{LLDRULE.NAME}	LLD ルールに基づく内部通知	
{LLDRULE.NAME.ORIG}	LLD ルールに基づく内部通知	

マクロ	サポート対象	説明
{LLDRULE.STATE}	→ LLD ルールに基づく 内部通知	ローレベルディスクバリアルールの最新の状態使用可能な値: Not supported と Normal .
{LLDRULE.STATE.ERROR}	→ LLD ルールに基づく 内部通知	LLD ルールがサポートされなくなった理由の詳細を含むエラーメッセージ
{MAP.ID}	→ マップ要素のラベル, URL の名前と値をマップ	LLD ルールがサポートされていない状態になり、すぐに再びサポートされるようになった場合、エラーフィールドは空になる可能性があります。
{MAP.NAME}	→ マップ要素のラベル, URL の名前と値をマップ	ネットワークマップ ID.
{PROXY.DESCRPTION}	→ マップ形状のテキストフィールド	ネットワークマップ名
{PROXY.DESCRPTION}	→ トリガーベースの通知とコマンド	3.4.0 以降でサポートされています
	→ 障害更新通知とコマンド	プロキシの説明次のいずれかに解決されます。
	→ 検出通知とコマンド	1) トリガー式の N 番目のアイテムのプロキシ (トリガーベースの通知)。ここで indexed マクロを使用できます。
	→ 自動登録の通知とコマンド	2) 検出を実行したプロキシ (検出通知)。ここではインデックスを作成せずに {PROXY.DESCRPTION} を使用します。
	→ 内部通知	3) アクティブなエージェントが登録されたプロキシ (自動登録通知)。ここでは、インデックスを作成せずに {PROXY.DESCRPTION} を使用します。
	→ 手動イベントアクションスクリプト	
{PROXY.NAME}	→ トリガーベースの通知とコマンド	このマクロは、数値インデックスと一緒に使用できます。{PROXY.DESCRPTION<1-9>} は、トリガー式で最初、2 番目、3 番目などのホストを指します。 インデックス付きマクロ を参照してください。
	→ 障害更新通知とコマンド	プロキシの名前次のいずれかに解決されます。
	→ 検出通知とコマンド	1) トリガー式の N 番目のアイテムのプロキシ (トリガーベースの通知)。ここで indexed マクロを使用できます。
	→ 自動登録の通知とコマンド	2) 検出を実行したプロキシ (検出通知)。ここでは、インデックスを作成せずに {PROXY.NAME} を使用します。
	→ 内部通知	3) アクティブなエージェントが登録されたプロキシ (自動登録通知)。ここでは、インデックスを作成せずに {PROXY.NAME} を使用します。
	→ 手動イベントアクションスクリプト	
{SERVICE.DESCRPTION}	→ サービスベースの通知とコマンド	このマクロは、数値インデックスと一緒に使用できます。{PROXY.NAME<1-9>} は、トリガー式の最初、2 番目、3 番目などのホストを指します。 インデックス付きマクロ を参照してください。
	→ サービス更新の通知とコマンド	サービスの説明 (解決されたマクロを含む)
{SERVICE.NAME}	→ サービスベースの通知とコマンド	サービスの名前 (解決されたマクロを含む)
	→ サービス更新の通知とコマンド	
{SERVICE.ROOTCAUSE}	→ サービスベースの通知とコマンド	サービスの失敗の原因となったトリガー問題イベントのリスト深刻度とホスト名でソートされています。次の詳細が含まれます: ホスト名、イベント名、深刻度、経過時間、サービスタグ、および値
	→ サービス更新の通知とコマンド	サービスイベントタグのコンマ区切りリストサービスイベントタグは、サービス構成セクションのタグで定義できます。タグが存在しない場合は空の文字列に展開されます。
{SERVICE.TAGS}	→ サービスベースの通知とコマンド	
	→ サービス更新の通知とコマンド	
{SERVICE.TAGSJSON}	→ サービスベースの通知とコマンド	サービスイベントタグオブジェクトを含む JSON 配列サービスイベントタグは、サービス構成セクションのタグで定義できます。タグが存在しない場合は、空の配列に展開されます。
	→ サービス更新の通知とコマンド	

マクロ	サポート対象	説明
{SERVICE.TAGS.<tag> name>}	サービススペースの通知とコマンド → サービス更新の通知とコマンド	タグ名によって参照されるサービスイベントタグの値。サービスイベントタグは、サービス構成セクションのタグで定義できます。 英数字以外の文字 (英語以外のマルチバイト UTF 文字を含む) を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。
{TIME}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → サービススペースの通知とコマンド → サービス更新通知とコマンド → 検出通知とコマンド → 自動登録の通知とコマンド → 内部通知 → トリガーイベント名 → 手動イベントアクションスクリプト	hh:mm:ss 形式の現在時刻
{TRIGGER.DESCRPTION}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガーの説明。 {TRIGGER.DESCRPTION} が通知テキストで使用されている場合、トリガーの説明でサポートされているすべてのマクロが展開されます。 {TRIGGER.COMMENT} は非推奨です。
{TRIGGER.EXPRESSION}<PLAIN>	トリガーベースの通知とコマンド → 障害更新通知とコマンド → 手動イベントアクションスクリプト → イベント名	部分的に評価されたトリガー式。 項目ベースの関数はイベント生成時に評価され、結果に置き換えられますが、他のすべての関数は式に記述されているとおりに表示されます。トリガー式のデバッグに使用できます。
{TRIGGER.EXPRESSION}<RECOVERY>	障害回復通知とコマンド → 障害更新通知とコマンド → 手動イベントアクションスクリプト	一部評価トリガー回復式 項目ベースの関数はイベント生成時に評価され、結果に置き換えられますが、他のすべての関数は式に記述されているとおりに表示されます。トリガー回復式のデバッグに使用できます。
{TRIGGER.EVENTS.AGK}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → マップ要素のラベル → 手動イベントアクションスクリプト	マップ内のマップ要素、または通知内の現在のイベントを生成したトリガーに対して確認されたイベントの数
{TRIGGER.EVENTS.PROBLEM.ACK}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → マップ要素のラベル → 手動イベントアクションスクリプト	状態を無視したすべてのトリガーの確認済みの障害イベントの数
{TRIGGER.EVENTS.PROBLEM.UNACK}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → マップ要素のラベル → 手動イベントアクションスクリプト	状態に関係なく、すべてのトリガーの未確認の障害イベントの数
{TRIGGER.EVENTS.UNACK}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → マップ要素のラベル → 手動イベントアクションスクリプト	マップ内のマップ要素、または通知内の現在のイベントを生成したトリガーの未確認イベントの数
{TRIGGER.HOSTGROUPNAME}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	(SQL クエリで) 並べ替えられた、トリガーが定義されているホストグループのカンマ区切りのリスト
{TRIGGER.PROBLEM.EVENTS.ACK}	障害状態の要素	障害状態のトリガーの確認済み障害イベントの数
{TRIGGER.PROBLEM.EVENTS.UNACK}	障害状態の要素	障害状態のトリガーの未確認の障害イベントの数
{TRIGGER.EXPRESSION}	トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガー式
{TRIGGER.EXPRESSION}<RECOVERY>	トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガー構成の OK イベント生成が '回復式' に設定されている場合はトリガー回復式 *; それ以外の場合は、空の文字列が返されます。 3.2.0 以降でサポートされています。

マクロ	サポート対象	説明
{TRIGGER.ID}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → マップ要素のラベル, URL の名前と値をマップ → トリガー URL → トリガータグ値 → 手動イベントアクションスクリプト	このアクションをトリガーしたトリガー ID 値 4.4.1 以降、トリガータグの値でサポートされています。
{TRIGGER.NAME}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガーの名前 (マクロが解決されている) 4.0.0 以降、{EVENT.NAME} をアクションで使用して、トリガーされたイベント/問題の名前をマクロが解決された状態で表示できることに注意してください。
{TRIGGER.NAME.ORIG}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガーの元の名前 (つまり、マクロが解決されていない)
{TRIGGER.NSEVERITY}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガー深刻度の数値使用可能な値: 0 - 未設定, 1 - 情報, 2 - 警告, 3 - 軽度の障害, 4 - 重度の障害, 5 - 致命的な障害
{TRIGGER.SEVERITY}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガー深刻度名 管理 → 一般設定 → イベントのステータスで定義できます。
{TRIGGER.STATE}	→ トリガーベースの内部通知	トリガーの最新の状態使用可能な値: Unknown および Normal .
{TRIGGER.STATE.ERROR}	→ トリガーベースの内部通知	トリガーがサポートされなくなった理由の詳細を含むエラーメッセージ。
{TRIGGER.STATUS}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → 手動イベントアクションスクリプト	トリガーがサポートされていない状態になり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。 操作ステップ実行時のトリガー値 PROBLEM または OK のいずれかになります。 {STATUS} は非推奨です。
{TRIGGER.TEMPLATE_NAME}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	* トリガーが定義されているテンプレートの (SQL クエリによって) 並べ替えられたコンマ区切りのリスト。トリガーがホストで定義されている場合は *UNKNOWN*
{TRIGGER.URL}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガーベースの内部通知 → 手動イベントアクションスクリプト	トリガー URL
{TRIGGER.VALUE}	→ トリガーベースの通知とコマンド → 障害更新通知とコマンド → トリガー式 → 手動イベントアクションスクリプト	現在のトリガー状態の数値: 0 - トリガーは OK 状態 1 - トリガーは障害状態
{TRIGGERS.UNACK}	→ マップ要素のラベル	マップ要素の未承認トリガー数。トリガーの状態は無視されます。 障害イベントの少なくとも 1 つが未承認の場合、トリガーは未承認と見なされます。
{TRIGGERS.PROBLEM_UNACK}	→ マップ要素のラベル	マップ要素の未承認障害トリガー数 障害イベントの少なくとも 1 つが未承認の場合、トリガーは未承認であると見なされます。
{TRIGGERS.ACK}	→ マップ要素のラベル	トリガーの状態を無視した、マップ要素の承認されたトリガー数
{TRIGGERS.PROBLEM_ACK}	→ マップ要素のラベル	すべての障害イベントが承認された場合、トリガーは承認されたと見なされます。 マップ要素の承認された障害トリガー数 すべての障害イベントが承認された場合、トリガーは承認されたと見なされます。

マクロ	サポート対象	説明
{USER.FULLNAME}	→ 障害更新通知とコマンド → 手動ホストアクションスクリプト (including confirmation text) → 手動イベントアクションスクリプト (including confirmation text)	イベント確認を追加した、またはスクリプトを開始したユーザーの名前、姓、およびユーザー名 3.4.0 以降の障害更新、5.0.2 以降のグローバルスクリプトでサポート
{USER.NAME}	→ 手動ホストアクションスクリプト (including confirmation text) → 手動イベントアクションスクリプト (including confirmation text)	スクリプトを開始したユーザー名 5.0.2 以降でサポートされています。
{USER.SURNAME}	→ 手動ホストアクションスクリプト (including confirmation text) → 手動イベントアクションスクリプト (including confirmation text)	スクリプトを開始したユーザーの姓。 5.0.2 以降でサポートされています
{USER.USERNAME}	→ 手動ホストアクションスクリプト (including confirmation text) → 手動イベントアクションスクリプト (including confirmation text)	スクリプトを開始したユーザーの名。 5.0.2 以降でサポート。 Zabbix 5.4.0 より前にサポートされていた {USER.ALIAS} は非推奨になりました。
{\$MACRO}	→ 参照: 場所によってサポートされているユーザーマクロ	ユーザー定義可能 マクロ。
{#MACRO}	→ 参照: 場所によってサポートされているユーザーマクロ	ローレベルディスカバリマクロ カスタマイズ Zabbix 4.0.0 以降、このマクロでマクロ値がサポートされています。 参照:式マクロ。
{?EXPRESSION}	→ トリガーイベント名 → トリガーベースの通知とコマンド → 障害更新通知とコマンド → マップ要素のラベル ³ → マップシェープラベル ³ → マップ内のリンクラベル ³ → グラフ名 ⁵	5.2.0 以降でサポートされています

脚注

¹ アイテムキーパラメータでサポートされている {HOST.*} マクロは、アイテム用に選択されたインターフェースに解決されます。インターフェースのないアイテムで使用すると、ホストの Zabbix エージェント、SNMP、JMX、または IPMI インターフェイスのいずれかに、この優先順位で解決されます。ホストにインターフェースがない場合は「不明」と解決されます。

² グローバルスクリプト、インターフェース IP/DNS フィールド、および Web シナリオでは、マクロはメインエージェントインターフェースに解決されますが、存在しない場合は、メイン SNMP インターフェースが使用されます。SNMP も存在しない場合は、メインの JMX インターフェースが使用されます。JMX も存在しない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースがない場合、マクロは「不明」と解決されます。

³ マップラベルのこのマクロでは、パラメータとして秒を使用する ** avg 、 last 、 max 、 および min** 関数のみがサポートされています。

⁴ {HOST.*} マクロは Web シナリオ変数、ヘッダー、* SSL 証明書ファイル、SSL キーフイルドフィールド、およびシナリオステップ URL、Post、ヘッダー、必須の文字列フィールドでサポートされています。Zabbix 5.4.0 以降 {HOST.*} マクロは Web シナリオ名前および Web シナリオステップ名前 * フィールドでサポートされなくなりました。

⁵ グラフ名のこのマクロでは、パラメーターとして秒を使用する ** avg 、 last 、 max 、 および min** 関数のみがサポートされます。{HOST.HOST<1-9>} マクロは、マクロ内のホストとして使用できます。例：

```
* last(/Cisco switch/ifAlias[#{SNMPINDEX}])
* last(/{HOST.HOST}/ifAlias[#{SNMPINDEX}])
```

⁶ 5.2.5 以降でサポートされています。

インデックス付きマクロ

{MACRO <1-9>} のインデックス付きマクロ構文はトリガー式のコンテキストでのみ機能します。式に表示される順序でホストまたは関数を参照するために使用できます。{HOST.IP1},{HOST.IP2},{HOST.IP3} などのマクロは、トリガー式の 1 番目、2 番目、および 3 番目のホストの IP に解決されます (トリガー式にこれらのホストが含まれている場合) {FUNCTION.VALUE1},{FUNCTION.VALUE2},{FUNCTION.VALUE3} などのマクロは、イベント時のトリガー式の 1 番目、2 番目、および 3 番目のアイテムベースの関数の値に解決されます (トリガーを提供します) 式にはこれらの関数が含まれています)

さらに {HOST.HOST<1-9>} マクロはグラフ名の {?func(/host/key,param)} 式マクロ内でもサポートされています。たとえばグラフ名の {?func(/{HOST.HOST2}/key,param)} は、グラフの 2 番目の項目のホストを参照します。

Warning:

インデックス付きマクロは、ここで説明した 2 つのケースを除いて、他のコンテキストでは解決されません。他のコンテキストでは、代わりにインデックスなしのマクロ (つまり {HOST.HOST}、{HOST.IP} など) を使用します。

2 ロケーションでサポートされているユーザーマクロ

概要

このセクションには**ユーザー定義可能**マクロがサポートされている場所のリストが含まれています。

Note:

アクション、ネットワークディスカバリ、プロキシ、およびこのページのその他の場所セクションにリストされているすべての場所でサポートされているのは、グローバルレベルのユーザーマクロのみです。上記の場所では、ホストレベルおよびテンプレートレベルのマクロは解決されません。

アクション

アクションでは、ユーザーマクロを次のフィールドで使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
トリガーベースの通知とコマンド	yes
トリガーベースの内部通知	yes
障害の更新通知	yes
サービスベースの通知とコマンド	yes
サービス更新通知	yes
期間	no
オペレーション	
	デフォルトの操作ステップ期間
	ステップ期間
	no
	no

Hosts/host prototypes

ホストおよび**ホストプロトタイプ**設定では、ユーザーマクロを次のフィールドで使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
インターフェース IP/DNS	DNS only
インターフェースポート	no
SNMP v1, v2	
	SNMP コミュニティ
	yes
SNMP v3	
	コンテキスト名
	yes
	セキュリティ名
	yes
	認証パスフレーズ
	yes
	プライバシーパスフレーズ
	yes
IPMI	
	ユーザー名
	yes
	パスワード
	yes
タグ ²	
	タグ名
	yes
	タグ値
	yes

アイテム / アイテムプロトタイプ

アイテムまたは**アイテムプロトタイプ**の設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合 ¹
キー		yes
監視間隔		no
監視間隔のカスタマイズ		no
履歴の保存期間		no
トレンドの保存期間		no
説明		yes
計算		
	式	yes
データベースモニタ		
	ユーザー名	yes
	パスワード	yes
	SQL クエリ	yes
HTTP エージェント		
	URL ³	yes
	クエリフィールド	yes
	タイムアウト	no
	リクエストボディ	yes
	ヘッダ (名前と値)	yes
	要求ステータスコード	yes
	HTTP プロキシ	yes
	HTTP 認証のユーザー名	yes
	HTTP 認証のパスワード	yes
	SSI 証明書ファイル	yes
	SSI 秘密鍵ファイル	yes
	SSI 秘密鍵パスワード	yes
	許可されたホスト	yes
JMX エージェント		
	JMX エンドポイント	yes
スクリプト		
	パラメータの名前と値	yes
SNMP エージェント		
	SNMP OID	yes
SSH エージェント		
	ユーザー名	yes
	公開鍵ファイル	yes
	秘密鍵ファイル	yes
	パスワード	yes
	スクリプト	yes
TELNET エージェント		
	ユーザー名	yes
	パスワード	yes
	実行するスクリプト	yes
Zabbix トラッパー		
	許可されたホスト	yes
タグ ²		
	タグ名	yes
	タグ値 yes	
前処理		
	ステップパラメーター (カスタムスクリプトを含む)	yes

ローレベルディスクカバリ

ローレベルディスクカバリルールでは、次のフィールドでユーザーマクロを使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
キー	yes
監視間隔	no
監視間隔のカスタマイズ	no
存在しなくなったリソースの保持期間	no
説明	yes

ロケーション		複数のマクロ/テキストとの混合 ¹
SNMP エージェント	SNMP OID	yes
SSH エージェント	ユーザー名	yes
	公開鍵ファイル	yes
	秘密鍵ファイル	yes
	パスワード	yes
	実行するスクリプト	yes
TELNET エージェント	ユーザー名	yes
	パスワード	yes
	実行するスクリプト	yes
Zabbix トラッパー	許可されたホスト	yes
データベースモニタ	ユーザー名	yes
	パスワード	yes
	SQL クエリ	yes
JMX エージェント	JMX エンドポイント	yes
HTTP エージェント	URL ³	yes
	クエリフィールド	yes
	タイムアウト	no
	リクエストボディ	yes
	ヘッダー (名前と値)	yes
	要求ステータスコード	yes
	HTTP 認証ユーザー名	yes
	HTTP 認証パスワード	yes
フィルター	正規表現	yes
オーバーライド	フィルター: 正規表現	yes
	実行内容の作成: 監視間隔 (アイテムのプロトタイプ)	no
	実行内容の作成: ヒストリの保存期間 (アイテムのプロトタイプ)	no
	実行内容の作成: テレンドの保存期間 (アイテムのプロトタイプ)	no

ネットワークディスカバリ

ネットワークディスカバリルールでは、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合 ¹
監視間隔		no
SNMP v1, v2	SNMP コミュニティ	yes
	SNMP OID	yes
SNMP v3	コンテキスト名	yes
	セキュリティ名	yes
	認証パスフレーズ	yes
	プライバシーパスフレーズ	yes
	SNMP OID	yes

プロキシ

プロキシの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合 ¹
インターフェースポート (バッシュプロキシ)		no

テンプレート

テンプレートの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
タグ ²	
タグ名	yes
タグ値	yes

トリガー

トリガーの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
名前	yes
運用データ	yes
条件式 (定数と関数パラメーターのみ。シークレットマクロはサポートされていません)	yes
説明	yes
URL ³	yes
クローズに利用するタグ名	yes
タグ ²	
	タグ名 yes
	タグ値 yes

Web シナリオ

Web シナリオの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
名前	yes
監視間隔	no
エージェント	yes
HTTP プロキシ	yes
変数 (値のみ)	yes
ヘッダー (名前と値)	yes
ステップ	
	名前 yes
	URL ³ yes
	変数 (値のみ) yes
	ヘッダー (名前と変数) yes
	タイムアウト no
	要求文字列 yes
	要求ステータスコード no
認証	
	ユーザー yes
	パスワード yes
	SSL 証明書ファイル yes
	SSL 秘密鍵ファイル yes
	SSL 秘密鍵パスワード yes
タグ ²	
	タグ名 yes
	タグ値 yes

他のロケーション

ユーザーマクロはここにリストされているロケーションに加えて、次のフィールドで使用できます。

ロケーション	複数のマクロ/テキストとの混合 ¹
グローバルスクリプト (script, SSH, Telnet, IPMI), 確認テキストを含む Webhooks	yes

ロケーション		
	JavaScript スクリプト	no
	JavaScript スクリプトパラメータ名	no
	JavaScript スクリプトパラメータ値	ye
監視データ → ダッシュボード	アイテム値ダッシュボードウィジェットの説明フィールド	ye
	URL ³ および動的 URL ダッシュボードウィジェットのフィールド	ye
管理 → ユーザー → メディア	アクティブ時	no
管理 → 一般設定 → 表示設定	ワーキングタイム	no
管理 → メディアタイプ → メッセージテンプレート	件名	ye
	メッセージ	ye

Zabbix でサポートされているすべてのマクロの完全なリストについては、[サポートされているマクロ](#)を参照してください。

脚注

¹ フィールド内の複数のマクロまたはテキストと混合されたマクロがそのロケーションでサポートされていない場合は、1つのマクロでフィールド全体を埋める必要があります。

² タグ名と値で使用されるマクロは、イベント生成プロセス中にのみ解決されます。

³ [シークレットマクロ](#)を含む URL は機能しません。それらのマクロは"*****" のように解決されます。

8 単位と記号

概要

1 日の秒数を表すために「86400」などの大きな数値を使用するのは、手間かつエラーが発生しやすくなります。これが Zabbix トリガー式とアイテムキーを単純化するためにいくつかの適切な単位記号（または接尾辞）を使用できる理由です。

例えば秒数の「86400」の代わりに「1d」とできます。接尾辞は乗数として機能します。

時間のサフィックス

時間では以下のサフィックスを使うことができます

- **s** - 秒（単位の指定がない場合と同じです）
- **m** - 分
- **h** - 時間
- **d** - 日
- **w** - 週
- **M** - 月（[トレンド関数](#)のみ）
- **y** - 年（[トレンド関数](#)のみ）

時間のサフィックスは整数のみをサポートします（「1h」は OK。「1.5h」や「1.5h」は NG。代わりに「90m」を使用してください）

時間のサフィックスは次で使用可能です。

- トリガー式の定数と関数パラメーター
- [計算項目式](#)の定数
- **zabbix[queue,<from>,<to>]**内部アイテムのパラメーター
- [集計計算](#)の期間パラメータ
- アイテムの設定（「更新間隔」「カスタム間隔」「履歴保管期間」および「トレンド保管期間」の各フィールド）
- プロトタイプアイテムの設定（「更新間隔」「カスタム間隔」「履歴保管期間」および「トレンド保管期間」の各フィールド）
- 低レベルの検出ルールの設定（「更新間隔」「カスタム間隔」「失われたリソース保持」の各フィールド）
- ネットワーク検出の設定（「更新間隔」フィールド）
- Web シナリオの設定（「更新間隔」「タイムアウト」の各フィールド）
- アクション操作の設定（「デフォルト操作ステップ間隔」「ステップ間隔」の各フィールド）
- ユーザープロファイル設定（「自動ログアウト」「更新」「メッセージタイムアウト」の各フィールド）
- モニタリング → ダッシュボードのグラフウィジェット（「タイムシフト」フィールド）
- 管理 → 一般 → ハウスキーピング（「保管期間」フィールド）
- 管理 → 一般 → トリガー表示オプション（「OK トリガーを表示」「ステータス変更トリガーが点滅」の各フィールド）

- 管理 → 一般 → その他 (「ログインブロック間隔」フィールドと Zabbix サーバーとの通信に関連するフィールド)
- Zabbix サーバー `ha_set_failover_delay= delay` ランタイムコントロールオプション

メモリのサフィックス

メモリサイズのサフィックスは次で使用可能です。

- トリガー式¹の定数と関数パラメーター
- 計算項目式²の定数

メモリサイズのサフィックスは以下を使用できます。

- **K** - キロバイト
- **M** - メガバイト
- **G** - ギガバイト
- **T** - テラバイト

その他

単位記号は、フロントエンドのデータを人間が読みやすい形式で表現するためにも使用されます。

Zabbix サーバーとフロントエンドの両方で、これらのシンボルがサポートされています。

- **K** - キロ
- **M** - メガ
- **G** - ギガ
- **T** - テラ

アイテムの値に **B** が使用されるとフロントエンドに表示される Bps は二進数です。それ以外の場合は十進数で表示されます。

さらに、フロントエンドは次の表示もサポートします。

- **P** - ペタ
- **E** - エクサ
- **Z** - ゼッタ
- **Y** - ヨッタ

使用例

いくつかの適切な接尾辞を使用することにより、理解と維持が容易なトリガー式を記述できます。たとえば次の式は

```
last(/host/system.uptime[])<86400s
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

次のように変更できます

```
last(/host/system.uptime[])<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

9 日時フォーマット

概要

日時を設定するには、次の形式を使用する必要があります。

`d-d, hh:mm-hh:mm`

記号は次のことを表します。

記号	説明
d	曜日:1-月、2-火、...、7-日
hh	時:00-24
mm	分:00-59

セミコロン (;) をセパレーターとして複数の期間を指定できます

`d-d, hh:mm-hh:mm; d-d, hh:mm-hh:mm...`

空白ではデフォルト値である 01-07,00:00-24:00 になります。

Attention:

指定時間の上限は含まれていません。したがって 09:00~18:00 を指定すると、最後の 1 秒は 17:59:59 になります。

例

営業時間月～金 9:00～18:00

1-5,09:00-18:00

平日営業時間と週末営業時間月～金 9:00～18:00 土～日 10:00～16:00

1-5,09:00-18:00; 6-7,10:00-16:00

10 コマンドの実行

Zabbix は、外部チェック、ユーザーパラメータ、system.run アイテム、カスタムアラートスクリプト、リモートコマンド、およびユーザースクリプトに共通の機能を使用します。

実行手順

コマンド/スクリプトは、Unix プラットフォームと Windows プラットフォームの両方で同様に実行されます。

1. Zabbix (親プロセス) は通信用のパイプを作成します。
2. Zabbix は作成される子プロセスの出力としてパイプを設定します。
3. Zabbix は子プロセスを作成します (コマンド/スクリプトを実行します)。
4. 子プロセス用に新しいプロセスグループ (Unix の場合) またはジョブ (Windows の場合) が作成されます。
5. Zabbix はタイムアウトが発生するか、誰も書き込みを行わなくなる (すべてのハンドル/ファイル記述子が閉じられる) までパイプから読み取ります。子プロセスは、ハンドル/ファイル記述子を終了または閉じる前に、さらに多くのプロセスを作成して終了できることに注意してください。
6. タイムアウトしていない場合、Zabbix は最初の子プロセスが終了するかタイムアウトが発生するまで待機します。
7. 最初の子プロセスが終了し、タイムアウトに達していない場合、Zabbix は最初の子プロセスの終了コードをチェックして 0 と比較します。(ゼロ以外の値は、Zabbix サーバーおよび Zabbix プロキシで実行されるカスタムアラートスクリプト、リモートコマンド、およびユーザースクリプトの場合のみ、実行の失敗と見なされます)
8. この時点で、すべてが完了し、プロセスツリー全体 (つまり、プロセスグループまたはジョブ) が終了したと見なされます。

Attention:

Zabbix は、最初の子プロセスが終了したときにコマンド/スクリプトが処理を完了し、他のプロセスが出力ハンドル/ファイル記述子を開いたままにしていることを前提としています。処理が完了すると、作成されたすべてのプロセスが終了します。

コマンド内のすべての二重引用符とバックスラッシュはバックスラッシュでエスケープされ、コマンドは二重引用符で囲まれます。

終了コードのチェック

終了コードは以下の条件のときにチェック可能です。

- Zabbix サーバーと Zabbix プロキシで実行されるカスタムアラートスクリプト、リモートコマンド、ユーザースクリプトの場合
- 0 以外の終了コードは、実行失敗と見なされます
- 標準エラーの内容と失敗した実行の標準出力が収集され、フロントエンド (実行結果が表示される場所) で利用できます。
- スクリプト実行出力を保存するために Zabbix サーバー上のリモートコマンド用に追加のログエントリが作成され、LogRemoteCommands エージェント **パラメータ** を使用して有効にできます。

失敗したコマンド/スクリプトのフロントエンドメッセージとログエントリ

- (あれば) 失敗した実行の標準エラーと標準出力の内容
- "Process exited with code: N." (空の出力、および 0 に等しくない終了コードの場合)
- "Process killed by signal: N." (シグナルによって終了するプロセスの場合。Linux のみ。)
- "Process terminated unexpectedly." (不明な理由でプロセスが終了したため)

詳細については以下も参照

- [External checks](#)
- [User parameters](#)
- [system.run items](#)

- [Custom alert scripts](#)
- [Remote commands](#)
- [Global scripts](#)

See also

- [External checks](#)
- [User parameters](#)
- [system.run items](#)
- [Custom alert scripts](#)
- [Remote commands](#)
- [Global scripts](#)

11 バージョン間の互換性

サポートされるエージェント

Zabbix 6.0 と互換性を持たせるには、Zabbix エージェントがバージョン 1.4 より古いものであってはならず、6.0 よりも新しいものであってはなりません。

3.0 より前のバージョンの[ログ取得](#)に関連するパラメータなど、一部のパラメータが変更されたため、古いエージェントの場合は設定を確認する必要がある場合があります。

最新メトリック、パフォーマンス向上、メモリ使用量削減の恩恵を受けるには、サポートされている最新のエージェントを使用してください。

サポートされるエージェント 2

バージョン 4.4 以降の古い Zabbix エージェント 2 は Zabbix6.0 と互換性があります。Zabbix エージェント 2 は 6.0 より新しいものであってはなりません。

Zabbix エージェント 2 バージョン 4.4 および 5.0 を使用する場合、サポートされていないアイテムの更新にはデフォルトの間隔である 10 分が使用されます。

最新メトリック、パフォーマンス向上、メモリ使用量削減の恩恵を受けるには、サポートされている最新のエージェント 2 を使用してください。

サポートされている Zabbix プロキシ

Zabbix 6.0 と互換性を持たせるには、プロキシが同じメジャーバージョンである必要があります。したがって、Zabbix6.0.x プロキシのみが Zabbix6.0.x サーバーで動作できます。

Attention:

アップグレードされていないプロキシがアップデート済みのサーバーにデータを報告することはできなくなりました。この方法は Zabbix によって推奨もサポートもされていませんでしたが、現在は正式に無効になっています。[アップグレード手順](#)も参照してください。

互換性のない Zabbix デモンバージョンの使用に関する警告がログに記録されます。

サポートされている XML ファイル

Zabbix 6.0 では、バージョン 1.8 以上の XML ファイルのインポートがサポートされています。

Attention:

XML エクスポート形式では、トリガーの依存関係は名前のみで保存されます。たとえば、重大度と式が異なる同じ名前のトリガーが複数あり、トリガー間に依存関係が定義されている場合、それらをインポートすることはできません。このような依存関係は XML ファイルから手動で削除し、インポート後に再度追加する必要があります。

12 データベースエラーハンドリング

Zabbix はバックエンドデータベースにアクセスできないことを検出すると、通知メッセージを送信し、データベース接続の試行を続けます。一部のデータベースエンジンでは、特定のエラーコードが検出されます。

MySQL

- CR_CONN_HOST_ERROR
- CR_SERVER_GONE_ERROR
- CR_CONNECTION_ERROR
- CR_SERVER_LOST
- CR_UNKNOWN_HOST
- ER_SERVER_SHUTDOWN
- ER_ACCESS_DENIED_ERROR
- ER_ILLEGAL_GRANT_FOR_TABLE
- ER_TABLEACCESS_DENIED_ERROR
- ER_UNKNOWN_ERROR

13 Windows 用の Zabbix sender ダイナミックリンクライブラリ

Windows 環境では、アプリケーションは外部プロセス (zabbix_sender.exe) を起動する代わりに、Zabbix sender ダイナミックリンクライブラリ (zabbix_sender.dll) を使用して、Zabbix サーバー/プロキシにデータを直接送信できます。

開発ファイルを含むダイナミックリンクライブラリは、bin\winXX\dev フォルダにあります。これを使用するには zabbix_sender.h ヘッダーファイルをインクルードし、zabbix_sender.lib ライブラリにリンクします。Zabbix sender API を使用したサンプルファイルは、build\win32\examples\zabbix_sender フォルダにあります。

次の機能は、Zabbix sender ダイナミックリンクライブラリによって提供されます。

```
int zabbix_sender_send_values(const char *address, unsigned short port, const char *source, const zabbix_
char **result);{.c}
```

Zabbix sender ダイナミックリンクライブラリでは、次のデータ構造が使用されます。

```
typedef struct
{
    /* host name, must match the name of target host in Zabbix */
    char *host;
    /* the item key */
    char *key;
    /* the item value */
    char *value;
}
zabbix_sender_value_t;

typedef struct
{
    /* number of total values processed */
    int total;
    /* number of failed values */
    int failed;
    /* time in seconds the server spent processing the sent values */
    double time_spent;
}
zabbix_sender_info_t;
```

14 Python library for Zabbix API

Overview

[zabbix_utils] (<https://github.com/zabbix/python-zabbix-utils/blob/main/README.md>) is a Python library for working with Zabbix API as well as with Zabbix sender and Zabbix get protocols.

It is supported for Zabbix 5.0, 6.0, 6.4 and later.

14 サービスモニタリングのアップグレード

概要 Zabbix 6.0 ではサービスモニタリング機能が大幅に作り直されました。(変更については[Zabbix 6.0.0の新機能](#)を参照)

このページでは、Zabbix 6.0以降へのアップグレード中に、以前の Zabbix バージョンで定義されたサービスと SLA がどのように変更されるかについて説明します。

サービス 古い Zabbix バージョンでは、サービスにはソフトとハードの 2 種類の依存関係がありました。アップグレード後、すべての依存関係は等しくなります。

サービス「子サービス」が以前にハード依存関係を介して「親サービス 1」にリンクされ、さらにソフト依存関係を介して「親サービス 2」にリンクされていた場合、アップグレード後「子サービス」は単に 2 つの親サービスにリンクされます（「親サービス 1」および「親サービス 2」）。

問題とサービス間のトリガーベースのマッピングは、タグベースのマッピングに置き換えられました。Zabbix 6.0 以降では、サービス設定フォームに新しいパラメータ Problem tags が追加され、問題のマッチングのために 1 つまたは複数のタグ名と値のペアを指定できるようになりました。サービスにリンクされているトリガーは、新しいタグ ServiceLink:< トリガー ID>:< トリガー名> を取得します。(タグ値は 32 文字に切り捨てられます) リンクされたサービスは、同じ値の ServiceLinkProblem tags を取得します。

ステータス計算ルール

「ステータス計算アルゴリズム」は次のルールでアップグレードされます。

- Do not calculate → Set status to OK
- Problem, if at least one child has a problem → Most critical of child nodes
- Problem, if all children have problems → Most critical if all children have problems

Note:

Zabbix 6.0 の正式リリース前のバージョンから Zabbix 6.0.0、6.0.1、または 6.0.2 にアップグレードした場合は、Zabbix 6.0 ドキュメントの[既知の問題](#)を参照してください

SLAs 以前は SLA ターゲットをサービスごとに個別に定義する必要がありましたが、Zabbix 6.0 以降、SLA は独立したエンティティになりました。このエンティティには、サービススケジュール、期待されるサービスレベル目標 (SLO)、および計算から除外するダウンタイム期間に関する情報が含まれています。設定が完了すると service tags を介して SLA を複数のサービスに割り当てることができます。

アップグレードによる変更

- サービスごとに定義された同一の SLA がグループ化され、グループごとに 1 つの SLA が作成されます。
- 影響を受ける各サービスは、特別なタグ SLA:<ID> を取得し、対応する SLA の Service tags パラメーターで同じタグが指定されます。
- SLA レポートの新しいメトリックであるサービス作成時間は、既存のサービスでは 2000 年 1 月 1 日 00:00 に設定されます。

15 その他の問題

ログインと systemd

ログインできないシステムユーザーとして zabbix ユーザーを**作成**することをお勧めします。一部のユーザーはこの推奨事項を無視し、同じアカウントを使用して Zabbix を実行しているホストにログイン (SSH を使用するなど) しますが、ログアウト時に Zabbix デモーンがクラッシュする可能性があります。この場合、Zabbix サーバーログに次のようなものが表示されます。

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Zabbix エージェントログ:

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

これは、/etc/systemd/logind.conf で構成されたデフォルトの systemd 設定 RemoveIPC=yes が原因で発生します。システムからログアウトすると、Zabbix によって以前に作成されたセマフォが削除され、クラッシュが発生します。

systemd ドキュメントからの引用:

RemoveIPC=

ユーザーが完全にログアウトしたときに、ユーザーに属するSystemVおよびPOSIXIPCオブジェクトを削除するかどうかこの問題には2つの解決策があります。

1. (推奨) Zabbix プロセス以外の目的で * zabbix * アカウントの使用を停止し、専用アカウントを作成します。
2. (非推奨) /etc/systemd/logind.conf で RemoveIPC= no を設定し、システムを再起動します。RemoveIPC はシステム全体のパラメータであり、変更するとシステム全体に影響することに注意してください。

プロキシの背後で Zabbix フロントエンドを使用する

Zabbix フロントエンドがプロキシサーバーの背後で実行されている場合、リバースプロキシパスと一致させるために、プロキシ設定ファイルの Cookie パスを書き換える必要があります。以下の例を参照してください。Cookie パスが書き換えられていない場合、ユーザーが Zabbix フロントエンドにログインしようとしたときに認証の問題が発生する可能性があります。

nginx の設定ファイル例

```
# ..
location / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Apache の設定ファイル例

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
# ..
```

16 Agent と agent 2 の比較

このセクションは Zabbix agent と Zabbix agent 2 の違いを説明します。

比較項目	Zabbix agent	Zabbix agent 2
開発言語	C 言語	Go 言語 (部分的に C 言語)
常駐化	可能	systemd 経由のみ (Windows 上は可能)
サポートされる拡張機能	C 言語で構成された loadable modules	Go 言語で構成された plugins
必須条件		
サポートされるプラットフォーム	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows XP 以降 (Windows server ならば 2003 以降)	最新版の supported Go version がインストールできる Linux, Windows のすべてのバージョン
サポートされる暗号ライブラリ	GnuTLS は 3.1.18 と最新 OpenSSL は 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. ::: 3.0.x は Zabbix 6.0.4 以降で対応します。 LibreSSL は 2.7.4, 2.8.2 でテストをしています。(特定の制限事項は Encryption のページで詳細を見てください。)	Linux: Zabbix 4.4.8 以降は OpenSSL 1.0.1 以降のライブラリをサポートします。 Windows: OpenSSL 1.1.1 以降のライブラリをサポートします。PSK(事前共有鍵) サポートを有効にすることが必須です。 LibreSSL はサポートしません。
モニタリングプロセス (OS 上の) プロセス	アクティブチェックプロセスはサーバやプロキシへの記録ごとに分離しています。	自動的に作られたスレッドを持つ 1 つのプロセスです。 スレッド数の最大値は「GOMAXPROCS」の環境変数で決定されます。

比較項目	Zabbix agent	Zabbix agent 2
測定対象	<p>UNIX: サポートされている <code>items</code> のリストを見てください。 (/manual/config/items/itemtypes/zabbix_agent).</p> <p>Windows: Windows 固有のものとして追加された <code>items</code> のリストを見てください。</p>	<p>UNIX: Zabbix agent にサポートされているすべての測定対象 さらに agent 2 は Zabbix-native モニタリングソリューションを提供します。対象: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd, 残りの監視対象の全リストは agent 2 固有の <code>items</code> を見てください。</p> <p>Windows: Zabbix agent にサポートされているすべての測定対象と net.tcp.service* HTTPS, LDAP。 さらに agent 2 は Zabbix-native モニタリングソリューションを提供します。対象: PostgreSQL, Redis</p>
並行性	1 つのサーバに対するアクティブチェックは順次的に実行されます。	異なるプラグインからのチェック、もしくは多数のチェックを内包するプラグインは並行的に実行されます。
取得時間指定/例外更新間隔	パッシブチェックのみサポートしていません。	パッシブ、アクティブチェックどちらもサポートしています。
3rd パーティツールからのトランプ追加特性	No	Yes
永続的保管 (測定値をファイルで一時保持)	No	Yes
log*[] アイテムでの固定ファイル測定	Yes (Unix のみ)	No
タイムアウト設定	Agent レベルでのみ定義されます。	Agent レベルで定義されたタイムアウト値をプラグインのタイムアウト値で上書きできます。
実行時に実行ユーザーを変更	Yes (Unix ライクのシステムのみ)	No (systemd でコントロールされます)
ユーザー定義の暗号化スイート	Yes	No

こちらも参照してください:

- Zabbix プロセスの説明: [Zabbix agent](#), [Zabbix agent 2](#)
- 設定項目: [Zabbix agent UNIX / Windows](#), [Zabbix agent 2 UNIX / Windows](#)

18 Escaping examples

Overview

This page provides examples of using correct escaping when using regular expressions in various contexts.

Note:

When using the trigger expression constructor, correct escaping in regular expressions is added automatically.

Examples

User macro with context

Regular expression: `\.+\\" [a-z]+
` User macro with context: `{${MACRO}:regex:"\\.+\\" [a-z]+"`

Notice:

- backslashes are **not escaped**;
- quotation marks are escaped.

LLD macro function

Regular expression: `\.+\\" [a-z]+
` LLD macro: `{{#MACRO}}.iregsub("\\.+\\" [a-z]+", \1)}`

Notice:

- backslashes are not escaped;
- quotation marks are escaped.

LLD macro function inside user macro context

Regular expression: `\.+\"[a-z]+`
 LLD macro: `{#MACRO}.iregsub(\".+\\\"[a-z]+\", \\1)`
 User macro with context: `{#MACRO}:{#MACRO}.iregsub(\".+\\\"[a-z]+\", \\1)}`

Notice:

- backslash escaping for LLD does not change;
 - upon inserting the LLD macro into user macro context, we need to put it into string:
1. Quotation marks are added around the macro expression;
 2. Quotation marks get escaped; in total, 3 new backslashes are introduced.

String parameter of non-history function

String content: `\.+\"[a-z]+`
 Expression: `concat("abc", "\\.\\"[a-z]+")`

Notice:

- String parameters require escaping both for backslashes and quotation marks.

String parameter of history function

String content: `\.+\"[a-z]+`
 Expression: `find(__ITEM_KEY__, "regex", "\.+\\\"[a-z]+")`

Notice:

- backslashes are not escaped;
- quotation marks are escaped.

LLD macro function inside string parameter of non-history function

Regular expression: `\.+\"[a-z]+`
 LLD macro: `{#MACRO}.iregsub(\".+\\\"[a-z]+\", \\1)`
 Expression: `concat("abc", "{#MACRO}.iregsub(\"\\\.\\"[a-z]+\", \\1)`

Notice:

- String parameters require escaping both for backslashes and quotation marks;
- Another layer of escaping is added, because the macro will be resolved only after string is unquoted;

LLD macro function inside string parameter of history function

Regular expression: `\.+\"[a-z]+`
 LLD macro: `{#MACRO}.iregsub(\".+\\\"[a-z]+\", \\1)`
 Expression: `find(__ITEM_KEY__, "eq", "{#MACRO}.iregsub(\"\\\.\\"[a-z]+\", \\1)`

Notice:

- backslashes are not escaped;
- quotation marks are escaped.

User macro with context inside string parameter of non-history function

Regular expression: `\.+\"[a-z]+`
 User macro with context: `{#MACRO:regex:\.+\\\"[a-z]+}`
 Expression: `concat("abc", "{#MACRO:regex:\\.\\"[a-z]+}")`

Notice:

- Same as in the previous example an additional layer of escaping is needed;
- Backslashes and quotation marks are escaped only for the top-level escaping (by virtue of it being a string parameter).

User macro with context inside string parameter of history function

Regular expression: `\.+\"[a-z]+`
 User macro with context: `{#MACRO:regex:\.+\\\"[a-z]+}`
 Expression: `find(__ITEM_KEY__, "eq", "{#MACRO:regex:\\.\\"[a-z]+}")`

Notice:

- backslashes are not escaped;
- quotation marks are escaped.

LLD macro function inside user macro context inside non-history function

Regular expression: `\.+\"[a-z]+`
 LLD macro: `{#MACRO}.iregsub(\".+\\\"[a-z]+\", \\1)`
 User macro with context: `{#MACRO}:{#MACRO}.iregsub(\".+\\\"[a-z]+\", \\1)`
 Expression: `concat("abc", "{#MACRO:\\.\\"[a-z]+}")`

Notice the three layers of escaping:

1. For LLD macro function, without escaping of backslashes;
2. For User macro with context, without escaping of backslashes;
3. For the string parameter of a function, with escaping of backslashes.

LLD macro function inside user macro context inside history function

Regular expression: `\.+\" [a-z]+`
 LLD macro: `{#{#MACRO}.iregsub(\".+\\\" [a-z]+\", \\1)}`
 User macro with context: `{#{#MACRO}:{#{#MACRO}.iregsub(\".+\\\" [a-z]+\", \\1)}`
 Expression: `find(__ITEM_KEY__, "eq", "{#{#MACRO}:\\\"{#{#MACRO}.iregsub(\".+\\\" [a-z]+\", \\1)}\\\"")`

Notice:

- backslashes are not escaped;
- quotation marks are escaped.

User macro with context just inside string

Regular expression: `\.+\" [a-z]+`
 User macro with context: `{#{#MACRO:regex:\".+\\\" [a-z]+}`
 Inside string of some expression, for example: `func(arg1, arg2, arg3)="{#{#MACRO:regex:\".+\\\" [a-z]+}\"`

Notice:

- Strings also require backslash escaping;
 - Strings also require quotation mark escaping;
 - Again a case with 2 levels of escaping:
1. Escaping for user macro context without backslash escaping;
 2. Escaping for it being a string with backslash escaping.

Zabbix リファレンスマニュアル

こちらは Zabbix プロセスのリファレンスマニュアルです。

zabbix_agent2

セクション: メンテナンスコマンド (8)

更新日: 2019-01-29

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_agent2 - Zabbix エージェント 2

概要

zabbix_agent2 [-c config-file]

zabbix_agent2 [-c config-file] -p

zabbix_agent2 [-c config-file] -t item-key

zabbix_agent2 [-c config-file] -R runtime-option

zabbix_agent2 -h

zabbix_agent2 -V

説明

zabbix_agent2 は様々なサービスのパラメーターを監視するためのアプリケーションです。

オプション

-c, --config config-file

デフォルトの設定ファイルの代わりに、別の config-file を使用します。

-R, --runtime-control runtime-option

runtime-option に従って管理機能を実行します。

ランタイム制御オプション: **userparameter reload**

現在の設定ファイルから UserParameter と Include のオプションをリロードする

loglevel increase

ログレベルを上げる

loglevel decrease

ログレベルを下げる

help

利用可能なランタイム制御オプションを一覧表示する

metrics

利用可能なメトリクスを一覧表示する

version

バージョンを表示する

-p, --print

既知のアイテムを表示して終了します。各アイテムに対して、一般的なデフォルトが使用されるか、テスト用の特定のデフォルトが提供されます。これらのデフォルトは、アイテムキーのパラメータとして角括弧内にリストされています。戻り値は角括弧で囲まれ、パイプ文字で区切られた戻り値の型が前に付けられます。ユーザーパラメータの型は常に **t** となります。これは、エージェントが可能なすべての戻り値を決定できるわけではないためです。パーミッションまたは環境が異なる可能性があるため、実行中のエージェントデーモンをクエリする場合、動作していると表示されるアイテムは、Zabbix サーバーまたは zabbix_get から動作することが保証されません。戻り値の型は以下の通りです:

d

小数部分を含む数値。

m

サポートされていません。これは、ログ監視アイテムや複数の収集値を必要とするアイテムなど、アクティブモードでのみ機能するアイテムをクエリしたことが原因である可能性があります。パーミッションの問題や、ユーザーパラメーターが正しくない場合も、サポートされていない状態になる可能性があります。

s

テキスト。長さに制限はありません。

t

テキスト。**s** と同じ。

u

符号なし整数。

-t, --test item-key

単一のアイテムをテストして終了します。出力の説明については、**--print** を参照してください。

-h, --help

このヘルプを表示して終了します。

-V, --version

バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_agent2.conf Zabbix エージェント 2 の設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)

参照

ドキュメント <https://www.zabbix.com/manuals>

[zabbix_agentd\(8\)](#), **[zabbix_get\(8\)](#)**, **[zabbix_js\(8\)](#)**, **[zabbix_proxy\(8\)](#)**, **[zabbix_sender\(8\)](#)**, **[zabbix_server\(8\)](#)**

著者

Zabbix LLC

目次

[名前](#)

[概要](#)

[説明](#)

[オプション](#)

[ファイル](#)

[参照](#)

[著者](#)

ドキュメント作成日: 14:07:57 GMT, 2021 年 11 月 22 日

zabbix_agentd

セクション: メンテナンスコマンド (8)

更新日: 2019-01-29

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_agentd - Zabbix エージェントデーモン

概要

```
zabbix_agentd [-c config-file]
zabbix_agentd [-c config-file] -p
zabbix_agentd [-c config-file] -t item-key
zabbix_agentd [-c config-file] -R runtime-option
zabbix_agentd -h
zabbix_agentd -V
```

説明

zabbix_agentd は、さまざまなサーバーパラメーターを監視するためのデーモンです。

オプション

-c, --config config-file

デフォルトの設定ファイルの代わりに config-file で指定されたファイルを使用します。

-f, --foreground

Zabbix エージェントをフォアグラウンドで実行します。

-R, --runtime-control runtime-option

runtime-option に従って管理機能を実行します。

ランタイム制御オプション

userparameter_reload[=target]

現在の設定ファイルから UserParameter と Include のオプションをリロードします。

log_level_increase[=target]

ログレベルを上げます。ターゲットが指定されていない場合、全てのプロセスに影響します。

log_level_decrease[=target]

ログレベルを下げます。ターゲットが指定されていない場合、全てのプロセスに影響します。

ログレベル制御対象

process-type

指定されたタイプのすべてのプロセス（アクティブチェック、コレクター、リスナー）

process-type,N

プロセスの種類と数 (例: listener,3)

pid

プロセス ID。最大 65535。値が大きい場合はターゲットを "process-type,N" と指定します。

-p, --print

既知のアイテムを出力して終了します。各アイテムに対して、一般的なデフォルトが使用されるか、テスト用の特定のデフォルトが提供されます。これらのデフォルトは、アイテムキーのパラメータとして角括弧内にリストされています。戻り値は角括弧で囲まれ、パイプ文字で区切られた戻り値の型が前に付けられます。ユーザーパラメータの型は常に **t** となります。これは、エージェントが可能なすべての戻り値を決定できるわけではないためです。パーミッションまたは環境が異なる可能性があるため、実行中のエージェントデーモンを照会する場合、動作していると表示されるアイテムは、Zabbix サーバーまたは `zabbix_get` から動作することが保証されません。戻り値の型は次の通りです。

d

小数部分を含む数値。

m

サポートされていません。これは、ログ監視アイテムや複数の収集値を必要とするアイテムなど、アクティブモードでのみ機能するアイテムをクエリしたことが原因である可能性があります。アクセス許可の問題やユーザーパラメーターが正しくない場合も、サポートされていない状態になる可能性があります。

s

テキスト。長さに制限はありません。

t

テキスト。**s** と同じ。

u

符号なし整数。

-t, --test item-key

単一のアイテムをテストして終了します。出力の説明については、**--print** を参照してください。

-h, --help

このヘルプを出力して終了します。

-V, --version

バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_agentd.conf Zabbix エージェント設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

著者

Alexei Vladishev <alex@zabbix.com>

目次

名前

概要

説明

オプション

ファイル

参照

著者

ドキュメント作成日: 20:50:13 GMT, 2021 年 11 月 22 日

zabbix_get

セクション: ユーザーコマンド (1)

更新日: 2021-06-01

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_get - Zabbix get ユーティリティ

概要

zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t timeout] [-k item-key] **zabbix_get -s** host-name-or-IP [-p port-number] [-I IP-address] [-t timeout] **--tls-connect cert --tls-ca-file** CA-file **--tls-crl-file** CRL-file **--tls-agent-cert-issuer** cert-issuer **--tls-agent-cert-subject** cert-subject **--tls-cert-file** cert-file **--tls-key-file** key-file **--tls-cipher13** cipher-string] **--tls-cipher** cipher-string] [-k item-key] **zabbix_get -s** host-name-or-IP [-p port-number] [-I IP-address] [-t timeout] **--tls-connect psk --tls-psk-identity** PSK-identity **--tls-psk-file** PSK-file **--tls-cipher13** cipher-string] **--tls-cipher** cipher-string] [-k item-key] **zabbix_get -h zabbix_get -V**

説明

zabbix_get は、Zabbix エージェントからデータを取得するためのコマンドラインユーティリティです。

オプション

-s, --host host-name-or-IP

ホスト名またはホストの IP アドレスを指定します。

-p, --port port-number

ホスト上で動作するエージェントのポート番号を指定します。デフォルトは 10050 です。

-I, --source-address IP-address

ソース IP アドレスを指定します。

-t, --timeout seconds

タイムアウトを指定します。有効範囲: 1-30 秒 (デフォルト: 30)

-k, --key item-key

値を取得するアイテムのキーを指定します。

--tls-connect value

エージェントへの接続方法。値:

unencrypted

暗号化せずに接続 (デフォルト)

psk

TLS と事前共有キーを使用して接続

cert

TLS と証明書を使用して接続

--tls-ca-file CA-file

ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス。

--tls-crl-file CRL-file

失効した証明書を含むファイルのフルパス。

--tls-agent-cert-issuer cert-issuer

許可されたエージェント証明書の発行者。

--tls-agent-cert-subject cert-subject

許可されたエージェント証明書のサブジェクト。

--tls-cert-file cert-file

証明書または証明書チェーンを含むファイルのフルパス。

--tls-key-file key-file

秘密鍵を含むファイルのフルパス。

--tls-psk-identity PSK-identity

PSK ID 文字列。

--tls-psk-file PSK-file

事前共有キーを含むファイルのフルパス。

--tls-cipher13 cipher-string

TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準をオーバーライドします。OpenSSL のバージョンが 1.1.1 未満の場合、このオプションは使用できません。

--tls-cipher cipher-string

GnuTLS 優先度文字列 (TLS 1.2 以降の場合) または OpenSSL 暗号文字列 (TLS 1.2 のみ)。デフォルトの暗号スイート選択基準をオーバーライドします。

-h, --help

ヘルプを出力して終了する。

-V, --version

バージョン情報を出力して終了する。

例

```
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
```

```
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file  
--tls-agent-cert-issuer "CN=Signing CA,OU=IT operations,O=Example Corp,DC=example,DC=com" --tls-agent-cert-  
subject "CN=server1,OU=IT operations,O=Example Corp,DC=example,DC=com" --tls-cert-file /home/zabbix/zabbix_get.crt  
--tls-key-file /home/zabbix/zabbix_get.key
```

```
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" --tls-connect psk --tls-psk-identity "PSK ID Zabbix  
agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk
```

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

著者

Alexei Vladishev <alex@zabbix.com>

目次

名前

概要

説明

オプション

例

参照

著者

ドキュメント作成日: 08:42:29 GMT, 2021 年 6 月 11 日

zabbix_js

セクション: ユーザーコマンド (1)

更新日: 2019-01-29

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_js - Zabbix JS ユーティリティ

概要

zabbix_js -s script-file **-p** input-param [**-l** log-level] [**-t** timeout]

zabbix_js -s script-file **-i** input-file [**-l** log-level] [**-t** timeout]

zabbix_js -h

zabbix_js -V

説明

zabbix_js は、組み込みスクリプトのテストに使用できるコマンドラインユーティリティです。

オプション

-s, --script script-file

実行するスクリプトのファイル名を指定します。ファイル名に '-' を指定するとスクリプトは標準入力から読み込まれます。

-p, --param input-param

入力パラメーターを指定します。

-i, --input input-file

入力パラメーターのファイル名を指定します。ファイル名に '-' を指定すると入力は標準入力から読み込まれます。

-l, --loglevel log-level

ログレベルを指定します。

-t, --timeout timeout

タイムアウトを秒単位で指定します。有効範囲: 1-60 秒 (デフォルト: 10)

-h, --help

ヘルプを表示して終了します。

-V, --version

バージョン情報を表示して終了します。

例

zabbix_js -s script-file.js -p example

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agent2(8), zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8)

目次

名前

概要

説明

オプション

例

参照

ドキュメント作成日: 21:23:35 GMT, 2020 年 3 月 18 日

zabbix_proxy

セクション: メンテナンスコマンド (8)

更新日: 2020-09-04

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_proxy - Zabbix プロキシデーモン

概要

zabbix_proxy [-c config-file]

zabbix_proxy [-c config-file] -R runtime-option

zabbix_proxy -h

zabbix_proxy -V

説明

zabbix_proxy は、デバイスから監視データを収集し、Zabbix サーバーに送信するデーモンです。

オプション

-c, --config config-file

デフォルトの設定ファイルの代わりに config-file で指定されたファイルを使用します。

-f, --foreground

Zabbix プロキシをフォアグラウンドで実行します。

-R, --runtime-control runtime-option

runtime-option に従って管理機能を実行します。

ランタイム制御オプション **config_cache_reload**

構成キャッシュをリロードします。キャッシュが読み込み中の場合は無視されます。アクティブな Zabbix プロキシは Zabbix サーバーに接続し、構成データを要求します。デフォルトの設定ファイル (**-c** オプションが指定されていない場合) が PID ファイルの検索に使用され、シグナルがプロセスに送信され、PID ファイルにリストされます。

snmp_cache_reload

SNMP キャッシュをリロードします。

housekeeper_execute

ハウスキーパーを実行します。実行中の場合は無視されます。

diaginfo[=section]

指定されたセクションの内部診断情報をログに記録します。対象セクションは historycache、preprocessing です。デフォルトではすべてのセクションの診断情報がログに記録されます。

log_level_increase[=target]

ログレベルを上げます。ターゲットが指定されていない場合、すべてのプロセスに影響します

log_level_decrease[=target]

ログレベルを下げます。ターゲットが指定されていない場合、すべてのプロセスに影響します

ログレベル制御対象 process-type

指定されたタイプのすべてのプロセス (configuration syncer, data sender, discoverer, heartbeat sender, history syncer, house-keeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector)

process-type,N

プロセスの種類と数 (例: poller,3)

pid

プロセス ID。最大 65535。値が大きい場合はターゲットを "process-type,N" と指定します。

-h, --help

ヘルプを出力して終了します。

-V, --version

バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_proxy.conf Zabbix プロキシ構成ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

著者

Alexei Vladishev <alex@zabbix.com>

- 目次
- 名前
- 概要
- 説明
- オプション
- ファイル
- 参照
- 著者

ドキュメント作成日: 16:12:22 GMT, 2020 年 9 月 04 日

zabbix_sender

セクション: ユーザーコマンド (1)

更新日: 2021-06-01

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_sender - Zabbix sender ユーティリティ

概要

```
zabbix_sender [-v] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] [-k key] [-o value]
zabbix_sender [-v] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] [-T] [-N] [-r] [-i input-file]
zabbix_sender [-v] [-c config-file] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] [-k key] [-o value] zabbix_sender [-v] [-c
config-file] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] [-T] [-N] [-r] [-i input-file]
zabbix_sender [-v] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
CRL-file] [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file --tls-key-file key-
file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-k key] [-o value]
zabbix_sender [-v] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
CRL-file] [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file --tls-key-file key-
file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-T] [-N] [-r] [-i input-file]
zabbix_sender [-v] [-c config-file] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-
file [--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-k key] [-o value]
zabbix_sender [-v] [-c config-file] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-
file [--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-T] [-N] [-r] [-i input-file]
zabbix_sender [-v] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-identity --
tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-k key] [-o value]
zabbix_sender [-v] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-identity
--tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-T] [-N] [-r] [-i input-file]
zabbix_sender [-v] [-c config-file] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect psk --tls-psk-identity
PSK-identity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-k key] [-o value]
zabbix_sender [-v] [-c config-file] [-z server] [-p port] [-I IP-address] [-t timeout] [-s host] --tls-connect psk --tls-psk-identity
PSK-identity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] [-T] [-N] [-r] [-i input-file]
zabbix_sender -h
zabbix_sender -V
```


- 改行エスケープシーケンス (\n) は、引用符で囲まれた文字列でサポートされています。

- 改行エスケープシーケンスは、エントリの末尾から削除されます。

-T, --with-timestamps

このオプションは、**--input-file** オプションでのみ使用できます。

入力ファイルの各行には、空白で区切られた 4 つのエントリが含まれている必要があります: < ホスト名 > < キー > < タイムスタンプ > < 値 >。タイムスタンプは Unix タイムスタンプ形式で指定する必要があります。ターゲットのアイテムにそれを参照するトリガーがある場合、すべてのタイムスタンプは昇順でなければなりません。そうしないと、イベントの計算が正しく行われません。

入力ファイルの行の例:

"Linux DB3" db.connections 1429533600 43

詳細については、オプション **--input-file** を参照してください。

メンテナンスタイプが"データなし"のホストにタイムスタンプ付きの値が送信された場合、この値は削除されます。ただし、期限切れのメンテナンス期間にタイムスタンプ付きの値を送信することは可能であり、それは受け入れられます。

-N, --with-ns このオプションは、**--with-timestamps** オプションでのみ使用できます。

入力ファイルの各行には、空白で区切られた 5 つのエントリが含まれている必要があります: <hostname> <key> <timestamp> <ns> <value>。

入力ファイルの行の例:

"Linux DB3" db.connections 1429533600 7402561 43

詳細については、オプション **--input-file** を参照してください。

-r, --real-time 値を受信したらすぐに 1 つずつ送信します。これは、標準入力から読み取る場合にも使用できます。*

--tls-connect value サーバーまたはプロキシへの接続方法。値:

接続方法の値: **unencrypted** 暗号化なしで接続 (デフォルト)

psk TLS と事前共有キーを使用して接続する

cert TLS と証明書を使用して接続する

--tls-ca-file CA-file ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

--tls-crl-file CRL-file 失効した証明書を含むファイルのフルパス名

--tls-server-cert-issuer cert-issuer 許可されたサーバー証明書の発行者

--tls-server-cert-subject cert-subject 許可されたサーバー証明書のサブジェクト

--tls-cert-file cert-file 証明書または証明書チェーンを含むファイルのフルパス名

--tls-key-file key-file 秘密鍵を含むファイルのフルパス名

--tls-psk-identity PSK-identity PSK ID 文字列

--tls-psk-file PSK-file 事前共有キーを含むファイルのフルパス名

--tls-cipher13 cipher-string TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準をオーバーライドします。OpenSSL のバージョンが 1.1.1 未満の場合、このオプションは使用できません。

--tls-cipher cipher-string GnuTLS 優先度文字列 (TLS 1.2 以降の場合) または OpenSSL 暗号文字列 (TLS 1.2 のみ)。デフォルトの暗号スイート選択基準をオーバーライドします。

-v, --verbose 冗長化モード。詳細については**-vv**

-h, --help ヘルプを出力して終了する。

-V, --version バージョン情報を出力して終了する。

終了ステータス

値が送信され、それらすべてがサーバーによって正常に処理された時の終了ステータスは 0 です。データが送信されたが、少なくとも 1 つの値の処理が失敗した時の終了ステータスは 2 です。データ送信が失敗した時の終了ステータスは 1 です。

例

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

監視対象ホストの **mysql.queries** 項目の値として **342.45** を送信します。エージェント構成ファイルで定義された監視対象ホストと Zabbix サーバーを使用します。

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host" -k mysql.queries -o 342.45

エージェント設定ファイルで定義された Zabbix サーバーを使用して **Monitored Host** ホストの **mysql.queries** 項目の値として **342.45** を送信します。

zabbix_sender -z 192.168.1.113 -i data_values.txt ファイル **data_values.txt** から IP **192.168.1.113** の Zabbix サーバーに値を送信します。ホスト名とキーはファイルで定義されます。

echo "- hw.serial.number 1287872261 SQ4321ASDF" | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

エージェント設定ファイルで指定されたコマンドラインから Zabbix サーバーにタイムスタンプ付きの値を送信します。入力データのダッシュは、ホスト名も同じ設定ファイルから使用する必要があることを示します。

echo "'Zabbix server' trapper.item ''" | zabbix_sender -z 192.168.1.113 -p 10000 -i -

コマンドラインから IP アドレス **192.168.1.113** のポート **10000** でリスンしている Zabbix サーバーに空の項目の値を送信します。空の値は、空の二重引用符で示します。

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o 342.45 --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-cert-file /home/zabbix/zabbix_agentd.crt --tls-key-file /home/zabbix/zabbix_agentd.key
Monitored Host ホストの **mysql.queries** 項目の値として **342.45** を、証明書付きの TLS を使用して IP **192.168.1.113** のサーバーに送信します。

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o 342.45 --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk

Monitored Host ホストの **mysql.queries** アイテムの値として **342.45** を IP **192.168.1.113** のサーバーに、TLS と事前共有キー (PSK) を使用して送信します。

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agentd(8), **zabbix_get(1)**, **zabbix_proxy(8)**, **zabbix_server(8)**, **zabbix_js(1)**, **zabbix_agent2(8)**, **zabbix_web_service(8)**

著者

Alexei Vladishev <alex@zabbix.com>

目次

[名前](#)

[概要](#)

[説明](#)

[オプション](#)

[終了ステータス](#)

[例](#)

[参照](#)

[著者](#)

ドキュメント作成日: 08:42:39 GMT, 2021 年 6 月 11 日

zabbix_server

セクション: メンテナンスコマンド (8)

更新日: 2020-09-04

[目次](#) [メインコンテンツに戻る](#)

名前

zabbix_server - Zabbix サーバーデーモン

概要

```
zabbix_server [-c config-file]
zabbix_server [-c config-file] -R runtime-option
zabbix_server -h
zabbix_server -V
```

説明

zabbix_server は、Zabbix ソフトウェアのコアデーモンです。

オプション

-c, --config config-file
デフォルトの設定ファイルの代わりに config-file で指定されたファイルを使用します。

-f, --foreground
Zabbix サーバーをフォアグラウンドで実行します。

-R, --runtime-control runtime-option
runtime-option に従って管理機能を実行します。

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を表示して終了します。

コマンドライン引数を使用した Zabbix サーバーの実行例：

```
zabbix_server -c /usr/local/etc/zabbix_server.conf
zabbix_server --help
zabbix_server -V
```

ランタイム制御

ランタイム制御オプション：

config_cache_reload

設定のキャッシュをリロードします。キャッシュが読み込み中の場合は無視されます。デフォルトの設定ファイル (**-c** オプションが指定されていない場合) が PID ファイルの検索に使用され、PID ファイルにリストされたプロセスにシグナルが送信されます。

snmp_cache_reload

SNMP のキャッシュをリロードします。

housekeeper_execute

ハウスキーパーを実行します。ハウスキーパーが実行中の場合は無視されます。

trigger_housekeeper_execute

トリガーハウスキーパーを実行します。トリガーハウスキーパーが実行中の場合は無視されます。

diaginfo[=section]

指定されたセクションの内部診断情報をログに記録します。対象セクションは historycache, preprocess です。ing, alerting, lld, valuecache です。デフォルトでは、すべてのセクションの診断情報がログに記録されます。

ha_status

高可用性 (HA) クラスターのステータスをログに記録します。

ha_remove_node[=target]

名前または ID で指定された高可用性 (HA) ノードを削除します。アクティブ/スタンバイ状態のノードは削除できません。

ha_set_failover_delay[=delay]

高可用性 (HA) フェールオーバーの遅延時間を設定します。時刻サフィックスがサポートされています。例: 10s, 1m.

secrets_reload

Vault からシークレットをリロードします。

service_cache_reload サービスマネージャーのキャッシュをリロードします。

snmp_cache_reload SNMP のキャッシュをリロードし、すべてのホストの SNMP プロパティ (engine time, engine boots, engine id, 認証情報) をクリアします。

prof_enable[=target]

プロファイリングを有効にします。ターゲットが指定されていない場合、すべてのプロセスに影響します。プロファイリングを有効にすると、関数名からすべての rwlocks/mutexes の詳細を見ることができます。Zabbix 6.0.13 以降でサポートされています。

prof_disable[=target]

プロファイリングを無効にします。ターゲットが指定されていない場合、すべてのプロセスに影響します。Zabbix 6.0.13 以降でサポートされています。

log_level_increase[=target]

ログレベルを上げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。

log_level_decrease[=target]

ログレベルを下げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。

ログレベル制御対象

process-type

指定されたタイプのすべてのプロセス (alerter, alert manager, configuration syncer, discoverer, escalator, history syncer, house-keeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, poller, preprocessing manager, preprocessing worker, proxy poller, self-monitoring, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N

プロセスの種類と数 (例: poller,3)

pid

プロセス ID。最大 65535。値が大きい場合はターゲットを "process-type,N" と指定します。

ファイル

/usr/local/etc/zabbix_server.conf Zabbix サーバー設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agentd(8), **zabbix_get(1)**, **zabbix_proxy(8)**, **zabbix_sender(1)**, **zabbix_js(1)**, **zabbix_agent2(8)**

著者

Alexei Vladishev <alex@zabbix.com>

目次
名前
概要
説明
オプション
ファイル
参照
著者

ドキュメント作成日: 16:12:14 GMT, 2020 年 9 月 04 日

zabbix_web_service

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
[目次](#) [メインコンテンツに戻る](#)

名前

`zabbix_web_service` - Zabbix web サービス

概要

zabbix_web_service [-c config-file]
zabbix_web_service -h
zabbix_web_service -V

説明

zabbix_web_service は、Zabbix コンポーネントに Web サービスを提供するためのアプリケーションです。

オプション

-c, --config config-file
デフォルトのコンフィグファイルの代わりに config-file で指定されたファイルを使用します。

-h, --help
ヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

ファイル

`/usr/local/etc/zabbix_web_service.conf`
Zabbix Web サービス設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)

参照

ドキュメント <https://www.zabbix.com/manuals>

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

著者

Zabbix LLC

目次

名前

概要

説明

オプション

ファイル

参照

著者

ドキュメント作成日: 12:58:30 GMT, 2021 年 6 月 11 日