АО "НТЦ ИТ РОСА"

**Платформа**

**централизованного управления жизненным циклом операционных систем**

**"РОСА ЦЕНТР УПРАВЛЕНИЯ"**

**Версия 2.3.0**

**Руководство системного администратора**

**Часть 3-2. Эксплуатация**

**ПОДСИСТЕМА МОНИТОРИНГА**

**Приложения**

РСЮК.10121-08 32 03-2

Листов 457

2025

АННОТАЦИЯ

Данное руководство предназначено для пользователей, в том числе системных администраторов, подсистемы мониторинга программного средства "Платформа централизованного управления жизненным циклом операционных систем "РОСА Центр Управления" РСЮК.10121 (далее – Подсистема).

В руководстве содержатся сведения информационно-справочного характера, необходимые для ознакомления пользователей с документами РОСА Центр Управления и Подсистемы:

* "РОСА Центр Управления. Руководство системного администратора. Часть 1. Установка и настройка" (шифр РСЮК.10121-08 32 01);
* "РОСА Центр Управления. Руководство системного администратора. Часть 2. Эксплуатация" (шифр РСЮК.10121-08 32 02);
* "РОСА Центр Управления. Руководство системного администратора. Часть 3-1. Эксплуатация. Подсистема мониторинга" (шифр РСЮК.10121-01 32 03-1).

Для разработки документа использованы ссылки на следующие стандарты:

* ГОСТ Р 2.105-2019 "Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам";
* ГОСТ 2.601 "Единая система программной документации. Виды программных документов";
* ГОСТ 19.101-77 "Единая система программной документации.  
  Виды программ и программных документов";
* ГОСТ 19.105-78 "Единая система программной документации. Общие требования к программным документам";
* ГОСТ 19.503-79 "Единая система программной документации. Руководство системного программиста".

Настоящий документ подготовлен в соответствии с технологической инструкцией "РОСА. Регламент формирования документации к программным продуктам" (шифр РСЮК.11001-02 90 01).

Содержание

[Содержание 3](#_Toc193280693)

[Приложение А. Конфигурация демонов 7](#_Toc193280694)

[А.1 Сервер 8](#_Toc193280695)

[А.2 Агент-1 (UNIX) 24](#_Toc193280696)

[А.3 Агент-2 (UNIX) 34](#_Toc193280697)

[А.4 Агент-1 (Windows) 43](#_Toc193280698)

[А.5 Агент-2 (Windows) 53](#_Toc193280699)

[А.6 Плагины Агента-1 (Windows) 62](#_Toc193280700)

[А.7 Java gateway 84](#_Toc193280701)

[А.8 Веб-сервис 85](#_Toc193280702)

[А.9 Включение (параметр "Include") 87](#_Toc193280703)

[Приложение Б. Протоколы 88](#_Toc193280704)

[Б.1 Протокол обмена данными между Сервером и Прокси 88](#_Toc193280705)

[Б.2 Пассивный Прокси 88](#_Toc193280706)

[Б.3 Протокол Агента-1 97](#_Toc193280707)

[Б.4 Протокол Агента-2 97](#_Toc193280708)

[Б.5 Протокол плагина Агента-2 100](#_Toc193280709)

[Б.6 Протокол sender 104](#_Toc193280710)

[Б.7 Заголовок 104](#_Toc193280711)

[Б.8 Протокол экспорта в режиме реального времени 107](#_Toc193280712)

[Б.8.1 Триггерные события 107](#_Toc193280713)

[Б.8.2 Значения элементов данных 108](#_Toc193280714)

[Б.8.3 Динамика изменений 109](#_Toc193280715)

[Приложение В. Типы процессов 111](#_Toc193280716)

[В.1 Типы процессов Сервера 111](#_Toc193280717)

[В.2 Типы процессов Прокси 112](#_Toc193280718)

[Приложение Г. Элементы данных 114](#_Toc193280719)

[Г.1 Поддерживаемые элементы данных по платформам 114](#_Toc193280720)

[Г.2 Параметры vm.memory.size 123](#_Toc193280721)

[Г.3 Пассивные и активные проверки Агентов 125](#_Toc193280722)

[Г.3.2.1 Пассивные проверки 125](#_Toc193280723)

[Г.3.2.2 Активные проверки 126](#_Toc193280724)

[Г.3.3 Получение списка активных проверок 126](#_Toc193280725)

[Г.3.4 Отправка собранных данных 128](#_Toc193280726)

[Г.4 Минимальный уровень прав для элементов данных Агента Windows 130](#_Toc193280727)

[Г.5 Кодировка получаемых значений 133](#_Toc193280728)

[Г.6 Поддержка больших файлов 133](#_Toc193280729)

[Г.7 Датчики 133](#_Toc193280730)

[Г.8 Параметр "тип памяти" в элементах данных proc.mem 136](#_Toc193280731)

[Г.9 Выбор процессов в элементах данных proc.mem и proc.num 140](#_Toc193280732)

[Г.9.1 Процессы, изменяющие свою командную строку 140](#_Toc193280733)

[Г.9.2 Потоки ядра Linux 143](#_Toc193280734)

[Г.10 Реализация проверок net.tcp.service и net.udp.service 145](#_Toc193280735)

[Г.11 Настройки недостижимости/недоступности интерфейса узла сети 146](#_Toc193280736)

[Г.11.1 Недостижимый интерфейс 146](#_Toc193280737)

[Г.11.2 Недоступный интерфейс 147](#_Toc193280738)

[Г.12 Удаленный мониторинг статистики 148](#_Toc193280739)

[Г.13 Настройка Kerberos 153](#_Toc193280740)

[Г.14 Параметры modbus.get 154](#_Toc193280741)

[Приложение Д. Типы элементов данных 157](#_Toc193280742)

[Д.1 Агент 157](#_Toc193280743)

[Д.1.1 Специфичные ключи элементов данных для Агента-2 199](#_Toc193280744)

[Д.1.2 Специфичные ключи элементов данных для Windows 222](#_Toc193280745)

[Д.2 SNMP-агент 231](#_Toc193280746)

[Д.2.1 Обработка массовых SNMP-запросов 235](#_Toc193280747)

[Д.2.2 Динамические индексы 237](#_Toc193280748)

[Д.2.3 Специальные OID 239](#_Toc193280749)

[Д.2.4 MIB-файлы 240](#_Toc193280750)

[Д.3 SNMP-трапы 242](#_Toc193280751)

[Д.3.1 Настройка SNMP-трапов 243](#_Toc193280752)

[Д.3.2 Настройка мониторинга SNMP-трапов 244](#_Toc193280753)

[Д.3.3 Требования к системе 246](#_Toc193280754)

[Д.3.4 Примеры с использованием разных версий SNMP-протоколов 247](#_Toc193280755)

[Д.4 Проверки IPMI 249](#_Toc193280756)

[Д.4.1 Настройка 249](#_Toc193280757)

[Д.4.2 Время ожидания и завершение сессии 251](#_Toc193280758)

[Д.4.3 Дискретные датчики IPMI 251](#_Toc193280759)

[Д.5 Простые проверки 255](#_Toc193280760)

[Д.5.1 ICMP-пинг 258](#_Toc193280761)

[Д.6 Мониторинг файлов журналов 260](#_Toc193280762)

[Д.6.1 Настройка 260](#_Toc193280763)

[Д.6.2 Извлечение совпадающей части регулярного выражения 265](#_Toc193280764)

[Д.6.3 Использование параметра "максзадержка" 265](#_Toc193280765)

[Д.6.4 Обработка ротации "copytruncate" файлов журналов 267](#_Toc193280766)

[Д.6.5 Постоянные файлы у log\*[]-элементов данных при нагрузке на I/O 267](#_Toc193280767)

[Д.6.6 Действия при ошибке связи между Агентом и Сервером 268](#_Toc193280768)

[Д.6.7 Обработка ошибок компиляции и времени выполнения для регулярных выражений 268](#_Toc193280769)

[Д.7 Вычисляемые элементы данных 272](#_Toc193280770)

[Д.7.1 Настраиваемые поля 272](#_Toc193280771)

[Д.7.2 Агрегированные вычисления 274](#_Toc193280772)

[Д.8 Внутренние проверки 277](#_Toc193280773)

[Д.9 Проверки через SSH 288](#_Toc193280774)

[Д.9.1 Диалог настройки командной строки 290](#_Toc193280775)

[Д.9.2 Настройка элемента данных 291](#_Toc193280776)

[Д.10 Проверки через Telnet 294](#_Toc193280777)

[Д.11 Внешние проверки 294](#_Toc193280778)

[Д.11.1 Результат внешней проверки 295](#_Toc193280779)

[Д.12 Траппер-элементы данных 296](#_Toc193280780)

[Д.13 JMX-мониторинг 298](#_Toc193280781)

[Д.13.1 Включение удаленного JMX-мониторинга для Java-приложений 298](#_Toc193280782)

[Д.13.2 Настройка JMX-интерфейсов и элементов данных в веб-интерфейсе 300](#_Toc193280783)

[Д.13.3 Детальная информация о ключах JMX-элементов данных 302](#_Toc193280784)

[Д.14 ODBC-мониторинг 303](#_Toc193280785)

[Д.14.1 Установка unixODBC 303](#_Toc193280786)

[Д.14.2 Установка драйверов unixODBC 304](#_Toc193280787)

[Д.14.3 Настройка unixODBC 304](#_Toc193280788)

[Д.14.4 Компиляция с поддержкой ODBC 305](#_Toc193280789)

[Д.14.5 Настройка элемента данных в веб-интерфейсе 306](#_Toc193280790)

[Д.14.6 Сообщения об ошибках 308](#_Toc193280791)

[Д.14.7 Рекомендуемые настройки UNIXODBC для MySQL 308](#_Toc193280792)

[Д.14.8 Рекомендуемые настройки UNIXODBC для PostgreSQL 308](#_Toc193280793)

[Д.14.9 Рекомендуемые настройки UNIXODBC для MSSQL 309](#_Toc193280794)

[Д.15 Зависимые элементы данных 309](#_Toc193280795)

[Д.16 HTTP-Агент 313](#_Toc193280796)

[Д.17 Проверки Prometheus 318](#_Toc193280797)

[Д.18 Скриптовые элементы данных 321](#_Toc193280798)

[Д.19 Браузер 323](#_Toc193280799)

[Приложение Е. Пользовательские команды 325](#_Toc193280800)

[Е.1 zabbix\_agent2 325](#_Toc193280801)

[Е.2 zabbix\_agentd 326](#_Toc193280802)

[Е.3 zabbix\_get 328](#_Toc193280803)

[Е.4 zabbix\_js 329](#_Toc193280804)

[Е.5 zabbix\_proxy 330](#_Toc193280805)

[Е.6 zabbix\_sender 331](#_Toc193280806)

[Е.7 zabbix\_server 336](#_Toc193280807)

[Е.8 zabbix\_web\_service 338](#_Toc193280808)

[Приложение Ж. Поддерживаемые функции 339](#_Toc193280809)

[Ж.1 Функции агрегации 339](#_Toc193280810)

[Ж.2 Функции цикла 345](#_Toc193280811)

[Ж.3 Побитовые функции 349](#_Toc193280812)

[Ж.4 Функции даты и времени 351](#_Toc193280813)

[Ж.5 Функции истории 352](#_Toc193280814)

[Ж.6 Функции динамики изменений 360](#_Toc193280815)

[Ж.7 Математические функции 366](#_Toc193280816)

[Ж.8 Операторные функции 372](#_Toc193280817)

[Ж.9 Функции прогнозирования 373](#_Toc193280818)

[Ж.10 Строковые функции 376](#_Toc193280819)

[Приложение И. Макросы 381](#_Toc193280820)

[И.1 Поддерживаемые макросы 381](#_Toc193280821)

[И.2 Нумерованные макросы 431](#_Toc193280822)

[И.3 Пользовательские макросы, поддерживаемые по местоположению 432](#_Toc193280823)

[Приложение К. Прочее 440](#_Toc193280824)

[К.1 Символы единиц измерения 440](#_Toc193280825)

[К.2 Настройка периодов времени 442](#_Toc193280826)

[К.3 Выполнение команд 443](#_Toc193280827)

[К.4 Обработка ошибок базы данных 444](#_Toc193280828)

[К.5 Динамическая библиотека sender для Windows 445](#_Toc193280829)

[К.6 Библиотека Python для API 446](#_Toc193280830)

[К.7 Вход в систему и systemd 446](#_Toc193280831)

[К.8 Отличия между Агентом-1 и Агентом-2 448](#_Toc193280832)

[К.9 Примеры экранирования 450](#_Toc193280833)

[Перечень терминов и сокращений 454](#_Toc193280834)

Приложение А. Конфигурация демонов

Примечание – Следует обратить внимание, что значения по умолчанию отражают значения по умолчанию самого демона, а не значения в поставляемых файлах конфигурации. Подсистема поддерживает файлы конфигурации только в кодировке UTF-8 без BOM. Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

1. Сервер

Параметры конфигурации Сервера описаны в таблице 1.

Сервер

| Параметр | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- |
| AlertScriptsPath | нет |  | /usr/local/share/zabbix/alertscripts | Местоположение пользовательских скриптов оповещений (зависит от указанной при компиляции переменной datadir). |
| AllowRoot | нет |  | 0 | Разрешение Серверу запускаться под root. Если отключено, а Сервер запускается из-под root, Сервер вместо этого будет пытаться переключиться на пользователя zabbix. Игнорируется, если Сервер запускается под обычным пользователем. 0 – не разрешено 1 – разрешено |
| AllowUnsupportedDBVersions | нет |  | 0 | Разрешение Серверу работать с неподдерживаемыми версиями баз данных. 0 – не разрешено 1 – разрешено |
| CacheSize | нет | 128K-64G | 32M | Размер кэша конфигурации, в байтах. Размер разделяемой памяти (shared memory) для хранения данных узлов сети, элементов данных и триггеров. |
| CacheUpdateFrequency | нет | 1-3600 | 60 | Частота выполнения Подсистемой процедуры обновления кэша конфигурации, в секундах. |
| DBHost | нет |  | localhost | Имя хоста базы данных. В случае MySQL localhost или пустая строка приведет к использованию сокета. В случае PostgreSQL только пустая строка приведет к попытке использования сокета. В случае Oracle пустая строка приведет к использованию метода подключения "Имя сетевой службы" (Net Service Name); в этом случае рассмотреть использование переменной окружения TNS\_ADMIN, чтобы указать директорию с файлом tnsnames.ora. |
| DBName | да |  |  | Имя базы данных. В случае Oracle, если используется метод подключения "Имя сетевой службы" (Net Service Name), указать имя службы из файла tnsnames.ora или выставить в пустую строку; если DBName выставлено в пустую строку, то задать переменную окружения "TWO\_TASK". |
| DBPassword | нет |  |  | Пароль к базе данных. Закомментировать эту строку, если пароль не используется. |
| DBPort | нет | 1024-65535 |  | Порт базы данных, когда не используется локальный сокет. В случае Oracle, если используется метод подключения "Имя сетевой службы" (Net Service Name), данный параметр игнорируется; вместо него будет использован номер порта из файла tnsnames.ora. |
| DBSchema | нет |  |  | Имя схемы. Используется для PostgreSQL. |
| DBSocket | нет |  |  | Путь к файлу сокета MySQL. |
| DBUser | нет |  |  | Пользователь базы данных. |
| DBTLSConnect | нет |  |  |  |
| DBTLSCAFile | нет (да, если DBTLSConnect выставлен в одно из: verify\_ca, verify\_full) |  |  | Полный путь к файлу, содержащему сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата БД. |
| DBTLSCertFile | нет |  |  | Полный путь к файлу, содержащему сертификат Сервера для аутентификации к базе данных. |
| DBTLSKeyFile | нет |  |  | Полный путь к файлу, содержащему закрытый ключ (private key) для аутентификации к базе данных. |
| DBTLSCipher | нет |  |  | Список алгоритмов шифрования, разрешенных со стороны Сервера для протокола TLS до версии TLSv1.2 включительно. Поддерживается только для MySQL. |
| DBTLSCipher13 | нет |  |  | Список алгоритмов шифрования, разрешенных со стороны Сервера для протокола TLSv1.3. Поддерживается только для MySQL, начиная с версии 8.0.16. |
| DebugLevel | нет | 0-5 | 3 | Задает уровень журналирования: 0 – основная информация о запуске и остановке процессов Подсистемы; 1 – критичная информация; 2 – информация об ошибках; 3 – предупреждения; 4 – для отладки (записывается очень много информации); 5 – расширенная отладка (записывается еще больше информации). |
| ExportDir | нет |  |  | Директория для экспорта в режиме реального времени событий, истории и динамики изменений в JSON-формате с разделением новой строкой. Если задано, активируется экспорт в режиме реального времени. |
| ExportFileSize | нет | 1M-1G | 1G | Максимальный размер каждого файла с экспортированными данными в байтах. Используется только для ротации, если задан параметр ExportDir. |
| ExportType | нет |  |  | Список (через запятую) типов объектов (событий, истории и динамики изменений – events, history, trends) для экспорта в режиме реального времени (по умолчанию – все типы). Имеет смысл, только если задано значение ExportDir. Обратить внимание, что если ExportType задано, а ExportDir – нет, то это ошибка конфигурации и Сервер не запустится. Например: ExportType=history,trends – экспортировать только данные истории и динамики изменений; ExportType=events – экспортировать только события. |
| ExternalScripts | нет |  | /usr/local/share/zabbix/externalscripts | Расположение внешних скриптов (зависит от указанной при компиляции переменной datadir). |
| Fping6Location | нет |  | /usr/sbin/fping6 | Расположение fping6. Следует убедиться, что владельцем бинарного файла fping6 является root и флаг SUID установлен. Оставить пустым ("Fping6Location="), если утилита fping поддерживает обработку адресов IPv6. |
| FpingLocation | нет |  | /usr/sbin/fping | Расположение fping. Следует убедиться, что владельцем бинарного файла fping6 является root и флаг SUID установлен. |
| HANodeName | нет |  |  | Имя ноды кластера высокой доступности. Если пусто, Сервер работает в стандартном режиме и создается нода с пустым именем. |
| HistoryCacheSize | нет | 128K-2G | 16M | Размер кэша истории, в байтах. Размер разделяемой памяти для хранения данных истории. |
| HistoryIndexCacheSize | нет | 128K-2G | 4M | Размер кэша индекса истории, в байтах. Размер разделяемой памяти для индексации данных истории, хранящихся в кэше истории. Кэшу индекса необходимо около 100 байтов на кэширование одного элемента данных. |
| HistoryStorageDateIndex | нет |  | 0 | Включение предварительной обработки значений истории в хранилище истории для сохранения значений в разных индексах на основе даты: 0 – выключено; 1 – включено. |
| HistoryStorageURL | нет |  |  | HTTP[S] URL хранилища истории. Этот параметр используется для настройки Elasticsearch. |
| HistoryStorageTypes | нет |  | uint,dbl,str,log,text | Список (через запятую) типов значений для отправки в хранилище истории. Этот параметр используется для настройки Elasticsearch. |
| HousekeepingFrequency | нет | 0-24 | 1 | Как часто Подсистема будет выполнять процедуру очистки базы (в часах). Автоматическая очистка базы данных удаляет из базы устаревшую информацию. Обратить внимание: для предотвращения перегрузки функции очистки (например, если периоды хранения данных истории и динамики изменения сильно уменьшены), не более чем 4 периода кратных HousekeepingFrequency часов устаревшей истории будет удалено за один цикл очистки по каждому элементу данных. Таким образом, если HousekeepingFrequency равен 1 часу, то за один цикл будет удалено не более чем 4 часа устаревшей истории (начиная с самой старой записи). Примечание: для снижения нагрузки на Сервер запуск процесса очистки отложен на 30 минут после запуска Сервера. Таким образом, если HousekeepingFrequency равен 1 часу, самая первая процедура очистки запустится через 30 минут после запуска Сервера, и затем будет повторяться с задержкой в один час. Имеется возможность отключить автоматическую очистку истории, указав HousekeepingFrequency равным 0. В этом случае процедуру очистки истории можно запустить только с помощью опции контроля управления housekeeper\_execute и периодом удаления устаревшей информации является 4-кратный период, начиная с последнего цикла удаления истории, но не менее чем 4 часа и не более 4 дней. |
| Include | нет |  |  | Можно включить в файл конфигурации отдельные файлы или все файлы из директории. Для включения только необходимых файлов из указанной директории поддерживается символ звездочки для поиска совпадения по маске. Например: /absolute/path/to/config/files/\*.conf. |
| JavaGateway | нет |  |  | IP-адрес (или имя хоста) Java gateway. Требуется, только если запущены поллеры Java. |
| JavaGatewayPort | нет | 1024-32767 | 10052 | Порт, который слушает Java gateway. |
| LISTENBacklog | нет | 0 – INT\_MAX | SOMAXCONN | Максимальное количество соединений в состоянии ожидания в очереди TCP. Значение по умолчанию является зашитой в коде константой, которая зависит от системы. Максимально поддерживаемое значение зависит от системы, слишком большие значения могут быть урезаны до "указанного в реализации максимума" ("implementation-specified maximum"). |
| LISTENIP | нет |  | 0.0.0.0 | Список (через запятую) IP-адресов, которые должен слушать траппер. Если этот параметр не указан, траппер будет слушать все сетевые интерфейсы. |
| LISTENPort | нет | 1024-32767 | 10051 | Порт, который слушает траппер. |
| LoadModule | нет |  |  | Модули, которые загружаются во время старта. Модули используются для расширения возможностей Сервера. Форматы: LoadModule=<module.so> LoadModule=<path/module.so> LoadModule=</abs\_path/module.so> Либо модули должны находиться в директории, указанной параметром LoadModulePath, либо путь должен быть указан перед именем модуля. Если указанный путь является абсолютным (начинается с "/"), то параметр LoadModulePath игнорируется. Допускается добавлять несколько параметров LoadModule. |
| LoadModulePath | нет |  |  | Абсолютный путь к директории с серверными модулями. По умолчанию зависит от опций компиляции. |
| LogFile | да, если LogType выставлен в file, иначе нет |  |  | Имя файла журнала. |
| LogFileSize | нет | 0-1024 | 1 | Максимальный размер файла журнала в МБ. 0 – отключение автоматической ротации журнала. Примечание: если лимит размера файла достигнут, а ротация по каким-либо причинам не удалась, существующий файл журнала усекается и начинается заново. |
| LogType | нет |  | file | Тип вывода журнала: – file – запись журнала в файл, указанный в параметре LogFile; – system – запись журнала в syslog; – console – вывод журнала в стандартный вывод. |
| LogSlowQueries | нет | 0-3600000 | 0 | Насколько долго могут выполняться запросы к БД до того, как они будут записаны в журнал (в миллисекундах). 0 – не журналировать медленные запросы. Эта опция активируется, начиная с DebugLevel=3. |
| MaxHousekeeperDelete | нет | 0-1000000 | 5000 | Удалять не более MaxHousekeeperDelete строк (в соответствии с [имя\_таблицы], [поле], [значение]) на одну задачу за один цикл автоматической очистки базы данных. Если значение равно 0, то ограничение не используется совсем. В этом случае нужна осторожность, чтобы не перегрузить базу данных.  Этот параметр применяется только для удаления данных истории и динамики изменений уже удаленных элементов данных. |
| NodeAddress | нет |  | 10051 | IP-адрес или имя хоста с опциональным номером порта, чтобы переопределить, каким образом веб-интерфейс должен подключаться к Серверу. Формат: <адрес>[:порт] Если IP или имя хоста не заданы, будет взято значение параметра LISTENIP. Если и LISTENIP на задано, будет использовано значение localhost. Эта опция может быть переопределена адресом, указанным в конфигурации веб-интерфейса. См. также параметр HANodeName. |
| PidFile | нет |  | /tmp/zabbix\_server.pid | Имя PID-файла. |
| ProblemHousekeepingFrequency | нет | 1-3600 | 60 | Как часто Подсистема будет удалять проблемы для удаленных триггеров (в секундах). |
| ProxyConfigFrequency | нет | 1-604800 | 3600 | Как часто Сервер будет отправлять данные конфигурации на Прокси (в секундах). Используется только с Прокси, которые работают в пассивном режиме. |
| ProxyDataFrequency | нет | 1-3600 | 1 | Как часто Сервер будет запрашивать данные истории с Прокси (в секундах). Используется только с Прокси, которые работают в пассивном режиме. |
| ServiceManagerSyncFrequency | нет | 1-3600 | 60 | Как часто Подсистема будет синхронизировать конфигурацию менеджера услуг (в секундах). |
| SNMPTrapperFile | нет |  | /tmp/zabbix\_traps.tmp | Временный файл, используемый для передачи данных Серверу от демона SNMP trap. Должен быть таким же, как и в файле zabbix\_trap\_receiver.pl или файле конфигурации SNMPTT. |
| SocketDir | нет |  | /tmp | Директория для хранения IPC-сокетов, которые используются внутренними службами Подсистемы. |
| SourceIP | нет |  |  | Локальный IP-адрес для: – исходящих подключений к Прокси и Агенту; – безагентных подключений (VMware, SSH, JMX, SNMP, Telnet и простые проверки); – подключений HTTP-Агента; – запросов HTTP из JavaScript для скриптовых элементов данных; – запросов HTTP из JavaScript во время предобработки; – отсылки почтовых уведомлений (соединения к серверу SMTP); – оповещений вебхук (HTTP соединения из JavaScript); – подключений к Хранилищу. |
| SSHKeyLocation | нет |  |  | Местоположение открытых и закрытых ключей для проверок и действий SSH. |
| SSLCertLocation | нет |  |  | Местоположение файлов клиентских сертификатов SSL для аутентификации клиента. Этот параметр используется только в веб-мониторинге. |
| SSLKeyLocation | нет |  |  | Местоположение файлов закрытых ключей SSL для аутентификации клиента. Этот параметр используется только в веб-мониторинге. |
| SSLCALocation | нет |  |  | Переопределение местоположения файлов удостоверяющих центров (certificate authority, CA) для верификации SSL-сертификатов Сервера. Если не задано, будет использоваться общесистемная директория. Обратить внимание, что значение этого параметра задаст опцию libcurl CURLOPT\_CAPATH. Для libcurl версий до 7.42.0 эта опция имеет эффект, только если libcurl скомпилирован для использования OpenSSL. Для получения более подробной информации см. [веб-страницу cURL [en]](http://curl.haxx.se/libcurl/c/CURLOPT_CAPATH.html). |
| StartAlerters | нет | 1-100 | 3 | Количество экземпляров запускаемых при старте процессов оповещений (alerter). |
| StartDBSyncers | нет | 1-100 | 4 | Количество экземпляров запускаемых при старте процессов записи истории в базу данных (history syncer). Примечание: следует осторожно изменять это значение, т.к. его увеличение может принести больше вреда, чем пользы. Значения по умолчанию должно быть достаточно, чтобы обслуживать до 4000 NVPS. |
| StartDiscoverers | нет | 0-250 | 1 | Количество экземпляров запускаемых при старте процессов сетевого обнаружения (discoverer). |
| StartEscalators | нет | 1-100 | 1 | Количество экземпляров запускаемых при старте процессов эскалации действий (escalator). |
| StartHistoryPollers | нет | 0-1000 | 5 | Количество экземпляров запускаемых при старте процессов поллеров истории (history poller). |
| StartHTTPPollers | нет | 0-1000 | 1 | Количество экземпляров запускаемых при старте процессов поллеров HTTP (http poller). |
| StartIPMIPollers | нет | 0-1000 | 0 | Количество экземпляров запускаемых при старте процессов поллеров IPMI (ipmi poller). |
| StartJavaPollers | нет | 0-1000 | 0 | Количество экземпляров запускаемых при старте процессов поллеров Java (java poller). |
| StartLLDProcessors | нет | 1-100 | 2 | Количество экземпляров запускаемых при старте процессов низкоуровневого обнаружения (lld worker). Процесс LLD manager автоматически запускается при старте процесса LLD worker. |
| StartODBCPollers | нет | 0-1000 | 1 | Количество экземпляров запускаемых при старте процессов поллеров ODBC (odbc poller). |
| StartPingers | нет | 0-1000 | 1 | Количество экземпляров запускаемых при старте процессов пингеров ICMP (icmp pinger). |
| StartPollersUnreachable | нет | 0-1000 | 1 | Количество экземпляров запускаемых при старте процессов поллеров недоступных устройств (unreachable poller) (в том числе IPMI и Java). По крайне один поллер недоступных устройств должен быть запущен, если запускаются обычные поллеры, IPMI или Java. |
| StartPollers | нет | 0-1000 | 5 | Количество экземпляров запускаемых при старте процессов поллеров (poller). |
| StartPreprocessors | нет | 1-1000 | 3 | Количество экземпляров запускаемых при старте рабочих процессов предобработки данных (preprocessing worker). Процесс менеджера задач предобработки автоматически запускается при старте процесса preprocessing worker. |
| StartProxyPollers | нет | 0-250 | 1 | Количество экземпляров запускаемых при старте процессов поллеров пассивных Прокси (proxy poller). |
| StartReportWriters | нет | 0-100 | 0 | Количество экземпляров запускаемых при старте процессов генерации отчетов (report writer). Если выставлено в 0, генерация отчетов по расписанию отключена. Процесс задач генерации отчетов по расписанию автоматически запускается при старте процесса report writer. |
| StartSNMPTrapper | нет | 0-1 | 0 | Если выставлено в 1, процесс траппера SNMP (snmp trapper) будет запущен. |
| StartTimers | нет | 1-1000 | 1 | Количество экземпляров запускаемых при старте процессов timer. Процессы timer обрабатывают периоды обслуживания. |
| StartTrappers | нет | 0-1000 | 5 | Количество экземпляров запускаемых при старте процессов трапперов (trapper). Трапперы принимают входящие соединения от sender, активных Агентов и активных Прокси. |
| StartVMwareCollectors | нет | 0-250 | 0 | Количество экземпляров запускаемых при старте процессов сборщиков данных VMware (vmware collector). |
| StatsAllowedIP | нет |  |  | Список (через запятую) IP-адресов, возможно в нотации CIDR, или DNS-имен внешних экземпляров Подсистемы. Запрос статистики будет приниматься только с перечисленных здесь адресов. Если параметр не задан, никакие запросы статистики приниматься не будут. Если поддержка IPv6 включена, то "127.0.0.1", "::127.0.0.1", "::ffff:127.0.0.1" рассматриваются как одинаковые, а "::/0" разрешит любой адрес IPv4 или IPv6. Можно использовать "0.0.0.0/0", чтобы разрешить любой адрес IPv4. Пример: StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mon.rosa.dev |
| Timeout | нет | 1-30 | 3 | Указывает, как долго ожидается ответ от Агента, SNMP устройства или внешней проверки (в секундах). |
| TLSCAFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCertFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификат или цепочку сертификатов, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCipherAll | нет |  |  | Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов и общего ключа (PSK). Пример: TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256 |
| TLSCipherAll13 | нет |  |  | Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов и общего ключа (PSK). Пример для GnuTLS: NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL::+SIGN-ALL:+CTYPE-X.509 Пример для OpenSSL: EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128 |
| TLSCipherCert | нет |  |  | Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов. Пример для GnuTLS: NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509 Пример для OpenSSL: EECDH+aRSA+AES128:RSA+aRSA+AES128 |
| TLSCipherCert13 | нет |  |  | Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов. |
| TLSCipherPSK | нет |  |  | Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе общего ключа (PSK). Пример для GnuTLS: NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL Пример для OpenSSL: kECDHEPSK+AES128:kPSK+AES128 |
| TLSCipherPSK13 | нет |  |  | Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе общего ключа (PSK). Пример: TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256 |
| TLSCRLFile | нет |  |  | Абсолютный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSKeyFile | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ Сервера (private key), используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TmpDir | нет |  | /tmp | Директория с временными данными. |
| TrapperTimeout | нет | 1-300 | 300 | Указывает, сколько секунд траппер может потратить на обработку новых данных. |
| TrendCacheSize | нет | 128K-2G | 4M | Размер кэша динамики изменений, в байтах. Размер разделяемой памяти для хранения данных динамики изменений. |
| TrendFunctionCacheSize | нет | 128K-2G | 4M | Размер кэша функций динамики изменений, в байтах. Размер разделяемой памяти для хранения данных вычисляемых функций динамики изменений. |
| UnavailableDelay | нет | 1-3600 | 60 | Как часто узел сети будет проверяться на доступность в период его недоступности, в секундах. |
| UnreachableDelay | нет | 1-3600 | 15 | Как часто узел сети будет проверяться на доступность в период его недостижимости, в секундах. |
| UnreachablePeriod | нет | 1-3600 | 45 | Через сколько секунд недостижимости узел сети считается недоступным. |
| User | нет |  | zabbix | Использование привилегий указанного, существующего пользователя системы. Имеет эффект, только если запускается под root и AllowRoot отключен. |
| ValueCacheSize | нет | 0,128K-64G | 8M | Размер кэша для хранения значений истории в байтах. Разделяемая память для кэширования запросов к данным истории элементов данных. Значение 0 отключит кэш значений (не рекомендуется). В случае, если кэш значений исчерпает разделяемую память, предупреждающее сообщение будет записано в журнал Сервера каждые 5 минут. |
| VaultDBPath | нет |  |  | Путь к Хранилищу, откуда будут извлекаться данные для аутентификации к базе данных по ключам "password" и "username". Пример: secret/zabbix/database Этот параметр может использоваться, только если параметры DBUser и DBPassword не указаны. |
| VaultToken | нет |  |  | Токен аутентификации Хранилища, который должен быть сгенерирован исключительно для Сервера с разрешениями только на чтение на пути, указанные в макросах Хранилища, и разрешениями только на чтение на пути, указанные в необязательном параметре конфигурации VaultDBPath. Если одновременно заданы и параметр VaultToken, и переменная окружения VAULT\_TOKEN, то это считается ошибкой. |
| VaultURL | нет |  | https://127.0.0.1:8200 | HTTP[S] URL сервера Хранилища. Если SSLCALocation не указан, будет использоваться общесистемная директория сертификатов удостоверяющих центров (certificate authority, CA). |
| VMwareCacheSize | нет | 256K-2G | 8M | Разделяемая память для хранения данных VMware. Можно использовать внутренние проверки VMware – zabbix[vmware,buffer,...] для мониторинга использования кэша VMware. Обратить внимание, что разделяемая память не выделяется, если не настроено на запуск ни одного экземпляра коллекторов VMware. |
| VMwareFrequency | нет | 10-86400 | 60 | Задержка в секундах между сбором данных с одной службы VMware. Эту задержку необходимо установить в наименьшее из значений интервала обновления у любого элемента данных мониторинга VMware. |
| VMwarePerfFrequency | нет | 10-86400 | 60 | Задержка в секундах между получением статистики по счетчикам производительности с одной службы VMware. Эту задержку необходимо установить в наименьшее из значений интервала обновления у любого элемента данных мониторинга VMware, который использует счетчики производительности VMware. |
| VMwareTimeout | нет | 1-300 | 10 | Максимальное количество секунд, которое коллектор vmware потратит на ожидание ответа от службы VMware (vCenter или ESX гипервизора). |
| WebServiceURL | нет |  |  | HTTP[S] URL к веб-сервису в формате <хост:порт>/report. Например: http://localhost:10053/report |

1. Агент-1 (UNIX)

Параметры конфигурации демона Агента-1 (UNIX) описаны в таблице 2.

Агент-1 (UNIX)

| Параметр | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- |
| Alias | нет |  |  | Задает алиас для ключа элемента данных. Его можно использовать для замены длинных и сложных ключей элементов данных на более короткие и простые. Можно добавлять несколько параметров Alias. Несколько параметров с одинаковым ключом Alias не разрешены. Различные ключи параметра Alias могут ссылаться на тот же ключ элемента данных. Алиасы можно использовать в параметре HostMetadataItem, но нельзя в параметре HostnameItem. |
| AllowKey | нет |  |  | Разрешение выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с DenyKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| AllowRoot | нет |  | 0 | Разрешение Агенту запускаться под root. Если отключено, а Агент запускается из-под root, Агент вместо этого будет пытаться переключиться на пользователя zabbix. Игнорируется, если Агент запускается под обычным пользователем. 0 – не разрешено 1 – разрешено |
| BufferSend | нет | 1-3600 | 5 | Не хранить данные в буфере дольше N секунд. |
| BufferSize | нет | 2-65535 | 100 | Максимальное количество значений в буфере памяти. При заполнении буфера Агент будет отправлять все собранные данные Серверу или Прокси. |
| DebugLevel | нет | 0-5 | 3 | Задает уровень журналирования: 0 – основная информация о запуске и остановке процессов Подсистемы; 1 – критичная информация; 2 – информация об ошибках; 3 – предупреждения; 4 – для отладки (записывается очень много информации); 5 – расширенная отладка (записывается еще больше информации). |
| DenyKey | нет |  |  | Запрет выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с AllowKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| EnableRemoteCommands | нет |  | 0 | Разрешены ли удаленные команды с Сервера. Этот параметр устарел, использовать вместо него AllowKey=system.run[\*] или DenyKey=system.run[\*]. Это внутренний алиас для параметров AllowKey/DenyKey, зависящий от значения: 0 – DenyKey=system.run[\*] 1 – AllowKey=system.run[\*] |
| HostInterface | нет | 0-255 символов |  | Опциональный параметр, который задает интерфейс узла сети. Интерфейс узла сети используется в процессе авторегистрации узла сети. Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов. Если не задано, значение будет получено из HostInterfaceItem. |
| HostInterfaceItem | нет |  |  | Опциональный параметр, который задает элемент данных, используемый для получения интерфейса узла сети. Интерфейс узла сети используется в процессе авторегистрации узла сети. Во время запроса на авторегистрацию Агент будет выводить в журнал предупреждение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов. Элемент данных system.run[] поддерживается независимо от значений параметров AllowKey/DenyKey. Данный параметр используется, только если HostInterface не определен. |
| HostMetadata | нет | 0-255 символов |  | Опциональный параметр, который задает метаданные узла сети. Метаданные узла сети используются только в процессе автоматической регистрации узлов сети (активный Агент). Если не определено, то значение берется от HostMetadataItem. Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение длины строки или не является UTF-8-строкой. |
| HostMetadataItem | нет |  |  | Опциональный параметр, который задает элемент данных Агента, используемый для получения метаданных узла сети. Этот параметр используется, только если HostMetadata не определен. Поддерживаются UserParameters и Alias. Поддерживается system.run[] независимо от значений AllowKey/DenyKey. Значение HostMetadataItem извлекается на каждую попытку авторегистрации и используется только в процессе автоматической регистрации узлов сети (активный Агент). При запросе на авторегистрацию Агент запишет в журнал предупреждение, если полученное от указанного элемента данных значение превышает ограничение в 255 символов. Значение, возвращаемое указанным элементом данных, должно являться UTF-8-строкой, иначе оно будет игнорироваться. |
| Hostname | нет |  | Задается параметром HostnameItem | Список (через запятую) уникальных регистрозависимых имен узла сети. Требуется для активных проверок и должно совпадать с именами узла сети, указанными на Сервере. Если не задано, значение будет браться из HostnameItem. Допустимые символы: буквенно-цифровые, ".", " ", "\_" и "-". Максимальная длина: 128 символов на имя узла сети, 2048 символов для всей строки. |
| HostnameItem | нет |  | system.Hostname | Опциональный параметр, который задает элемент данных Агента, используемый для получения имени узла сети. Этот параметр используется, только если Hostname не определен. Не поддерживает UserParameters или Alias, но поддерживает system.run[], независимо от значений AllowKey/DenyKey. |
| Include | нет |  |  | Можно включить в файл конфигурации отдельные файлы или все файлы из директории. Для включения только необходимых файлов из указанной директории, поддерживается символ звездочки для поиска совпадения по маске. Например: /absolute/path/to/config/files/\*.conf. |
| LISTENBacklog | нет | 0 – INT\_MAX | SOMAXCONN | Максимальное количество соединений в состоянии ожидания в очереди TCP. Значение по умолчанию является зашитой в коде константой, которая зависит от системы. Максимально поддерживаемое значение зависит от ОС, слишком большие значения могут быть урезаны до "указанного в реализации максимума" ("implementation-specified maximum"). |
| LISTENIP | нет |  | 0.0.0.0 | Список (через запятую) IP-адресов, которые должен слушать Агент. |
| LISTENPort | нет | 1024-32767 | 10050 | Агент будет слушать этот порт для подключений с Сервера. |
| LoadModule | нет |  |  | Модули, которые загружаются на Агенте во время старта. Модули используются для расширения возможностей Агента. Форматы: LoadModule=<module.so> LoadModule=<path/module.so> LoadModule=</abs\_path/module.so> Либо модули должны находиться в директории, указанной параметром LoadModulePath, либо путь должен быть указан перед именем модуля. Если указанный путь является абсолютным (начинается с "/"), то параметр LoadModulePath игнорируется. Допускается добавлять несколько параметров LoadModule. |
| LoadModulePath | нет |  |  | Абсолютный путь к директории с модулями Агента. По умолчанию зависит от опций компиляции. |
| LogFile | Да, если LogType задан как file, иначе – нет. |  |  | Имя файла журнала. |
| LogFileSize | нет | 0-1024 | 1 | Максимальный размер файла журнала в МБ. 0 – отключение автоматической ротации журнала. Примечание: если лимит размера файла достигнут, а ротация по каким-либо причинам не удалась, существующий файл журнала усекается и начинается заново. |
| LogType | нет |  | file | Тип вывода журнала: – file – запись журнала в файл, указанный в параметре LogFile; – system – запись журнала в syslog; – console – запись журнала в стандартный вывод. |
| LogRemoteCommands | нет |  | 0 | Включение журналирования выполняемых оболочкой команд как предупреждений. 0 – отключено 1 – включено Команды будут записаны в журнал, только если они выполняются удаленно. Записи в журнале не создаются, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem. |
| MaxLinesPerSecond | нет | 1-1000 | 20 | Максимальное количество новых строк в секунду, которые Агент будет отправлять Серверу или Прокси при обработке активных проверок log и eventlog. Указанное значение будет переопределено параметром maxlines, указанным в ключах элементов данных log и eventlog. Обратить внимание: при поиске нужной строки в элементах данных журналов Подсистема будет обрабатывать в 10 раз больше новых строк, чем указано в параметре MaxLinesPerSecond. |
| PidFile | нет |  | /tmp/zabbix\_agentd.pid | Имя PID-файла. |
| RefreshActiveChecks | нет | 60-3600 | 120 | Как часто обновлять список активных проверок, в секундах. Обратить внимание, что после неуспешного обновления активных проверок, следующая попытка будет предпринята через 60 секунд. |
| Server | да, если StartAgents явно не выставлено в 0 |  |  | Список (через запятую) IP-адресов , опционально в CIDR-нотации, или имен хостов Серверов и Прокси. Входящие соединения будут приниматься только с хостов, указанных в этом списке. Если включена поддержка IPv6, то "127.0.0.1", "::127.0.0.1", "::ffff:127.0.0.1" обрабатываются одинаково, а "::/0" разрешает все IPv4 и IPv6-адреса. "0.0.0.0/0" можно использовать, чтобы разрешить любой IPv4-адрес. Обратить внимание, что "IPv4-совместимые IPv6-адреса" (префикс 0000::/96) поддерживаются, но являются устаревшими согласно [RFC4291 [en]](https://tools.ietf.org/html/rfc4291#section-2.5.5). Пример: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mon.rosa.dev Пробелы разрешены. |
| ServerActive | нет |  |  | Адрес Сервера/Прокси или конфигурация кластера для получения активных проверок. Адрес Сервера/Прокси – это IP-адрес или DNS-имя и опционально порт, разделенные двоеточием. Конфигурация кластера – один или более адресов Серверов, разделенные символом "точка с запятой". Можно указать несколько Серверов/Кластеров/Прокси, разделяя их запятой. От каждого Сервера/Кластера должно быть указано не более одного Прокси. Если указан Прокси, то Сервер/Кластер для этого Прокси не должны быть указаны. Несколько адресов через запятую можно указать для использования нескольких независимых Серверов параллельно. Пробелы разрешены. Если порт не указан, используется порт по умолчанию. Адреса IPv6 должны быть заключены в квадратные скобки, если для этого хоста указан порт. Если порт не указан, квадратные скобки для адресов IPv6 опциональны. Если этот параметр не указан, активные проверки отключены. |
| SourceIP | нет |  |  | Локальный IP-адрес для: - исходящих подключений к Серверу или Прокси; - создания подключений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т.п.) |
| StartAgents | нет | 0-100 | 3 | Количество экземпляров запускаемых при старте процессов zabbix\_agentd, обрабатывающих пассивные проверки. Если выставлено в 0, отключает пассивные проверки, и Агент не будет слушать никакие TCP порты. |
| Timeout | нет | 1-30 | 3 | Тратить не более Timeout секунд при обработке. |
| TLSAccept | да, если заданы TLS сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет |  |  | Какие входящие подключения принимать. Используется для пассивных проверок. Можно указывать несколько значений через запятую: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. |
| TLSCAFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCertFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCipherAll | нет |  |  | Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов и общего ключа (PSK). |
| TLSCipherAll13 | нет |  |  | Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов и общего ключа (PSK). |
| TLSCipherCert | нет |  |  | Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов. |
| TLSCipherCert13 | нет |  |  | Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе сертификатов. |
| TLSCipherPSK | нет |  |  | Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе общего ключа (PSK). |
| TLSCipherPSK13 | нет |  |  | Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора алгоритмов шифрования по умолчанию для шифрования на основе общего ключа (PSK). |
| TLSConnect | да, если заданы TLS -сертификат или параметры PSK (даже при незашифрованном соединении),иначе – нет |  |  | Как Агент должен подключаться к Серверу или Прокси. Используется для активных проверок. Можно указать только одно из значений: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. |
| TLSCRLFile | нет |  |  | Абсолютный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSKeyFile | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ Агента (private key), используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSPSKFile | нет |  |  | Абсолютный путь к файлу, содержащему общий ключ (pre-shared key) Агента. Используется для зашифрованных коммуникаций с Сервером. |
| TLSPSKIdentity | нет |  |  | Строка идентификатора PSK используется для зашифрованных соединений с Сервером. |
| TLSServerCertIssuer | нет |  |  | Разрешенный эмитент сертификата Сервера (Прокси). |
| TLSServerCertSubject | нет |  |  | Разрешенный субъект сертификата Сервера (Прокси). |
| UnsafeUserParameters | нет | 0,1 | 0 | Разрешить передавать аргументами в пользовательские параметры все символы. 0 – не разрешать 1 – разрешить Не разрешены следующие символы: \ " " ` \* ? [ ] { } ~ $ ! & ; ( ) < > | # @ Кроме того, не разрешены символы новой строки. |
| User | нет |  | zabbix | Использование привилегий указанного существующего пользователя системы. Имеет эффект, только если запускается под root и AllowRoot отключен. |
| UserParameter | нет |  |  | Пользовательский параметр для мониторинга. Можно указать несколько пользовательских параметров. Формат: UserParameter=<ключ>,<shell команда> Обратить внимание, что команда не должна возвращать только пустую строку или EOL. Команды оболочки могут иметь относительные пути, если указан параметр UserParameterDir. Примеры: UserParameter=system.test,who|wc -l UserParameter=check\_cpu,./custom\_script.sh |
| UserParameterDir | нет |  |  | Путь поиска по умолчанию для команд UserParameter. Если используется, Агент перед выполнением команды сменит свою рабочую директорию на указанную здесь. Тем самым, команды UserParameter могут иметь относительный префикс "./" вместо полного пути. Допустимо только одно значение. Пример: UserParameterDir=/opt/myscripts |

1. Агент-2 (UNIX)

Параметры конфигурации демона Агента-2 (UNIX) описаны в таблице 3.

Агент-2 (UNIX)

| Параметр | | Обязательный | Диапазон | По умолчанию | | Описание |
| --- | --- | --- | --- | --- | --- | --- |
| Alias | | нет |  |  | | Задает алиас для ключа элемента данных. Его можно использовать для замены длинных и сложных ключей элементов данных на более короткие и простые. Можно добавлять несколько параметров Alias. Несколько параметров с одинаковым ключом Alias не разрешены. Различные ключи параметра Alias могут ссылаться на тот же ключ элемента данных. Алиасы можно использовать в параметре HostMetadataItem, но нельзя в параметре HostnameItem. |
| AllowKey | | нет |  |  | | Разрешение выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с DenyKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| BufferSend | | нет | 1-3600 | 5 | | Интервал времени в секундах, определяющий, как часто значения из буфера отсылаются на Сервер. Обратить внимание: если буфер заполнен, данные будут отосланы раньше. |
| BufferSize | | нет | 2-65535 | 100 | | Максимальное количество значений в буфере памяти. При заполнении буфера Агент будет отправлять все собранные данные Серверу или Прокси. Этот параметр должен использоваться только если постоянный буфер отключен (EnablePersistentBuffer=0). |
| ControlSocket | | нет |  | /tmp/agent.sock | | Сокет управления, используется при отправке команд управления при помощи опции "-R". |
| DebugLevel | нет | | 0-5 | 3 | | Задает уровень журналирования: 0 – основная информация о запуске и остановке процессов Подсистемы; 1 – критичная информация; 2 – информация об ошибках; 3 – предупреждения; 4 – для отладки (записывается очень много информации); 5 – расширенная отладка (записывается еще больше информации). |
| DenyKey | нет | |  |  | | Запрет выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с AllowKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| EnablePersistentBuffer | нет | | 0-1 | 0 | | Включить использование локального постоянного хранилища для активных элементов данных. 0 – отключено 1 – включено Если постоянное хранилище отключено, будет использован буфер памяти. |
| ForceActiveChecksOnStart | нет | | 0-1 | 0 | | Выполнить активные проверки сразу после перезапуска для первой полученной конфигурации. 0 – отключено 1 – включено Также доступно как параметр конфигурации плагина, например: Plugins.Uptime.System.ForceActiveChecksOnStart=1 |
| HostInterface | нет | | 0-255 символов |  | | Необязательный параметр, определяющий интерфейс узла сети. Интерфейс узла сети используется в процессе автоматической регистрации узла сети. Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов. Если не определено, значение будет получено от HostInterfaceItem. |
| HostInterfaceItem | нет | |  |  | | Необязательный параметр, который определяет элемент данных, используемый для получения интерфейса узла сети. Интерфейс узла сети используется в процессе автоматической регистрации узла сети. Во время запроса на автоматическую регистрацию Агент регистрирует предупреждающее сообщение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов. Элемент данных system.run[] поддерживается независимо от значений параметров AllowKey/DenyKey. Эта опция используется, только если HostInterface не определен. |
| HostMetadata | нет | | 0-255 символов |  | | Опциональный параметр, который задает метаданные узла сети. Метаданные узла сети используются только в процессе автоматической регистрации узлов сети. Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение длины строки или не является UTF-8-строкой. Если не определено, то значение берется от HostMetadataItem. |
| HostMetadataItem | нет | |  |  | | Опциональный параметр, который задает элемент данных, используемый для получения метаданных узла сети. Значение HostMetadataItem извлекается на каждую попытку авторегистрации для процесса автоматической регистрации узла сети. При запросе на авторегистрацию Агент запишет в журнал предупреждение, если полученное от указанного элемента данных значение превышает ограничение в 255 символов. Этот параметр используется, только если HostMetadata не определен. Поддерживаются UserParameters и алиасы. Поддерживается system.run[] независимо от значений AllowKey/DenyKey. Значение, возвращаемое указанным элементом данных, должно являться UTF-8-строкой, иначе оно будет игнорироваться. |
| Hostname | нет | |  | Задается параметром HostnameItem | Список (через запятую) уникальных, регистрозависимых имен узла сети. Требуется для активных проверок и должно совпадать с именами узла сети, указанными на Сервере. Если не задано, значение будет браться из HostnameItem. Допустимые символы: буквенно-цифровые, ".", " ", "\_" и "-". Максимальная длина: 128 символов на имя узла сети, 2048 символов для всей строки. | |
| HostnameItem | нет | |  | system.Hostname | Элемент данных, который используется для формирования Hostname, если этот параметр не указан. Игнорируется, если задан параметр Hostname. Не поддерживает UserParameters или алиасы, но поддерживает system.run[], независимо от значений AllowKey/DenyKey. | |
| LISTENIP | нет | |  | 0.0.0.0 | Список (через запятую) IP-адресов, которые должен слушать Агент. Первый IP-адрес отсылается на Сервер при подключении к нему для получения списка активных проверок. | |
| LISTENPort | нет | | 1024-32767 | 10050 | Агент будет слушать этот порт для подключений с Сервера. | |
| LogFile | Да, если LogType задан как file, иначе – нет. | |  | /tmp/zabbix\_agent2.log | Имя файла журнала, если LogType равен значению "file". | |
| LogFileSize | нет | | 0-1024 | 1 | Максимальный размер файла журнала в МБ. 0 – отключение автоматической ротации журнала. Примечание: если лимит размера файла достигнут, а ротация по каким-либо причинам не удалась, существующий файл журнала усекается и начинается заново. | |
| LogType | нет | |  | file | Задает место, куда будут записываться сообщения журнала: – system – в syslog; – file – в файл, указанный параметром LogFile; – console – в стандартный вывод. | |
| PersistentBufferFile | нет | |  |  | Файл, в котором Агент-2 должен хранить базу данных SQLite. Имя файла должно быть указано с абсолютным путем. Этот параметр используется, только если включен постоянный буфер (EnablePersistentBuffer=1). | |
| PersistentBufferPeriod | нет | | 1m-365d | 1h | Период времени, в течение которого должны храниться данные, когда нет соединения с Сервером или Прокси. Более старые данные будут потеряны. Журнальные данные будут сохранены. Этот параметр используется, только если включен постоянный буфер (EnablePersistentBuffer=1). | |
| PidFile | нет | |  | /tmp/zabbix\_agent2.pid | Имя PID-файла. | |
| Plugin | нет | |  |  | Большинство плагинов имеют свои собственные файлы конфигурации. Файл конфигурации Агента содержит параметры плагинов, перечисленные ниже. | |
| Plugins.Log.MaxLinesPerSecond | нет | | 1-1000 | 20 | Максимальное количество новых строк, которые Агент будет отправлять в секунду на Сервер или Прокси при обработке активных проверок log и eventlog. Указанное значение будет переопределено параметром maxlines, указанным в ключах элементов данных log или eventlog. Обратить внимание: при поиске нужной строки в элементах данных журналов Подсистема будет обрабатывать в 10 раз больше новых строк, чем указано в параметре MaxLinesPerSecond. | |
| Plugins.SystemRun.LogRemoteCommands | нет | |  | 0 | Включение журналирования выполняемых оболочкой команд как предупреждений. 0 – отключено 1 – включено Команды будут записаны в журнал, только если они выполняются удаленно. Записи в журнале не создаются, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem. | |
| PluginSocket | нет | |  | /tmp/agent.plugin.sock | Путь к файлу UNIX-сокета для коммуникаций с подгружаемыми плагинами. | |
| PluginTimeout | нет | | 1-30 | Глобальный тайм-аут | Тайм-аут для соединений с подгружаемыми плагинами. | |
| RefreshActiveChecks | нет | | 60-3600 | 120 | Как часто обновлять список активных проверок, в секундах. Обратить внимание, что после неуспешного обновления активных проверок, следующая попытка будет предпринята через 60 секунд. | |
| Server | да | |  |  | Список (через запятую) IP-адресов , опционально в CIDR-нотации, или имен хостов Серверов и Прокси. Входящие соединения будут приниматься только с хостов, указанных в этом списке. Если включена поддержка IPv6, то "127.0.0.1", "::ffff:127.0.0.1" обрабатываются одинаково, а "::/0" разрешает все IPv4 и IPv6-адреса. "0.0.0.0/0" можно использовать, чтобы разрешить любой IPv4-адрес. Пример: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mon.rosa.dev Пробелы разрешены. | |
| ServerActive | нет | |  |  | Адрес Сервера/Прокси или конфигурация Кластера для получения активных проверок. Адрес Сервера/Прокси – это IP-адрес или DNS-имя и опционально порт, разделенные двоеточием. Конфигурация Кластера – один или более адресов Серверов, разделенные символом "точка с запятой". Можно указать несколько Серверов/Кластеров/Прокси Подсистемы, разделяя их запятой. От каждого Сервера/Кластера должно быть указано не более одного Прокси. Если указан Прокси, то Сервер/Кластер для этого Прокси не должны быть указаны. Несколько адресов через запятую можно указать для использования нескольких независимых Серверов параллельно. Пробелы разрешены. Если порт не указан, используется порт по умолчанию. Адреса IPv6 должны быть заключены в квадратные скобки, если для этого хоста указан порт. Если порт не указан, квадратные скобки для адресов IPv6 опциональны. Если этот параметр не указан, активные проверки отключены. | |
| SourceIP | нет | |  |  | Локальный IP-адрес для: - исходящих подключений к Серверу или Прокси; - создания подключений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т.п.). | |
| StatusPort | нет | | 1024-32767 |  | Если задан, Агент будет слушать указанный порт для HTTP-запросов состояния (http://localhost:<порт>/status). | |
| Timeout | нет | | 1-30 | 3 | Тратить не более Timeout секунд при обработке. | |
| TLSAccept | да, если заданы TLS- сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет | |  |  | Какие входящие подключения принимать. Используется для пассивных проверок. Можно указывать несколько значений через запятую: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. | |
| TLSCAFile | нет | |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между компонентами Подсистемы. | |
| TLSCertFile | нет | |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, используется для зашифрованных соединений между компонентами Подсистемы. | |
| TLSConnect | да, если заданы TLS сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет | |  |  | Как Агент должен подключаться к Серверу или Прокси. Используется для активных проверок. Можно указать только одно из значений: – unencrypted – подключаться без шифрования (по умолчанию); – psk – подключаться, используя TLS и общий ключ (pre-shared key, PSK); – cert – подключаться, используя TLS и сертификат. | |
| TLSCRLFile | нет | |  |  | Абсолютный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных коммуникаций между компонентами Подсистемы. | |
| TLSKeyFile | нет | |  |  | Абсолютный путь к файлу, содержащему закрытый ключ Агента (private key), используется для зашифрованных коммуникаций между компонентами Подсистемы. | |
| TLSPSKFile | нет | |  |  | Абсолютный путь к файлу, содержащему общий ключ (pre-shared key) Агента. Используется для зашифрованных коммуникаций с Сервером. | |
| TLSPSKIdentity | нет | |  |  | Строка идентификатора PSK используется для зашифрованных соединений с Сервером. | |
| TLSServerCertIssuer | нет | |  |  | Разрешенный эмитент сертификата Сервера (Прокси). | |
| TLSServerCertSubject | нет | |  |  | Разрешенный субъект сертификата Сервера (Прокси). | |
| UnsafeUserParameters | нет | | 0,1 | 0 | Разрешить передавать аргументами в пользовательские параметры все символы. Не разрешены следующие символы: \ " " ` \* ? [ ] { } ~ $ ! & ; ( ) < > | # @ Кроме того, не разрешены символы новой строки. | |
| UserParameter | нет | |  |  | Пользовательский параметр для мониторинга. Можно указать несколько пользовательских параметров. Формат: UserParameter=<ключ>,<shell команда> Обратить внимание, что команда не должна возвращать только пустую строку или EOL. Команды оболочки могут иметь относительные пути, если указан параметр UserParameterDir. | |
| UserParameterDir | нет | |  |  | Путь поиска по умолчанию для команд UserParameter. Если используется, Агент перед выполнением команды сменит свою рабочую директорию на указанную здесь. Тем самым, команды UserParameter могут иметь относительный префикс "./" вместо полного пути. Допустимо только одно значение. | |

1. Агент-1 (Windows)

Параметры конфигурации демона Агента-1 (Windows) описаны в таблице 4.

Агент-1 (Windows)

| Параметр | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- |
| Alias | нет |  |  | Задает алиас для ключа элемента данных. Его можно использовать для замены длинных и сложных ключей элементов данных на более короткие и простые. Можно добавлять несколько параметров Alias. Несколько параметров с одинаковым ключом Alias не разрешены. Различные ключи параметра Alias могут ссылаться на тот же ключ элемента данных. Алиасы можно использовать в параметре HostMetadataItem, но нельзя в параметре HostnameItem. |
| AllowKey | нет |  |  | Разрешение выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с DenyKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| BufferSend | нет | 1-3600 | 5 | Не хранить данные в буфере дольше N секунд. |
| BufferSize | нет | 2-65535 | 100 | Максимальное количество значений в буфере памяти. При заполнении буфера Агент будет отправлять все собранные данные Серверу или Прокси. |
| DebugLevel | нет | 0-5 | 3 | Задает уровень журналирования: 0 – основная информация о запуске и остановке процессов Подсистемы; 1 – критичная информация; 2 – информация об ошибках; 3 – предупреждения; 4 – для отладки (записывается очень много информации);  5 – расширенная отладка (записывается еще больше информации). |
| DenyKey | нет |  |  | Запрет выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с AllowKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| EnableRemoteCommands | нет |  | 0 | Разрешены ли удаленные команды с Сервера. Этот параметр устарел, использовать вместо него AllowKey=system.run[\*] или DenyKey=system.run[\*]. Это внутренний алиас для параметров AllowKey/DenyKey, зависящий от значения: 0 – DenyKey=system.run[\*] 1 – AllowKey=system.run[\*]. |
| HostInterface | нет | 0-255 символов |  | Опциональный параметр, который задает интерфейс узла сети. Интерфейс узла сети используется в процессе авторегистрации узла сети. Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов. Если не задано, значение будет получено из HostInterfaceItem. |
| HostInterfaceItem | нет |  |  | Опциональный параметр, который задает элемент данных, используемый для получения интерфейса узла сети. Интерфейс узла сети используется в процессе авторегистрации узла сети. Во время запроса на авторегистрацию Агент будет выводить в журнал предупреждение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов. Элемент данных system.run[] поддерживается независимо от значений параметров AllowKey/DenyKey. Данный параметр используется , только если HostInterface не определен. |
| HostMetadata | нет | 0-255 символов |  | Опциональный параметр, который задает метаданные узла сети. Метаданные узла сети используются только в процессе автоматической регистрации узлов сети (активный Агент). Если не определено, то значение берется от HostMetadataItem. Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение длины строки или не является UTF-8-строкой. |
| HostMetadataItem | нет |  |  | Опциональный параметр, который задает элемент данных Агента, используемый для получения метаданных узла сети. Этот параметр используется, только если HostMetadata не определен. Поддерживаются UserParameters, счетчики производительности (performance counters) и алиасы. Поддерживается system.run[] независимо от значений AllowKey/DenyKey. Значение HostMetadataItem извлекается на каждую попытку авторегистрации и используется только в процессе автоматической регистрации узлов сети (активный Агент). При запросе на авторегистрацию Агент запишет в журнал предупреждение, если полученное от указанного элемента данных значение превышает ограничение в 255 символов. Значение, возвращаемое указанным элементом данных, должно являться UTF-8-строкой, иначе оно будет игнорироваться. |
| Hostname | нет |  | Задается параметром HostnameItem | Список (через запятую) уникальных, регистрозависимых имен узла сети. Требуется для активных проверок и должно совпадать с именами узла сети, указанными на Сервере. Если не задано, значение будет браться из HostnameItem. Допустимые символы: буквенно-цифровые, ".", " ", "\_" and "-". Максимальная длина: 128 символов на имя узла сети, 2048 символов для всей строки. |
| HostnameItem | нет |  | system.Hostname | Опциональный параметр, который задает элемент данных Агента, используемый для получения имени узла сети. Этот параметр используется, только если Hostname не определен. Не поддерживает UserParameters, счетчики производительности (performance counters) или алиасы, но поддерживает system.run[], независимо от значений AllowKey/DenyKey. |
| Include | нет |  |  | Можно включить в файл конфигурации отдельные файлы или все файлы из директории (расположен по умолчанию в директории "C:\Program Files\Zabbix Agent", если Агент устанавливается с использованием установщика пакетов Windows MSI; расположен в директории, указанной во время установки, если Агент устанавливается как zip-архив). Все включаемые файлы должны иметь корректный синтаксис, иначе Агент не запустится. Для включения только необходимых файлов из указанной директории, поддерживается символ звездочки для поиска совпадения по маске. Например: "C:\Program Files\Zabbix Agent\zabbix\_agentd.d\\*.conf". |
| LISTENBacklog | нет | 0 – INT\_MAX | SOMAXCONN | Максимальное количество соединений в состоянии ожидания в очереди TCP. Значение по умолчанию является зашитой в коде константой, которая зависит от системы. Максимально поддерживаемое значение зависит от ОС, слишком большие значения могут быть урезаны до "указанного в реализации максимума" ("implementation-specified maximum"). |
| LISTENIP | нет |  | 0.0.0.0 | Список (через запятую) IP-адресов, которые должен слушать Агент. |
| LISTENPort | нет | 1024-32767 | 10050 | Агент будет слушать этот порт для подключений с Сервера. |
| LogFile | Да, если LogType задан как file, иначе нет. |  | C:\zabbix\_agentd.log | Имя файла журнала Агента. |
| LogFileSize | нет | 0-1024 | 1 | Максимальный размер файла журнала в МБ. 0 – отключение автоматической ротации журнала. Примечание: если лимит размера файла достигнут, а ротация по каким-либо причинам не удалась, существующий файл журнала усекается и начинается заново. |
| LogType | нет |  | file | Тип вывода журнала: – file – запись журнала в файл, указанный в параметре LogFile; – system – запись журнала в журнал событий Windows (Windows Event Log); – console – запись журнала в стандартный вывод. |
| LogRemoteCommands | нет |  | 0 | Включение журналирования выполняемых оболочкой команд как предупреждений. 0 – отключено 1 – включено |
| MaxLinesPerSecond | нет | 1-1000 | 20 | Максимальное количество новых строк в секунду, которые Агент будет отправлять Серверу или Прокси при обработке активных проверок log, logrt и eventlog. Указанное значение будет переопределено параметром maxlines, указанным в ключах элементов данных log, logrt и eventlog. Обратить внимание: при поиске нужной строки в элементах данных журналов Подсистема будет обрабатывать в 10 раз больше новых строк, чем указано в параметре MaxLinesPerSecond. |
| PerfCounter | нет |  |  | Определяет новый параметр <имя\_параметра>, который является усредненным значением системного счетчика производительности <путь\_к\_счетчику\_производительности> за указанный период времени <период> (в секундах). Синтаксис: <имя\_параметра>,"<путь\_к\_счетчику\_производительности>",<период> Например, если требуется получать среднее количество прерываний процессора в секунду за последнюю минуту, можно задать новый параметр interrupts следующим образом: PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60 Следует обратить внимание на двойные кавычки вокруг пути счетчика производительности. Имя параметра (interrupts) используется как ключ элемента данных при создании элемента данных. Образцы для вычисления среднего значения берутся каждую секунду. Можно выполнить "typeperf -qx", чтобы получить список всех счетчиков производительности, доступных в Windows. |
| PerfCounterEn | нет |  |  | Определяет новый параметр <имя\_параметра>, который является усредненным значением системного счетчика производительности <путь\_к\_счетчику\_производительности> за указанный период времени <период> (в секундах). Синтаксис: <имя\_параметра>,"<путь\_к\_счетчику\_производительности>",<период> По сравнению с PerfCounter пути к счетчику производительности должны быть записаны по-английски. Поддерживается только на Windows Server 2008/Vista и выше. Например, если требуется получать среднее количество прерываний процессора в секунду за последнюю минуту, можно задать новый параметр interrupts следующим образом: PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60 Следует обратить внимание на двойные кавычки вокруг пути счетчика производительности. Имя параметра (interrupts) используется как ключ элемента данных при создании элемента данных. Образцы для вычисления среднего значения берутся каждую секунду. Можно найти список английских строк путем просмотра следующего ключа реестра: HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009. |
| RefreshActiveChecks | нет | 60-3600 | 120 | Как часто обновлять список активных проверок, в секундах. Обратить внимание, что после неуспешного обновления активных проверок, следующая попытка будет предпринята через 60 секунд. |
| Server | да, если StartAgents явно не выставлено в 0 |  |  | Список (через запятую) IP-адресов, опционально в CIDR-нотации, или DNS-имен хостов Серверов и Прокси. Входящие соединения будут приниматься только с хостов, указанных в этом списке. Если включена поддержка IPv6, то "127.0.0.1", "::127.0.0.1", "::ffff:127.0.0.1" обрабатываются одинаково, а "::/0" разрешает все IPv4 и IPv6-адреса. "0.0.0.0/0" можно использовать, чтобы разрешить любой IPv4-адрес. Обратить внимание, что "IPv4-совместимые IPv6-адреса" (префикс 0000::/96) поддерживаются, но являются устаревшими согласно [RFC4291 [en]](https://tools.ietf.org/html/rfc4291#section-2.5.5). Пример: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mon.rosa.dev Пробелы разрешены. |
| ServerActive | нет | (\*) |  | Адрес Сервера/Прокси или конфигурация кластера для получения активных проверок. Адрес Сервера/Прокси – это IP-адрес или DNS-имя и опционально порт, разделенные двоеточием. Конфигурация кластера – один или более адресов Серверов, разделенные символом "точка с запятой". Можно указать несколько Серверов/кластеров/Прокси Подсистемы, разделяя их запятой. От каждого Сервера/кластера должно быть указано не более одного Прокси. Если указан Прокси, то Сервер/кластер для этого Прокси не должны быть указаны. Несколько адресов через запятую можно указать для использования нескольких независимых Серверов параллельно. Пробелы разрешены. Если порт не указан, используется порт по умолчанию. Адреса IPv6 должны быть заключены в квадратные скобки, если для этого хоста указан порт. Если порт не указан, квадратные скобки для адресов IPv6 опциональны. Если этот параметр не указан, активные проверки отключены. |
| SourceIP | нет |  |  | Локальный IP-адрес для: - исходящих подключений к Серверу или Прокси; - создания подключений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т.п.) |
| StartAgents | нет | 0-63 (\*) | 3 | Количество экземпляров запускаемых при старте процессов zabbix\_agentd, обрабатывающих пассивные проверки. Если выставлено в 0, отключает пассивные проверки, и Агент не будет слушать никакие TCP порты. |
| Timeout | нет | 1-30 | 3 | Тратить не более Timeout секунд при обработке. |
| TLSAccept | да, если заданы TLS сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет |  |  | Какие входящие подключения принимать. Используется для пассивных проверок. Можно указывать несколько значений через запятую: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. |
| TLSCAFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCertFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSConnect | да, если заданы TLS-сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет |  |  | Как Агент должен подключаться к Серверу или Прокси. Используется для активных проверок. Можно указать только одно значение: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. |
| TLSCRLFile | нет |  |  | Абсолютный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSKeyFile | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ Агента (private key), используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSPSKFile | нет |  |  | Абсолютный путь к файлу, содержащему общий ключ (pre-shared key) Агента. Используется для зашифрованных коммуникаций с Сервером. |
| TLSPSKIdentity | нет |  |  | Строка идентификатора PSK, используется для зашифрованных соединений с Сервером. |
| TLSServerCertIssuer | нет |  |  | Разрешенный эмитент сертификата Сервера (Прокси). |
| TLSServerCertSubject | нет |  |  | Разрешенный субъект сертификата Сервера (Прокси). |
| UnsafeUserParameters | нет | 0-1 | 0 | Разрешить передавать аргументами в пользовательские параметры все символы. 0 – не разрешать 1 – разрешать Не разрешены следующие символы: \ " " ` \* ? [ ] { } ~ $ ! & ; ( ) < > | # @ Кроме того, не разрешены символы новой строки. |
| UserParameter | нет |  |  | Пользовательский параметр для мониторинга. Можно указать несколько пользовательских параметров. Формат: UserParameter=<ключ>,<shell команда> Обратить внимание, что команда не должна возвращать только пустую строку или EOL. Команды оболочки могут иметь относительные пути, если указан параметр UserParameterDir. Примеры: UserParameter=system.test,who|wc -l UserParameter=check\_cpu,./custom\_script.sh |
| UserParameterDir | нет |  |  | Путь поиска по умолчанию для команд UserParameter. Если используется, Агент перед выполнением команды сменит свою рабочую директорию на указанную здесь. Тем самым, команды UserParameter могут иметь относительный префикс "./" вместо полного пути. Допустимо только одно значение. Пример: UserParameterDir=/opt/myscripts |

Примечание – (\*) Количество активных Серверов, перечисленных в параметре ServerActive, и количество запускаемых при старте экземпляров для пассивных проверок, указанное в StartAgents, должно быть меньше 64.

1. Агент-2 (Windows)

Параметры конфигурации демона Агента-2 (Windows) описаны в таблице 5.

Агент-2 (Windows)

| Параметр | | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- | --- |
| Alias | | нет |  |  | Задает алиас для ключа элемента данных. Его можно использовать для замены длинных и сложных ключей элементов данных на более короткие и простые. Можно добавлять несколько параметров Alias. Несколько параметров с одинаковым ключом Alias не разрешены. Различные ключи параметра Alias могут ссылаться на тот же ключ элемента данных. Алиасы можно использовать в параметре HostMetadataItem, но нельзя в параметре HostnameItem. |
| AllowKey | | нет |  |  | Разрешение выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с DenyKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| BufferSend | | нет | 1-3600 | 5 | Интервал времени в секундах, определяющий, как часто значения из буфера отсылаются на Сервер. Следует обратить внимание, что, если буфер заполнен, данные будут отосланы раньше. |
| BufferSize | | нет | 2-65535 | 100 | Максимальное количество значений в буфере памяти. При заполнении буфера Агент будет отправлять все собранные данные Серверу или Прокси. Этот параметр должен использоваться, только если постоянный буфер отключен (EnablePersistentBuffer=0). |
| ControlSocket | | нет |  | \\.\pipe\agent.sock | Сокет управления, используется при отправке команд управления при помощи опции "-R". |
| DebugLevel | | нет | 0-5 | 3 | Задает уровень журналирования: 0 – основная информация о запуске и остановке процессов Подсистемы 1 – критичная информация 2 – информация об ошибках 3 – предупреждения 4 – для отладки (записывается очень много информации) 5 – расширенная отладка (записывается еще больше информации) |
| DenyKey | | нет |  |  | Запрет выполнения тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа – это выражение с подстановочным знаком, которое поддерживает символ "\*" для соответствия с любым количеством любых символов. Можно задать несколько правил соответствия ключей в сочетании с AllowKey. Параметры обрабатываются по очереди в соответствии с порядком их появления. |
| EnablePersistentBuffer | | нет | 0-1 | 0 | Включить использование локального постоянного хранилища для активных элементов данных. 0 – отключено 1 – включено Если постоянное хранилище отключено, будет использован буфер памяти. |
| ForceActiveChecksOnStart | | нет | 0-1 | 0 | Выполнить активные проверки сразу после перезапуска для первой полученной конфигурации. 0 – отключено 1 – включено Также доступно как параметр конфигурации плагина, например:  Plugins.Uptime.System.ForceActiveChecksOnStart=1 |
| HostInterface | | нет | 0-255 символов |  | Необязательный параметр, определяющий интерфейс узла сети. Интерфейс узла сети используется в процессе автоматической регистрации узла сети. Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов. Если не определено, значение будет получено от HostInterfaceItem. |
| HostInterfaceItem | | нет |  |  | Необязательный параметр, который определяет элемент данных, используемый для получения интерфейса узла сети. Интерфейс узла сети используется в процессе автоматической регистрации узла сети. Во время запроса на автоматическую регистрацию Агент регистрирует предупреждающее сообщение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов. Элемент данных system.run[] поддерживается независимо от значений параметров AllowKey/DenyKey. Эта опция используется, только если HostInterface не определен. |
| HostMetadata | | нет | 0-255 символов |  | Опциональный параметр, который задает метаданные узла сети. Метаданные узла сети используются только в процессе автоматической регистрации узлов сети. Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение длины строки или не является UTF-8-строкой. Если не определено, то значение берется от HostMetadataItem. |
| HostMetadataItem | | нет |  |  | Опциональный параметр, который задает элемент данных, используемый для получения метаданных узла сети. Значение HostMetadataItem извлекается на каждую попытку авторегистрации для процесса автоматической регистрации узла сети. При запросе на авторегистрацию Агент запишет в журнал предупреждение, если полученное от указанного элемента данных значение превышает ограничение в 255 символов. Этот параметр используется только если HostMetadata не определен. Поддерживаются UserParameters и алиасы. Поддерживается system.run[] независимо от значений AllowKey/DenyKey. Значение, возвращаемое указанным элементом данных, должно являться строкой UTF-8, иначе оно будет игнорироваться. |
| Hostname | | нет |  | Задается параметром HostnameItem | Список (через запятую) уникальных, регистрозависимых имен узла сети. Требуется для активных проверок и должно совпадать с именами узла сети, указанными на Сервере. Если не задано, значение будет браться из HostnameItem. Допустимые символы: буквенно-цифровые, ".", " ", "\_" и "-". Максимальная длина: 128 символов на имя узла сети, 2048 символов для всей строки. |
| HostnameItem | | нет |  | system.Hostname | Элемент данных, который используется для формирования Hostname, если этот параметр не указан. Игнорируется, если задан параметр Hostname. Не поддерживает UserParameters или алиасы, но поддерживает system.run[], независимо от значений AllowKey/DenyKey. |
| Include | | нет |  |  | Можно включить в файл конфигурации отдельные файлы или все файлы из директории (расположен по умолчанию в "C:\Program Files\Zabbix Agent 2", если Агент устанавливается с использованием установщика пакетов Windows MSI; расположен в директории, указанной во время установки, если Агент устанавливается как zip-архив). Все включаемые файлы должны иметь корректный синтаксис, иначе Агент не запустится. Путь может быть относительным по отношению к местоположению файла zabbix\_agent2.conf (например, Include=.\zabbix\_agent2.d\plugins.d\\*.conf). Для включения только необходимых файлов из указанной директории, поддерживается символ звездочки для поиска совпадения по маске. Например: C:\Program Files\Zabbix Agent2\zabbix\_agent2.d\\*.conf. |
| LISTENIP | | нет |  | 0.0.0.0 | Список (через запятую) IP-адресов, которые должен слушать Агент. Первый IP-адрес отсылается на Сервер при подключении к нему для получения списка активных проверок. |
| LISTENPort | | нет | 1024-32767 | 10050 | Агент будет слушать этот порт для подключений с Сервера. |
| LogFile | | Да, если LogType задан как file, иначе нет. |  | c:\zabbix\_agent2.log | Имя файла журнала, если LogType равен значению "file". |
| LogFileSize | | нет | 0-1024 | 1 | Максимальный размер файла журнала в МБ. 0 – отключение автоматической ротации журнала. Примечание: если лимит размера файла достигнут, а ротация по каким-либо причинам не удалась, существующий файл журнала усекается и начинается заново. |
| LogType | | нет |  | file | Задает место, куда будут записываться сообщения журнала: – file – в файл, указанный параметром LogFile; – console – в стандартный вывод. |
| PersistentBufferFile | | нет |  |  | Файл, в котором Агент-2 должен хранить базу данных SQLite. Имя файла должно быть указано с абсолютным путем. Этот параметр используется, только если включен постоянный буфер (EnablePersistentBuffer=1). |
| PersistentBufferPeriod | | нет | 1m-365d | 1h | Период времени, в течение которого должны храниться данные, когда нет соединения с Сервером или Прокси. Более старые данные будут потеряны. Журнальные данные будут сохранены. Этот параметр используется, только если включен постоянный буфер (EnablePersistentBuffer=1). |
| Plugins | | нет |  |  | Большинство плагинов имеют свои собственные файлы конфигурации. Файл конфигурации Агента содержит параметры плагинов, перечисленные ниже. |
|  | Plugins.Log.MaxLinesPerSecond | нет | 1-1000 | 20 | Максимальное количество новых строк, которые Агент будет отправлять в секунду на Сервер или Прокси при обработке активных проверок log и eventlog. Указанное значение будет переопределено параметром maxlines, указанным в ключах элементов данных log или eventlog. Следует обратить внимание: при поиске нужной строки в элементах данных журналов будет обрабатывать в 10 раз больше новых строк, чем указано в параметре MaxLinesPerSecond. |
| Plugins.SystemRun.LogRemoteCommands | нет |  | 0 | Включение журналирования выполняемых оболочкой команд как предупреждений. 0 – отключено 1 – включено Команды будут записаны в журнал, только если они выполняются удаленно. Записи в журнале не создаются, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem. |
| PluginSocket | | нет |  | \\..plugin.sock | Путь к файлу UNIX-сокета для коммуникаций с подгружаемыми плагинами. |
| PluginTimeout | | нет | 1-30 | Глобальный тайм-аут | Тайм-аут для соединений с подгружаемыми плагинами. |
| RefreshActiveChecks | | нет | 60-3600 | 120 | Как часто обновлять список активных проверок, в секундах. Следует обратить внимание, что после неуспешного обновления активных проверок следующая попытка будет предпринята через 60 секунд. |
| Server | | да |  |  | Список (через запятую) IP-адресов, опционально в CIDR-нотации, или DNS-имен Серверов и Прокси. Входящие соединения будут приниматься только с хостов, указанных в этом списке. Если включена поддержка IPv6, то "127.0.0.1", "::ffff:127.0.0.1" обрабатываются одинаково, а "::/0" разрешает все IPv4 и IPv6-адреса. "0.0.0.0/0" можно использовать, чтобы разрешить любой IPv4-адрес. Пример: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mon.rosa.dev Пробелы разрешены. |
| ServerActive | | нет |  |  | Адрес Сервера/Прокси или конфигурация кластера для получения активных проверок. Адрес Сервера/Прокси – это IP-адрес или DNS-имя и опционально порт, разделенные двоеточием. Конфигурация кластера – один или более адресов Серверов, разделенные символом "точка с запятой". Можно указать несколько Серверов/кластеров/Прокси, разделяя их запятой. От каждого Сервера/кластера должно быть указано не более одного Прокси. Если указан Прокси, то Сервер/кластер для этого Прокси не должны быть указаны. Несколько адресов через запятую можно указать для использования нескольких независимых Серверов параллельно. Пробелы разрешены. Если порт не указан, используется порт по умолчанию. Адреса IPv6 должны быть заключены в квадратные скобки, если для этого хоста указан порт. Если порт не указан, квадратные скобки для адресов IPv6 опциональны. Если этот параметр не указан, активные проверки отключены. |
| SourceIP | | нет |  |  | Локальный IP-адрес для: - исходящих подключений к Серверу или Прокси; - создания подключений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т.п.). |
| StatusPort | | нет | 1024-32767 |  | Если задан, Агент будет слушать указанный порт для HTTP-запросов состояния (http://localhost:<порт>/status). |
| Timeout | | нет | 1-30 | 3 | Тратить не более Timeout секунд при обработке. |
| TLSAccept | | да, если заданы TLS-сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет |  |  | Какие входящие подключения принимать. Используется для пассивных проверок. Можно указывать несколько значений через запятую: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. |
| TLSCAFile | | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCertFile | | нет |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSConnect | | да, если заданы TLS-сертификат или параметры PSK (даже при незашифрованном соединении), иначе – нет |  |  | Как Агент должен подключаться к Серверу или Прокси. Используется для активных проверок. Можно указать только одно значение: – unencrypted – принимать подключения без шифрования (по умолчанию); – psk – принимать подключения с TLS и общим ключом (pre-shared key, PSK); – cert – принимать подключения с TLS и сертификатом. |
| TLSCRLFile | | нет |  |  | Абсолютный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSKeyFile | | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ Агента (private key), используется для зашифрованных коммуникаций между компонентами Подсистемы. |
| TLSPSKFile | | нет |  |  | Абсолютный путь к файлу, содержащему общий ключ (pre-shared key) Агента. Используется для зашифрованных коммуникаций с Сервером. |
| TLSPSKIdentity | | нет |  |  | Строка идентификатора PSK, используется для зашифрованных соединений с Сервером. |
| TLSServerCertIssuer | | нет |  |  | Разрешенный эмитент сертификата Сервера (Прокси). |
| TLSServerCertSubject | | нет |  |  | Разрешенный субъект сертификата Сервера (Прокси). |
| UnsafeUserParameters | | нет | 0,1 | 0 | Разрешить передавать аргументами в пользовательские параметры все символы. Не разрешены следующие символы: \ " " ` \* ? [ ] { } ~ $ ! & ; ( ) < > | # @ Кроме того, не разрешены символы новой строки. |
| UserParameter | | нет |  |  | Пользовательский параметр для мониторинга. Можно указать несколько пользовательских параметров. Формат: UserParameter=<ключ>,<shell команда> Следует обратить внимание, что команда не должна возвращать только пустую строку или EOL. Команды оболочки могут иметь относительные пути, если указан параметр UserParameterDir. Примеры: UserParameter=system.test,who|wc -l UserParameter=check\_cpu,./custom\_script.sh |
| UserParameterDir | | нет |  |  | Путь поиска по умолчанию для команд UserParameter. Если используется, Агент перед выполнением команды сменит свою рабочую директорию на указанную здесь. Тем самым, команды UserParameter могут иметь относительный префикс "./" вместо полного пути. Допустимо только одно значение. Пример: UserParameterDir=/opt/myscripts |

1. Плагины Агента-1 (Windows)

Параметры конфигурации демона Агента-1 (Windows) описаны в таблице 6.

Плагины Агента-2

| Параметр | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- |
| Плагин Ceph |  |  |  |  |
| Plugins.Ceph.Default.ApiKey | нет |  |  | Ключ API по умолчанию для подключения к Ceph; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.Ceph.Default.User | нет |  |  | Имя пользователя по умолчанию для подключения к Ceph; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.Ceph.Default.Uri | нет |  | https://localhost:8003 | URI по умолчанию для подключения к Ceph; используется, если значение не указано в ключе элемента данных или в именованной сессии. URL не должен содержать встроенные credential (они будут проигнорированы). Должен соответствовать формату URI. Поддерживается только схема https; схема может быть опущена. Порт может быть опущен (по умолчанию 8003). Примеры: https://127.0.0.1:8003, localhost |
| Plugins.Ceph.InsecureSkipVerify | нет | false / true | false | Определяет, должен ли http-клиент верифицировать серверные цепочку сертификатов и имя хоста. Если true, TLS принимает любой предъявленный Сервером сертификат и любое имя хоста в этом сертификате. В этом режиме TLS уязвим к атакам типа man-in-the-middle (следует использовать только для тестирования). |
| Plugins.Ceph.KeepAlive | нет | 60-900 | 300 | Максимальное время ожидания (в секундах), прежде чем неиспользуемые плагином соединения закрываются. |
| Plugins.Ceph.Sessions.<ИмяСессии>.ApiKey | нет |  |  | Ключ API именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Ceph.Sessions.<ИмяСессии>.User | нет |  |  | Имя пользователя именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Ceph.Sessions.<ИмяСессии>.Uri | нет |  |  | Строка подключения именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Не должна содержать встроенные credential (они будут проигнорированы). Должна соответствовать формату URI. Поддерживается только схема https; схема может быть опущена. Порт может быть опущен (по умолчанию 8003). Примеры: https://127.0.0.1:8003, localhost |
| Plugins.Ceph.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин Docker |  |  |  |  |
| Plugins.Docker.Endpoint | нет |  | unix:///var/run/docker.sock | Местоположение unix-сокета демона Docker. Должно содержать схему (поддерживается только unix://). |
| Plugins.Docker.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин Memcached |  |  |  |  |
| Plugins.Memcached.Default.Password | нет |  |  | Пароль по умолчанию для подключения к Memcached; используется, если значение не указано ни в ключе элемента данных, ни в именованной сессии. |
| Plugins.Memcached.Default.Uri | нет |  | tcp://localhost:11211 | URI по умолчанию для подключения к Memcached; используется, если значение не указано ни в ключе элемента данных, ни в именованной сессии. Не должно содержать встроенных credential (они будут проигнорированы). Должно соответствовать формату URI. Поддерживаются схемы: tcp, unix; схема может быть опущена. Порт может быть опущен (по умолчанию 11211). Примеры: tcp://localhost:11211 localhost unix:/var/run/memcached.sock |
| Plugins.Memcached.Default.User | нет |  |  | Имя пользователя по умолчанию для подключения к Memcached; используется, если значение не указано ни в ключе элемента данных, ни в именованной сессии. |
| Plugins.Memcached.KeepAlive | нет | 60-900 | 300 | Максимальное время ожидания (в секундах), прежде чем неиспользуемые соединения плагина будут закрыты. |
| Plugins.Memcached.Sessions.<ИмяСессии>.Password | нет |  |  | Пароль именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Memcached.Sessions.<ИмяСессии>.Uri | нет |  |  | Строка подключения именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Не должна содержать встроенных credential (они будут проигнорированы). Должна соответствовать формату URI. Поддерживаются схемы: tcp, unix; схема может быть опущена. Порт может быть опущен (по умолчанию 11211). Примеры: tcp://localhost:11211 localhost unix:/var/run/memcached.sock |
| Plugins.Memcached.Sessions.<ИмяСессии>.User | нет |  |  | Имя пользователя именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Memcached.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин Modbus |  |  |  |  |
| Plugins.Modbus.Sessions.<ИмяСессии>.Endpoint | нет |  |  | Точка входа – строка подключения, состоящая из схемы протокола, адреса хоста и порта или имени последовательного порта и параметров. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Modbus.Sessions.<ИмяСессии>.SlaveID | нет |  |  | Идентификатор ведомого именованной сессии. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. Пример: Plugins.Modbus.Sessions.MB1.SlaveID=20 Следует обратить внимание, что этот параметр именованной сессии проверяется, только если значение id ведомого в параметре ключа элемента данных пустое. |
| Plugins.Modbus.Sessions.<ИмяСессии>.Timeout | нет |  |  | Тайм-аут именованной сессии. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. Пример: Plugins.Modbus.Sessions.MB1.Timeout=2 |
| Plugins.Modbus.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин MongoDB  Опции  Параметр Описание  -V --version Вывести версию плагина и информацию о лицензии.  -h --help Вывести справочную информацию (сокращенно). |  |  |  |  |
| Plugins.MongoDB.Default.Password | нет |  |  | Пароль по умолчанию для подключения к MongoDB; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.MongoDB.Default.Uri | нет |  |  | URI по умолчанию для подключения к MongoDB; используется, если значение не указано в ключе элемента данных или в именованной сессии. Не должно содержать встроенных credential (они будут проигнорированы). Должно соответствовать формату URI. Поддерживается только схема tcp; схема может быть опущена. Порт может быть опущен (по умолчанию 27017). Примеры: tcp://127.0.0.1:27017, tcp:localhost, localhost |
| Plugins.MongoDB.Default.User | нет |  |  | Имя пользователя по умолчанию для подключения к MongoDB; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.MongoDB.KeepAlive | нет | 60-900 | 300 | Максимальное время ожидания (в секундах), прежде чем неиспользуемые плагином соединения закрываются. |
| Plugins.MongoDB.Sessions.<ИмяСессии>.Password | нет |  |  | Пароль именованной сессии. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MongoDB.Sessions.<ИмяСессии>.TLSCAFile | нет (да, если Plugins.MongoDB.Sessions.<ИмяСессии>.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MongoDB.Sessions.<ИмяСессии>.TLSCertFile | да, если Plugins.MongoDB.Sessions.<ИмяСессии>.TLSKeyFile указано |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, используется для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MongoDB.Sessions.<ИмяСессии>.TLSConnect | нет |  |  | Тип шифрования для коммуникаций между Агентом-2 и наблюдаемыми базами данных. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. Поддерживаются значения: – required – требовать TLS-подключений; – verify\_ca – проверять сертификаты; – verify\_full – проверять сертификаты и IP-адрес. |
| Plugins.MongoDB.Sessions.<ИмяСессии>.TLSKeyFile | да, если Plugins.MongoDB.Sessions.<ИмяСессии>.TLSCertFile указано |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) базы данных, используется для зашифрованных коммуникаций между Агентом-2 и наблюдаемыми базами данных. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MongoDB.Sessions.<ИмяСессии>.Uri | нет |  |  | Строка подключения именованной сессии. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. Не должна содержать встроенных credential (они будут проигнорированы). Должна соответствовать формату URI. Поддерживается только схема tcp; схема может быть опущена. Порт может быть опущен (по умолчанию 27017). Примеры: tcp://127.0.0.1:27017, tcp:localhost, localhost |
| Plugins.MongoDB.Sessions.<ИмяСессии>.User | нет |  |  | Имя пользователя именованной сессии. **<ИмяСессии>** – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MongoDB.System.Path | нет |  |  | Путь к внешнему исполняемому файлу плагина. |
| Plugins.MongoDB.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин MQTT |  |  |  |  |
| Plugins.MQTT.Default.Password | нет |  |  | Пароль по умолчанию для подключения к MQTT; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.MQTT.Default.TLSCAFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла для зашифрованных соединений между Агентом-2 и брокером MQTT; используется, если значение не указано в именованной сессии. |
| Plugins.MQTT.Default.TLSCertFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов для зашифрованных соединений между Агентом-2 и брокером MQTT; используется, если значение не указано в именованной сессии. |
| Plugins.MQTT.Default.TLSKeyFile | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) MQTT для зашифрованных коммуникаций между Агентом-2 и брокером MQTT; используется, если значение не указано в именованной сессии. |
| Plugins.MQTT.Default.Topic | нет |  |  | Тема по умолчанию для подписки MQTT; используется, если значение не указано в ключе элемента данных или в именованной сессии. Тема может содержать подстановочные символы ("+","#") Примеры: path/to/file, path/to/#, path/+/topic |
| Plugins.MQTT.Default.Url | нет |  | tcp://localhost:1883 | Строка подключения к брокеру MQTT по умолчанию; используется, если значение не указано в ключе элемента данных или в именованной сессии. Не должна включать параметры запроса. Должна соответствовать формату URL. Поддерживаются схемы: tcp (по умолчанию), ws, tls; схема может быть опущена. Порт может быть опущен (по умолчанию 1883). Примеры: tcp://host:1883 localhost ws://host:8080 |
| Plugins.MQTT.Default.User | нет |  |  | Имя пользователя по умолчанию для подключения к MQTT; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.MQTT.Sessions.<ИмяСессии>.Password | нет |  |  | Пароль именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MQTT.Sessions.<ИмяСессии>.TLSCAFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между Агентом-2 и брокером MQTT. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MQTT.Sessions.<ИмяСессии>.TLSCertFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, которые используются для зашифрованных соединений между Агентом-2 и брокером MQTT. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MQTT.Sessions.<ИмяСессии>.TLSKeyFile | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) MQTT, который используется для зашифрованных коммуникаций между Агентом-2 и брокером MQTT. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MQTT.Sessions.<ИмяСессии>.Topic | нет |  |  | Тема именованной сессии для подписки MQTT. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Тема может содержать подстановочные символы ("+","#") Примеры: path/to/file path/to/# path/+/topic |
| Plugins.MQTT.Sessions.<ИмяСессии>.Url | нет |  |  | Строка подключения именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Не должна включать параметры запроса. Должна соответствовать формату URL. Поддерживаются темы: tcp (по умолчанию), ws, tls; схема может быть опущена. Порт может быть опущен (по умолчанию 1883). Примеры: tcp://host:1883 localhost ws://host:8080 |
| Plugins.MQTT.Sessions.<ИмяСессии>.User | нет |  |  | Имя пользователя именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.MQTT.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин MySQL |  |  |  |  |
| Plugins.Mysql.CallTimeout | нет | 1-30 | глобальный тайм-аут | Максимальное время ожидания завершения запроса в секундах. |
| Plugins.Mysql.CustomQueriesPath | нет |  | пусто | Абсолютный путь к директории, которая используется для хранения пользовательских запросов. |
| Plugins.Mysql.Default.Password | нет |  |  | Пароль по умолчанию для подключения к MySQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.Mysql.Default.TLSCAFile | нет (да, если Plugins.Mysql.Default.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. |
| Plugins.Mysql.Default.TLSCertFile | нет (да, если Plugins.Mysql.Default.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. |
| Plugins.Mysql.Default.TLSConnect | нет |  |  | Тип шифрования для коммуникаций между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. Поддерживаются значения: – required – требовать TLS-подключений; – verify\_ca – проверять сертификаты; – verify\_full – проверять сертификаты и IP-адрес. |
| Plugins.Mysql.Default.TLSKeyFile | нет (да, если Plugins.Mysql.Default.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) базы данных для зашифрованных коммуникаций между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. . |
| Plugins.Mysql.Default.Uri | нет |  | tcp://localhost:3306 | URI по умолчанию для подключения к MySQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. Не должно содержать встроенных credential (они будут проигнорированы). Должно соответствовать формату URI. Поддерживаются схемы: tcp, unix; схема может быть опущена. Порт может быть опущен (по умолчанию 3306). Примеры: tcp://localhost:3306 localhost unix:/var/run/mysql.sock. |
| Plugins.Mysql.Default.User | нет |  |  | Имя пользователя по умолчанию для подключения к MySQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.Mysql.KeepAlive | нет | 60-900 | 300 | Максимальное время ожидания (в секундах), прежде чем неиспользуемые плагином соединения закрываются. |
| Plugins.Mysql.Sessions.<ИмяСессии>.Password | нет |  |  | Пароль именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Mysql.Sessions.<ИмяСессии>.TLSCAFile | нет (да, если Plugins.Mysql.Sessions.<ИмяСессии>.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Mysql.Sessions.<ИмяСессии>.TLSCertFile | да, если параметр Plugins.Mysql.Sessions.<ИмяСессии>.TLSKeyFile задан. |  |  | Абсолютный путь к файлу, который содержит сертификат Агента или цепочку сертификатов, используется для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Mysql.Sessions.<ИмяСессии>.TLSConnect | нет |  |  | Тип шифрования для коммуникаций между Агентом-2 и наблюдаемыми базами данных. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Поддерживаются значения: – required – требовать TLS-подключений; – verify\_ca – проверять сертификаты; – verify\_full – проверять сертификаты и IP-адрес. |
| Plugins.Mysql.Sessions.<ИмяСессии>.TLSKeyFile | да, если параметр Plugins.Mysql.Sessions.<ИмяСессии>.TLSCertFile задан. |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) базы данных, который используется для зашифрованных коммуникаций между Агентом-2 и наблюдаемыми базами данных. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Mysql.Sessions.<ИмяСессии>.Uri | нет |  |  | Строка подключения именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Не должна содержать встроенных credential (они будут проигнорированы). Должна соответствовать формату URI. Поддерживаются схемы: tcp, unix; схема может быть опущена. Порт может быть опущен (по умолчанию 3306). Примеры: tcp://localhost:3306 localhost unix:/var/run/mysql.sock |
| Plugins.Mysql.Sessions.<ИмяСессии>.User | нет |  |  | Имя пользователя именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Mysql.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин PostgreSQL  -V --version Вывести версию плагина и информацию о лицензии.  -h --help Вывести справочную информацию (сокращенно). |  |  |  |  |
| Plugins.PostgreSQL.Default.CacheMode | нет |  | prepare | Режим кэширования для соединений PostgreSQL. Поддерживаются значения: – prepare (по умолчанию) – будет создавать параметризованные запросы (prepared statements) на сервере PostgreSQL; – describe – будет использовать анонимные параметризованные запросы (anonymous prepared statement) для описания запросов без создания запросов на Сервере. Следует обратить внимание, что describe главным образом полезен, если среда не допускает параметризованные запросы, например, при работе пулера соединений типа PgBouncer. |
| Plugins.PostgreSQL.CallTimeout | нет | 1-30 | глобальный тайм-аут | Максимальное время ожидания (в секундах) завершения запроса. |
| Plugins.PostgreSQL.CustomQueriesPath | нет |  | отключено | Абсолютный путь к директории, содержащей файлы .sql для пользовательских запросов. |
| Plugins.PostgreSQL.Default.Database | нет |  |  | База данных по умолчанию для подключения к PostgreSQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.PostgreSQL.Default.Password | нет |  |  | Пароль по умолчанию для подключения к PostgreSQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.PostgreSQL.Default.TLSCAFile | нет (да, если Plugins.PostgreSQL.Default.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. |
| Plugins.PostgreSQL.Default.TLSCertFile | нет (да, если Plugins.PostgreSQL.Default.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификат PostgreSQL или цепочку сертификатов для зашифрованных соединений между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. |
| Plugins.PostgreSQL.Default.TLSConnect | нет |  |  | Тип шифрования для коммуникаций между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. Поддерживаются значения: – required – подключаться, используя TLS как режим транспорта без проверки подлинности; – verify\_ca – подключаться с использованием TLS и проверкой сертификата; – verify\_full – подключаться с использованием TLS, проверкой сертификата и проверкой того, что идентификатор базы данных (CN), указанный параметром DBHost, соответствует своему сертификату. Неуказанный тип шифрования означает нешифрованное подключение. |
| Plugins.PostgreSQL.Default.TLSKeyFile | нет (да, если Plugins.PostgreSQL.Default.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) PostgreSQL, используется для зашифрованных коммуникаций между Агентом-2 и наблюдаемыми базами данных; используется, если значение не указано в именованной сессии. |
| Plugins.PostgreSQL.Default.Uri | нет |  |  | URI по умолчанию для подключения к PostgreSQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. Не должно содержать встроенных credential (они будут проигнорированы). Должно соответствовать формату URI. Поддерживаются схемы: tcp, unix. Примеры: tcp://127.0.0.1:5432 tcp://localhost unix:/var/run/postgresql/.s.PGSQL.5432. |
| Plugins.PostgreSQL.Default.User | нет |  |  | Имя пользователя по умолчанию для подключения к PostgreSQL; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.PostgreSQL.KeepAlive | нет | 60-900 | 300 | Максимальное время ожидания (в секундах), прежде чем неиспользуемые плагином соединения закрываются. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.CacheMode | нет |  | prepare | Режим кэширования для соединений PostgreSQL. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Поддерживаются значения: – prepare (по умолчанию) – будет создавать параметризованные запросы (prepared statements) на сервере PostgreSQL; – describe – будет использовать анонимные параметризованные запросы (anonymous prepared statement) для описания запросов без создания запросов на Сервере. Следует обратить внимание, что describe главным образом полезен, когда среда не допускает параметризованные запросы, например, при работе пулера соединений типа PgBouncer. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.Database | нет |  |  | База данных именованной сессии для подключения. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.Password | нет | Должно соответствовать формату пароля. |  | Пароль именованной сессии для подключения. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSCAFile | нет (да, если Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSConnect имеет одно из значений: verify\_ca, verify\_full) |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSCertFile | да, если параметр Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSKeyFile указан. |  |  | Абсолютный путь к файлу, который содержит сертификат PostgreSQL или цепочку сертификатов. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSConnect | нет |  |  | Тип шифрования для подключения к PostgreSQL. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Поддерживаются значения: – required – подключаться, используя TLS как режим транспорта без проверки подлинности; – verify\_ca – подключаться с использованием TLS и проверкой сертификата; verify\_full – подключаться с использованием TLS, проверкой сертификата и проверкой того, что идентификатор базы данных (CN), указанный параметром DBHost, соответствует своему сертификату. Неуказанный тип шифрования означает нешифрованное подключение. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSKeyFile | да, если параметр Plugins.PostgreSQL.Sessions.<ИмяСессии>.TLSCertFile указан. |  |  | Абсолютный путь к файлу, содержащему закрытый ключ (private key) PostgreSQL. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.Uri | нет |  |  | Строка подключения именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Не должна содержать встроенных credential (они будут проигнорированы). Должна соответствовать формату URI. Поддерживаются схемы: tcp, unix. Примеры: tcp://127.0.0.1:5432 tcp://localhost unix:/var/run/postgresql/.s.PGSQL.5432 |
| Plugins.PostgreSQL.Sessions.<ИмяСессии>.User | нет |  |  | Имя пользователя именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.PostgreSQL.System.Path | да |  |  | Путь до внешнего исполняемого файла плагина. |
| Plugins.PostgreSQL.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин Redis |  |  |  |  |
| Plugins.Redis.Default.Password | нет |  |  | Пароль по умолчанию для подключения к Redis; используется, если значение не указано в ключе элемента данных или в именованной сессии. |
| Plugins.Redis.Default.Uri | нет |  | tcp://localhost:6379 | URI по умолчанию для подключения к Redis; используется, если значение не указано в ключе элемента данных или в именованной сессии. Не должно содержать встроенных credential (они будут проигнорированы). Должно соответствовать формату URI. Поддерживаются схемы: tcp, unix;  Порт может быть опущен (по умолчанию 6379). Примеры: tcp://localhost:6379 localhost unix:/var/run/redis.sock |
| Plugins.Redis.KeepAlive | нет | 60-900 | 300 | Максимальное время ожидания (в секундах), прежде чем неиспользуемые плагином соединения закрываются. |
| Plugins.Redis.Sessions.<ИмяСессии>.Password | нет |  |  | Пароль именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. |
| Plugins.Redis.Sessions.<ИмяСессии>.Uri | нет |  |  | Строка подключения именованной сессии. <ИмяСессии> – определяет имя сессии, используемое в ключах элементов данных. Не должна содержать встроенных credential (они будут проигнорированы). Должна соответствовать формату URI. Поддерживаются схемы: tcp, unix; схема может быть опущена. Порт может быть опущен (по умолчанию 6379). Примеры: tcp://localhost:6379 localhost unix:/var/run/redis.sock |
| Plugins.Redis.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |
| Плагин Smart |  |  |  |  |
| Plugins.Smart.Path | нет |  | smartctl | Путь к исполняемому файлу smartctl. |
| Plugins.Smart.Timeout | нет | 1-30 | глобальный тайм-аут | Время ожидания выполнения запроса (как долго запрос будет ожидать выполнения, прежде чем будет принудительно завершен). |

1. Java gateway

Параметры конфигурации демона Java gateway описаны в таблице 7.

Java gateway

| Переменная | Свойство | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- | --- |
| LISTEN\_IP | zabbix.LISTENIP | нет |  | 0.0.0.0 | IP-адрес, который слушает Java Gateway. |
| LISTEN\_PORT | zabbix.LISTENPort | нет | 1024-32767 | 10052 | Порт, который слушает Java Gateway. |
| PID\_FILE | zabbix.pidFile | нет |  | /tmp/zabbix\_java.pid | Имя PID-файла. Если не задано, Java Gateway запустится как консольное приложение. |
| PROPERTIES\_FILE | zabbix.propertiesFile | нет |  |  | Имя файла свойств. Может использоваться, чтобы задать дополнительные свойства, используя формат ключ-значение, таким образом, что они не будут видимы в командной строке или перезапишут существующие свойства. Например: "javax.net.ssl.trustStorePassword=<пароль>" |
| START\_POLLERS | zabbix.StartPollers | нет | 1-1000 | 5 | Количество запускаемых рабочих потоков. |
| TIMEOUT | zabbix.Timeout | нет | 1-30 | 3 | Как долго ждать завершения сетевых операций. |

1. Веб-сервис

Параметры конфигурации демона Java gateway описаны в таблице 8.

Веб-сервис

| Параметр | Обязательный | Диапазон | По умолчанию | Описание |
| --- | --- | --- | --- | --- |
| AllowedIP | да |  |  | Список (через запятую) IP-адресов, опционально в CIDR-нотации, или DNS-имен Серверов и Прокси. Входящие соединения будут приниматься только с хостов, указанных в этом списке. Если включена поддержка IPv6, то 127.0.0.1,::127.0.0.1,::ffff:127.0.0.1 обрабатываются одинаково, а::/0 разрешает все IPv4 и IPv6-адреса. 0.0.0.0/0 можно использовать, чтобы разрешить любой IPv4-адрес. Пример: 127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mon.rosa.dev |
| DebugLevel | нет | 0-5 | 3 | Задает уровень журналирования: 0 – основная информация о запуске и остановке процессов Подсистемы; 1 – критичная информация; 2 – информация об ошибках; 3 – предупреждения; 4 – для отладки (записывается очень много информации); 5 – расширенная отладка (записывается еще больше информации). |
| LISTENPort | нет | 1024-32767 | 10053 | Порт, который слушает веб-сервис для подключений с Сервера. |
| LogFile | Да, если LogType задан как file, иначе нет. |  |  | Имя файла журнала для параметра LogType: "file". Пример: /tmp/zabbix\_web\_service.log |
| LogFileSize | нет | 0-1024 | 1 | Максимальный размер файла журнала в МБ. 0 – отключение автоматической ротации журнала. |
| LogType | нет | system / file / console | file | Указывает, куда пишутся сообщения журнала: – system – в syslog; – file – в файл, указанный в параметре LogFile; – console – в стандартный вывод. |
| Timeout | нет | 1-30 | 3 | Тратить не более Timeout секунд при обработке. |
| TLSAccept | нет | unencrypted / cert | unencrypted | Какие входящие подключения принимать: – unencrypted – принимать подключения без шифрования (по умолчанию); – cert – принимать подключения с TLS и сертификатом. |
| TLSCAFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификаты удостоверяющих центров (certificate authority, CA) верхнего уровня для проверки сертификата узла, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSCertFile | нет |  |  | Абсолютный путь к файлу, который содержит сертификат веб-сервиса или цепочку сертификатов, используется для зашифрованных соединений между компонентами Подсистемы. |
| TLSKeyFile | нет |  |  | Абсолютный путь к файлу, содержащему закрытый ключ веб-сервиса (private key), используется для зашифрованных коммуникаций между компонентами Подсистемы. |

1. Включение (параметр "Include")

Дополнительные файлы или директории можно добавлять к конфигурации Сервера/Прокси/Агента, используя параметр Include.

Если параметр Include используется для включения файла, этот файл должен быть доступен на чтение.

Если параметр Include используется для включения директории, то:

* все файлы из этой директории должны быть доступны на "чтение";
* определенный порядок чтения файлов не предусмотрен, поэтому не следует указывать один параметр в нескольких разных Include-файлах для переопределения настроек;
* все файлы из этой директории включаются в конфигурацию;
* следует с осторожностью использовать файлы архивных копий, создаваемые некоторыми текстовыми редакторами. Например, если при редактировании файла include/my\_specific.conf добавится файл с резервной копией include/my\_specific\_conf.BAK, то затем будут включены оба эти файла. Нужно переместить include/my\_specific.conf.BAK за пределы Include-директории. В Linux для обнаружения ненужных файлов содержимое Include-директории можно проверить командой:

ls -al

Если параметр Include используется для включения файлов с использованием шаблона, то:

* все попадающие под шаблон файлы должны быть доступны на чтение;
* определенный порядок чтения файлов не предусмотрен, поэтому не следует указывать один параметр в нескольких разных Include-файлах для переопределения настроек.

Приложение Б. Протоколы

1. Протокол обмена данными между Сервером и Прокси

Обмен данными "Сервер ↔ Прокси" основывается на формате JSON.

Сообщения запроса и ответа должны начинаться с заголовка и длины данных.

1. Пассивный Прокси

Запрос конфигурации пассивного Прокси "proxy config" отсылается Сервером, чтобы снабдить Прокси данными конфигурации. Этот запрос отсылается каждые ProxyConfigFrequency секунд (параметр конфигурации Сервера) (таблица 9).

Параметры запроса конфигурации

| Имя | | | | Тип значения | Описание |
| --- | --- | --- | --- | --- | --- |
| **Cервер→Прокси:** | | | | | |
| request | | | | строка | "proxy config" |
| <таблица> | | | | объект | Один или более объектов с данными <таблицы>. |
|  | fields | | | массив | Массив имен полей. |
|  | - | | строка | Имя поля. |
| data | | | массив | Массив строк. |
|  | - | | массив | Массив столбцов. |
|  | - | строка,число | Значение столбца с типом, зависящим от типа столбца в схеме базы данных. |
| **Прокси→Cервер:** | | | | | |
| response | | | | строка | Информация об успешности запроса (success или failed). |
| version | | | | строка | Версия Прокси (<major>.<minor>.<build>). |

Запрос данных пассивного Прокси "proxy data" используется для получения от Прокси данных о доступности интерфейсов узлов сети, а также данных истории, обнаружений и авторегистрации. Этот запрос отсылается каждые ProxyDataFrequency секунд (параметр конфигурации Сервера) (таблица 10).

Параметры запроса данных

| Имя | | Тип значения | Описание |
| --- | --- | --- | --- |
| **Cервер→Прокси:** | | | |
| request | | строка | "proxy data" |
| **Прокси→Cервер:** | | | |
| session | | строка | Токен сессии данных. |
| interface availability | | массив | (опционально) Массив объектов данных о доступности интерфейсов. |
|  | interfaceid | число | Идентификатор интерфейса. |
| available | число | Доступность интерфейса: 0, INTERFACE\_AVAILABLE\_UNKNOWN – неизвестно 1, INTERFACE\_AVAILABLE\_TRUE – доступен 2, INTERFACE\_AVAILABLE\_FALSE – недоступен |
| error | строка | Сообщение об ошибке интерфейса либо пустая строка. |
| history data | | массив | (опционально) Массив объектов данных истории. |
|  | itemid | число | Идентификатор элемента данных. |
| clock | число | Отметка времени элемента данных (секунды). |
| ns | число | Отметка времени элемента данных (наносекунды). |
| value | строка | (опционально) Значение элемента данных. |
| id | число | Идентификатор значения (возрастающий счетчик, уникальный в пределах одной сессии данных). |
| timestamp | число | (опционально) Отметка времени (timestamp) элементов данных журнального типа. |
| source | строка | (опционально) Источник (source) элемента данных журнального типа. |
| severity | число | (опционально) Важность (severity) элемента данных журнального типа. |
| eventid | число | (опционально) Идентификатор (eventid) элемента данных журнального типа. |
| state | строка | (опционально) Состояние элемента данных: 0, ITEM\_STATE\_NORMAL 1, ITEM\_STATE\_NOTSUPPORTED |
| lastlogsize | число | (опционально) Последний размер файла журнала (lastlogsize) элемента данных журнального типа. |
| mtime | число | (опционально) Отметка времени последнего обновления (mtime) элемента данных журнального типа. |
| discovery data | | массив | (опционально) Массив объектов данных обнаружения. |
|  | clock | число | Отметка времени данных обнаружения. |
| druleid | число | Идентификатор правила обнаружения. |
| dcheckid | число | Идентификатор проверки обнаружения или null для данных правила обнаружения. |
| type | число | Тип проверки обнаружения: -1 – данные проверки обнаружения 0, SVC\_SSH – проверка сервиса SSH 1, SVC\_LDAP – проверка сервиса LDAP 2, SVC\_SMTP – проверка сервиса SMTP 3, SVC\_FTP – проверка сервиса FTP 4, SVC\_HTTP – проверка сервиса HTTP 5, SVC\_POP – проверка сервиса POP 6, SVC\_NNTP – проверка сервиса NNTP 7, SVC\_IMAP – проверка сервиса IMAP 8, SVC\_TCP – проверка доступности порта TCP 9, SVC\_AGENT – Агент Подсистемы 10, SVC\_SNMPv1 – Агент SNMPv1 11, SVC\_SNMPv2 – Агент SNMPv2 12, SVC\_ICMPPING – пинг ICMP 13, SVC\_SNMPv3 – Агент SNMPv3 14, SVC\_HTTPS – проверка сервиса HTTPS 15, SVC\_TELNET – проверка доступности Telnet |
| ip | строка | IP-адрес хоста. |
| dns | строка | DNS-имя хоста. |
| port | число | (опционально) Номер порта сервиса. |
| key\_ | строка | (опционально) Ключ элемента данных для проверок обнаружения с типом 9 SVC\_AGENT |
| value | строка | (опционально) Полученное от сервиса значение, для большинства сервисов может быть пустым. |
| status | число | (опционально) Статус сервиса: 0, DOBJECT\_STATUS\_UP – Сервис работает (UP) 1, DOBJECT\_STATUS\_DOWN – Сервис не работает (DOWN) |
| auto registration | | массив | (опционально) Массив объектов данных авторегистрации. |
|  | clock | число | Отметка времени данных авторегистрации. |
| host | строка | Имя узла сети. |
| ip | строка | (опционально) IP-адрес узла сети. |
| dns | строка | (опционально) DNS-имя, отрезолвенное из IP-адреса. |
| port | строка | (опционально) Порт узла сети. |
| host\_metadata | строка | (опционально) Метаданные узла сети, отосланные Агентом (на основе параметров конфигурации Агента HostMetadata или HostMetadataItem). |
| tasks | | массив | (опционально) Массив задач. |
|  | type | число | Тип задачи: 0, ZBX\_TM\_TASK\_PROCESS\_REMOTE\_COMMAND\_RESULT – результат удаленной команды |
| status | число | Статус выполнения удаленной команды: 0, ZBX\_TM\_REMOTE\_COMMAND\_COMPLETED – удаленная команда завершилась успешно 1, ZBX\_TM\_REMOTE\_COMMAND\_FAILED – удаленная команда завершилась неудачно |
| error | строка | (опционально) Сообщение об ошибке. |
| parent\_taskid | число | Идентификатор родительской задачи. |
| more | | число | (опционально) 1 – имеются дополнительные данные истории для отправки. |
| clock | | число | (опционально) Отметка времени обмена данными (секунды). |
| ns | | число | (опционально) Отметка времени обмена данными (наносекунды). |
| version | | строка | Версия Прокси (<major>.<minor>.<build>). |
| Cервер→Прокси: | | | |
| response | | строка | Информация об успешности запроса (success или failed). |
| tasks | | массив | (опционально) Массив задач. |
|  | type | число | Тип задачи: 1, ZBX\_TM\_TASK\_PROCESS\_REMOTE\_COMMAND – удаленная команда |
| clock | число | Время создания задачи. |
| ttl | число | Время в секундах до истечения срока задачи. |
| commandtype | число | Тип удаленной задачи: 0, ZBX\_SCRIPT\_TYPE\_CUSTOM\_SCRIPT – применить пользовательский скрипт 1, ZBX\_SCRIPT\_TYPE\_IPMI – использовать IPMI 2, ZBX\_SCRIPT\_TYPE\_SSH – использовать SSH 3, ZBX\_SCRIPT\_TYPE\_TELNET – использовать Telnet 4, ZBX\_SCRIPT\_TYPE\_GLOBAL\_SCRIPT – использовать глобальный скрипт (в данный момент функционал эквивалентен пользовательскому скрипту) |
| command | строка | Команда для удаленного выполнения. |
| execute\_on | число | Целевой объект выполнения для пользовательского скрипта: 0, ZBX\_SCRIPT\_EXECUTE\_ON\_AGENT – выполнить скрипт на Агенте 1, ZBX\_SCRIPT\_EXECUTE\_ON\_SERVER – выполнить скрипт на Сервере 2, ZBX\_SCRIPT\_EXECUTE\_ON\_PROXY – выполнить скрипт на Прокси |
| port | число | (опционально) Порт для команд Telnet и SSH. |
| authtype | число | (опционально) Тип аутентификации для команд SSH. |
| username | строка | (опционально) Имя пользователя для команд Telnet и SSH. |
| password | строка | (опционально) Пароль для команд Telnet и SSH. |
| publickey | строка | (опционально) Открытый ключ (public key) для команд SSH. |
| privatekey | строка | (опционально) Закрытый ключ (private key) для команд SSH. |
| parent\_taskid | число | Идентификатор родительской задачи. |
| hostid | число | Идентификатор целевого узла сети. |

Активный Прокси

Контрольный запрос активного Прокси "proxy heartbeat" отсылается Прокси Сервером, чтобы сообщить, что он в рабочем состоянии. Этот запрос отсылается каждые HeartbeatFrequency секунд (параметр конфигурации Прокси) (таблица 11).

Параметры контрольного запроса

| Имя | Тип значения | Описание |
| --- | --- | --- |
| **Прокси→Cервер:** | | |
| request | строка | "proxy heartbeat" |
| host | строка | Имя Прокси-сервера. |
| version | строка | Версия Прокси-сервера (<major>.<minor>.<build>). |
| **Cервер→Прокси:** | | |
| response | строка | Информация об успешности запроса (success или failed). |

Запрос конфигурации активного Прокси "proxy config" отсылается Прокси Сервером для получения его конфигурационных данных. Этот запрос отсылается каждые ConfigFrequency секунд (параметр конфигурации Прокси) (таблица 12).

Параметры запроса конфигурации

| Имя | | | | Тип значения | Описание |
| --- | --- | --- | --- | --- | --- |
| **Прокси→Cервер:** | | | | | |
| request | | | | строка | "proxy config" |
| host | | | | строка | Имя Прокси-сервера. |
| version | | | | строка | Версия Прокси-сервера (<major>.<minor>.<build>). |
| **Cервер→Прокси:** | | | | | |
| request | | | | строка | "proxy config" |
| <таблица> | | | | объект | Один или более объектов с данными <таблицы>. |
|  | fields | | | массив | Массив имен полей. |
|  | - | | строка | Имя поля. |
| data | | | массив | Массив строк. |
|  | - | | массив | Массив столбцов. |
|  | - | строка,число | Значение столбца с типом, зависящим от типа столбца в схеме базы данных. |
| **Прокси→Cервер:** | | | | | |
| response | | | | строка | Информация об успешности запроса (success или failed). |

Запрос на передачу данных активного Прокси "proxy data" отсылается Прокси Сервером, чтобы предоставить данные о доступности интерфейсов узлов сети, а также данных истории, обнаружений и авторегистрации. Этот запрос отсылается каждые DataSenderFrequency секунд (параметр конфигурации Прокси). Следует обратить внимание, что активный Прокси будет опрашивать Сервер каждую секунду относительно задач на удаленное выполнение (с пустым запросом "proxy data") (таблица 13).

Параметры запроса на передачу

| Имя | | Тип значения | Описание |
| --- | --- | --- | --- |
| **Прокси→Cервер:** | | | |
| request | | строка | "proxy data" |
| host | | строка | Имя Прокси Сервера. |
| session | | строка | Токен сессии данных. |
| interface availability | | массив | (опционально) Массив объектов данных о доступности интерфейсов. |
|  | interfaceid | число | Идентификатор интерфейса. |
| available | число | Доступность интерфейса: 0, INTERFACE\_AVAILABLE\_UNKNOWN – неизвестно 1, INTERFACE\_AVAILABLE\_TRUE – доступен 2, INTERFACE\_AVAILABLE\_FALSE – недоступен |
| error | строка | Сообщение об ошибке интерфейса либо пустая строка. |
| history data | | массив | (опционально) Массив объектов данных истории. |
|  | itemid | число | Идентификатор элемента данных. |
| clock | число | Отметка времени элемента данных (секунды). |
| ns | число | Отметка времени элемента данных (наносекунды). |
| value | строка | (опционально) Значение элемента данных. |
| id | число | Идентификатор значения (возрастающий счетчик, уникальный в пределах одной сессии данных). |
| timestamp | число | (опционально) Отметка времени (timestamp) элементов данных журнального типа. |
| source | строка | (опционально) Источник (source) элемента данных журнального типа. |
| severity | число | (опционально) Важность (severity) элемента данных журнального типа. |
| eventid | число | (опционально) Идентификатор (eventid) элемента данных журнального типа. |
| state | строка | (опционально) Состояние элемента данных: 0, ITEM\_STATE\_NORMAL 1, ITEM\_STATE\_NOTSUPPORTED |
| lastlogsize | число | (опционально) Последний размер файла журнала (lastlogsize) элемента данных журнального типа. |
| mtime | число | (опционально) Отметка времени последнего обновления (mtime) элемента данных журнального типа. |
| discovery data | | массив | (опционально) Массив объектов данных обнаружения. |
|  | clock | число | Отметка времени данных обнаружения. |
| druleid | число | Идентификатор правила обнаружения. |
| dcheckid | число | Идентификатор проверки обнаружения или null для данных правила обнаружения. |
| type | число | Тип проверки обнаружения: -1 – данные проверки обнаружения 0, SVC\_SSH – проверка сервиса SSH 1, SVC\_LDAP – проверка сервиса LDAP 2, SVC\_SMTP – проверка сервиса SMTP 3, SVC\_FTP – проверка сервиса FTP 4, SVC\_HTTP – проверка сервиса HTTP 5, SVC\_POP – проверка сервиса POP 6, SVC\_NNTP – проверка сервиса NNTP 7, SVC\_IMAP – проверка сервиса IMAP 8, SVC\_TCP – проверка доступности порта TCP 9, SVC\_AGENT – Агент Подсистемы 10, SVC\_SNMPv1 – Агент SNMPv1 11, SVC\_SNMPv2 – Агент SNMPv2 12, SVC\_ICMPPING – пинг ICMP 13, SVC\_SNMPv3 – Агент SNMPv3 14, SVC\_HTTPS – проверка сервиса HTTPS 15, SVC\_TELNET – проверка доступности Telnet |
| ip | строка | IP-адрес хоста. |
| dns | строка | DNS-имя хоста. |
| port | число | (опционально) Номер порта сервиса. |
| key\_ | строка | (опционально) Ключ элемента данных для проверок обнаружения с типом 9 SVC\_AGENT |
| value | строка | (опционально) Полученное от сервиса значение, для большинства сервисов может быть пустым. |
| status | число | (опционально) Статус сервиса: 0, DOBJECT\_STATUS\_UP – Сервис работает (UP) 1, DOBJECT\_STATUS\_DOWN – Сервис не работает (DOWN) |
| autoregistration | | массив | (опционально) Массив объектов данных авторегистрации. |
|  | clock | число | Отметка времени данных авторегистрации. |
| host | строка | Имя узла сети. |
| ip | строка | (опционально) IP-адрес узла сети. |
| dns | строка | (опционально) DNS-имя, отрезолвенное из IP-адреса. |
| port | строка | (опционально) Порт узла сети. |
| host\_metadata | строка | (опционально) Метаданные узла сети, отосланные Агентом (на основе параметров конфигурации Агента HostMetadata или HostMetadataItem). |
| tasks | | массив | (опционально) Массив задач. |
|  | type | число | Тип задачи: 0, ZBX\_TM\_TASK\_PROCESS\_REMOTE\_COMMAND\_RESULT – результат удаленной команды |
| status | число | Статус выполнения удаленной команды: 0, ZBX\_TM\_REMOTE\_COMMAND\_COMPLETED – удаленная команда завершилась успешно 1, ZBX\_TM\_REMOTE\_COMMAND\_FAILED – удаленная команда завершилась неудачно |
| error | строка | (опционально) Сообщение об ошибке. |
| parent\_taskid | число | Идентификатор родительской задачи. |
| more | | число | (опционально) 1 – имеются дополнительные данные истории для отправки. |
| clock | | число | (опционально) Отметка времени обмена данными (секунды). |
| ns | | число | (опционально) Отметка времени обмена данными (наносекунды). |
| version | | строка | Версия Прокси Сервера (<major>.<minor>.<build>). |
| **Cервер→Прокси:** | | | |
| response | | строка | Информация об успешности запроса (success или failed). |
| upload | | строка | Управление загрузкой для исторических данных (история, авторегистрация, доступность узлов сети, сетевое обнаружение). Возможные значения: – enabled – нормальная работа;  – disabled – Сервер не принимает данные (возможно, из-за превышения ограничения внутреннего кэша). |
| tasks | | массив | (опционально) Массив задач. |
|  | type | число | Тип задачи: 1, ZBX\_TM\_TASK\_PROCESS\_REMOTE\_COMMAND – удаленная команда |
| clock | число | Время создания задачи. |
| ttl | число | Время в секундах до истечения срока задачи. |
| commandtype | число | Тип удаленной задачи: 0, ZBX\_SCRIPT\_TYPE\_CUSTOM\_SCRIPT – применить пользовательский скрипт 1, ZBX\_SCRIPT\_TYPE\_IPMI – использовать IPMI 2, ZBX\_SCRIPT\_TYPE\_SSH – использовать SSH 3, ZBX\_SCRIPT\_TYPE\_TELNET – использовать Telnet 4, ZBX\_SCRIPT\_TYPE\_GLOBAL\_SCRIPT – использовать глобальный скрипт (в данный момент функционал эквивалентен пользовательскому скрипту) |
| command | строка | Команда для удаленного выполнения. |
| execute\_on | число | Целевой объект выполнения для пользовательского скрипта: 0, ZBX\_SCRIPT\_EXECUTE\_ON\_AGENT – выполнить скрипт на Агенте 1, ZBX\_SCRIPT\_EXECUTE\_ON\_SERVER – выполнить скрипт на Сервере 2, ZBX\_SCRIPT\_EXECUTE\_ON\_PROXY – выполнить скрипт на Прокси |
| port | число | (опционально) Порт для команд Telnet и SSH. |
| authtype | число | (опционально) Тип аутентификации для команд SSH. |
| username | строка | (опционально) Имя пользователя для команд Telnet и SSH. |
| password | строка | (опционально) Пароль для команд Telnet и SSH. |
| publickey | строка | (опционально) Открытый ключ (public key) для команд SSH. |
| privatekey | строка | (опционально) Закрытый ключ (private key) для команд SSH. |
| parent\_taskid | число | Идентификатор родительской задачи. |
| hostid | число | Идентификатор целевого узла сети. |

1. Протокол Агента-1

Для получения более подробной информации следует обратиться к п.Г.3.

1. Протокол Агента-2

В этом разделе представлена информация о запросах и ответах Агента-2:

* "Агент-2 → Сервер" – запрос на активную проверку;
* "Сервер → Агент-2" – ответ активной проверки;
* "Агент-2 → Сервер" – запрос данных Агента;
* "Сервер → Агент-2" – ответ на данные Агента.

Запрос на активную проверку используется для получения активных проверок для последующей обработки Агентом. Запрос посылается Агентом при старте и затем с интервалом RefreshActiveChecks (таблица 14).

Параметры запросов на активную проверку

| Поле | Тип | Обязательное | Значение |
| --- | --- | --- | --- |
| request | строка | да | active checks |
| host | строка | да | Имя узла сети. |
| version | строка | да | Версия Агента: <major>.<minor>. |
| host\_metadata | строка | нет | Параметр конфигурации HostMetadata или значение метрики HostMetadataItem. |
| interface | строка | нет | Параметр конфигурации HostInterface или значение метрики HostInterfaceItem. |
| ip | строка | нет | Первый IP параметра конфигурации LISTENIP, если задан. |
| port | число | нет | Значение параметра конфигурации LISTENPort, если задано и отличается от порта Агента по умолчанию. |

Ответ активных проверок отсылается Сервером обратно Агенту после обработки запроса на активные проверки (таблица 15).

Параметры ответа активные проверок

| Поле | | Тип | Обязательное | Значение |
| --- | --- | --- | --- | --- |
| response | | строка | да | success | failed |
| info | | строка | нет | Информация об ошибке в случае сбоя. |
| data | | массив объектов | нет | Активные проверки элементов данных. |
|  | key | строка | нет | Ключ элемента данных с раскрытыми макросами. |
| itemid | число | нет | Идентификатор элемента данных. |
| delay | строка | нет | Интервал обновления элемента данных. |
| lastlogsize | число | нет | Последний размер файла журнала (lastlogsize) элемента данных. |
| mtime | число | нет | Отметка времени последней модификации (mtime) элемента данных. |
| regexp | | массив объектов | нет | Глобальные регулярные выражения. |
|  | name | строка | нет | Имя глобального регулярного выражения. |
| expression | строка | нет | Глобальное регулярное выражение. |
| expression\_type | число | нет | Тип глобального регулярного выражения. |
| exp\_delimiter | строка | нет | Разделитель глобального регулярного выражения. |
| case\_sensitive | число | нет | Флаг чувствительности к регистру глобального регулярного выражения. |

Запрос данных Агента содержит собранные значения элементов данных (таблица 16).

Параметры запроса данных

| Поле | | Тип | Обязательное | Значение |
| --- | --- | --- | --- | --- |
| request | | строка | да | agent data |
| host | | строка | да | Имя узла сети. |
| version | | строка | да | Версия Агента: <major>.<minor>. |
| session | | строка | да | Уникальный идентификатор сессии, генерируется каждый раз при старте Агента. |
| data | | массив объектов | да | Значения элементов данных. |
|  | id | число | да | Идентификатор значения (возрастающий счетчик, используемый для контроля дублированных значений в случае сетевых проблем). |
| itemid | число | да | Идентификатор элемента данных. |
| value | строка | нет | Значение элемента данных. |
| lastlogsize | число | нет | Последний размер файла журнала (lastlogsize) элемента данных. |
| mtime | число | нет | Отметка времени последнего обновления (mtime) элемента данных. |
| state | число | нет | Состояние элемента данных. |
| source | строка | нет | Источник (source) значения журнала событий. |
| eventid | число | нет | Идентификатор (eventid) значения журнала событий. |
| severity | число | нет | Важность (severity) значения журнала событий. |
| timestamp | число | нет | Отметка времени (timestamp) значения журнала событий. |
| clock | число | да | Отметка времени значения (секунд с начала эпохи). |
| ns | число | да | Наносекунды отметки времени значения. |

Ответ на данные Агента отсылается Сервером назад Агенту после обработки запроса данных Агента (таблица 17).

Параметра ответы на данные

| Поле | Тип | Обязательное | Значение |
| --- | --- | --- | --- |
| response | строка | да | success | failed |
| info | строка | да | Результаты обработки элементов данных. |

1. Протокол плагина Агента-2

Протокол Агента-2 основан на коде, размере и модели данных (таблицы 18-20).

Код

| Тип | Размер | Комментарии |
| --- | --- | --- |
| Byte | 4 | Тип полезной нагрузки, в настоящее время  поддерживается только JSON. |

Размер

| Тип | Размер | Комментарии |
| --- | --- | --- |
| Byte | 4 | Размер текущей полезной нагрузки в байтах. |

Данные полезной нагрузки

| Тип | Размер | Комментарии |
| --- | --- | --- |
| Byte | Определяется полем Размер | Данные в формате JSON. |

**Определение данных полезной нагрузки**

Параметры, присутствующие во всех вопросах/ответах, – в таблице 21.

Общие параметры

| Имя | Тип | Комментарии |
| --- | --- | --- |
| id | uint32 | Для запросов – возрастающий идентификатор, используемый, чтобы связать запросы с ответами. Уникален в пределах направления запроса (т.е. от Агента к плагину или от плагина к Агенту). Для ответов – идентификатор соответствующего запроса. |
| type | uint32 | Тип запроса. |

Запрос журналирования отсылается плагином, чтобы записать журнальное сообщение в журнальный файл Агента:

|  |  |
| --- | --- |
| направление | плагин → Агент |
| ответ | нет |

Параметры, специфичные для запросов журналирования, – в таблице 22.

Параметры, специфичные для запросов журналирования

| Имя | Тип | Комментарии |
| --- | --- | --- |
| severity | uint32 | Важность сообщения (уровень журналирования). |
| message | string | Сообщение для журнала. |

Запрос регистрации отсылается Агентом в фазе начальной загрузки Агента, чтобы получить предоставляемые метрики для регистрации плагина:

|  |  |
| --- | --- |
| направление | Агент → плагин |
| ответ | да |

Параметры, специфичные для запросов регистрации, – в таблице 23.

Параметры запроса регистрации

| Имя | Тип | Комментарии |
| --- | --- | --- |
| version | string | Версия протокола <major>.<minor> |

Ответ плагина на запрос на регистрацию:

|  |  |
| --- | --- |
| направление | плагин → Агент |
| ответ | неприменимо |

Параметры, специфичные для ответов на регистрацию, – в таблице 24.

Параметры ответов на регистрацию

| Имя | Тип | Комментарии |
| --- | --- | --- |
| name | string | Имя плагина. |
| metrics | array of strings (опционально) | Метрики с описаниями, как используется в плагине. Возвращает RegisterMetrics(). Отсутствует, если возвращается ошибка. |
| interfaces | uint32 (опционально) | Битовая маска интерфейсов, поддерживаемых плагином. Отсутствует, если возвращается ошибка. |
| error | string (опционально) | Сообщение об ошибке возвращается, если плагин не может запуститься. Отсутствует, если возвращаются метрики. |

Запрос на выполнение функции запуска (Start) интерфейса Runner:

|  |  |
| --- | --- |
| направление | Агент → плагин |
| ответ | нет |

Запрос на запуск не имеет специфичных параметров, он содержит только параметры общих данных.

Запрос на останов отсылается Агентом, чтобы остановить плагин:

|  |  |
| --- | --- |
| направление | Агент → плагин |
| ответ | нет |

Запрос на запуск не имеет специфичных параметров, он содержит только параметры общих данных.

Запрос на выполнение функции экспорта (Export) интерфейса Exporter:

|  |  |
| --- | --- |
| направление | Агент → плагин |
| ответ | нет |

Параметры, специфичные для запросов экспорта, – в таблице 25.

Параметры запроса экспорта

| Имя | Тип | Комментарии |
| --- | --- | --- |
| key | string | Ключ плагина. |
| parameters | array of strings (опционально) | Параметры для функции Export. |

Ответ экспорта из функции Export интерфейса Exporter:

|  |  |
| --- | --- |
| направление | плагин → Агент |
| ответ | неприменимо |

Параметры, специфичные для ответов экспорта, ­– в таблице 26.

Параметры ответов экспорта

| Имя | Тип | Комментарии |
| --- | --- | --- |
| value | string (опционально) | Значение ответа из функции Export. Отсутствует, если возвращается ошибка. |
| error | string (опционально) | Сообщение об ошибке, если функция Export не была выполнена успешно. Отсутствует, если возвращается значение (value). |

Запрос на конфигурацию на выполнение функции Configure интерфейса Configurator:

|  |  |
| --- | --- |
| направление | Агент → плагин |
| ответ | неприменимо |

Параметры, специфичные для запросов Configure, – в таблице 27.

Параметры запроса конфигурации

| Имя | Тип | Комментарии |
| --- | --- | --- |
| global\_options | объект JSON | Объект JSON, содержащий глобальные опции конфигурации Агента. |
| private\_options | объект JSON (опционально) | Объект JSON, содержащий персональные опции конфигурации плагина, если есть. |

Запрос на валидацию на выполнение функции Validate интерфейса Configurator:

|  |  |
| --- | --- |
| направление | Агент → плагин |
| ответ | да |

Параметры, специфичные для запросов Validate, – в таблице 28.

Параметры запроса на валидацию

| Имя | Тип | Комментарии |
| --- | --- | --- |
| private\_options | объект JSON (опционально) | Объект JSON, содержащий персональные опции конфигурации плагина, если есть. |

Ответ на валидацию от функции Validate интерфейса Configurator:

|  |  |
| --- | --- |
| направление | плагин → Агент |
| ответ | неприменимо |

Параметры, специфичные для ответов на Validate, – в таблице 29.

Параметры ответов на валидацию

| Имя | Тип | Комментарии |
| --- | --- | --- |
| error | string (опционально) | Сообщение об ошибке, возвращаемое, если функция Validate не была выполнена успешно. Отсутствует при успешном завершении. |

1. Протокол sender

Для получения более подробной информации следует обратиться к п.Д.12.

1. Заголовок

Заголовок присутствует во всех сообщениях запросов и ответов между компонентами Подсистемы. Требуется определить длину сообщения, его сжатие и величину.

Протокол связи Подсистемы имеет ограничение размера пакета в 1 ГБ на одно соединение. Это ограничение в 1 ГБ относится как к длине полученных пакетных данных, так и к длине несжатых данных.

При отправке конфигурации на Прокси ограничение размера пакета увеличено до 4 ГБ, чтобы позволить синхронизацию больших конфигураций. Если длина данных до сжатия превышает 4 ГБ, Сервер автоматически начинает использовать формат больших пакетов (флаг 0x04), что повышает ограничение размера пакета до 16 ГБ.

Следует обратить внимание, что хотя для отправки любых данных можно использовать большой формат пакетов, только процесс синхронизации конфигурации Прокси (configuration syncer) может обрабатывать пакеты размером более 1 ГБ.

Заголовок состоит из четырех полей. Все числа в заголовке имеют прямой порядок байтов (little-endian) (таблица 30).

Описание полей заголовка

| Поле | Размер | Размер (большой пакет) | Описание |
| --- | --- | --- | --- |
| <ПРОТОКОЛ> | 4 | 4 | "ZBXD" или 5A 42 58 44 |
| <ФЛАГИ> | 1 | 1 | Флаги протокола: 0x01 – протокол связи Подсистемы 0x02 – сжатие 0x04 – большой пакет |
| <ДЛИНАДАННЫХ> | 4 | 8 | Длина данных. |
| <РЕЗЕРВ> | 4 | 8 | При использовании сжатия (флаг 0x02) – длина несжатых данных Когда сжатие не используется – 00 00 00 00 |

Ниже представлены несколько фрагментов кода, демонстрирующих, как к отсылаемым данным (переменная data) добавить заголовок протокола, чтобы пакет, который нужно отправить в Подсистему, получился правильно интерпретируемым. Эти фрагменты кода подразумевают, что данные не превышают 1 ГБ, поэтому формат больших пакетов не используется.

**Python**

packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data

или

def zbx\_create\_header(plain\_data\_size, compressed\_data\_size=None):

protocol = b"ZBXD"

flags = 0x01

if compressed\_data\_size is None:

datalen = plain\_data\_size

reserved = 0

else:

flags |= 0x02

datalen = compressed\_data\_size

reserved = plain\_data\_size

return protocol + struct.pack("<BII", flags, datalen, reserved)

packet = zbx\_create\_header(len(data)) + data

**Perl**

my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;

или

sub zbx\_create\_header($;$)

{

my $plain\_data\_size = shift;

my $compressed\_data\_size = shift;

my $protocol = "ZBXD";

my $flags = 0x01;

my $datalen;

my $reserved;

if (!defined($compressed\_data\_size))

{

$datalen = $plain\_data\_size;

$reserved = 0;

}

else

{

$flags |= 0x02;

$datalen = $compressed\_data\_size;

$reserved = $plain\_data\_size;

}

return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);

}

my $packet = zbx\_create\_header(length($data)) . $data;

**PHP**

$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;

или

function zbx\_create\_header($plain\_data\_size, $compressed\_data\_size = null)

{

$protocol = "ZBXD";

$flags = 0x01;

if (is\_null($compressed\_data\_size))

{

$datalen = $plain\_data\_size;

$reserved = 0;

}

else

{

$flags |= 0x02;

$datalen = $compressed\_data\_size;

$reserved = $plain\_data\_size;

}

return $protocol . chr($flags) . pack("VV", $datalen, $reserved);

}

$packet = zbx\_create\_header(strlen($data)) . $data;

**Bash**

datalen=$(printf "%08x" ${#data})

datalen=\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}

printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"

1. Протокол экспорта в режиме реального времени

В этом разделе представлены подробности реализации протокола экспорта в режиме реального времени в формате JSON с новой строкой в качестве разделителя для:

* триггерных событий;
* значений элементов данных;
* динамики изменений.

Все файлы имеют расширение .ndjson. Каждая строка файла экспорта является объектом JSON.

1. Триггерные события

Для события проблемы экспортируется информация из таблицы 31.

Поля триггерного события проблемы

| Поле | | | Тип | Описание |
| --- | --- | --- | --- | --- |
| clock | | | число | Количество секунд с начала эпохи до момента обнаружения проблемы (целая часть). |
| ns | | | число | Количество наносекунд, которые необходимо добавить к clock, чтобы получить точное время обнаружения проблемы. |
| value | | | число | 1 (всегда). |
| eventid | | | число | Идентификатор события проблемы. |
| name | | | строка | Имя события проблемы. |
| severity | | | число | Важность события проблемы (0 – Не классифицировано, 1 – Информация, 2 – Предупреждение, 3 – Средняя, 4 – Высокая, 5 – Чрезвычайная). |
| hosts | | | массив | Список узлов сети, участвующих в выражении триггера; в массиве должен быть хотя бы один элемент. |
|  | - | | объект | |
|  | host | строка | Имя узла сети. |
| name | строка | Видимое имя узла сети. |
| groups | | | массив | Список групп всех узлов сети, участвующих в выражении триггера; в массиве должен быть хотя бы один элемент. |
|  | - | | строка | Имя группы узлов сети. |
| tags | | | массив | Список тегов проблемы (может быть пустым). |
|  | - | | объект | |
|  | tag | строка | Имя тега. |
| value | строка | Значение тега (может быть пустым). |

Для события восстановления экспортируется информация из таблицы 32.

Поля триггерного события восстановления

| Поле | Тип | Описание |
| --- | --- | --- |
| clock | число | Количество секунд с начала эпохи до момента восстановления проблемы (целая часть). |
| ns | число | Количество наносекунд, которые нужно добавить к clock, чтобы получить точное время восстановления проблемы. |
| value | число | 0 (всегда). |
| eventid | число | Идентификатор события восстановления. |
| p\_eventid | число | Идентификатор события проблемы. |

1. Значения элементов данных

Для собранного значения элемента данных экспортируется информация из таблицы 33.

Поля значения элемента данных

| Поле | | Тип | Описание |
| --- | --- | --- | --- |
| host | | объект | Имя узла сети этого элемента данных. |
|  | host | строка | Имя узла сети. |
| name | строка | Видимое имя узла сети. |
| groups | | массив | Список групп узлов сети для узла сети этого элемента данных; в массиве должен быть по крайней мере один элемент. |
|  | - | строка | Имя группы узла сети. |
| itemid | | число | ID элемента данных. |
| name | | строка | Видимое имя элемента данных. |
| clock | | число | Количество секунд с начала эпохи до момента сбора значения (целая часть). |
| ns | | число | Количество наносекунд, которые нужно добавить к clock для получения точного времени сбора значения. |
| timestamp (только Журнал (лог)) | | число | 0, если недоступно. |
| source (только Журнал (лог)) | | строка | Пустая строка если недоступно. |
| severity (только Журнал (лог)) | | число | 0, если недоступно. |
| eventid (только Журнал (лог)) | | число | 0, если недоступно. |
| value | | число (для числовых элементов данных) или строка (для текстовых элементов данных) | Собранное значение элемента данных. |
| type | | число | Тип собранного значения: 0 – Числовой (с плавающей точкой), 1 – Символ, 2 – Журнал (лог), 3 – Числовой (целое положительное), 4 – Текст |

1. Динамика изменений

Для подсчитанного значения динамики изменений экспортируется информация из таблицы 34.

Поля динамики изменений

| Поле | | Тип | Описание |
| --- | --- | --- | --- |
| host | | объект | Имя узла сети этого элемента данных. |
|  | host | строка | Имя узла сети. |
| name | строка | Видимое имя узла сети. |
| groups | | массив | Список групп узлов сети этого элемента данных; в массиве должен быть по крайней мере один элемент. |
|  | - | строка | Имя узла сети. |
| itemid | | число | ID элемента данных. |
| name | | строка | Видимое имя элемента данных. |
| clock | | число | Количество секунд с начала эпохи до момента получения значения (целая часть). |
| count | | число | Количество значений, собранных за данный час. |
| min | | число | Наименьшее значение элемента данных за данный час. |
| avg | | число | Среднее значение элемента данных за данный час. |
| max | | число | Наибольшее значение элемента данных за данный час. |
| type | | число | Тип значения: 0 – числовой (с плавающей точкой), 3 – числовой беззнаковый |

Приложение В. Типы процессов

1. Типы процессов Сервера

* alert manager – менеджер задач оповещений;
* alert syncer – процесс записи оповещений в БД;
* alerter – процесс отправки оповещений;
* availability manager – процесс для обновления доступности узлов сети;
* configuration syncer – процесс управления кэшем данных конфигурации в оперативной памяти;
* discoverer – процесс обнаружения устройств;
* escalator – процесс эскалации действий;
* ha manager – процесс управления кластером высокой доступности;
* history poller – процесс обработки вычисляемых, агрегированных и внутренних проверок, требующих подключения к базе данных;
* history syncer – процесс, который записывает историю в БД;
* housekeeper – процесс удаления устаревших данных истории;
* http poller – поллер веб-мониторинга;
* icmp pinger – поллер проверок icmpping;
* ipmi manager – менеджер IPMI-поллеров;
* ipmi poller – поллер для проверок через IPMI;
* java poller – поллер для Java-проверок;
* lld manager – процесс менеджера задач низкоуровневого обнаружения;
* lld worker – рабочий процесс для задач низкоуровневого обнаружения;
* odbc poller – поллер для ODBC-проверок;
* poller – обычный поллер для пассивных проверок;
* preprocessing manager – менеджер задач предобработки;
* preprocessing worker – рабочий процесс предобработки данных;
* proxy poller – поллер для пассивных Прокси;
* report manager – процесс менеджера задач генерации отчетов по расписанию;
* report writer – процесс генерации отчетов по расписанию;
* self-zabbix – процесс сбора внутренней статистики Сервера;
* service manager – процесс управления услугами для получения информации о проблемах, тегах проблем и событиях восстановления после проблем от процессов history syncer, task manager и alert manager;
* snmp trapper – траппер сбора/обработки SNMP-трапов;
* task manager – процесс для удаленного выполнения задач, которые запрашиваются другими компонентами (например, функционал закрытия проблемы, подтверждения проблемы, принудительной проверки значения элемента данных, удаленной команды);
* timer – процесс обработки обслуживаний;
* trapper – процесс-улавливатель для активных проверок, трапов и взаимодействия с Прокси;
* unreachable poller – поллер недоступных устройств.

1. Типы процессов Прокси

* availability manager – процесс для обновления доступности узлов сети;
* configuration syncer – процесс управления кэшем данных конфигурации в оперативной памяти;
* data sender – процесс отправки данных с Прокси;
* discoverer – процесс обнаружения устройств;
* heartbeat sender – процесс уведомления Сервера Прокси Сервером о своем состоянии;
* history poller – процесс обработки вычисляемых, агрегированных и внутренних проверок, требующих подключения к базе данных;
* history syncer – процесс, который записывает историю в БД;
* housekeeper – процесс удаления устаревших данных истории;
* http poller – поллер веб-мониторинга;
* icmp pinger – поллер проверок icmpping;
* ipmi manager – менеджер IPMI-поллеров;
* ipmi poller – поллер для проверок через IPMI;
* java poller – поллер для Java-проверок;
* odbc poller – поллер для ODBC-проверок;
* poller – обычный поллер для пассивных проверок;
* preprocessing manager – менеджер задач предобработки;
* preprocessing worker – рабочий процесс предобработки данных;
* self-zabbix – процесс сбора внутренней статистики Сервера;
* snmp trapper – траппер сбора/обработки SNMP-трапов;
* task manager – процесс для удаленного выполнения задач, которые запрашиваются другими компонентами (например, функционал закрытия проблемы, подтверждения проблемы, принудительной проверки значения элемента данных, удаленной команды);
* trapper – процесс-улавливатель для активных проверок, трапов и Прокси;
* unreachable poller – поллер недоступных устройств.

Приложение Г. Элементы данных

1. Поддерживаемые элементы данных по платформам

В таблице 35 показаны поддерживаемые Агентом элементы данных на различных платформах:

* элементы данных с пометкой "X" поддерживаются;
* элементы данных с пометкой "-" не поддерживаются;
* если элемент данных с пометкой "?", то поддержка не определена;
* если элемент данных с пометкой "r", то требуются права доступа уровня root;
* параметры, которые заключены в угловые скобки "<>**"**, являются необязательными.

В таблицу не включены элементы данных Агента, поддерживаемые или относящиеся только к Windows.

Поддерживаемые Агентом элементы данных на различных платформах

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NetBSD | | | | | | | | | | | | ▼▼ |
| OpenBSD | | | | | | | | | | | ▼▼ |
| Mac OS X | | | | | | | | | | ▼▼ |
| Tru64 | | | | | | | | | ▼▼ |
| AIX | | | | | | | | ▼▼ |
| HP-UX | | | | | | | ▼▼ |
| Solaris | | | | | | ▼▼ |
| FreeBSD | | | | | ▼▼ |
| Linux 2.6 (и новее) | | | | ▼▼ |
| Linux 2.4 | | | ▼▼ |
| Windows | | ▼▼ |
| ▼ Item ▼ | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| **agent.HostMetadata** | | X | X | X | X | X | X | X | X | X | X | X |
| **agent.Hostname** | | X | X | X | X | X | X | X | X | X | X | X |
| **agent.ping** | | X | X | X | X | X | X | X | X | X | X | X |
| **agent.variant** | | X | X | X | X | X | X | X | X | X | X | X |
| **agent.version** | | X | X | X | X | X | X | X | X | X | X | X |
| **kernel.maxfiles** | | - | X | X | X | - | - | - | ? | X | X | X |
| **kernel.maxproc** | | - | - | X | X | X | - | - | ? | X | X | X |
| **kernel.openfiles** | | - | X | X | ? | ? | ? | ? | ? | ? | ? | ? |
| **log[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>,<постоянное\_хранилище>]** | | X4 | X | X | X | X | X | X | X | X | X | X |
| постоянное\_хранилище ▲ | | - | X | X | X | X | X | X | X | X | X | X |
| **log.count[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>,<постоянное\_хранилище>]** | | X4 | X | X | X | X | X | X | X | X | X | X |
| постоянное\_хранилище ▲ | | - | X | X | X | X | X | X | X | X | X | X |
| **logrt[регулярное\_выражение\_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>,<опции>,<постоянное\_хранилище>]** | | X4 | X | X | X | X | X | X | X | X | X | X |
| постоянное\_хранилище ▲ | | - | X | X | X | X | X | X | X | X | X | X |
| **logrt.count[регулярное\_выражение\_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>,<опции>,<постоянное\_хранилище>]** | | X4 | X | X | X | X | X | X | X | X | X | X |
| постоянное\_хранилище ▲ | | - | X | X | X | X | X | X | X | X | X | X |
| **modbus.get[точка входа,<id ведомого>,<функция>,<адрес>,<количество>,<тип>,<порядок следования байтов>,<сдвиг>]** | | X | X | X | - | - | - | - | - | - | - | - |
| **net.dns[<ip>,имя,<тип>,<время ожидания>,<количество>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.dns.record[<ip>,имя,<тип>,<время ожидания>,<количество>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.if.collisions[if]** | | - | X | X | X | X | - | X | - | X | X | r |
| **net.if.discovery** | | X | X | X | X | X | X | X | - | - | X | X |
| **net.if.in[if,<режим>]** | | X | X | X | X | X | X1 | X | - | X | X | r |
| режим ▲ | Bytes (по умолчанию) | X | X | X | X | X | X | X | - | X | X | r |
| packets | X | X | X | X | X | X | X | - | X | X | r |
| errors | X | X | X | X | X2 | X | X | - | X | X | r |
| dropped | X | X | X | X | - | X | - | - | X | X | r |
| overruns | - | X | X | - | - | - | - | - | - | - | - |
| frame | - | X | X | - | - | - | - | - | - | - | - |
| compressed | - | X | X | - | - | - | - | - | - | - | - |
| multicast | - | X | X | - | - | - | - | - | - | - | - |
| **net.if.out[if,<режим>]** | | X | X | X | X | X | X1 | X | - | X | X | r |
| режим ▲ | Bytes (по умолчанию) | X | X | X | X | X2 | X | X | - | X | X | r |
| packets | X | X | X | X | X | X | X | - | X | X | r |
| errors | X | X | X | X | X2 | X | X | - | X | X | r |
| dropped | X | X | X | - | - | X | - | - | - | - | - |
| overruns | - | X | X | - | - | - | - | - | - | - | - |
| collision | - | X | X | - | - | - | - | - | - | - | - |
| carrier | - | X | X | - | - | - | - | - | - | - | - |
| compressed | - | X | X | - | - | - | - | - | - | - | - |
| **net.if.total[if,<режим>]** | | X | X | X | X | X | X1 | X | - | X | X | r |
| режим ▲ | Bytes (по умолчанию) | X | X | X | X | X2 | X | X | - | X | X | r |
| packets | X | X | X | X | X | X | X | - | X | X | r |
| errors | X | X | X | X | X2 | X | X | - | X | X | r |
| dropped | X | X | X | - | - | X | - | - | - | - | - |
| overruns | - | X | X | - | - | - | - | - | - | - | - |
| compressed | - | X | X | - | - | - | - | - | - | - | - |
| **net.tcp.LISTEN[port]** | | X | X | X | X | X | - | - | - | X | - | - |
| **net.tcp.port[<ip>,порт]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.tcp.service[сервис,<ip>,<порт>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.tcp.service.perf[сервис,<ip>,<порт>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.tcp.socket.count[<локальный\_адрес>,<локальный\_порт>,<удаленный\_адрес>,<удаленный\_порт>,<состояние>]** | | X7 | X | X | - | - | - | - | - | - | - | - |
| **net.udp.LISTEN[порт]** | | - | X | X | X | X | - | - | - | X | - | - |
| **net.udp.service[сервис,<ip>,<порт>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.udp.service.perf[сервис,<ip>,<порт>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **net.udp.socket.count[<локальный\_адрес>,<локальный\_порт>,<удаленный\_адрес>,<удаленный\_порт>,<состояние>]** | | X7 | X | X | - | - | - | - | - | - | - | - |
| **proc.cpu.util[<имя>,<пользователь>,<тип>,<cmdline>,<режим>,<зона>]** | | - | X | X | - | X 3 | - | - | - | - | - | - |
| тип ▲ | total (по умолчанию) | - | X | X | - | X | - | - | - | - | - | - |
| user | - | X | X | - | X | - | - | - | - | - | - |
| system | - | X | X | - | X | - | - | - | - | - | - |
| режим ▲ | avg1 (по умолчанию) | - | X | X | - | X | - | - | - | - | - | - |
| avg5 | - | X | X | - | X | - | - | - | - | - | - |
| avg15 | - | X | X | - | X | - | - | - | - | - | - |
| зона ▲ | current (по умолчанию) | - | - | - | - | X | - | - | - | - | - | - |
| all | - | - | - | - | X | - | - | - | - | - | - |
| **proc.mem[<имя>,<пользователь>,<режим>,<cmdline><тип памяти>]** | | - | X | X | X | X3 | - | X | X | - | X | X |
| режим ▲ | sum (по умолчанию) | - | X | X | X | X | - | X | X | - | X | X |
| avg | - | X | X | X | X | - | X | X | - | X | X |
| max | - | X | X | X | X | - | X | X | - | X | X |
| min | - | X | X | X | X | - | X | X | - | X | X |
| тип памяти ▲ | | - | X | X | X | X | - | X | - | - | - | - |
| **proc.num[<имя>,<пользователь>,<состояние>,<cmdline>,<зона>]** | | X | X | X | X | X3 | X | X | X | - | X | X |
| состояние ▲ | all (по умолчанию) | - | X | X | X | X | X | X | X | - | X | X |
| disk | - | X | X | X | - | - | - | - | - | X | X |
| sleep | - | X | X | X | X | X | X | X | - | X | X |
| zomb | - | X | X | X | X | X | X | X | - | X | X |
| run | - | X | X | X | X | X | X | X | - | X | X |
| trace | - | X | X | X | - | - | - | - | - | X | X |
| cmdline ▲ | | - | X | X | X | X | X | X | X | - | X | X |
| зона ▲ | current (по умолчанию) | - | - | - | - | X | - | - | - | - | - | - |
| all | - | - | - | - | X | - | - | - | - | - | - |
| **sensor[устройство,сенсор,<режим>]** | | - | X | X | - | - | - | - | - | - | X | - |
| **system.boottime** | | - | X | X | X | X | - | - | - | X | X | X |
| **system.cpu.discovery** | | X | X | X | X | X | X | X | X | X | X | X |
| **system.cpu.intr** | | - | X | X | X | X | - | X | - | - | X | X |
| **system.cpu.load[<cpu>,<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| cpu ▲ | all (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| percpu | X | X | X | X | X | X | X | - | X | X | X |
| режим ▲ | avg1 (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| avg5 | X | X | X | X | X | X | X | X | X | X | X |
| avg15 | X | X | X | X | X | X | X | X | X | X | X |
| **system.cpu.num[<тип>]** | | X | X | X | X | X | X | X | - | X | X | X |
| тип ▲ | online (по умолчанию) | X | X | X | X | X | X | X | - | X | X | X |
| max | - | X | X | X | X | - | - | - | X | - | - |
| **system.cpu.switches** | | - | X | X | X | X | - | X | - | - | X | X |
| **system.cpu.util[<cpu>,<тип>,<режим>,<логический\_или\_физический>]** | | X | X | X | X | X | X | X | X | - | X | X |
| тип ▲ | user (по умолчанию) | - | X | X | X | X | X | X | X | - | X | X |
| nice | - | X | X | X | - | X | - | X | - | X | X |
| idle | - | X | X | X | X | X | X | X | - | X | X |
| system (по умолчанию для Windows) | X | X | X | X | X | X | X | X | - | X | X |
| iowait | - | - | X | - | X | - | X | - | - | - | - |
| interrupt | - | - | X | X | - | - | - | - | - | X | - |
| softirq | - | - | X | - | - | - | - | - | - | - | - |
| steal | - | - | X | - | - | - | - | - | - | - | - |
| guest | - | - | X | - | - | - | - | - | - | - | - |
| guest\_nice | - | - | X | - | - | - | - | - | - | - | - |
| режим ▲ | avg1 (по умолчанию) | X | X | X | X | X | X | X | X | - | X | X |
| avg5 | X | X | X | X | X | X | X | - | - | X | X |
| avg15 | X | X | X | X | X | X | X | - | - | X | X |
| логический\_или\_физический ▲ | logical (по умолчанию) | - | - | - | - | - | - | X | - | - | - | - |
| physical | - | - | - | - | - | - | X | - | - | - | - |
| **system.Hostname[<тип>,<преобразование>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **system.hw.chassis[<информация>]** | | - | X | X | - | - | - | - | - | - | - | - |
| **system.hw.cpu[<cpu>,<информация>]** | | - | X | X | - | - | - | - | - | - | - | - |
| **system.hw.devices[<тип>]** | | - | X | X | - | - | - | - | - | - | - | - |
| **system.hw.macaddr[<интерфейс>,<формат>]** | | - | X | X | - | - | - | - | - | - | - | - |
| **system.localtime[<тип>]** | | X | X | X | X | X | X | X | X | X | X | X |
| тип ▲ | utc (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| local | X | X | X | X | X | X | X | X | X | X | X |
| **system.run[команда,<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| режим ▲ | wait (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| nowait | X | X | X | X | X | X | X | X | X | X | X |
| **system.stat[ресурс,<тип>]** | | - | - | - | - | - | - | X | - | - | - | - |
| **system.sw.arch** | | X | X | X | X | X | X | X | X | X | X | X |
| **system.sw.os[<информация>]** | | - | X | X | - | - | - | - | - | - | - | - |
| **system.sw.packages[<регулярное выражение>,<менеджер>,<формат>]** | | - | X | X | - | - | - | - | - | - | - | - |
| **system.swap.in[<устройство>,<тип>]** (указание устройства поддерживается только на Linux) | | - | X | X | - | X | - | - | - | - | X | - |
| тип ▲ (pages будет работать, только если устройство не указано) | count (по умолчанию для всех, кроме Linux) | - | X | X | - | X | - | - | - | - | X | - |
| sectors | - | X | X | - | - | - | - | - | - | - | - |
| pages (по умолчанию для Linux) | - | X | X | - | X | - | - | - | - | X | - |
| **system.swap.out[<устройство>,<тип>]** (указание устройства поддерживается только на Linux) | | - | X | X | - | X | - | - | - | - | X | - |
| тип ▲ (pages будет работать, только если устройство не указано) | count (по умолчанию для всех, кроме Linux) | - | X | X | - | X | - | - | - | - | X | - |
| sectors | - | X | X | - | - | - | - | - | - | - | - |
| pages (по умолчанию для Linux) | - | X | X | - | X | - | - | - | - | X | - |
| **system.swap.size[<устройство>,<тип>]** (указание устройства поддерживается только на FreeBSD, для других платформ должно быть пусто либо "all") | | X | X | X | X | X | - | X | X | - | X | - |
| тип ▲ | free (по умолчанию) | X | X | X | X | X | - | X | X | - | X | - |
| total | X | X | X | X | X | - | X | X | - | X | - |
| used | X | X | X | X | X | - | X | X | - | X | - |
| pfree | X | X | X | X | X | - | X | X | - | X | - |
| pused | X6 | X | X | X | X | - | X | X | - | X | - |
| **system.uname** | | X | X | X | X | X | X | X | X | X | X | X |
| **system.uptime** | | X | X | X | X | X | - | X | ? | X | X | X |
| **system.users.num** | | - | X | X | X | X | X | X | X | X | X | X |
| **systemd.unit.discovery** | | - | X | X | - | - | - | - | - | - | - | - |
| **systemd.unit.get** | | - | X | X | - | - | - | - | - | - | - | - |
| **systemd.unit.info** | | - | X | X | - | - | - | - | - | - | - | - |
| **vfs.dev.discovery** | | - | X | X | - | - | - | - | - | - | - | - |
| **vfs.dev.read[<устройство>,<тип>,<режим>]** | | - | X | X | X | X | - | X | - | - | X | - |
| тип ▲ | sectors | - | X | X | - | - | - | - | - | - | - | - |
| operations (по умолчанию для OpenBSD, AIX) | - | X | X | X | X | - | X | - | - | X | - |
| Bytes (по умолчанию для Solaris) | - | - | - | X | X | - | X | - | - | X | - |
| sps (по умолчанию для Linux) | - | X | X | - | - | - | - | - | - | - | - |
| ops | - | X | X | X | - | - | - | - | - | - | - |
| bps (по умолчанию для FreeBSD) | - | - | - | X | - | - | - | - | - | - | - |
| режим ▲ (совместимо только с типами: sps, ops, bps) | avg1 (по умолчанию) | - | X | X | X | - | - | - | - | - | - | - |
| avg5 | - | X | X | X | - | - | - | - | - | - | - |
| avg15 | - | X | X | X | - | - | - | - | - | - | - |
| **vfs.dev.write[<устройство>,<тип>,<режим>]** | | - | X | X | X | X | - | X | - | - | X | - |
| тип ▲ | sectors | - | X | X | - | - | - | - | - | - | - | - |
| operations (по умолчанию для OpenBSD, AIX) | - | X | X | X | X | - | X | - | - | X | - |
| Bytes (по умолчанию для Solaris) | - | - | - | X | X | - | X | - | - | X | - |
| sps (по умолчанию для Linux) | - | X | X | - | - | - | - | - | - | - | - |
| ops | - | X | X | X | - | - | - | - | - | - | - |
| bps (по умолчанию для FreeBSD) | - | - | - | X | - | - | - | - | - | - | - |
| режим ▲ (совместимо только с типами: sps, ops, bps) | avg1 (по умолчанию) | - | X | X | X | - | - | - | - | - | - | - |
| avg5 | - | X | X | X | - | - | - | - | - | - | - |
| avg15 | - | X | X | X | - | - | - | - | - | - | - |
| **vfs.dir.count[директория,<regex\_вкл>,<regex\_искл>,<типы\_вкл>,<типы\_искл>,<макс\_глубина>,<мин\_размер>,<макс\_размер>,<мин\_возраст>,<макс\_возраст>,<regex\_искл\_директории>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.dir.get[директория,<regex\_вкл>,<regex\_искл>,<типы\_вкл>,<типы\_искл>,<макс\_глубина>,<мин\_размер>,<макс\_размер>,<мин\_возраст>,<макс\_возраст>,<regex\_искл\_директории>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.dir.size[директория,<regex\_вкл>,<regex\_искл>,<режим>,<макс\_глубина>,<regex\_искл\_директории>]** | | X | X | X | ? | ? | ? | ? | ? | ? | ? | ? |
| **vfs.file.cksum[файл,<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.contents[файл,<кодировка>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.exists[файл,<типы\_вкл>,<типы\_искл>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.get[файл]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.md5sum[файл]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.owner[файл,<тип\_владельца>,<тип\_результата>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.permissions[файл]** | | - | X | X | ? | ? | ? | ? | ? | ? | ? | ? |
| **vfs.file.regexp[файл,регулярное выражение,<кодировка>,<вывод>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.regmatch[файл,регулярное выражение,<кодировка>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.size[файл,<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **vfs.file.time[файл,<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| режим ▲ | modify (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| access | X | X | X | X | X | X | X | X | X | X | X |
| change | X5 | X | X | X | X | X | X | X | X | X | X |
| **vfs.fs.discovery** | | X | X | X | X | X | X | X | - | X | X | X |
| **vfs.fs.get** | | X | X | X | X | X | X | X | - | X | X | X |
| **vfs.fs.inode[fs,<режим>]** | | - | X | X | X | X | X | X | X | X | X | X |
| mode ▲ | total (по умолчанию) | - | X | X | X | X | X | X | X | X | X | X |
| free | - | X | X | X | X | X | X | X | X | X | X |
| used | - | X | X | X | X | X | X | X | X | X | X |
| pfree | - | X | X | X | X | X | X | X | X | X | X |
| pused | - | X | X | X | X | X | X | X | X | X | X |
| **vfs.fs.size[fs,<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| режим ▲ | total (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| free | X | X | X | X | X | X | X | X | X | X | X |
| used | X | X | X | X | X | X | X | X | X | X | X |
| pfree | X | X | X | X | X | X | X | X | X | X | X |
| pused | X | X | X | X | X | X | X | X | X | X | X |
| **vm.memory.size[<режим>]** | | X | X | X | X | X | X | X | X | X | X | X |
| режим ▲ | total (по умолчанию) | X | X | X | X | X | X | X | X | X | X | X |
| active | - | - | - | X | - | X | - | - | X | X | X |
| anon | - | - | - | - | - | - | - | - | - | - | X |
| buffers | - | X | X | X | - | - | - | - | - | X | X |
| cached | X | X | X | X | - | - | X | - | - | X | X |
| exec | - | - | - | - | - | - | - | - | - | - | X |
| file | - | - | - | - | - | - | - | - | - | - | X |
| free | X | X | X | X | X | X | X | X | X | X | X |
| inactive | - | - | - | X | - | - | - | - | X | X | X |
| pinned | - | - | - | - | - | - | X | - | - | - | - |
| shared | - | X | - | X | - | - | - | - | - | X | X |
| wired | - | - | - | X | - | - | - | - | X | X | X |
| used | X | X | X | X | X | X | X | X | X | X | X |
| pused | X | X | X | X | X | X | X | X | X | X | X |
| available | X | X | X | X | X | X | X | X | X | X | X |
| pavailable | X | X | X | X | X | X | X | X | X | X | X |
| **web.page.get[хост,<путь>,<порт>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **web.page.perf[хост,<путь>,<порт>]** | | X | X | X | X | X | X | X | X | X | X | X |
| **web.page.regexp[хост,<путь>,<порт>,регулярное выражение,<длина>,<вывод>]** | | X | X | X | X | X | X | X | X | X | X | X |

Примечания–

**1** Элементы данных net.if.in, net.if.out и net.if.total не предоставляют статистику по loopback-интерфейсам (например, lo0).

**2** Значения перечисленных элементов данных не поддерживаются для loopback-интерфейсов на Solaris вплоть до Solaris 10 6/06 включительно, поскольку статистика байт, утилизации и ошибок не хранится и/или не отдается ядром. Если мониторят Подсистема через net-snmp, значения можно получить через старый код net-snmp от cmu-snmp, датированный 1997 годом, после ошибки чтения значений байтов из статистики интерфейса возвращается счетчик пакетов (который существует на loopback-интерфейсе), умноженный на произвольное значение 308. Это дает предположение, что средний размер пакета 308 октетов, которое является очень грубым приближением, поскольку ограничение MTU на системах Solaris для loopback-интерфейсов составляет 8892 байта. Эти значения не должны рассматриваться, как корректные и даже близкие к точным. Они предположительные, но следует учесть, что net-snmp возвращает значения этих полей.

**3** Командная строка на Solaris, получаемая с /proc/pid/psinfo, ограничена 80 байтами и содержит командную строку, которая была на момент запуска процесса.

**4** Не поддерживается для Windows Event Log.

**5** На Windows XP vfs.file.time[файл,change] может быть равным vfs.file.time[файл,access].

**6** Поддерживается только Агентом-2; не поддерживается Агентом-1.

**7** Поддерживается только Агентом-2 на 64-битных Windows; не поддерживается Агентом-1.

1. Параметры vm.memory.size

В этом разделе представлены некоторые подробности относительно элемента данных vm.memory.size[<режим>] Агента.

В этом элементе данных разрешены следующие параметры:

* active – память, используемая в данный момент или бывшая в использовании совсем недавно, и поэтому еще находится в RAM;
* anon – память, не связанная с файлами (повторное чтение из них невозможно);
* available – доступная память, вычисляется по-разному в зависимости от платформы (таблица 36);
* buffers – кэш для таких вещей, как метаданные файловой системы;
* cached – кэш для различных вещей;
* exec – исполняемый код, в основном из (программ) файлов;
* file – кэш содержимого недавно использованных файлов;
* free – память, которая доступна без каких-либо проблем любому объекту, запрашивающему память;
* inactive – память, помеченная как неиспользуемая;
* pavailable – available-память в процентах по отношению к total (рассчитывается как available/total\*100);
* pinned – то же, что и wired;
* pused – used-память в процентах по отношению к total (рассчитывается как used/total\*100);
* shared – память, которая может быть доступна сразу нескольким процессам;
* slab – общий объем памяти, которая используется ядром для кэширования структур данных для собственного использования;
* total – общий объем доступной физической памяти;
* used – используемая память, вычисляется по-разному в зависимости от платформы;
* wired – память, помеченная всегда оставаться в RAM. Она не может быть перемещена на диск.

Некоторые из этих параметров работают только для конкретных платформ.

Вычисления available и used в зависимости от платформы приведены в таблице 36.

Вычисления по платформам

| Платформа | available | used |
| --- | --- | --- |
| AIX | free + cached | реальное использование памяти |
| FreeBSD | inactive + cached + free | active + wired + cached |
| HP UX | free | total – free |
| Linux<3.14 | free + buffers + cached | total – free |
| Linux 3.14+ (также перенесено для 3.10 на RHEL 7) | /proc/meminfo, см. описание "MemAvailable" в [документации](https://www.kernel.org/doc/Documentation/filesystems/proc.txt) ядра Linux. Следует обратить внимание, что "free + buffers + cached" больше не равно "available", так как не весь кэш страницы может быть освобожден и в расчетах используется минимальный объем свободной памяти, зарезервированной системой (low watermark). | total – free |
| NetBSD | inactive + execpages + file + free | total – free |
| OpenBSD | inactive + free + cached | active + wired |
| OSX | inactive + free | active + wired |
| Solaris | free | total – free |
| Win32 | free | total – free |

Примечание – Сумма vm.memory.size[used] и vm.memory.size[available] не обязательно равна общему количеству памяти.

Например, для FreeBSD:

* активная, неактивная, wired, кэшируемая памяти считаются использованными, так как содержат некоторую полезную информацию;
* неактивная, кэшируемая, свободная памяти считаются доступными, так как такая память может быть незамедлительно освобождена процессу, который запросил больше памяти.

Таким образом, неактивная память помечается и как занятая, так и как свободная одновременно. Из-за этого элемент данных vm.memory.size[used] используется исключительно в информационных целях, а элемент данных vm.memory.size[available] предназначен для использования в триггерах.

1. Пассивные и активные проверки Агентов

Этот раздел подробно описывает пассивные и активные проверки, которые выполняются Агентом.

Для взаимодействия с Агентом Подсистема использует протокол на основе JSON.

1. Пассивные проверки

Пассивная проверка – это простой запрос данных. Сервер или Прокси запрашивает какие-либо данные (например, загрузку CPU), и Агент отправляет результат обратно Серверу.

За определением длины заголовка и данных следует обратиться к подробностям протокола (см. п.Б.1).

**Запрос Сервера:**

<ключ элемента данных>

**Ответ Агента:**

<ДАННЫЕ>[\0<ОШИБКА>]

где часть в квадратных скобках опциональна и отправляется только по неподдерживаемым элементам данных.

Например, для поддерживаемых элементов данных:

1. сервер открывает TCP-соединение;
2. сервер отправляет <ЗАГОЛОВОК><ДЛИНАДАННЫХ>agent.ping;
3. Агент читает запрос и отвечает <ЗАГОЛОВОК><ДЛИНАДАННЫХ>1;
4. сервер обрабатывает данные, чтобы извлечь значение "1";
5. TCP-соединение закрывается.

Для неподдерживаемых элементов данных:

1. сервер открывает TCP-соединение;
2. сервер отправляет <ЗАГОЛОВОК><ДЛИНАДАННЫХ>vfs.fs.size[/nono];
3. Агент читает запрос и отвечает <ЗАГОЛОВОК><ДЛИНАДАННЫХ>ZBX\_NOTSUPPORTED\0Cannot obtain filesystem information: [2] No such file or directory;
4. сервер обрабатывает данные, меняет состояние элемента данных на неподдерживаемое с приведенным сообщением об ошибке;
5. TCP-соединение закрывается.
6. Активные проверки

Активные проверки требуют более сложной обработки. Агент сначала должен получить с Сервера (или Серверов) список элементов данных для независимой обработки.

Серверы для получения активных проверок перечислены в параметре ServerActive файла конфигурации Агента. Частота запросов данных проверок настраивается параметром RefreshActiveChecks в этом же файле конфигурации. Если обновление активных проверок завершится с ошибкой, запрос повторится через 60 секунд (задано в коде).

Затем Агент периодически отправляет новые значения на Сервер(ы).

Если Агент находится за брандмауэром, можно рассмотреть возможность использования исключительно активных проверок, т.к. в этом случае не понадобится менять настройки брандмауэра для разрешения начальных входящих соединений.

Г.3.3 Получение списка активных проверок

Запрос на активные проверки используется для получения списка активных проверок, обрабатываемых Агентом. Этот запрос отсылается Агентом после старта и затем с интервалом RefreshActiveChecks (таблица 37).

Поля запроса активных проверок

| Поле | Тип | Обязательный | Значение |
| --- | --- | --- | --- |
| request | строка | да | active checks |
| host | строка | да | Имя узла сети. |
| host\_metadata | строка | нет | Значение метрики параметра конфигурации HostMetadata или HostMetadataItem. |
| HostInterface | строка | нет | Значение метрики параметра конфигурации HostInterface или HostInterfaceItem. |
| ip | строка | нет | Первый IP параметра конфигурации LISTENIP, если задан. |
| port | число | нет | Значение параметра конфигурации LISTENPort, если задано и не является портом, прослушиваемым Агентом по умолчанию. |

Ответ активных проверок посылается Сервером обратно Агенту после обработки запроса на активные проверки (таблица 38).

Поля ответа активных проверок

| Поле | | Тип | Обязательный | Значение |
| --- | --- | --- | --- | --- |
| response | | строка | да | success | failed |
| info | | строка | нет | Сообщение об ошибке в случае сбоя. |
| data | | массив объектов | нет | Элементы данных для активных проверок. |
|  | key | строка | нет | Ключ элемента данных с раскрытыми макросами. |
| key\_orig | строка | нет | Ключ элемента данных с нераскрытыми макросами. |
| itemid | число | нет | Идентификатор элемента данных. |
| delay | строка | нет | Интервал обновления элемента данных. |
| lastlogsize | число | нет | Последний размер файла журнала (lastlogsize) элемента данных. |
| mtime | число | нет | Отметка времени последнего обновления (mtime) элемента данных. |
| refresh\_unsupported | | число | нет | Интервал обновления неподдерживаемого элемента данных. |
| regexp | | массив объектов | нет | Глобальные регулярные выражения. |
|  | name | строка | нет | Имя глобального регулярного выражения. |
| expression | строка | нет | Глобальное регулярное выражение. |
| expression\_type | число | нет | Тип глобального регулярного выражения. |
| exp\_delimiter | строка | нет | Разделитель глобального регулярного выражения. |
| case\_sensitive | число | нет | Признак чувствительности к регистру глобального регулярного выражения. |

Сервер должен ответить успешно, например:

1. Агент открывает TCP-соединение;
2. Агент запрашивает список проверок;
3. Сервер отвечает списком элементов данных (ключ элемента данных, интервал обновления);
4. Агент анализирует ответ;
5. TCP-соединение закрывается;
6. Агент начинает периодический сбор данных.

Примечание – Следует обратить внимание, что при использовании активных проверок (чувствительные) данные конфигурации могут стать доступными лицам, имеющим доступ к порту траппера Сервера. Это возможно, так как любой пользователь может представиться активным Агентом и запросить данные конфигурации элементов данных; аутентификация не производится, если только не используются опции шифрования.

Г.3.4 Отправка собранных данных

Запрос Агента с данными содержит собранные значения элементов данных (таблица 39).

Поля собранных данных

| Поле | | Тип | Обязательный | Значение |
| --- | --- | --- | --- | --- |
| request | | строка | да | "agent data" |
| session | | строка | да | Уникальный идентификатор сессии, генерируемый каждый раз при старте Агента. |
| data | | массив объектов | да | Значения элементов данных. |
|  | id | число | да | Идентификатор значения (возрастающий счетчик, используемый для проверки дубликатов значений в случае сетевых проблем). |
| host | строка | да | Имя узла сети. |
| key | строка | да | Ключ элемента данных. |
| value | строка | нет | Значение элемента данных. |
| lastlogsize | число | нет | Последний размер журнала (lastlogsize) элемента данных. |
| mtime | число | нет | Отметка времени последнего обновления (mtime) элемента данных. |
| state | число | нет | Состояние элемента данных. |
| source | строка | нет | Источник (source) значения журнала событий. |
| eventid | число | нет | Идентификатор (eventid) значения журнала событий. |
| severity | число | нет | Важность (severity) значения журнала событий. |
| timestamp | число | нет | Отметка времени (timestamp) значения журнала событий. |
| clock | число | да | Отметка времени значения (секунд с начала эпохи). |
| ns | число | да | Наносекунды отметки времени значения. |

Каждому значению назначается виртуальный ID. Значение ID – простой возрастающий счетчик, уникальный в пределах одной сессии данных (идентифицируемой токеном сессии). Этот ID используется, чтобы отбрасывать дубликаты значений, которые могли быть отосланы в средах с плохой связью.

Ответ на данные Агента отсылается Сервером назад Агенту после обработки запроса Агента с данными (таблица 40).

Поля ответа на данные

| Поле | Тип | Обязательный | Значение |
| --- | --- | --- | --- |
| response | строка | да | success | failed |
| info | строка | да | Результаты обработки элементов данных. |

Если отправка некоторых значений завершилась неудачей на Сервере (например, по причине того, что узел сети или элемент данных был деактивирован или удален), Агент не будет повторять отправку этих элементов данных, например:

1. Агент открывает TCP-соединение;
2. Агент отправляет список значений;
3. сервер обрабатывает данные и отправляет обратно статус;
4. TCP-соединение закрывается.

Следует обратить внимание на то, как в примере выше указывается неподдерживаемое состояние для vfs.fs.size[/nono] при помощи значения state, равного "1", и сообщения об ошибке в свойстве value.

Сообщение об ошибке будет обрезано до 2048 символов на стороне Сервера.

1. Минимальный уровень прав для элементов данных Агента Windows

При использовании Подсистемы можно воспользоваться получением метрик с хоста, на котором установлен Агент. Чтобы использовать принцип минимальных привилегий, необходимо определить метрики, которые собираются Агентом.

Таблицы 41-42 позволят выбрать минимальные права, которые будут гарантировать корректную работу Агента.

Если для работы Агента выбран пользователь, отличный от LocalSystem, то для работы Агента в качестве службы Windows новый пользователь должен иметь права "Вход в качестве службы**"** из меню панели навигации "Групповая политика → Назначение прав пользователя**"** и право создавать, записывать и удалять файл журнала Агента. Пользователь Active Directory должен быть добавлен в группу "Performance Monitor Users**"**.

При работе с правами Агента, основываясь на группе "Минимально технически приемлемо**"**, требуется предварительное предоставление прав на объекты мониторинга.

Основные элементы данных Агента, поддерживаемые в Windows

| Ключ элемента данных | Группа пользователей | |
| --- | --- | --- |
|  | **Рекомендуется** | **Минимально технически приемлемо (ограниченные возможности)** |
| agent.Hostname | Гости | Гости |
| agent.ping | Гости | Гости |
| agent.variant | Гости | Гости |
| agent.version | Гости | Гости |
| log | Администраторы | Гости |
| log.count | Администраторы | Гости |
| logrt | Администраторы | Гости |
| logrt.count | Администраторы | Гости |
| net.dns | Гости | Гости |
| net.dns.record | Гости | Гости |
| net.if.discovery | Гости | Гости |
| net.if.in | Гости | Гости |
| net.if.out | Гости | Гости |
| net.if.total | Гости | Гости |
| net.tcp.LISTEN | Гости | Гости |
| net.tcp.port | Гости | Гости |
| net.tcp.service | Гости | Гости |
| net.tcp.service.perf | Гости | Гости |
| net.udp.service | Гости | Гости |
| net.udp.service.perf | Гости | Гости |
| proc.num | Администраторы | Гости |
| system.cpu.discovery | Пользователи монитора производительности | Пользователи монитора производительности |
| system.cpu.load | Пользователи монитора производительности | Пользователи монитора производительности |
| system.cpu.num | Гости | Гости |
| system.cpu.util | Пользователи монитора производительности | Пользователи монитора производительности |
| system.Hostname | Гости | Гости |
| system.localtime | Гости | Гости |
| system.run | Администраторы | Гости |
| system.sw.arch | Гости | Гости |
| system.swap.size | Гости | Гости |
| system.uname | Гости | Гости |
| system.uptime | Пользователи монитора производительности | Пользователи монитора производительности |
| vfs.dir.count | Администраторы | Гости |
| vfs.dir.get | Администраторы | Гости |
| vfs.dir.size | Администраторы | Гости |
| vfs.file.cksum | Администраторы | Гости |
| vfs.file.contents | Администраторы | Гости |
| vfs.file.exists | Администраторы | Гости |
| vfs.file.md5sum | Администраторы | Гости |
| vfs.file.regexp | Администраторы | Гости |
| vfs.file.regmatch | Администраторы | Гости |
| vfs.file.size | Администраторы | Гости |
| vfs.file.time | Администраторы | Гости |
| vfs.fs.discovery | Администраторы | Гости |
| vfs.fs.size | Администраторы | Гости |
| vm.memory.size | Гости | Гости |
| web.page.get | Гости | Гости |
| web.page.perf | Гости | Гости |
| web.page.regexp | Гости | Гости |
| zabbix.stats | Гости | Гости |

Ключи элементов данных, специфичные для Windows

| Ключ элемента данных | Группа пользователей | |
| --- | --- | --- |
|  | **Рекомендуется** | **Минимально технически приемлемо (ограниченные возможности)** |
| eventlog | Читатели журнала событий | Гости |
| net.if.list | Гости | Гости |
| perf\_counter | Пользователи монитора производительности | Пользователи монитора производительности |
| proc\_info | Администраторы | Гости |
| service.discovery | Гости | Гости |
| service.info | Гости | Гости |
| services | Гости | Гости |
| wmi.get | Администраторы | Гости |
| vm.vmemory.size | Гости | Гости |

1. Кодировка получаемых значений

Сервер ожидает, что каждое получаемое текстовое значение будет в кодировке UTF-8. Это относится к любому типу проверок: Агент, ssh, telnet и т.д.

Различные системы/устройства и проверки могут возвращать в значениях non-ASCII символы. Для таких случаев почти все доступные в Подсистеме ключи имеют дополнительный параметр ключа элемента данных <кодировка>. Этот параметр ключа необязателен, но он должен быть указан, если получаемое значение не в кодировке UTF-8 и содержит non-ASCII символы. Иначе результат может быть неожиданным и непредсказуемым.

Описание поведения различных БД в этих случаях:

* MySQL – если значение содержит non-ASCII символ в не UTF-8-кодировке, то этот символ и следующий за ним символ будут отброшены при записи этого значения базой данных. Никакие предупреждающие сообщения не записываются в zabbix\_server.log;
* PostgreSQL – если значение содержит non-ASCII символ в не UTF-8-кодировке, то это приведет к ошибке в SQL-запросе (PGRES\_FATAL\_ERROR:ERROR invalid Byte sequence for encoding), и данные не будут записаны. Соответствующее сообщение с предупреждением будет записано в zabbix\_server.log.

1. Поддержка больших файлов

Поддержка больших файлов (Large file support, сокращенно – LFS) – это термин, применяемый к возможности работать с файлами больше чем 2ГБ на 32-битных операционных системах. Это на мониторинг файлов журналов и на все элементы данных vfs.file.\*. Поддержка больших файлов зависит от возможностей системы во время компиляции Подсистемы, но поддержка полностью отключена для 32-битных систем Solaris по причине несовместимости с procfs и swapctl.

1. Датчики

Каждый контроллер имеет свою собственную директорию в дереве sysfs /sys/devices. Чтобы найти все контроллеры, нужно перейти по символическим ссылкам на устройства из /sys/class/hwmon/hwmon\*, где \* – это конкретный номер (0,1, 2, ...).

Показания датчиков находятся либо в директории /sys/class/hwmon/hwmon\*/ для виртуальных устройств, либо в директории /sys/class/hwmon/hwmon\*/device для невиртуальных устройств. Файл с именем name, расположенный в директориях hwmon\* или hwmon\*/device, содержит имя чипа, которое соответствует имени используемого контроллером драйвера ядра.

В каждом файле имеется только одно значение показания датчика. Общая схема именования файлов, которые содержат показания датчиков, внутри любой из вышеупомянутых папок <тип><номер>\_<элемент>, где:

* тип – для чипов датчиков это "in" (напряжение), "temp" (температура), "fan" (скорость вентилятора) и т.д.;
* элемент – "input" (измеренное значение), "max" (верхний порог), "min" (нижний порог) и т.д.;
* номер – всегда используется для экземпляров, которые могут присутствовать более одного раза (обычно начинается с 1, за исключением датчиков напряжения, которые начинаются с 0). Если файлы не относятся к конкретным экземплярам, они имеют простое наименование без номера.

Информацию о датчиках, доступных на хосте, можно получить с помощью инструментов sensor-detect и sensors (пакет lm-sensors). Sensors-detect помогает определить, какие модули необходимы для доступных датчиков. Когда модули загружены, можно использовать утилиту sensors, чтобы показать значения со всех контроллеров. Маркировка значений датчиков, используемая данной утилитой, может отличаться от общепринятой схемы именования (<тип><номер>\_<элемент>):

* если имеется файл с именем <тип><номер>\_метка, то метка внутри этого файла будет использоваться вместо имени <тип><номер>\_<элемент>;
* если файл <тип><номер>\_метка отсутствует, то утилита выполняет поиск внутри /etc/sensors.conf (или /etc/sensors3.conf, или другой) для подстановки имени.

Эта маркировка позволяет пользователю определить тип используемого оборудования. Если нет ни файла <тип><номер>\_метка, ни метки внутри файла конфигурации, то тип оборудования может быть определен по параметру name (hwmon\*/device/name). Реальные имена датчиков, которые воспринимает zabbix\_agent, можно получить, запустив утилиту sensors с параметром "-u" (sensors -u).

В утилите sensor доступные датчики разделены по типу шины (ISA adapter, PCI adapter, SPI adapter, Virtual device, ACPI interface, HID adapter).

В Linux 2.4 показания датчиков извлекаются из директории /proc/sys/dev/sensors:

* устройство – имя устройства (если используется <режим>, это регулярное выражение);
* сенсор – имя датчика (если используется <режим>, это регулярное выражение);
* режим – возможные значения: avg, max, min (если этот параметр не указан, устройство и сенсор обрабатываются буквально).

Пример ключа:

sensor[w83781d-i2c-0-2d,temp1]

В Linux 2.6 и более поздних показания датчиков извлекаются из директории /sys/class/hwmon:

* device – имя устройства (не регулярное выражение). Имя устройства может быть фактическим именем устройства (например, 0000:00:18.3) или именем, полученным при использовании утилиты sensors (например, k8temp-pci-00c3). Выбор того, какое из этих имен использовать, оставлен за пользователем;
* sensor – имя датчика (не регулярное выражение);
* mode – возможные значения: avg, max, min (если этот параметр не указан, устройство и сенсор обрабатываются буквально).

Примеры ключей:

sensor[k8temp-pci-00c3,temp,max] или sensor[0000:00:18.3,temp1]

sensor[smsc47b397-isa-0880,in,avg] или sensor[smsc47b397.2176,in1]

При получении имен датчиков метки датчиков не всегда могут быть использованы непосредственно в том виде, в каком они выводятся утилитой sensors, т.к. именование меток может отличаться у каждого производителя контроллеров. Например, вывод утилиты sensors может содержать следующие строки:

$ sensors

in0: +2.24 V (min = +0.00 V, max = +3.32 V)

Vcore: +1.15 V (min = +0.00 V, max = +2.99 V)

+3.3V: +3.30 V (min = +2.97 V, max = +3.63 V)

+12V: +13.00 V (min = +0.00 V, max = +15.94 V)

M/B Temp: +30.0°C (low = -127.0°C, high = +127.0°C)

Из этих строк только одна строка может использоваться непосредственно:

$ zabbix\_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]

2.240000

Попытки использовать остальные метки, такие как Vcore или +12V, работать не будут.

$ zabbix\_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]

ZBX\_NOTSUPPORTED

Чтобы найти фактическое имя датчика, которое можно использовать в Подсистеме для получения его показаний, запускают "sensors -u". В полученном выводе можно увидеть следующее:

$ sensors -u

...

Vcore:

in1\_input: 1.15

in1\_min: 0.00

in1\_max: 2.99

in1\_alarm: 0.00

...

+12V:

in4\_input: 13.00

in4\_min: 0.00

in4\_max: 15.94

in4\_alarm: 0.00

...

Таким образом, Vcore может быть опрошен как in1, а +12V должен опрашиваться как in4.1:

$ zabbix\_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]

1.301000

В Подсистеме можно получать не только показания напряжения (in), но также тока (curr), температуры (temp) и скорости вращения вентилятора (fan).

1. Параметр "тип памяти" в элементах данных proc.mem

Параметр "тип памяти**"** поддерживается на платформах Linux, AIX, FreeBSD и Solaris.

Три общих значения параметра "тип памяти" поддерживаются на всех этих платформах: pmem, rss и vsize. Кроме того, для некоторых платформ поддерживаются специфичные для этих платформ значения "типа памяти".

Для AIX значения, поддерживаемые для параметра "тип памяти", приведены в таблице 43.

Значения для AIX

| Поддерживаемое значение | Описание | Источник в структуре procentry64 | Пытается быть совместимым с |
| --- | --- | --- | --- |
| vsize | Размер виртуальной памяти | pi\_size |  |
| pmem | Процент физической памяти | pi\_prm | ps -o pmem |
| rss | Размер резидентного набора | pi\_trss + pi\_drss | ps -o rssize |
| size | Размер процесса (код + данные) | pi\_dvm | "ps gvw" колонка SIZE |
| dsize | Размер данных | pi\_dsize | |
| tsize | Размер текста (кода) | pi\_tsize | "ps gvw" колонка TSIZ |
| sdsize | Размер данных из разделяемой библиотеки | pi\_sdsize | |
| drss | Размер резидентного набора данных | pi\_drss |  |
| trss | Размер резидентного набора текста | pi\_trss |  |

Примечания:

* При выборе параметров для ключа элемента данных proc.mem[] на AIX следует указать узкие критерии выбора процесса. Иначе есть риск того, что в результат proc.mem[] окажутся сосчитаны нежелательные процессы.

Пример:

$ zabbix\_agentd -t proc.mem[,,,НесуществующийПроцесс,rss]

proc.mem[,,,НесуществующийПроцесс,rss] [u|2879488]

Данный пример иллюстрирует, как указание только параметра командной строки (в виде регулярного выражения) приводит к тому, что Агент считает и себя тоже.

* Не следует использовать "ps -ef" для просмотра процессов, показывающий только процессы, не относящиеся к ядру. Используют "ps -Af", чтобы увидеть все процессы, которые будет видеть Агент.
* proc.mem[] Агента выбирает процессы на примере использования topasrec:

$ ps -Af | grep topasrec

root 1074798410Mar 16-0:00 /usr/bin/topasrec-L -s 300 -R 1 -r 6 -o /var/perf daily/ -ypersistent=1 -O type=bin -yStart\_time=04:08:54,Mar16,2023

proc.mem[] имеет аргументы:

proc.mem[<имя>,<пользователь>,<режим>,<cmdline>,<тип памяти>]

Первый критерий – имя процесса (аргумент <имя>). В этом примере Агент увидит его как topasrec. Для соответствия нужно либо указать topasrec, либо оставить его пустым. Второй критерий – имя пользователя (аргумент <пользователь>). Для соответствия нужно либо указать root, либо оставить его пустым. Третий критерий, используемый при выборе процесса, – это аргумент <cmdline>. Агент увидит его значение как "/usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -yStart\_time=04:08:54,Mar16,2023". Для сопоставления нужно либо указать регулярное выражение, соответствующее этой строке, либо оставить его пустым.

Аргументы <режим> и <тип памяти> применяются после использования трех критериев, упомянутых выше.

Для FreeBSD значения, поддерживаемые для параметра "тип памяти", приведены в таблице 44.

Значения для FreeBSD

| Поддерживаемое значение | Описание | Источник в структуре kinfo\_proc | Пытается быть совместимым с |
| --- | --- | --- | --- |
| vsize | Размер виртуальной памяти | kp\_eproc.e\_vm.vm\_map.size или ki\_size | ps -o vsz |
| pmem | Процент физической памяти | вычисляется из rss | ps -o pmem |
| rss | Размер резидентного набора | kp\_eproc.e\_vm.vm\_rssize или ki\_rssize | ps -o rss |
| size | Размер процесса (код + данные + стек) | tsize + dsize + ssize |  |
| tsize | Размер текста (кода) | kp\_eproc.e\_vm.vm\_tsize или ki\_tsize | ps -o tsiz |
| dsize | Размер данных | kp\_eproc.e\_vm.vm\_dsize или ki\_dsize | ps -o dsiz |
| ssize | Размер стека | kp\_eproc.e\_vm.vm\_ssize или ki\_ssize | ps -o ssiz |

Для Linux значения, поддерживаемые для параметра "тип памяти", приведены в таблице 45.

Значения для FreeBSD

| Поддерживаемое значение | Описание | Источник в файле /proc/<pid>/status |
| --- | --- | --- |
| vsize | Размер виртуальной памяти | VmSize |
| pmem | Процент физической памяти | (VmRSS/total\_memory) \* 100 |
| rss | Размер резидентного набора | VmRSS |
| data | Размер сегмента данных | VmData |
| exe | Размер сегмента кода | VmExe |
| hwm | Пиковый размер резидентного набора | VmHWM |
| lck | Размер заблокированной памяти | VmLck |
| lib | Размер разделяемых библиотек | VmLib |
| peak | Пиковый размер виртуальной памяти | VmPeak |
| pin | Размер закрепленных страниц | VmPin |
| pte | Размер записей таблицы страниц | VmPTE |
| size | Размер сегментов кода + данных + стека процесса | VmExe + VmData + VmStk |
| stk | Размер сегмента стека | VmStk |
| swap | Размер используемого места в разделе подкачки | VmSwap |

Примечания для Linux:

* Не все значения параметра "тип памяти" поддерживаются старыми версиями ядер Linux. Например, ядра Linux 2.4 не поддерживают значения hwm, pin, peak, pte и swap.
* Процесс самодиагностики активных проверок Агента с proc.mem[...,...,...,...,data] отображает значение, которое на 4 КБ больше, чем отдает строка VmData в файле Агента /proc/<pid>/status. Во время измерения себя сегмент данных Агента увеличивается на 4 КБ и затем возвращается к предыдущему размеру.

Для Solaris значения, поддерживаемые для параметра "тип памяти", приведены в таблице 46.

Значения для Solaris

| Поддерживаемое значение | Описание | Источник в структуре psinfo | Пытается быть совместимым с |
| --- | --- | --- | --- |
| vsize (по умолчанию) | Размер образа процесса | pr\_size | ps -o vsz |
| pmem | Процент физической памяти | pr\_pctmem | ps -o pmem |
| rss | Размер резидентного набора Он может быть недооценен – см. описание rss в man ps. | pr\_rssize | ps -o rss |

1. Выбор процессов в элементах данных proc.mem и proc.num
2. **Процессы, изменяющие свою командную строку**

Некоторые программы используют изменение своей командной строки как метод отображения своей текущей активности. Пользователь может увидеть эту активность, запуская команды ps и top. Примеры таких программ: PostgreSQL, Sendmail, Zabbix.

Пример из Linux. Наблюдение количество процессов Агента.

Команда ps отобразит интересующие процессы следующим образом:

$ ps -fu Zabbix

UID PID PPID C STIME TTY TIME CMD

...

zabbix 6318 1 0 12:01 ? 00:00:00 sbin/zabbix\_agentd -c /home/zabbix/ZBXNEXT-1078/zabbix\_agentd.conf

zabbix 6319 6318 0 12:01 ? 00:00:01 sbin/zabbix\_agentd: collector [idle 1 sec]

zabbix 6320 6318 0 12:01 ? 00:00:00 sbin/zabbix\_agentd: LISTENer #1 [waiting for connection]

zabbix 6321 6318 0 12:01 ? 00:00:00 sbin/zabbix\_agentd: LISTENer #2 [waiting for connection]

zabbix 6322 6318 0 12:01 ? 00:00:00 sbin/zabbix\_agentd: LISTENer #3 [waiting for connection]

zabbix 6323 6318 0 12:01 ? 00:00:00 sbin/zabbix\_agentd: active checks #1 [idle 1 sec] ...

Выбор процессов по имени и пользователю выполняет задачу:

$ zabbix\_get -s localhost -k "proc.num[zabbix\_agentd,zabbix]"6

После переименования исполняемого файла zabbix\_agentd в zabbix\_agentd\_30 команда перезапускается.

Здесь ps отображает:

$ ps -fu Zabbix

UID PID PPID C STIME TTY TIME CMD

...

zabbix 6715 1 0 12:53 ? 00:00:00 sbin/zabbix\_agentd\_30 -c /home/zabbix/ZBXNEXT-1078/zabbix\_agentd.conf

zabbix 6716 6715 0 12:53 ? 00:00:00 sbin/zabbix\_agentd\_30: collector [idle 1 sec]

zabbix 6717 6715 0 12:53 ? 00:00:00 sbin/zabbix\_agentd\_30: LISTENer #1 [waiting for connection]

zabbix 6718 6715 0 12:53 ? 00:00:00 sbin/zabbix\_agentd\_30: LISTENer #2 [waiting for connection]

zabbix 6719 6715 0 12:53 ? 00:00:00 sbin/zabbix\_agentd\_30: LISTENer #3 [waiting for connection]

zabbix 6720 6715 0 12:53 ? 00:00:00 sbin/zabbix\_agentd\_30: active checks #1 [idle 1 sec] ...

Здесь выбор процессов по имени и пользователю выдает неверный результат:

$ zabbix\_get -s localhost -k "proc.num[zabbix\_agentd\_30,zabbix]"1

В данном случае простое переименование исполняемого файла на более длинное имя приводит к совершенно другому результату.

Агент начинает с проверки имени процесса. Открывается файл /proc/<pid>/status и проверяется строка Name. В этом случае строки Name это:

$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status

/proc/6715/status:Name: zabbix\_agentd\_3

/proc/6716/status:Name: zabbix\_agentd\_3

/proc/6717/status:Name: zabbix\_agentd\_3

/proc/6718/status:Name: zabbix\_agentd\_3

/proc/6719/status:Name: zabbix\_agentd\_3

/proc/6720/status:Name: zabbix\_agentd\_3

Имя процесса в файле status обрезается до 15 символов.

Аналогичный результат можно увидеть при помощи команды ps:

$ ps -u zabbix PID TTY TIME CMD

...

6715 ? 00:00:00 zabbix\_agentd\_3

6716 ? 00:00:01 zabbix\_agentd\_3

6717 ? 00:00:00 zabbix\_agentd\_3

6718 ? 00:00:00 zabbix\_agentd\_3

6719 ? 00:00:00 zabbix\_agentd\_3

6720 ? 00:00:00 zabbix\_agentd\_3

...

Очевидно, что этот вывод не идентичен значению zabbix\_agentd\_30 в параметре name ключа proc.num[]. Не найдя совпадение по имени процесса в файле status, Агент обращается к файлу /proc/<pid>/cmdline.

То, как Агент просматривает файл cmdline, может быть проиллюстрировано при помощи выполнения команды:

$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk "{gsub(/\x0/,"<NUL>"); print};"; done

sbin/zabbix\_agentd\_30<NUL>-c<NUL>/home/zabbix/ZBXNEXT-1078/zabbix\_agentd.conf<NUL>

sbin/zabbix\_agentd\_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL>...

sbin/zabbix\_agentd\_30: LISTENer #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL>...

sbin/zabbix\_agentd\_30: LISTENer #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL>...

sbin/zabbix\_agentd\_30: LISTENer #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL>...

sbin/zabbix\_agentd\_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL>...

В рассматриваемом случае файлы /proc/<pid>/cmdline содержат невидимые непечатаемые нулевые байты, которые используются для завершения строки в языке C. В этом примере нулевые байты отображаются как <NUL>.

Агент проверяет cmdline основного процесса и извлекает zabbix\_agentd\_30, которое соответствует значению zabbix\_agentd\_30 параметра name. Таким образом, основной процесс учитывается элементом данных proc.num[zabbix\_agentd\_30,zabbix].

При проверке следующего процесса Агент из файла cmdline получает zabbix\_agentd\_30: collector [idle 1 sec], и это не соответствует значению zabbix\_agentd\_30 в параметре name. Таким образом, будет сосчитан только основной процесс, который не меняет свою командную строку. Остальные процессы Агента модифицируют свои командные строки и игнорируются.

Этот пример показывает, что в данном случае параметр name нельзя использовать в proc.mem[] и proc.num[] для выбора процессов.

Использование параметра cmdline с надлежащим регулярным выражением даст правильный результат:

$ zabbix\_get -s localhost -k "proc.num[,zabbix,,zabbix\_agentd\_30[ :]]"6

Следует с осторожностью использовать элементы данных proc.mem[] и proc.num[] при наблюдении за программами, которые модифицируют свои командные строки.

Перед тем как поместить параметры name и cmdline в элементы данных proc.mem[] и proc.num[], возможно, потребуется протестировать эти параметры, используя элемент данных proc.num[] и команду ps.

1. **Потоки ядра Linux**

Нельзя выбрать потоки при помощи параметров cmdline в элементах данных proc.mem[] и proc.num[].

Например, один из потоков ядра:

$ ps -ef| grep kthreadd

root 2 0 0 09:33 ? 00:00:00 [kthreadd]

Его можно выбрать при помощи параметра имя:

$ zabbix\_get -s localhost -k "proc.num[kthreadd,root]"

1

Но выбор при помощи параметра cmdline не работает:

$ zabbix\_get -s localhost -k "proc.num[,root,,kthreadd]"

0

Причина такого поведения кроется в том, что Агент берет регулярное выражение, которое указано в параметре cmdline, и применяет его к содержимому относящегося к процессу файла /proc/<pid>/cmdline. В случае потоков ядра их файлы /proc/<pid>/cmdline – пустые. Поэтому параметр cmdline никогда не совпадет.

Потоки ядра Linux подсчитываются элементом данных proc.num[], но не сообщают информацию о памяти в элементе данных proc.mem[] item, например:

$ ps -ef | grep kthreadd

root 2 0 0 09:51 ? 00:00:00 [kthreadd]

$ zabbix\_get -s localhost -k "proc.num[kthreadd]"

1

$ zabbix\_get -s localhost -k "proc.mem[kthreadd]"

ZBX\_NOTSUPPORTED: Cannot get amount of "VmSize" memory.

Если имеется пользовательский процесс с таким же именем, как и у потока ядра, то:

$ ps -ef | grep kthreadd

root 2 0 0 09:51 ? 00:00:00 [kthreadd]

zabbix 9611 6133 0 17:58 pts/1 00:00:00 ./kthreadd

$ zabbix\_get -s localhost -k "proc.num[kthreadd]"

2

$ zabbix\_get -s localhost -k "proc.mem[kthreadd]"

4157440

proc.num[] подсчитал как поток ядра, так и пользовательский процесс.

proc.mem[] сообщил информацию о памяти только по пользовательскому процессу и вычислил память потока ядра, как будто оно было нулевым. Это отличается от случая выше, когда сообщалось ZBX\_NOTSUPPORTED.

Следует внимательно использовать элементы данных proc.mem[] и proc.num[] при совпадении имени процесса с одним из потоков.

Перед добавлением параметров в элементы данных proc.mem[] и proc.num[], возможно, потребуется протестировать эти параметры, используя элемент данных proc.num[] и команду ps.

1. Реализация проверок net.tcp.service и net.udp.service

В этом разделе подробно описана реализация проверок net.tcp.service и net.udp.service для различных сервисов, которые можно указать в параметре service.

Параметры элемента данных net.tcp.service:

* ftp – создает TCP-соединение и ожидает, что первые 4 символа ответа будут "220 ", после чего посылает "QUIT\r\n". По умолчанию используется порт 21, если он не указан;
* http – создает TCP-соединение без ожидания и отсылки чего-либо. По умолчанию используется порт 80, если он не указан;
* https – использует (и работает только с) libcurl, не проверяет подлинность сертификата, не проверяет имя хоста в SSL-сертификате, только извлекает заголовок ответа (запрос HEAD). По умолчанию используется порт 443, если он не указан;
* imap – создает TCP-соединение и ожидает, что первые 4 символа ответа будут "\* OK", после чего посылает "a1 LOGOUT\r\n". По умолчанию используется порт 143, если он не указан;
* ldap – открывает соединение на сервер LDAP и выполняет операцию поиска (LDAP search) с фильтром, выставленным в (objectClass=\*). Ожидает успешного извлечения первого параметра у первого элемента. По умолчанию используется порт 389, если он не указан;
* nntp – создает TCP-соединение и ожидает, что первые 3 символа ответа будут "200" или "201", после чего посылает "QUIT\r\n". По умолчанию используется порт 119, если он не указан;
* pop – создает TCP-соединение и ожидает, что первые 3 символа ответа будут "+OK", после чего посылает "QUIT\r\n". По умолчанию используется порт 110, если он не указан;
* smtp – создает TCP-соединение и ожидает, что первые 3 символа ответа будут "220", после которых идет пробел, конец строки или минус. Строки, содержащие минус, относятся к многострочному ответу, и ответ будет заново читаться до тех пор, пока не будет получена строка без минуса. После этого отсылается "QUIT\r\n". По умолчанию используется порт 25, если он не указан;
* ssh – создает TCP-соединение. Если соединение было установлено, обе стороны обмениваются строками идентификации (SSH-major.minor-XXXX), где major и minor являются версиями протокола, а XXXX – строкой. Подсистема проверяет, найдена ли соответствующая спецификации строка, и затем отсылает назад строку "SSH-major.minor-zabbix\_agent\r\n" или "0\n" при несовпадении. По умолчанию используется порт 22, если он не указан;
* tcp – создает TCP-соединение без ожидания и отсылки чего-либо. В отличие от остальных проверок, требует указания параметра "порт";
* telnet – создает TCP-соединение и ожидает приглашение на ввод имени пользователя (":" в конце строки). По умолчанию используется порт 23, если он не указан.

Параметры элемента данных net.udp.service:

* ntp – отсылает по UDP пакет SNTP и проверяет ответ в соответствии с RFC 4330. По умолчанию используется порт 123, если он не указан.

1. Настройки недостижимости/недоступности интерфейса узла сети

Несколько параметров конфигурации определяют, каким должно быть поведение Сервера, если агентские проверки (Подсистемы, SNMP, IPMI, JMX) завершаются с ошибками и узел сети становится недостижимым.

1. Недостижимый интерфейс

Интерфейс узла сети считается недостижимым после ошибки при проверке (сетевая ошибка, превышение времени ожидания) посредством Подсистемы, SNMP, IPMI или JMX Агентов. Следует учесть, что активные проверки Агента никак не влияют на доступность интерфейса.

С этого момента UnreachableDelay определяет, как часто в этой ситуации недостижимости интерфейс будет проверяться повторно, используя один из элементов данных (включая правила LLD), причем такие повторные проверки выполняются уже с помощью unreachable-поллеров (или IPMI-поллеров для IPMI-проверок). По умолчанию – 15 секунд до следующей проверки.

В журнале Сервера недостижимость обозначается записями такого рода:

Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for 15 seconds

Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait for 15 seconds

Следует обратить внимание, что здесь указывается, какой именно элемент данных выполнился с ошибкой, а также тип этого элемента данных (Агент).

Параметр Timeout также будет влиять на то, как быстро интерфейс будет проверен повторно в течение периода недостижимости. Если время ожидания составляет 20 секунд и UnreachableDelay составляет 30 секунд, то следующая проверка будет через 50 секунд после первой попытки.

Параметр UnreachablePeriod определяет общую длительность периода недостижимости. По умолчанию UnreachablePeriod равен 45 секундам. UnreachablePeriod должен быть в несколько раз больше, чем UnreachableDelay, чтобы интерфейс проверялся повторно более одного раза, прежде чем он станет недоступным.

По окончании периода недостижимости интерфейс опрашивается заново со снижением приоритета элемента данных, который перевел интерфейс в недостижимое состояние. Если устанавливается связь с интерфейсом, который был недостижимым, мониторинг автоматически возвращается к норме:

resuming Zabbix agent checks on host "New host": connection restored

После того как интерфейс становится доступным, опрос всех элементов данных немедленно не осуществляется узлом сети по двум причинам:

* это может создать слишком большую нагрузку на узел сети;
* время восстановления интерфейса не всегда совпадает с плановым опросом элемента данных согласно заданному расписанию проверок.

Поэтому, когда интерфейс становится доступным, опрос элементов данных производится не сразу, а перепланируется в соответствии с расписанием их следующих проверок.

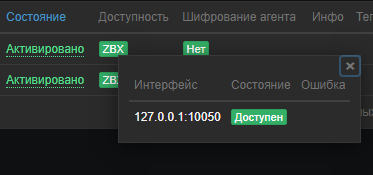
1. Недоступный интерфейс

После того как UnreachablePeriod завершился, а интерфейс так и не стал доступным, он считается недоступным.

В журнале Сервера это обозначается записями такого рода:

temporarily disabling Zabbix agent checks on host "New host": interface unavailable

При этом в веб-интерфейсе иконка доступности узла сети меняет свой цвет с зеленого/серого на желтый/красный (при наведении указателя "мыши" на иконку доступности узла сети можно видеть информацию о недостижимости интерфейса) (рисунок 1).



Доступность узла сети

Параметр UnavailableDelay определяет, как часто интерфейс проверяется повторно в течение его недоступности.

По умолчанию – раз в 60 секунд (таким образом, в этом случае "temporarily disabling" из файла журнала будет означать деактивацию проверок сроком на одну минуту).

Когда соединение c интерфейсом будет восстановлено, мониторинг тоже автоматически вернется к норме:

enabling Zabbix agent checks on host "New host": interface became available

1. Удаленный мониторинг статистики

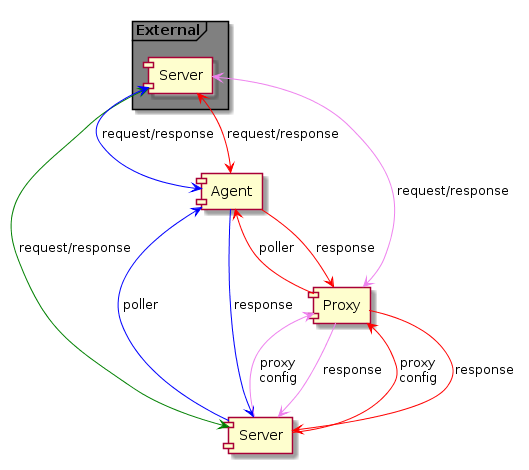
Некоторые внутренние метрики Сервера и Прокси можно сделать доступными удаленно для другого экземпляра Подсистемы или сторонних утилит. Может быть полезно, чтобы поставщики поддержки/сервисов имели возможность удаленно мониторить Сервер/Прокси своих клиентов, или в организациях, где Подсистема не является основным инструментом мониторинга, чтобы внутренние метрики Подсистемы могли контролироваться сторонней системой при использовании зонтичного мониторинга.

Внутренняя статистика Подсистемы открыта для настраиваемого списка адресов, перечисленных в новом параметре StatsAllowedIP Сервера/Прокси. Запросы будут приниматься только с этих адресов.

Для настройки запроса внутренней статистики с другого экземпляра Подсистемы можно использовать два элемента данных:

* Внутренний элемент данных zabbix[stats,<ip>,<порт>] – для прямых удаленных запросов к Серверу/Прокси. <ip> и <порт> используются для идентификации необходимого целевого экземпляра.
* Элемент данных Агента zabbix.stats[<ip>,<порт>] для удаленных запросов к Серверу/Прокси через Агента. <ip> и <порт> используются для идентификации необходимого целевого экземпляра.

Рисунок 2 иллюстрирует использование обоих элементов данных в зависимости от контекста.



Использование элементов данных

*  – Сервер → внешний экземпляр (zabbix[stats,<ip>,<порт>]);
*  – Сервер → Прокси → внешний экземпляр (zabbix[stats,<ip>,<порт>]);
*  – Сервер → Агент → внешний экземпляр (zabbix.stats[<ip>,<порт>]);
*  – Сервер → Прокси → Агент → внешний экземпляр (zabbix.stats[<ip>,<порт>]).

Чтобы убедиться, что целевой экземпляр разрешает запросы к себе со стороны внешнего экземпляра, указывают адрес внешнего экземпляра в параметре StatsAllowedIP на целевом экземпляре.

Используя предоставляемые метрики, элементы данных собирают статистику одним запросом и возвращают JSON, который является основой зависимых элементов данных и из которого они получают данные. Любым из двух элементов данных возвращаются следующие внутренние метрики:

* zabbix[boottime];
* zabbix[hosts];
* zabbix[items];
* zabbix[items\_unsupported];
* zabbix[preprocessing\_queue] (только Сервер);
* zabbix[process,<тип>,<режим>,<состояние>] (статистика только на основе типов процессов);
* zabbix[rcache,<кэш>,<режим>];
* zabbix[requiredperformance];
* zabbix[triggers] (только Сервер);
* zabbix[uptime];
* zabbix[vcache,buffer,<режим>] (только Сервер);
* zabbix[vcache,cache,<параметр>];
* zabbix[version];
* zabbix[vmware,buffer,<режим>];
* zabbix[wcache,<кэш>,<режим>] (тип кэша trends; только Сервер).

Доступные шаблоны для удаленного мониторинга внутренних метрик Сервера или Прокси с внешнего экземпляра:

* Remote Zabbix server;
* Remote Zabbix proxy.

Следует обратить внимание, что при использовании шаблонов удаленного мониторинга нескольких внешних экземпляров для мониторинга каждого внешнего экземпляра потребуется отдельный узел сети.

Получение запросов внутренних метрик с внешнего экземпляра Подсистемы обрабатывается процессом траппер, который проверяет запрос, собирает метрики, создает буфер данных JSON и отправляет назад подготовленный JSON, например, с Сервера:

{

"response": success,

"data":

{

"boottime": N,

"uptime": N,

"hosts": N,

"items": N,

"items\_unsupported": N,

"preprocessing\_queue": N,

"process": {

"alert manager": {

"busy": {

"avg": N,

"max": N,

"min": N

},

"idle": {

"avg": N,

"max": N,

"min": N

},

"count": N

},

...

},

"queue": N,

"rcache": {

"total": N,

"free": N,

"pfree": N,

"used": N,

"pused": N

},

"requiredperformance": N,

"triggers": N,

"uptime": N,

"vcache": {

"buffer": {

"total": N,

"free": N,

"pfree": N,

"used": N,

"pused": N

},

"cache": {

"requests": N,

"hits": N,

"misses": N,

"mode": N

}

},

"vmware": {

"total": N,

"free": N,

"pfree": N,

"used": N,

"pused": N

},

"version": "N",

"wcache": {

"values": {

"all": N,

"float": N,

"uint": N,

"str": N,

log: N,

"text": N,

"not supported": N

},

"history": {

"pfree": N,

"free": N,

"total": N,

"used": N,

"pused": N

},

"index": {

"pfree": N,

"free": N,

"total": N,

"used": N,

"pused": N

},

"trend": {

"pfree": N,

"free": N,

"total": N,

"used": N,

"pused": N

}

}

}

}

Также имеются два других элемента данных, которые позволяют получить внутреннюю статистику очереди удаленным запросом с другого экземпляра Подсистемы:

* Внутренний элемент данных zabbix[stats,<ip>,<порт>,queue,<от>,<до>] – для прямых запросов внутренней очереди с удаленного Сервера/Прокси;
* Элемент данных Агента zabbix.stats[<ip>,<порт>,queue,<от>,<до>] – для запросов внутренней очереди удаленного Сервера/Прокси через Агента.

1. Настройка Kerberos

Аутентификацию Kerberos можно использовать в веб-мониторинге и HTTP-элементах данных в Подсистеме.

В этом разделе описывается пример настройки Kerberos с Сервером для выполнения веб-мониторинга страницы www.example.com под пользователем zabbix по шагам:

1. установить пакет Kerberos:

* для Debian/Ubuntu:

apt install krb5-user

* для RHEL:

dnf install krb5-workstation

1. настроить файл конфигурации Kerberos:

``` {.java} cat /etc/krb5.conf

[libdefaults] default\_realm = EXAMPLE.COM

Следующие переменные krb5.conf предназначены только для MIT Kerberos:

kdc\_timesync = 1

ccache\_type = 4

forwardable = true

proxiable = true

[realms]

EXAMPLE.COM = {

}

[domain\_realm]

.example.com=EXAMPLE.COM

example.com=EXAMPLE.COM

**```**

1. создать билет Kerberos для пользователя zabbix, выполнив следующую команду под пользователем zabbix:

kinit zabbix

Важно выполнить команду выше именно под пользователем zabbix. Если выполнить ее под root, аутентификация не будет работать.

1. создать веб-сценарий или элемент данных типа "HTTP Агент" с типом аутентификации Kerberos.

Опционально можно протестировать при помощи следующей команды:

curl -v --negotiate -u : http://example.com

Следует обратить внимание, что для длительного веб-мониторинга потребуется позаботиться о продлении срока действия билета Kerberos. По умолчанию время истечения срока действия билета – 10 часов.

1. Параметры modbus.get

В таблице 47 представлены детальные описания параметров элемента данных modbus.get[].

Параметры modbus.get

| Параметр | Описание | По умолчанию |
| --- | --- | --- |
| точка входа | Протокол и адрес точки входа, определяемый в виде протокол://строка\_подключения Возможные значения протокола: rtu, ascii (только Агент-2), tcp Формат строки подключения: – для tcp – адрес:порт – для последовательной линии: rtu, ascii – имя\_порта:скорость:параметры где: – скорость – 1200, 9600 и т.д.; – параметры – количество бит (5,6,7 или 8), четность (n,e или o для нет(none)/чет(even)/нечет(odd)), стоп-битов (1 или 2). | протокол: нет  протокол: rtu/ascii: имя\_порта: нет скорость: 115200 параметры: 8n1  протокол: tcp: адрес: нет порт: 502 |
| id ведомого | Адрес modbus устройства, для которого он предназначен (от 1 до 247),  устройство tcp (не GW) будет игнорировать это поле | для последовательной линии: 1 для tcp: 255 (0xFF) |
| функция | Пусто или значение поддерживаемой функции: 1 – Чтение Флага, 2 – Чтение Дискретного Входа, 3 – Чтение Регистров Временного Хранения, 4 – Чтение Входных Регистров | пусто |
| адрес | Адрес первого регистра, флага или входа. Если параметр "функция" пустой, то "адрес" должен быть в диапазоне для: Флага – 00001 – 09999 Дискретного входа – 10001 – 19999 Входного регистра – 30001 – 39999 Регистра Временного Хранения – 40001 – 49999 Если параметр "функция" не пуст, то поле "адрес" будет от 0 до 65535 использоваться без модификации (PDU) | пустая функция: 00001  непустая функция: 0 |
| количество | Количество последовательных "типов", которые будут считаны с устройства, где: для флага или дискретного входа "тип" = 1 бит в остальных случаях: (количество\*тип)/2 = реальное количество регистров для чтения Если "сдвиг" не равен 0, то значение будет добавлено к "реальному количеству" Допустимый диапазон для "реального количества" составляет 1:65535 | 1 |
| тип | Тип данных: – для Чтения Флага и Чтения Дискретного Входа – bit – для Чтения Регистров Временного Хранения и Чтения Входных Регистров: int8 – 8 бит uint8 – 8 бит (без знака) int16 – 16 бит uint16 – 16 бит (без знака) int32 – 32 бита uint32 – 32 бита (без знака) float – 32 бита uint64 – 64 бита (без знака) double – 64 бита | bit uint16 |
| порядок следования байтов | Тип порядка следования байтов: be – Big Endian le – Little Endian mbe – Mid-Big Endian mle – Mid-Little Endian  Ограничения: для 1 бит – be для 8 бит – be,le для 16 бит – be,le | be |
| сдвиг | Количество регистров, начиная с "адреса", результаты которых будут отброшены. Размер каждого регистра составляет 16 бит (требуется для поддержки оборудования, не поддерживающего произвольный доступ для чтения). | 0 |

Приложение Д. Типы элементов данных

1. Агент

Эти проверки используют общение с Агентом для сбора данных.

Существуют пассивные и активные проверки (п.Г.3). При настройке элемента данных, можно выбрать требуемый тип:

* Агент – для пассивных проверок;
* Агент (активный) – для активных проверок.

В таблице 48-58 приводится подробная информация о поддерживаемых ключах элементов данных, которые можно использовать с элементами данных Агента, сгруппированная по семействам элементов данных.

Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками "<>", опциональны.

Следует обратить внимание, что при тестировании или использовании ключей элементов данных в zabbix\_agentd или zabbix\_get с командной строкой также нужно учитывать синтаксис командной оболочки.

Например, если какой-то параметр ключа должен быть заключен в двойные кавычки, придется явно экранировать эти двойные кавычки, иначе они будут обрезаны командной оболочкой как специальные символы и не будут переданы утилите:

zabbix\_agentd -t "vfs.dir.count[/var/log,,,"file,dir",,0]"

zabbix\_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]

Ключи элемента данных

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| agent.HostMetadata | | | |
| Метаданные узла сети Агента. | Строка |  | Получение значения параметров HostMetadata или HostMetadataItem, или пустая строка, если ничего не задано. |
| agent.Hostname | | | |
| Имя хоста Агента. | Строка |  | Возвращает: – пассивной проверкой – имя первого узла сети, указанное в параметре Hostname в файле конфигурации Агента; – активной проверкой – текущее значение имени хоста. |
| agent.ping | | | |
| Проверка доступности Агента. | Ничего – недоступен 1 – доступен |  | Использовать триггерную функцию nodata() для проверки недоступности узла сети. |
| agent.variant | | | |
| Проверка разновидности Агента (Агент-1 или Агент-2) | Целое число |  | Примеры возвращаемых значений: 1 – Агент-1; 2 – Агент-2. |
| agent.version | | | |
| Версия Агента. | Строка |  | Пример возвращаемого значения: 1.0.0 |
| zabbix.stats[<ip>,<порт>] | | | |
| Возвращает набор внутренних метрик с удаленного Сервера или Прокси. | JSON объект | ip – список IP/DNS/сетевой маски опрашиваемых удаленно Серверов / Прокси (по умолчанию 127.0.0.1) порт – порт опрашиваемого удаленно Сервера / Прокси (по умолчанию 10051) | Обратить внимание, что запрос статистики будет приниматься только с адресов, которые перечислены в параметре StatsAllowedIP Сервера/Прокси на удаленном экземпляре. Этим элементом данных возвращается выбранный список внутренних метрик. |
| zabbix.stats[<ip>,<порт>,queue,<от>,<до>] | | | |
| Возвращает количество наблюдаемых элементов данных в очереди, которые задерживаются на удаленном Сервере или Прокси. | JSON объект | – ip – список IP/DNS/сетевой маски опрашиваемых удаленно Серверов / Прокси (по умолчанию 127.0.0.1); – порт – порт опрашиваемого удаленно Сервера / Прокси (по умолчанию 10051); – queue – константа (используется как есть); – от – с задержкой как минимум (по умолчанию от 6 секунд); – до – с задержкой как максимум (по умолчанию до бесконечности) | Обратить внимание, что запрос статистики будет приниматься только с адресов, которые перечислены в параметре StatsAllowedIP Сервера/Прокси на удаленном экземпляре. |

Примечания:

* (Специфичное для Linux) Агент должен иметь права "только на чтение" на файловую систему /proc. Патчи к ядру от www.grsecurity.org ограничивают права доступа непривилегированных пользователей.
* vfs.dev.read[], vfs.dev.write[]: Агент будет прерывать потерянные подключения к устройствам, если значения элементов данных не опрашиваются более 3 часов. Это может произойти, если система имеет устройства с динамически меняющимися путями или если устройство было удалено вручную. Следует обратить также внимание, что эти элементы данных при использовании интервала обновления от 3 часов и более всегда будут возвращать значение "0".
* vfs.dev.read[], vfs.dev.write[]: если первым параметром используется параметр по умолчанию all, то ключ вернет суммарную статистику, включая все блочные устройства, такие как sda, sdb и их разделы (sda1, sda2, sdb3…) и несколько устройств (MD raid) на основе этих блочных устройств/разделов и логические разделы (LVM) на основе этих блочных устройств/разделов. В этих случаях возвращаемые значения следует рассматривать как относительные значения (изменяемые во времени), но не как абсолютные значения.
* SSL (HTTPS) поддерживается, только если Агент скомпилирован с поддержкой cURL. Иначе элемент данных станет неподдерживаемым.

Чтобы убедиться, что полученные данные не повреждены, можно указать корректную кодировку для обработки проверки (например, в "vfs.file.contents") в параметре кодировка. Список поддерживаемых кодировок (идентификаторов кодовых страниц) можно найти в документации к [libiconv](http://www.gnu.org/%E2%80%8Bsoftware/%E2%80%8Blibiconv/%E2%80%8B) (GNU Project) или в документации к Microsoft Windows SDK по "Идентификаторам кодовых страниц"​. ​

Если кодировка не задана в параметре ​кодировка, то применяются следующие стратегии преобразования:

* если кодировка не задана (или указана пустая строка), то предполагается наличие UTF-8-кодировки и данные обрабатываются "как есть";
* анализ BOM – применяется к элементам данных "vfs.file.contents", "vfs.file.regexp", "vfs.file.regmatch". Предпринимается попытка определить корректную кодировку при помощи использования маркера последовательности байтов (BOM) в начале файла. Если BOM отсутствует, применяется стандартное преобразование.

Для поиска проблем с элементами данных Агента при использовании с пассивным Агентом, возможно, значение конфигурации Сервера Timeout потребуется выставить выше, чем Timeout в файле конфигурации Агента. Иначе элемент данных может не получить никакого значения по причине того, что время ожидания запроса на стороне Сервера истечет раньше, чем на стороне Агента.

Ключи элемента данных ядра

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Возвращаемое значение | Параметры | Комментарии |
| kernel.maxfiles | | | |
| Максимальное количество открытых файлов, поддерживаемое ОС. | Целое число |  |  |
| kernel.maxproc | | | |
| Максимальное количество процессов, поддерживаемое ОС. | Целое число |  |  |
| kernel.openfiles | | | |
| Получение количества открытых в данное время файловых дескрипторов. | Целое число |  |  |

Ключи элемента данных журналов

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Возвращаемое значение | Параметры | Комментарии |
| **log**[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>,<опции>,<постоянное\_хранилище>] | | | |
| Мониторинг файлов журналов (логов). | Журнал (лог) | –файл – абсолютный путь и имя файла журнала; –регулярное выражение – регулярное выражение, описывающее необходимый шаблон содержимого; –кодировка – идентификатор кодовой страницы –макс. кол-во строк – максимальное количество новых строк в секунду, которое Агент будет отправлять Серверу или Прокси. Этот параметр переопределяет значение "MaxLinesPerSecond" в zabbix\_agentd.conf; –режим – возможные значения: all (по умолчанию), skip – пропуск обработки старых данных (влияет только на недавно созданные элементы данных, которые еще не получали никаких данных); – вывод – опциональный шаблон форматирования вывода. \0 – управляющая последовательность заменяется найденной частью текста (начиная от первого символа совпадения, заканчивая символом, где совпадение заканчивается), при этом \N (где N=1...9) – управляющая последовательность заменяется N-ой совпадающей группой (или пустой строкой, если N превышает количество найденных групп); – максзадержка – максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 – (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 – игнорировать более старые строки с целью получения наиболее новых строк, проанализированных в течение "максзадержка" секунд; ­– опции  – дополнительные опции; – постоянное\_хранилище  (только в zabbix\_agentd на UNIX-системах; не поддерживается в Агент-2) – абсолютный путь к директории, где хранить данные долговременного хранения. | Элемент данных должен быть настроен как активная проверка. Если файл не существует или доступ к нему не разрешен, элемент данных перейдет в неподдерживаемое состояние.  Если параметр вывод оставить пустым – будет возвращена вся строка, содержащая текст, соответствующий регулярному выражению. Обратить внимание, что все глобальные регулярные выражения, исключая "Результат ИСТИНА", всегда возвращают всю строку, соответствующую выражению, и параметр вывод игнорируется. Извлечение содержимого согласно параметру вывод производится на стороне Агента. |
| log.count[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>,<опции>,<постоянное\_хранилище>] | | | |
| Подсчет количества совпадающих строк в наблюдаемом файле журнала (логе). | Целое число | –файл – абсолютный путь и имя файла журнала; –регулярное выражение – регулярное выражение, описывающее необходимый шаблон содержимого; –кодировка – идентификатор кодовой страницы; –макс. кол-во строк – максимальное количество новых строк в секунду, которое Агент будет анализировать (не может превышать 10000). Значением по умолчанию является 10\*"MaxLinesPerSecond" в zabbix\_agentd.conf. –режим – возможные значения: all (по умолчанию), skip – пропуск обработки старых данных (влияет только на недавно созданные элементы данных); –максзадержка – максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 – (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 – игнорировать более старые строки с целью получения наиболее новых строк, проанализированных в течение "максзадержка" секунд;  –опции – дополнительные опции; –постоянное\_хранилище  (только в zabbix\_agentd на UNIX системах; не поддерживается в Агент-2) – абсолютный путь к директории, где хранить данные долговременного хранения. См. также дополнительные заметки о файлах долговременного хранения. | Элемент данных должен быть настроен как активная проверка.  Количество совпадающих строк вычисляется за счет новых строк, начиная со времени последней проверки журнала Агентом; таким образом, количество зависит от интервала обновления элемента данных. Если файл не существует или доступ к нему не разрешен, элемент данных переходит в неподдерживаемое состояние.  Этот элемент данных не поддерживается для Журнала событий Windows. |
| logrt[регулярное\_выражение\_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>,<опции>,<постоянное\_хранилище>] | | | |
| Мониторинг файлов журналов (логов) с поддержкой ротации файлов журналов. | Журнал (лог) | –регулярное\_выражение\_файла – абсолютный путь к файлу и регулярное выражение, описывающее его имя. Обратить внимание, регулярным выражением является только имя файла; регулярное выражение – регулярное выражение, описывающее необходимый шаблон содержимого; кодировка – идентификатор кодовой страницы; макс. кол-во строк – максимальное количество новых строк в секунду, которое Агент будет отправлять Серверу или Прокси. Этот параметр переопределяет значение "MaxLinesPerSecond" в zabbix\_agentd.conf; –режим – возможные значения: all (по умолчанию), skip – пропуск обработки старых данных (влияет только на недавно созданные элементы данных, которые еще не получали никаких данных); –вывод – опциональный шаблон форматирования вывода. \0 – управляющая последовательность заменяется найденной частью текста (начиная от первого символа совпадения, заканчивая символом, где совпадение заканчивается), то \N (где N=1...9) – управляющая последовательность заменяется N-ой совпадающей группой (или пустой строкой, если N превышает количество найденных групп); максзадержка – максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 – (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 – игнорировать более старые строки с целью получения наиболее новых строк, проанализированных в течение "максзадержка" секунд.  ­–опции – тип ротации файлов журналов и другие опции. Возможные значения: rotate (по умолчанию), copytruncate – следует обратить внимание, что copytruncate нельзя использовать вместе с maxdelay. В этом случае maxdelay должен быть равен 0 или не указан; mtime-reread – не уникальные записи, перечитывать, если время модификации или размер изменились (по умолчанию), mtime-noreread – не уникальные записи, перечитывать, только если размер файла изменяется (игнорирование изменения времени модификации); –постоянное\_хранилище ( только в zabbix\_agentd на UNIX системах; не поддерживается в Agent2) – абсолютный путь к директории, где хранить данные долговременного хранения. | Элемент данных должен быть настроен как активная проверка. Ротация журналов основывается на времени последней модификации файлов. Обратить внимание, что logrt разработан для работы с одним активным файлом журнала, с несколькими другими совпадающими уже устаревшими неактивными файлами. Если, например, директория содержит множество активных файлов журналов, по каждому такому файлу необходимо создавать отдельный logrt элемент данных. Иначе, если один logrt элемент данных подхватит слишком много файлов, это может привести к исчерпанию памяти и сбою в мониторинге.  Если параметр вывод оставить пустым – будет возвращена вся строка, содержащая соответствующий регулярному выражению текст. Обратить внимание, что все глобальные регулярные выражения, исключая "Результат ИСТИНА", всегда возвращают всю соответствующую выражению строку, и параметр вывод игнорируется. Извлечение содержимого согласно параметру вывод производится на стороне Агента. |
| logrt**.count**[регулярное\_выражение\_файла ,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>,<опции>,<постоянное\_хранилище>] | | | |
| Подсчет количества совпадающих строк в наблюдаемом файле журнала (логе) с поддержкой ротации файлов журналов. | Целое число | **–**регулярное\_выражение\_файла **–** абсолютный путь к файлу и регулярное выражение, описывающее его имя. Обратить внимание, регулярным выражением является только имя файла; –регулярное выражение – регулярное выражение, описывающее необходимый шаблон содержимого; –кодировка – идентификатор кодовой страницы; –макс. кол-во строк – максимальное количество новых строк в секунду, которое Агент будет анализировать (не может превышать 10000). Значением по умолчанию является 10\*"MaxLinesPerSecond" в zabbix\_agentd.conf; –режим – возможные значения: all (по умолчанию), skip – пропуск обработки старых данных (влияет только на недавно созданные элементы данных); –максзадержка – максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 – (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 – игнорировать более старые строки с целью получения наиболее новых строк, проанализированных в течение "максзадержка" секунд; –опции – тип ротации файлов журналов и другие опции. Возможные значения: rotate (по умолчанию), copytruncate – следует обратить внимание, что copytruncate нельзя использовать вместе с maxdelay. В этом случае maxdelay должен быть равен 0 или не указан;  mtime-reread – не уникальные записи, перечитывать, если время модификации или размер изменились (по умолчанию), mtime-noreread – не уникальные записи, перечитывать, только если размер файла изменяется (игнорирование изменения времени модификации); –постоянное\_хранилище (только в zabbix\_agentd на UNIX системах; не поддерживается в Агент-2) – абсолютный путь к директории, где хранить данные долговременного хранения. См. также дополнительные заметки о файлах долговременного хранения. | Элемент данных должен быть настроен как активная проверка.  Количество совпадающих строк вычисляется за счет новых строк, начиная со времени последней проверки журнала Агентом; таким образом, количество зависит от интервала обновления элемента данных.  Этот элемент данных не поддерживается для журнала событий Windows. |

Ключи элемента данных Modbus

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Возвращаемое значение | Параметры | Комментарии |
| modbus.get[точка входа,<id ведомого>,<функция>,<адрес>,<количество>,<тип>,<порядок следования байтов>,<сдвиг>] | | | |
| Чтение данных Modbus. | JSON объект | –точка входа – точка входа, определяемая в виде протокол://строка\_подключения; –id ведомого – ID ведомого; –функция – функция Modbus; –адрес – адрес первого регистра, флага или входа; –количество – количество записей для чтения; –тип – тип данных; –порядок следования байтов – настройка порядка следования байтов; –сдвиг – количество регистров, начиная с "адреса", результаты которых будут отброшены. |  |

Ключи элемента сетевых данных

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Возвращаемое значение | Параметры | Комментарии |
| net.dns[<ip>,имя,<тип>,<время ожидания>,<количество>,<протокол>] | | | |
| Проверка работы DNS- сервиса | 0 – DNS не работает (Сервер не ответил или ответ DNS некорректен)  1 – DNS работает | –ip – IP-адрес DNS-сервера (оставить пустым, чтобы использовались DNS-сервера по умолчанию, на Windows поддерживается Агентом-2, игнорируется Агентом-1); –имя – проверяемое имя DNS; –тип – запрашиваемый тип записи (по умолчанию SOA); –время ожидания – время ожидания ответа в секундах (по умолчанию 1 секунда); –количество – количество попыток запросов (по умолчанию 2); –протокол – используемый протокол при выполнении DNS-запросов: udp (по умолчанию) или tcp | Возможные значения для тип: ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS (не поддерживается Агентом на Windows, Агентом-2 на всех ОС), HINFO, MINFO, TXT, AAAA, SRV. Национальные доменные имена не поддерживаются; вместо них использовать имена в IDNA-кодировке. |
| net.dns.record[<ip>,имя,<тип>,<время ожидания>,<количество>,<протокол>] | | | |
| Выполнение DNS- запроса. | Строка символов с требуемым типом информации | –ip – IP-адрес DNS-сервера (оставить пустым, чтобы использовались DNS-сервера по умолчанию); –имя – проверяемое имя DNS; –тип – запрашиваемый тип записи (по умолчанию SOA); –время ожидания – время ожидания ответа в секундах (по умолчанию 1 секунда); –количество – количество попыток запросов (по умолчанию 2); –протокол – используемый протокол при выполнении DNS-запросов: udp (по умолчанию) или tcp. | Возможные значения для тип: ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS (не поддерживается Агентом-1 на Windows, Агентом-2 на всех ОС), HINFO, MINFO, TXT, AAAA, SRV. Национальные доменные имена не поддерживаются; вместо них использовать имена в IDNA-кодировке. |
| net.if.collisions[if] | | | |
| Коллизии за пределами окна (out-of-window). | Целое число | –if – имя сетевого интерфейса |  |
| net.if.discovery | | | |
| Список сетевых интерфейсов. Используется низкоуровневым обнаружением. | Объект JSON |  | В некоторых версиях Windows (например, Server 2008) может потребоваться установка последних обновлений для поддержки не-ASCII символов в именах интерфейсов. |
| net.if.in[if,<режим>] | | | |
| Статистика по входящему трафику сетевого интерфейса. | Целое число | –if – имя сетевого интерфейса (UNIX); полное описание сетевого интерфейса или IPv4-адрес; или, если в фигурных скобках, GUID сетевого интерфейса (Windows); –режим – возможные значения: Bytes – количество байтов (по умолчанию); packets – количество пакетов; errors – количество ошибок; dropped – количество отброшенных пакетов; overruns (fifo) – количество ошибок FIFO-буфера; frame – количество ошибок кадрирования пакетов; compressed – количество сжатых пакетов, полученных драйвером устройства; multicast – количество кадров многоадресного вещания, которые получены драйвером устройства. | В Windows элемент данных берет значения из 64-битных счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битные счетчики недоступны, Агент будет использовать 32-битные счетчики. Можно получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list. Можно использовать этот ключ с шагом предобработки "Изменение в секунду" для получения статистики байтов в секунду. |
| net.if.out[if,<режим>] | | | |
| Статистика по исходящему трафику сетевого интерфейса. | Целое число | –if – имя сетевого интерфейса (UNIX); полное описание сетевого интерфейса или IPv4-адрес; или, если в фигурных скобках, GUID сетевого интерфейса (Windows); –режим – возможные значения: Bytes – количество байтов (по умолчанию); packets – количество пакетов; errors – количество ошибок; dropped – количество отброшенных пакетов; overruns (fifo) – количество ошибок FIFO-буфера; collisions (colls) – количество коллизий, обнаруженных на интерфейсе; carrier – количество потерь несущей, обнаруженных драйвером устройства; compressed – количество сжатых пакетов, переданных драйвером устройства. | В Windows элемент данных берет значения из 64-битных счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битные счетчики недоступны, Агент будет использовать 32-битные счетчики. Можно получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list. Можно использовать этот ключ с шагом предобработки "Изменение в секунду" для получения статистики байтов в секунду. |
| net.if.total[if,<режим>] | | | |
| Суммарная статистика входящего и исходящего трафика на сетевом интерфейсе. | Целое число | –if – имя сетевого интерфейса (UNIX); полное описание сетевого интерфейса или IPv4-адрес; или, если в фигурных скобках, GUID сетевого интерфейса (Windows); –режим – возможные значения: Bytes – количество байтов (по умолчанию); packets – количество пакетов; errors – количество ошибок; dropped – количество отброшенных пакетов; overruns (fifo) – количество ошибок FIFO-буфера; compressed – количество сжатых пакетов, переданных или полученных драйвером устройства. | В Windows элемент данных берет значения из 64-битных счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битные счетчики недоступны, Агент будет использовать 32-битные счетчики. Можно получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list. Можно использовать этот ключ с шагом предобработки "Изменение в секунду" для получения статистики байтов в секунду. Обратить внимание: отброшенные пакеты поддерживаются, только если оба ключа net.if.in и net.if.out работают с отброшенными пакетами на вашей платформе. |
| net.tcp.LISTEN[порт] | | | |
| Проверка, находится ли TCP- порт в состоянии LISTEN. | 0 – не находится в состоянии LISTEN 1 – находится в состоянии LISTEN | –порт – номер TCP-порта |  |
| net.tcp.port[<ip>,порт] | | | |
| Проверка возможности создания TCP подключения на указанный номер порта. | 0 – не удалось подключиться  1 – удалось подключиться | –ip – IP-адрес или DNS-имя (по умолчанию 127.0.0.1); –порт – номер порта. | Пример: => net.tcp.port[,80] → можно использовать для проверки доступности веб-сервера, работающего на 80 порту.  Для простого тестирования производительности TCP использовать net.tcp.service.perf[tcp,<ip>,<порт>]  Обратить внимание, что эти проверки могут привести к дополнительным записям в системных файлах журналов (обычно сессии SMTP и SSH журналируются). |
| net.tcp.service[сервис,<ip>,<порт>] | | | |
| Проверка, запущен ли сервис и принимает ли он TCP подключения. | 0 – сервис недоступен  1 – сервис запущен | –сервис – один из: ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet;  **–**ip – IP-адрес или DNS-имя (по умолчанию 127.0.0.1); –порт – номер порта (по умолчанию используется стандартный номер порта сервиса). | Пример: => net.tcp.service[ftp,,45] можно использовать для тестирования доступности FTP-сервера на TCP 45 порту. Обратить внимание, что эти проверки могут привести к дополнительным записям в системных файлах журналов (обычно сессии SMTP и SSH журналируются). Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. В качестве обходного решения для подобных проверок использовать net.tcp.port. Проверка LDAP и HTTPS на платформе Windows поддерживается только Агентом-2. Обратить внимание, что telnet проверка ищет запрос на вход (с ":" в конце). |
| net.tcp.service.perf[сервис,<ip>,<порт>] | | | |
| Проверка производительности TCP-сервиса. | 0 – сервис недоступен  секунды – количество секунд, потраченное на подключение к сервису | –сервис – один из: ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet ; –ip – IP-адрес или DNS-имя (по умолчанию 127.0.0.1); –порт – номер порта (по умолчанию используется стандартный номер порта сервиса). | Пример: => net.tcp.service.perf[ssh] можно использовать для тестирования скорости изначального ответа от SSH-сервера. Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. В качестве обходного решения для подобных проверок использовать net.tcp.service.perf[tcp,<ip>,<порт>]. Проверка LDAP и HTTPS на платформе Windows поддерживается только Агентом-2. Обратить внимание, что telnet проверка ищет запрос на вход (с ":" в конце). |
| net.tcp.socket.count[<локальный\_адрес>,<локальный\_порт>,<удаленный\_адрес>,<удаленный\_порт>,<состояние>] | | | |
| Получение количества TCP-сокетов, соответствующих заданным параметрам. | Целое число | –локальный\_адрес – локальный IPv4/6-адрес или CIDR-подсеть; –локальный\_порт – номер локального порта или имя сервиса; –удаленный\_адрес – удаленный IPv4/6-адрес или CIDR-подсеть; –удаленный\_порт – номер удаленного порта или имя сервиса; –состояние – состояние соединения (established, syn\_sent, syn\_recv, fin\_wait1, fin\_wait2, time\_wait, close, close\_wait, last\_ack, LISTEN, closing) | Этот элемент данных поддерживается только на Linux как Агентом-1, так и Агентом-2. В Агенте-2 этот ключ также поддерживается на 64-битной Windows. Например: => net.tcp.socket.count[,80,,,established]– проверка, имеется ли локальный TCP порт 80 в состоянии established |
| net.udp.LISTEN[порт] | | | |
| Проверка, находится ли UDP-порт в состоянии LISTEN. | 0 – не находится в состоянии LISTEN 1 – находится в состоянии LISTEN | –порт – номер UDP-порта | Например: => net.udp.LISTEN[68] |
| net.udp.service[сервис,<ip>,<порт>] | | | |
| Проверка, запущен ли сервис и принимает ли он UDP подключения. | 0 – сервис недоступен  1 – сервис запущен | –сервис – ntp; –ip – IP-адрес или DNS-имя (по умолчанию 127.0.0.1); –порт – номер порта (по умолчанию используется стандартный номер порта сервиса) | Пример: => net.udp.service[ntp,,45] можно использовать для тестирования доступности NTP сервиса на 45 порту UDP. |
| net.udp.service.perf[сервис,<ip>,<порт>] | | | |
| Проверка производительности UDP сервиса. | 0 – сервис недоступен секунды – количество секунд, потраченное на ожидание ответа от сервиса | –сервис – ntp;  –ip – IP-адрес или DNS-имя (по умолчанию 127.0.0.1); –порт – номер порта (по умолчанию используется стандартный номер порта сервиса) | Пример: => net.udp.service.perf[ntp] можно использовать для тестирования времени отклика от NTP-сервиса. |
| net.udp.socket.count[<локальный\_адрес>,<локальный\_порт>,<удаленный\_адрес>,<удаленный\_порт>,<состояние>] | | | |
| Получение количества UDP-сокетов, соответствующих заданным параметрам. | Целое число | –локальный\_адрес – локальный IPv4/6-адрес или CIDR-подсеть; –локальный\_порт – номер локального порта или имя сервиса; –удаленный\_адрес – удаленный IPv4/6-адрес или CIDR-подсеть; –удаленный\_порт – номер удаленного порта или имя сервиса; –состояние – состояние соединения (established, unconn). | Этот элемент данных поддерживается только на Linux как Агентом-1, так и Агентом-2. В Агенте-2 этот ключ также поддерживается на 64-битной Windows. Например: => net.tcp.socket.count[,,,,established] возвращает количество UDP-сокетов в состоянии established |

Ключ элемента данных процессов

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Возвращаемое значение | Параметры | Комментарии |
| proc.cpu.util[<имя>,<пользователь>,<тип>,<cmdline>,<режим>,<зона>] | | | |
| Использование CPU процесса в процентах. | Число с плавающей точкой | –имя – имя процесса (по умолчанию все процессы);  –пользователь – имя пользователя (по умолчанию все пользователи); –тип – тип использования CPU: total (по умолчанию), user, system; –cmdline – фильтр по командной строке (является регулярным выражением); –режим – режим сбора данных: avg1 (по умолчанию), avg5, avg15; –зона – целевая зона: current (по умолчанию), all. Этот параметр поддерживается только на платформе Solaris. | Примеры: => proc.cpu.util[,root] – Использование CPU по всем процессам, работающим под пользователем root; => proc.cpu.util[zabbix\_server,zabbix] – Использование CPU по всем процессам zabbix\_server работающими под пользователем zabbix. Возвращаемое значение основывается на использовании одного ядра CPU в процентах. Например, утилизация CPU процесса, полностью использующего два ядра, равна 200%. Данные использования CPU процессами собираются коллекторами, которые поддерживают максимально до 1024 уникальных (по имени, пользователю и командной строке) запросов. Запросы, к которым не было обращения в течение последних 24 часов, удаляются из коллектора. Обратить внимание, что в случае, если Агент скомпилирован на Solaris без поддержки зон, если задать параметр зона значением current (или по умолчанию) и запустить его на более новом Solaris, где зоны поддерживаются, то Агент вернет NOTSUPPORTED (Агент не может ограничить результаты в пределах только текущей зоны). Значение all в этом случае поддерживается. |
| proc.mem[<имя>,<пользователь>,<режим>,<cmdline>,<тип памяти>] | | | |
| Количество памяти, используемое процессом, в байтах. | Целое число – с режимом равным max, min, sum  Дробное число – с режимом равным avg | –имя – имя процесса (по умолчанию все процессы); –пользователь – имя пользователя (по умолчанию все пользователи); –режим – возможные значения: avg, max, min, sum (по умолчанию); –cmdline – фильтр по командной строке (является регулярным выражением); –тип памяти – тип памяти, используемый процессом. | Пример: => proc.mem[,root] – память, используемая всеми процессами запущенными под пользователем root. Обратить внимание: если несколько процессов используют разделяемую память, сумма используемой процессами памяти может дать в результате большие, нереальные значения. Если этот элемент данных вызывается с командной строки и содержит параметр командной строки (например, при использовании тестового режима Агента: zabbix\_agentd -t proc.mem[,,,apache2]), будет засчитан один дополнительный процесс, так как Агент посчитает самого себя. |
| proc.num[<имя>,<пользователь>,<состояние>,<cmdline>,<зона>] | | | |
| Количество процессов. | Целое число | –имя – имя процесса (по умолчанию все процессы); –пользователь – имя пользователя (по умолчанию все пользователи); –состояние – возможные значения: all (по умолчанию), disk – непрерывный сон, run – в процессе работы, sleep – прерываемый сон, trace – остановлен, zomb – зомби –cmdline – фильтр по командной строке (является регулярным выражением); –зона – целевая зона: current (по умолчанию), all. Этот параметр поддерживается только на платформе Solaris. | Примеры ключей: => proc.num[,mysql] – количество процессов, выполняемых под пользователем mysql; => proc.num[apache2,www-data] – количество процессов apache2, выполняемых под пользователем www-data. В Windows, поддерживаются только параметры имя и пользователь. Если этот элемент данных вызывается с командной строки и содержит параметр командной строки (например, при использовании тестового режима Агента: zabbix\_agentd -t proc.num[,,,apache2]), будет засчитан один дополнительный процесс, так как Агент посчитает самого себя. Обратить внимание, что в случае, когда Агент скомпилирован на Solaris без поддержки зон, если задать параметр зона значением current (или по умолчанию) и запустить его на более новом Solaris, где зоны поддерживаются, то Агент вернет NOTSUPPORTED (Агент не может ограничить результаты в пределах только текущей зоны). Значение all в этом случае поддерживается. |

Ключ элемента данных датчиков

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Возвращаемое значение | Параметры | Комментарии |
| sensor[устройство,сенсор,<режим>] | | | |
| Чтение аппаратного сенсора. | Число с плавающей точкой | –устройство – имя устройства; может быть регулярным выражением, если режим не указан; –сенсор – имя сенсора; может быть регулярным выражением, если режим не указан; –режим – возможные значения: avg, max, min (если этот параметр не указан, то устройство и сенсор обрабатываются дословно). | В Linux 2.4 читается /proc/sys/dev/sensors. Пример: => sensor[w83781d-i2c-0-2d,temp1] |
| В Linux 2.6+ читается /sys/class/hwmon. |
| В OpenBSD, читается MIB hw.sensors. Примеры: => sensor[cpu0,temp0] – температура одного CPU => sensor[cpu[0-2]$,temp,avg] – средняя температура первых трех CPU |

Ключ элемента данных о системе

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| system.boottime | | | |
| Штамп времени (timestamp) загрузки системы. | Целое число (UNIX timestamp) |  |  |
| system.cpu.discovery | | | |
| Список найденных CPU/CPU ядер. Используется низкоуровневым обнаружением. | Объект JSON |  |  |
| system.cpu.intr | | | |
| Прерывания устройств. | Целое число |  |  |
| system.cpu.load[<cpu>,<режим>] | | | |
| [Загрузка CPU](https://ru.wikipedia.org/wiki/Load_Average). | Число с плавающей точкой | –cpu – возможные значения: all (по умолчанию), percpu (общая загрузка делится на количество CPU онлайн); –режим – возможные значения: avg1 (усреднение за одну минуту, по умолчанию), avg5, avg15 | Пример: => system.cpu.load[,avg5] |
| system.cpu.num[<тип>] | | | |
| Количество CPU. | Целое число | –тип – возможные значения: online (по умолчанию), max | Пример: => system.cpu.num |
| system.cpu.switches | | | |
| Количество переключений контекста. | Целое число |  |  |
| system.cpu.util[<cpu>,<тип>,<режим>,<логический\_или\_физический>] | | | |
| Утилизация CPU в процентах. | Число с плавающей точкой | –cpu – <номер CPU> или all (по умолчанию); –тип – возможные значения: user (по умолчанию), idle, nice, system (по умолчанию для Windows), iowait, interrupt, softirq, steal, guest (на ядрах Linux 2.6.24 и выше), guest\_nice (на ядрах Linux 2.6.33 и выше); –режим – возможные значения: avg1 (усреднение за одну минуту, по умолчанию), avg5, avg15; –логический\_или\_физический (только на AIX) – возможные значения: logical (по умолчанию), physical. Этот параметр поддерживается только на AIX. | На Windows значение получается с использованием счетчика производительности "Processor Time". Обратить внимание, начиная с Windows 8 его Менеджер Задач отображает утилизацию CPU, используя счетчик производительности "Processor Utility". Пример: => system.cpu.util[0,user,avg5] |
| system.Hostname[<тип>, <преобразование>] | | | |
| Системное имя хоста. | Строка | –тип – возможные значения: netbios (по умолчанию на Windows), host (по умолчанию на Linux), shorthost возвращает часть хоста до первой точки, всю строку для имен без точек); –преобразование – возможные значения: none (по умолчанию), lower (конвертация в нижний регистр) | Значение опрашивается на платформе Windows либо функцией GetComputerName() (для netbios), либо функцией getHostname() (для **host**), а на остальных системах – забирая значение nodename из вывода системного API uname(). |
| system.hw.chassis[<информация>] | | | |
| Информация о шасси. | Строка | –информация – одно из full (по умолчанию), model, serial, type или vendor | Пример: system.hw.chassis[full] Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop] Этот ключ зависит от наличия [SMBIOS](https://ru.wikipedia.org/wiki/SMBIOS) таблицы. Будет выполнена попытка чтения DMI таблицы из sysfs; если доступ к sysfs будет неудачным, то будет попытка чтения напрямую из памяти. Требуются праваroot, потому что значение читается из sysfs или памяти. |
| system.hw.cpu[<cpu>,<информация>] | | | |
| Информация о CPU. | Строка или целое число | –cpu – <номер CPU> или all (по умолчанию); –информация – возможные значения: full (по умолчанию), curfreq, maxfreq, model или vendor | Пример: => system.hw.cpu[0,vendor] → AuthenticAMD Информация собирается из /proc/cpuinfo и из /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo\_max\_freq. Если указаны номер CPU и curfreq или maxfreq, то возвращается числовое значение (Гц). |
| system.hw.devices[<тип>] | | | |
| Список PCI или USB устройств. | Текст | –тип – pci (по умолчанию) или usb | Пример: => system.hw.devices[pci] → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge [..] Возвращает вывод утилит lspci или lsusb (выполняются без каких-либо параметров). |
| system.hw.macaddr[<интерфейс>,<формат>] | | | |
| Список MAC адресов. | Строка | –интерфейс – all (по умолчанию) или регулярное выражение; –формат – full (по умолчанию) или short. | Список MAC-адресов интерфейсов, чьи имена совпадают с заданным в параметре интерфейс регулярным выражением (all список всех интерфейсов). Пример:  => system.hw.macaddr["eth0$",full] => [eth0] 00:11:22:33:44:55 Если формат задан значением short, имена интерфейсов и одинаковые MAC-адреса не попадут в список. |
| system.localtime[<тип>] | | | |
| Системное время. | Целое число – с тип как utc  Строка – с тип как local. | –type – возможные значения: utc – (по умолчанию) время с начала Эпохи (00:00:00 UTC, Январь 1, 1970), измеренное в секундах. local – время в формате "гггг-мм-дд,чч:мм:сс.ннн,+чч:мм" | Необходимо использовать только как пассивную проверку. Например: => system.localtime[local] – создать элемент данных, используя этот ключ, и далее использовать его для отображения времени узла сети в элементе Комплексного экрана Часы. |
| system.run[команда,<режим>] | | | |
| Выполнение указанной команды на узле сети. | Текст с результатом выполнения команды 1 – c режимом со значением nowait (независимо от результата выполнения команды) | –команда – выполняемая команда; –режим – возможные значения: wait – ожидать завершения выполнения (по умолчанию), nowait – не ожидать | Возвращаемое значение элемента данных является стандартным выводом совместно со стандартным выводом ошибок, которые получены от команды. Код завершения не проверяется. Для корректной обработки, возвращаемое командой значение должно иметь тип данных текст. Пустой результат также допустим. Возвращаемое значение ограничено 16 МБ (включая конечные пробелы, которые усекаются); также применяются ограничения баз данных. Обратить внимание, что элементы данных system.run по умолчанию запрещены.  Пример: => system.run[ls -l /] – подробный список файлов в корневой директории. |
| system.stat[ресурс,<тип>] | | | |
| Статистика системы. | Целое число или число с плавающей точкой | –ent – количество процессорных единиц, выделенных на ресурс (с плавающей точкой); –kthr,<тип> – информация о состояниях потоков ядра: r – среднее количество запускаемых потоков ядра (дробное) b – среднее количество потоков ядра, помещенных в очередь ожидания Менеджера Виртуальный Памяти (с плавающей точкой); –memory,<тип> – информация об использовании виртуальной и реальной памяти: avm – активные виртуальные страницы (целое) fre – размер свободного списка (целое); –page,<тип> – информация об ошибках страниц и активности страниц: fi – операции страничного ввода файлов в секунду (с плавающей точкой) fo – операции страничного вывода файлов в секунду (с плавающей точкой) pi – страницы, выгружаемые из подкачки (с плавающей точкой) po – страницы, загружаемые в подкачку (с плавающей точкой) fr – замененные страницы (с плавающей точкой) sr – количество страниц, сканированных алгоритмом замещения страниц (с плавающей точкой); –faults,<тип> – соотношение ловушек и прерываний: in – прерывания устройства (с плавающей точкой); sy – системные вызовы (с плавающей точкой); cs – переключения контекста потока ядра (с плавающей точкой); –cpu,<тип> – распределение процессорного времени в процентах: us – пользовательское время (с плавающей точкой); sy – системное время (с плавающей точкой); id – время простоя (с плавающей точкой); wa – время простоя, в ходе которого система обрабатывала запросы I/O диска/NFS (с плавающей точкой); pc – количество использованных физических процессоров (с плавающей точкой); ec – процентное соотношение размеченного занятого пространства (с плавающей точкой); lbusy – отображает утилизацию в процентах для логических процессор(ов), которая произошла при выполнении уровней пользователя и системы (с плавающей точкой); app – отображает количество доступных физических процессоров в распределенном пуле (с плавающей точкой); –disk,<тип> – статистика о дисках: bps – отображает суммарное количество данных переданных (чтением или записью) на диск в байтах в секунду (целое); tps – показывает количество передач за секунду, которые произошли на физическом диске/ленте (с плавающей точкой). | Этот элемент данных поддерживается только на AIX. Учесть следующие ограничения этих ключей: => system.stat[cpu,app] – поддерживается только на AIX LPAR с типом Shared; => system.stat[cpu,ec] – поддерживается на AIX LPAR с типом Shared и Dedicated (Dedicated всегда возвращает 100 (процентов)); => system.stat[cpu,lbusy] – поддерживается только на AIX LPAR с типом Shared; => system.stat[cpu,pc] – поддерживается на AIX LPAR с типом Shared и Dedicated; => system.stat[ent] – поддерживается на AIX LPAR с типом Shared и Dedicated. |
| system.sw.arch | | | |
| Информация об архитектуре программного обеспечения. | Строка | | Пример: => system.sw.arch → i686  Для получения информации используется функция uname(). |
| system.sw.os[<информация>] | | | |
| Информация об операционной системе. | Строка | –информация – возможные значения: full (по умолчанию), short или name | Пример: => system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11 Информация извлекается из (Следует обратить внимание, что не все файлы и опции представлены во всех дистрибутивах): /proc/version (full) /proc/version\_signature (short) Параметр PRETTY\_NAME из /etc/os-release на системах, которые его поддерживают, или /etc/issue.net (name) |
| system.sw.packages[<регулярное выражение>,<менеджер>,<формат>] | | | |
| Список установленных пакетов. | Текст | –регулярное выражение – all (по умолчанию) или регулярное выражение; –менеджер – all (по умолчанию) или конкретный менеджер пакетов; –формат – full (по умолчанию) или short. | Список (в алфавитном порядке) установленных пакетов, имена которых совпадают с регулярным выражением, заданным в пакет (all списки всех).  Пример: => system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal Поддерживаемые менеджеры пакетов (выполняемые команды): dpkg (dpkg --get-selections) pkgtool (ls /var/log/packages) rpm (rpm -qa) pacman (pacman -Q) Если формат указан значением full, пакеты группируются по менеджерам пакетов (каждый менеджер с отдельной строки, которая начинается с его имени, заключенного в квадратные скобки). Если формат указан значением short, пакеты не группируются и отображаются одной строкой. |
| system.swap.in[<устройство>,<тип>] | | | |
| Статистика файла подкачки в (с устройства в память). | Целое число | –устройство – устройство, используемое для файла подкачки (по умолчанию all); –тип – возможные значения: count (количество swapins), sectors (секторов swapped in), pages (страниц swapped in). | Пример: => system.swap.in[,pages]  Источником этой информации является: /proc/swaps, /proc/partitions, /proc/stat (Linux 2.4) /proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6) |
| system.swap.out[<устройство>,<тип>] | | | |
| Статистика файла подкачки из (из памяти в устройство). | Целое число. | –устройство – устройство, используемое для файла подкачки (по умолчанию all); –тип – возможные значения: count (количество swapouts), sectors (секторов swapped out), pages (страниц swapped out). | Пример: => system.swap.out[,pages]  Источником этой информации является: //proc/swaps, /proc/partitions, /proc/stat (Linux 2.4) /proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6) |
| system.swap.size[<устройство>,<тип>] | | | |
| Размер файла подкачки в байтах или процентах от общего размера. | Целое число – для количества байт  Число с плавающей точкой – для процентов. | –устройство – устройство, используемое для файла подкачки (по умолчанию all); –тип – возможные значения: free (свободно в файле подкачки, по умолчанию), pfree (свободно в файле подкачки, в процентах), pused (используемое место в файле подкачки, в процентах), total (полный размер файла подкачки), used (используется места в файле подкачки) Обратить внимание, что pfree, pused не поддерживаются на Windows, если размер файла подкачки нулевой. | Пример: => system.swap.size[,pfree] – свободное место в файле подкачки в процентах  Если устройство не указано, Агент будет брать во внимание только swap-устройства (файлы), физическая память будет игнорироваться. Например, на Solaris системах команда "swap -s" включает часть физической памяти и swap-устройства (в отличие от "swap -l"). Обратить внимание, этот ключ может возвращать некорректные данные размера места в файле подкачки, а также в процентах, на виртуализированных (VMware ESXi, VirtualBox) Windows платформах. В этом случае можно использовать ключ perf\_counter[\700(\_Total)\702] для получения корректного использования файла подкачки в процентах. |
| system.uname | | | |
| Идентификация системы. | Строка |  | Пример возвращаемого значения (UNIX): FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386 Пример возвращаемого значения (Windows): Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86 В UNIX значение этого элемента данных извлекается при помощи системного вызова uname(). Ранее эта информация запрашивалась с помощью системного вызова "uname -a". Значение этого элемента данных может отличаться от вывода "uname -a" и может не включать дополнительную информацию, которую "uname -a" выводит на основании на других источников. В Windows значение этого элемента данных извлекается при помощи классов WMI Win32\_OperatingSystem и Win32\_Processor. Ранее эта информация запрашивалась при помощи непостоянных вызовов Windows API и недокументированных ключей реестра. Имя ОС (включая редакцию) может быть переведено на пользовательский язык интерфейса. Некоторые версии Windows содержат символы товарных знаков и дополнительные пробелы. Обратить внимание, что этот элемент данных в Windows возвращает архитектуру ОС, при этом в UNIX он возвращает архитектуру CPU. |
| system.uptime | | | |
| Время работы в секундах. | Целое число |  | При настройке элемента данных использовать единицы измерения s или uptime для получения читабельных значений. |
| system.users.num | | | |
| Количество пользователей, находящихся в системе. | Целое число |  | Для получения значений используется команда who на стороне Агента. |

Ключ элемента данных виртуальных файловых систем

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| vfs.dev.discovery | | | |
| Список блочных устройств и их типов. Используется низкоуровневым обнаружением. | JSON объект |  | Этот элемент данных поддерживается только на платформе Linux. |
| vfs.dev.read[<устройство>,<тип>,<режим>] | | | |
| Статистика чтения диска. | Целое число при тип равном: sectors, operations, Bytes  Число с плавающей точкой при тип равном: sps, ops, bps  Обратить внимание: если используется интервал обновления три часа и более**2**, будет всегда возвращаться значение "0" | –устройство – дисковое устройство (по умолчанию all **3**); –тип – возможные значения: sectors, operations, Bytes, sps, ops, bps Обратить внимание, что поддержка и умолчания параметра тип зависят от платформы. См. подробную информацию о зависимости от платформы. sps, ops, bps соответствуют: секторам, операциям, байтам в секунду соответственно –режим – возможные значения: avg1 (усреднение за минуту, по умолчанию), avg5 (усреднение за 5 минут), avg15 (усреднение за 15 минут). Третий параметр поддерживается, только если тип один из: sps, ops, bps. | Можно использовать относительные имена устройств (например, sda), а также необязательный префикс /dev/ (например, /dev/sda).  Поддерживаются логические тома LVM.  Значения по умолчанию параметра "тип" для различных ОС: AIX – operations FreeBSD – bps Linux – sps OpenBSD – operations Solaris – Bytes  Пример: => vfs.dev.read[,operations]  ops, bps и sps на поддерживаемых платформах ранее было ограничено 8 устройствами (7 отдельных устройств и одно all). |
| vfs.dev.write[<устройство>,<тип>,<режим>] | | | |
| Статистика записи на диск. | Целое число при тип равном: sectors, operations, Bytes  Число с плавающей точкой при тип равном: sps, ops, bps  Обратить внимание: если используется интервал обновления три часа и более**2**, будет всегда возвращаться значение "0" | –устройство – дисковое устройство (по умолчанию all **3**); –тип – возможные значения: sectors, operations, Bytes, sps, ops, bps Обратить внимание, что поддержка и умолчания параметра тип зависят от платформы.  sps, ops, bps соответствуют: секторам, операциям, байтам в секунду соответственно; –режим – возможные значения: avg1 (усреднение за минуту, по умолчанию), avg5 (усреднение за 5 минут), avg15 (усреднение за 15 минут). Третий параметр поддерживается только если тип один из: sps, ops, bps. | Можно использовать относительные имена устройств (например, sda), а также необязательный префикс /dev/ (например, /dev/sda).  Поддерживаются логические тома LVM.  Значения по умолчанию параметра "тип" для различных ОС: AIX – operations FreeBSD – bps Linux – sps OpenBSD – operations Solaris – Bytes  Пример:  => vfs.dev.write[,operations] ops, bps и sps на поддерживаемых платформах ранее было ограничено 8 устройствами (7 отдельных устройств и одно all). |
| vfs.dir.count[директория,<regex\_вкл>,<regex\_искл>,<типы\_вкл>,<типы\_искл>,<макс\_глубина>,<мин\_размер>,<макс\_размер>,<мин\_возраст>,<макс\_возраст>,<regex\_искл\_директории>] | | | |
| Количество записей в директории. | Целое число | –директория – абсолютный путь к директории; –regex\_вкл – регулярное выражение, описывающее шаблон включаемых имен сущностей (файлов, директорий и символических ссылок); если пусто, включается все (по умолчанию); –regex\_искл – регулярное выражение, описывающее шаблон исключаемых имен сущностей (файлов, директорий и символических ссылок); если пусто (по умолчанию), исключения не применяются; –типы\_вкл – перечисление типов записей в директории, которые будут подсчитываться, возможные значения: file – обычный файл, dir – поддиректория, sym – символическая ссылка, sock – сокет, bdev – блочное устройство, cdev – устройство посимвольного ввода-вывода, fifo- FIFO, dev- синоним к "bdev,cdev", all- все вышеупомянутые типы (по умолчанию), т.е. "file,dir,sym,sock,bdev,cdev,fifo". Несколько типов необходимо разделять запятой и весь список заключать в двойные кавычки; –типы\_искл – перечисление типов записей (см. <типы\_вкл>) в директории, которые не будут подсчитываться. Если какие-либо типы записей указаны как в <типы\_вкл>, так и в <типы\_искл>, записи этих типов в директории не будут подсчитаны; –макс\_глубина – максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) – без ограничения, 0 – не опускаться в поддиректории; –мин\_размер – минимальный размер (в байтах) подсчитываемых файлов. Файлы, которые меньше этого значения, не будут подсчитаны. Можно использовать суффиксы памяти; –макс\_размер – максимальный размер (в байтах) подсчитываемых файлов. Файлы, которые больше этого значения, не будут подсчитаны. Можно использовать суффиксы памяти; –мин\_возраст – минимальный возраст (в секундах) подсчитываемой записи директории. Более свежие записи не будут подсчитаны. Можно использовать суффиксы времени; –макс\_возраст – максимальный возраст (в секундах) подсчитываемой записи директории. Записи этого возраста и более старые (время модификации) не будут подсчитаны. Можно использовать суффиксы времени; –regex\_искл\_директории – регулярное выражение, описывающее шаблон имен исключаемых директорий. Все содержимое директории будет исключено (в отличие от regex\_искл) | Переменные среды, такие как %APP\_HOME%, $HOME и %TEMP%, не поддерживаются.  Псевдо-директории "." и ".." никогда не учитываются.  При обходе директорий символические ссылки никогда используются.  В Windows символические ссылки директорий пропускаются.  Оба параметра regex\_вкл и regex\_искл применяются к файлам и директориям при вычислении общего количества, но игнорируются при выборе поддиректорий для обхода (если regex\_вкл равен "(?i)^.+\.zip$" и макс\_глубина не задана, то будут обойдены все поддиректории, но посчитаны только файлы с типом zip).  Время выполнения ограничено значением времени ожидания в конфигурации Агента. Так как сканирование большой директории может занять более длительное время, чем время ожидания, то никакие данные не возвратятся, и элемент данных будет отмечен как "Не поддерживается". Результат частичного подсчета не возвращается.  При фильтрации по размеру только обычные файлы имеют осмысленные размеры. В Linux и BSD директории также имеют ненулевые размеры (в основном несколько Кб). Устройства имеют нулевые размеры; например, размер /dev/sda1 не отражает размер соответствующего раздела. Поэтому, при использовании параметров <мин\_размер> и <макс\_размер> рекомендуется указать <типы\_вкл> равным "file", чтобы избежать неожиданностей.  Примеры: ⇒ vfs.dir.count[/dev] – мониторинг количества устройств в /dev (Linux) ⇒ vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] – мониторинг количества файлов во временной директории (Windows) |
| vfs.dir.get[директория,<regex\_вкл>,<regex\_искл>,<типы\_вкл>,<типы\_искл>,<макс\_глубина>,<мин\_размер>,<макс\_размер>,<мин\_возраст>,<макс\_возраст>,<regex\_искл\_директории>] | | | |
| Список содержимого директории. | JSON | –директория – абсолютный путь к директории; –regex\_вкл – регулярное выражение, описывающее шаблон включаемых имен сущностей (файлов, директорий и символических ссылок); если пусто, включается все (по умолчанию); –regex\_искл – регулярное выражение, описывающее шаблон исключаемых имен сущностей (файлов, директорий и символических ссылок); если пусто (по умолчанию), исключения не применяются; –типы\_вкл – перечисление типов записей в директории, которые будут добавлены в список, возможные значения: file – обычный файл, dir – поддиректория, sym – символическая ссылка, sock – сокет, bdev – блочное устройство, cdev – устройство посимвольного ввода-вывода, fifo- FIFO, dev- синоним к "bdev,cdev", all- все вышеупомянутые типы (по умолчанию), т.е. "file,dir,sym,sock,bdev,cdev,fifo". Несколько типов необходимо разделять запятой и весь список заключить в двойные кавычки; –типы\_искл – перечисление типов записей (см. <типы\_вкл>) в директории, которые не будут добавлены в список. Если какие-либо типы записей указаны как в <типы\_вкл>, так и в <типы\_искл>, записи этих типов в директории не будут добавлены в список; –макс\_глубина – максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) – без ограничения, 0 – не опускаться в поддиректории; –мин\_размер – минимальный размер (в байтах) добавляемых в список файлов. Файлы, которые меньше этого значения, не будут добавлены в список. Можно использовать суффиксы памяти; –макс\_размер – максимальный размер (в байтах) добавляемых в список файлов. Файлы, которые больше этого значения, не будут добавлены в список. Можно использовать суффиксы памяти; –мин\_возраст – минимальный возраст (в секундах) добавляемой в список записи директории. Более свежие записи не будут добавлены в список. Можно использовать суффиксы времени; –макс\_возраст – максимальный возраст (в секундах) добавляемой в список записи директории. Записи этого возраста и более старые (время модификации) не будут добавлены в список. Можно использовать суффиксы времени; –regex\_искл\_директории – регулярное выражение, описывающее шаблон имен исключаемых директорий. Все содержимое директории будет исключено (в отличие от regex\_искл) | Переменные среды, такие как %APP\_HOME%, $HOME и %TEMP%, не поддерживаются.  Псевдо-директории "." и ".." никогда не учитываются.  При обходе директорий символические ссылки никогда используются.  В Windows символические ссылки директорий пропускаются.  Оба параметра regex\_вкл и regex\_искл применяются к файлам и директориям при генерации списка содержимого директории, но игнорируются при выборе поддиректорий для обхода (если regex\_вкл равен "(?i)^.+\.zip$" и макс\_глубина не задана, то будут обойдены все поддиректории, но в список будут добавлены только файлы с типом zip).  Время выполнения ограничено значением времени ожидания в конфигурации Агента. Так как сканирование большой директории может занять более длительное время, чем время ожидания, то никакие данные не возвратятся, и элемент данных будет отмечен как "Не поддерживается". Частичный список не возвращается. При фильтрации по размеру только обычные файлы имеют осмысленные размеры. В Linux и BSD директории также имеют ненулевые размеры (в основном несколько Кб). Устройства имеют нулевые размеры; например, размер /dev/sda1 не отражает размер соответствующего раздела. Поэтому, при использовании параметров <мин\_размер> и <макс\_размер> рекомендуется указать <типы\_вкл> равным "file", чтобы избежать неожиданностей.  Примеры: ⇒ vfs.dir.get[/dev] – получение списка устройств в /dev (Linux) ⇒ vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] – получение списка файлов во временной директории (Windows) |
| vfs.dir.size[директория,<regex\_вкл>,<regex\_искл>,<режим>,<макс\_глубина>,<regex\_искл\_директории>] | | | |
| Размер директории (в байтах). | Целое число | –директория – абсолютный путь к директории regex\_вкл – регулярное выражение, описывающее шаблон включаемых имен сущностей (файлов, директорий и символических ссылок); если пусто, включается все (по умолчанию) regex\_искл – регулярное выражение, описывающее шаблон исключаемых имен сущностей (файлов, директорий и символических ссылок); если пусто (по умолчанию), исключения не применяются; –режим – возможные значения: apparent (по умолчанию) – получение реальных размеров файлов вместо использования диска (работает как du -sb директория), disk – получение использования диска (работает как du -s -B1 директория). В отличие от команды du, элемент данных vfs.dir.size при вычислении размера директории учитывает скрытые файлы (работает как du -sb .[^.]\* \* в пределах директории). макс\_глубина – максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) – без ограничения, 0 – не опускаться в поддиректории; –regex\_искл\_директории – регулярное выражение, описывающее шаблон имен исключаемых директорий. Все содержимое директории будет исключено (в отличие от regex\_искл) | Подсчитываются только те директории, к которым пользователь zabbix имеет по крайней мере права чтения.  В Windows любая символическая ссылка пропускается, а жесткие ссылки учитываются только один раз.  При наличии больших директорий или медленных дисков этот элемент данных может превысить время ожидания из-за настройки Timeout в файлах конфигурации Агента и Сервера/Прокси. При необходимости увеличить эти значения времени ожидания.  Примеры: ⇒ vfs.dir.size[/tmp,log] – вычисление размеров всех файлов в /tmp, которые содержат в своем имени log ⇒ vfs.dir.size[/tmp,log,^.+\.old$] – вычисление размеров всех файлов в /tmp, которые содержат в своем имени log, исключая файлы с именами, оканчивающимися на ".old"  Ограничение размера файлов зависит от наличия поддержки больших файлов. |
| vfs.file.cksum[файл,<режим>] | | | |
| Контрольная сумма файла, вычисленная по алгоритму, используемому в UNIX cksum. | Целое число – с режимом в значении crc32  Строка – с режимом в значениях md5, sha256 | –файл – абсолютный путь к файлу; –режим – crc32 (по умолчанию), md5, sha256 | Пример: => vfs.file.cksum[/etc/passwd]  Пример возвращаемых значений (для crc32/md5/sha256 соответственно): 675436101 9845acf68b73991eb7fd7ee0ded23c44 ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.contents[файл,<кодировка>] | | | |
| Получение содержимого файла. | Текст | –файл – абсолютный путь к файлу; –кодировка – идентификатор кодовой страницы. | Возвращает пустую строку, если файл пустой или содержит только LF/CR символы. Возвращаемое значение ограничено 16 МБ (включая конечные пробелы, которые усекаются); также применяются ограничения баз данных. Маркер последовательности байтов (BOM) исключен из вывода. Пример: => vfs.file.contents[/etc/passwd] |
| vfs.file.exists[файл,<типы\_вкл>,<типы\_искл>] | | | |
| Проверка существования файла | 0 – файл не найден  1 – файл заданного типа существует. | –файл – абсолютный путь к файлу; –типы\_вкл – список включаемых типов файлов, возможные значения: file (обычный файл, по умолчанию (если типы\_искл не задан)), dir (директория), sym (символическая ссылка), sock (сокет), bdev (блочное устройство), cdev (устройство посимвольного ввода-вывода), fifo (FIFO), dev (синоним к "bdev,cdev"), all (все вышеупомянутые типы, по умолчанию если типы\_искл задан); –типы\_искл – список исключаемых типов файлов, См. типы\_вкл для получения возможных значений (по умолчанию никакие типы не исключаются) | Несколько типов необходимо разделять запятой и весь список заключить в "" (кавычки). На Windows двойные кавычки необходимо экранировать обратной косой чертой "\" и весь ключ элемента данных должен быть заключен в двойные кавычки при использовании утилиты zabbix\_get.exe или agent2 в командной строке.  Если какие-либо типы записей указаны как в <типы\_вкл>, так и в <типы\_искл>, файлы этих типов будут исключены.  Примеры: => vfs.file.exists[/tmp/application.pid] => vfs.file.exists[/tmp/application.pid,"file,dir,sym"] => vfs.file.exists[/tmp/application\_dir,dir] Ограничение размера файла зависит от поддержки больших файлов. Обратить внимание, элемент данных может стать неподдерживаемым на Windows, если искомая директория ищется в несуществующей директории, например, vfs.file.exists[C:\нет\dir,dir] (где "нет" не существует). |
| vfs.file.get[файл] | | | |
| Получение информации о файле. | Объект JSON | –файл – абсолютный путь к файлу | Поддерживаемые типы файлов в UNIX подобных системах: обычный файл, директория, символическая ссылка, сокет, блочное устройство, устройство посимвольного ввода-вывода, FIFO Поддерживаемые типы файлов на Windows: обычный файл, директория, символическая ссылка Например: => vfs.file.get[/etc/passwd] – получение JSON с информацией о файле /etc/passwd (тип, пользователь, права доступа, SID, uid и т.п.) |
| vfs.file.md5sum[файл] | | | |
| MD5 контрольная сумма файла | Строка (MD5 хеш-сумма файла) | –файл – абсолютный путь к файлу | Пример: => vfs.file.md5sum[/usr/local/etc/zabbix\_agentd.conf] Пример возвращаемого значения: b5052decb577e0fffd622d6ddc017e82 Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.owner[файл,<тип\_владельца>,<тип\_результата>] | | | |
| Получение владельца файла. | Строка символов | –файл – абсолютный путь к файлу; –тип\_владельца – user (по умолчанию) или group (только в UNIX); –тип\_результата – name (по умолчанию) или id; в случае с id – получение uid/gid на UNIX, SID на Windows. | Пример: => vfs.file.owner[/tmp/zabbix\_server.log] – получение владельца файла /tmp/zabbix\_server.log; => vfs.file.owner[/tmp/zabbix\_server.log,,id] – получение ID владельца файла /tmp/zabbix\_server.log. Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.permissions[файл] | | | |
| Получение строки из 4 цифр, содержащую восьмеричное число с правами доступа UNIX. | Строка | –файл – абсолютный путь к файлу | Не поддерживается на Windows. Пример: => vfs.file.permissions[/etc/passwd] – получение прав доступа к /etc/passwd, например, "0644" Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.regexp[файл,регулярное выражение,<кодировка>,<начальная строка>,<конечная строка>,<вывод>] | | | |
| Поиск строки в файле. | Строка, содержащая совпадающую подстроку, или то, что определено в дополнительном параметре вывод. | –файл – абсолютный путь к файлу; –регулярное выражение – регулярное выражение, описывающее требуемый шаблон; –кодировка – идентификатор кодовой страницы; –начальная строка – номер первой строки для поиска (по умолчанию, первая строка файла); –конечная строка – номер последней строки для поиска (по умолчанию, последняя строка файла); –вывод – дополнительный шаблон форматирования вывода; –\0 – управляющая последовательность заменяется найденной частью текста (начиная от первого символа совпадения, заканчивая символом, где совпадение заканчивается), то \N (где **N=1..9**) – управляющая последовательность заменяется N-ной совпадающей группой (или пустой строкой, если N превышает количество найденных групп). | Возвращается только первая совпавшая строка. Если совпадений с выражением не найдено, будет возвращена пустая строка.  Маркер последовательности байтов (BOM) исключается из вывода.  Извлечение содержимого при использовании параметра вывод выполняется Агентом.  Параметры начальная строка, конечная строка и вывод.   Примеры: => vfs.file.regexp[/etc/passwd,zabbix] => vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1] => vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] – получение ID пользователя zabbix  Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.regmatch[файл,регулярное выражение,<кодировка>,<начальная строка>,<конечная строка>] | | | |
| Поиск строки в файле. | 0 – совпадение не найдено  1 – найдено | –файл – абсолютный путь к файлу; –регулярное выражение – регулярное выражение, описывающее требуемый шаблон; –кодировка – идентификатор кодовой страницы; –начальная строка – номер первой строки для поиска (по умолчанию, первая строка файла); –конечная строка – номер последней строки для поиска (по умолчанию, последняя строка файла). | Маркер последовательности байтов (BOM) игнорируется.  Параметры начальная строка, конечная строка .  Например: => vfs.file.regmatch[/var/log/app.log,error]  Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.size[файл,<режим>] | | | |
| Размер файла (в байтах). | Целое число | –файл – абсолютный путь к файлу; –режим – возможные значения: Bytes (по умолчанию) или lines (пустые строки также считаются) | Пользователь zabbix должен иметь права на чтение файла  Например: => vfs.file.size[/var/log/syslog]  Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.file.time[файл,<режим>] | | | |
| Информация о времени файла. | Целое число (UNIX timestamp) | –файл – полный путь к файлу; –режим – возможные значения: modify (по умолчанию) – время последней модификации содержимого файла, access – время последнего чтения файла, change – время последнего изменения свойств файла. | Пример: => vfs.file.time[/etc/passwd,modify]  Ограничение размера файла зависит от поддержки больших файлов. |
| vfs.fs.discovery | | | |
| Список примонтированных файловых систем и их типы. Используется низкоуровневым обнаружением. | Объект JSON |  |  |
| vfs.fs.get | | | |
| Список примонтированных файловых систем, их типов, размеров дисков и статистики по inode. Можно использовать низкоуровневым обнаружением. | Объект JSON |  | Этот элемент данных может сообщать о файловых системах с нулевым количеством файловых дескрипторов (inode), что может случаться на файловых системах с динамическим количеством файловых дескрипторов (например, btrfs). |
| vfs.fs.inode[fs,<режим>] | | | |
| Количество или процент inodes. | Целое число – для количества  Число с плавающей точной – для процентов | **­**–fs – файловая система; –режим – возможные значения: total (по умолчанию), free, used, pfree (свободно, в процентах), pused (использовано, в процентах). | Этот элемент данных не будет переходить в неподдерживаемое состояние в режимах pfree/pused, если количество inode станет нулевым, что может случаться на файловых системах с динамическим количеством файловых дескрипторов (например, btrfs). Вместо этого для таких файловых систем значения pfree/pused будут сообщаться как "100" и "0" соответственно. Пример: => vfs.fs.inode[/,pfree] |
| vfs.fs.size[fs,<режим>] | | | |
| Размер диска в байтах или процентах от общего размера. | Целое число – для байтов  Число с плавающей точкой – для процентов. | **–fs** – файловая система; –режим – возможные значения: total (по умолчанию), free, used, pfree (доступно, в процентах), pused (использовано, в процентах). | Если файловая система не примонтирована, будет возвращен размер локальной файловой системы, где расположена точка монтирования. Пример: => vfs.fs.size[/tmp,free] Зарезервированное место на файловой системе принимается во внимание и не включено при использовании режима free. |

Ключ элемента данных виртуальной памяти

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| vm.memory.size[<режим>] | | | |
| Размер памяти в байтах или в процентах от общего количества. | Целое число – для байтов  Число с плавающей точкой – для процентов. | **–режим** – возможные значения: total (по умолчанию), active, anon, buffers, cached, exec, file, free, inactive, pinned, shared, slab, wired, used, pused (использовано, в процентах), available, pavailable (доступно, в процентах) | Этот элемент данных принимает три категории параметров: 1) total – общее количество памяти. 2) зависимые от платформы типы памяти: active, anon, buffers, cached, exec, file, free, inactive, pinned, shared, slab, wired. 3) оценка на уровне пользователя как много памяти используется и доступно: used, pused, available, pavailable. |

Ключи элемента данные веб-мониторинга

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| web.page.get[хост,<путь>,<порт>] | | | |
| Получение содержимого веб-страницы. | Исходная веб-страница в виде текста (включая заголовки) | –хост – имя хоста или URL (в виде схема://хост:порт/путь, где обязателен только хост). Разрешенные схемы URL: http, https4. При отсутствии схемы будет обрабатываться как http. Если указан URL, то путь и порт должны быть пустыми. При подключении к Серверам, которые требуют аутентификацию, возможность указать имя пользователя / пароль, например: http://user:password@www.example.com, имеется только с поддержкой cURL 4. В именах хостов поддерживается [punycode](https://ru.wikipedia.org/wiki/Punycode); –путь – путь к HTML документу (по умолчанию /); –порт – номер порта (по умолчанию 80 для HTTP) | Этот элемент данных становится неподдерживаемым, если заданный ресурс в хост не существует или недоступен. хост может быть именем хоста, доменным именем, адресом IPv4 или IPv6. Но для поддержки IPv6-адресов Агент должен быть скомпилирован с включенной поддержкой IPv6. |
| web.page.perf[хост,<путь>,<порт>] | | | |
| Время полной загрузки веб-страницы (в секундах). | Число с плавающей точкой | –хост – имя хоста или URL (в виде схема://хост:порт/путь, где обязателен только хост). Разрешенные схемы URL: http, https4. При отсутствии схемы будет обрабатываться как http. Если указан URL, то путь и порт должны быть пустыми. При подключении к Серверам, которые требуют аутентификацию, возможность указать имя пользователя / пароль, например: http://user:password@www.example.com, имеется только с поддержкой cURL 4. В именах хостов поддерживается [punycode](https://ru.wikipedia.org/wiki/Punycode); –путь – путь к HTML документу (по умолчанию /); –порт – номер порта (по умолчанию 80 для HTTP). | Этот элемент данных становится неподдерживаемым, если заданный ресурс в хост не существует или недоступен.  хост может быть именем хоста, доменным именем, адресом IPv4 или IPv6. Но для поддержки IPv6-адресов Агент должен быть скомпилирован с включенной поддержкой IPv6. |
| web.page.regexp[хост,<путь>,<порт>,регулярное выражение,<длина>,<вывод>] | | | |
| Поиск строки на веб-странице. | Совпадающая строка, или указанная опциональным параметром вывод | –хост – имя хоста или URL (в виде схема://хост:порт/путь, где обязателен только хост). Разрешенные схемы URL: http, https4. При отсутствии схемы будет обрабатываться как http. Если указан URL, то путь и порт должны быть пустыми. При подключении к Серверам, которые требуют аутентификацию, возможность указать имя пользователя / пароль, например: http://user:password@www.example.com, имеется только с поддержкой cURL 4. В именах хостов поддерживается [punycode](https://ru.wikipedia.org/wiki/Punycode); –путь – путь к HTML документу (по умолчанию /); –порт – номер порта (по умолчанию 80 для HTTP); –регулярное выражение – регулярное выражение, описывающее требуемый шаблон; –длина – максимальное количество возвращаемых символов; –вывод – дополнительный шаблон форматирования вывода. \0 – управляющая последовательность заменяется найденной частью текста (начиная от первого символа совпадения, заканчивая символом, где совпадение заканчивается), при этом \N (где N=1..9) – управляющая последовательность заменяется N-ой совпадающей группой (или пустой строкой, если N превышает количество найденных групп). | Этот элемент данных становится неподдерживаемым, если заданный ресурс в хост не существует или недоступен.  хост может быть именем хоста, доменным именем, адресом IPv4 или IPv6. Но для поддержки IPv6-адресов Агент должен быть скомпилирован с включенной поддержкой IPv6.  Извлечение содержимого при использовании параметра вывод производится на стороне Агента. |

1. Специфичные ключи элементов данных для Агента-2

Агент-2 поддерживает все ключи элементов данных, которые поддерживаются Агентом в UNIX и Windows. В таблицах 59-69 представлена подробная информация о дополнительных ключах элементов данных, которые можно использовать только с Агентом-2; эти ключи сгруппированы по плагинам, которым они принадлежат.

Параметры без угловых скобок обязательны. Параметры, обозначенные угловыми скобками, "< >" опциональны.

Ключ Ceph

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| Ceph.df.details [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Использование данных кластера и распределение между пулами. | Объект JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |
| Ceph.osd.stats [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Агрегированная статистика и статистика по каждому OSD. | Объект JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |
| Ceph.osd.discovery [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Список обнаруженных OSD. Используется низкоуровневым обнаружением. | Объект JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |
| Ceph.osd.dump [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Пороги использования и состояния OSD. | Объект JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |
| Ceph.ping [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Проверка, можно ли установить подключение к Ceph. | 0 – подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками) 1 – подключение успешно. | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |
| Ceph.pool.discovery [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Список обнаруженных пулов. Используется низкоуровневым обнаружением. | Объект JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |
| Ceph.status [строка\_подкл, <пользователь>, <ключ\_api>] | | | |
| Общее состояние кластера. | Объект JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – Учетные данные для входа в Ceph. |  |

Ключи Docker

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| docker.container\_info [<ID>,<info>] | | | |
| Низкоуровневая информация о контейнере. | Вывод API вызова [ContainerInspect](https://docs.docker.com/engine/api/v1.28/#operation/ContainerInspect), преобразованный в JSON | ID – ID или имя контейнера info – объем возвращаемой информации. Возможные значения: short (по умолчанию) или full. | Agent2 пользователь (zabbix) должен быть добавлен в [группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) "docker" для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.container\_stats [<ID>] | | | |
| Статистика использования ресурсов контейнером. | Вывод API вызова [ContainerStats](https://docs.docker.com/engine/api/v1.28/#operation/ContainerStats) и использование CPU в процентах, преобразованные в JSON | ID – ID или имя контейнера | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.containers | | | |
| Список контейнеров. | Вывод API запроса [ContainerList](https://docs.docker.com/engine/api/v1.28/#operation/ContainerList), преобразованный в JSON | - | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.containers.discovery[<опции>] | | | |
| Список контейнеров. Используется низкоуровневым обнаружением. | Объект JSON | опции – задает, следует ли обнаруживать все или только запущенные контейнеры. Поддерживаемые значения: true – возвращать все контейнеры; false – возвращать только запущенные контейнеры (по умолчанию). | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.data\_usage | | | |
| Информация о текущем использовании данных. | Вывод API запроса [SystemDataUsage](https://docs.docker.com/engine/api/v1.28/#operation/SystemDataUsage), преобразованный в JSON | - | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.images | | | |
| Список образов. | Вывод API запроса [ImageList](https://docs.docker.com/engine/api/v1.28/#operation/ImageList), преобразованный в JSON | - | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.images.discovery | | | |
| Список образов. Используется низкоуровневым обнаружением. | Объект JSON | - | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.info | | | |
| Информация о системе. | Вывод API вызова [SystemInfo](https://docs.docker.com/engine/api/v1.28/#operation/SystemInfo), преобразованный в JSON | - | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |
| docker.ping | | | |
| Проверка, активен или нет Docker демон. | 1 – подключение установлено  0 – подключение разорвано | - | Agent2 пользователь (zabbix) должен быть добавлен в "docker"-[группу](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) для достаточных привилегий. Иначе проверка завершится с ошибкой. |

Ключи Memcached

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| memcached.ping[строка\_подкл,<пользователь>,<пароль>] | | | |
| Проверка можно ли установить подключение. | 1 – подключение активно  0 – подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками) | строка\_подкл – URI или имя сессии. |  |
| memcached.stats[строка\_подкл,<пользователь>,<пароль>,<тип>] | | | |
| Получение вывода команды STATS. | JSON – вывод, преобразованный в JSON | строка\_подкл – URI или имя сессии. пользователь, пароль – учетные данные для входа в Memchached. тип – тип возвращаемой статистики: items, sizes, slabs или settings (по умолчанию пусто, возвращается общая статистика). |  |

Ключи MongoDB

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| mongodb.collection.stats[строка\_подкл,<имя\_пользователя>,<пароль>,<база\_данных>,коллекция] | | | |
| Получение различной статистики хранилища по заданной коллекции. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. база\_данных – имя базы данных (по умолчанию: admin). коллекции — имя коллекции. |  |
| mongodb.collections.discovery[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение списка обнаруженных коллекций. Используется низкоуровневым обнаружением. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.collections.usage[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение статистики использования по коллекциям. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.connpool.stats[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение информации касательно открытых исходящих подключений с текущего экземпляра базы данных к другим членам кластера сегментов или набора реплик. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.db.stats[строка\_подкл,<имя\_пользователя>,<пароль>,<база\_данных>] | | | |
| Получение статистики, отражающей состояние системы заданной базы. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. база\_данных – имя базы (по умолчанию: admin). |  |
| mongodb.db.discovery[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение списка обнаруженных баз данных. Используется низкоуровневым обнаружением. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.jumbo\_chunks.count[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение количества jumbo chunk. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.oplog.stats[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение состояния набора реплик, используя данные, полученные от oplog. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.ping[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Проверка, можно ли установить подключение. | 1 – подключение успешно 0 – подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками). | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.rs.config[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение текущей конфигурации набора реплик. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.rs.status[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение состояния набора реплик с точки видимости члена, откуда вызывается метод. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.server.status[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение состояния базы данных. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.sh.discovery[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение списка обнаруженных сегментов, находящихся в кластере. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |
| mongodb.version[строка\_подкл,<имя\_пользователя>,<пароль>] | | | |
| Получение версии сервера базы данных. | String | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные к MongoDB. |  |

Ключи MQTT

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| mqtt.get[<url\_ брокера>,тема,<имя\_пользователя>,<пароль>] | | | |
| Подписка на заданную тему или темы (с шаблонами), которые предоставляются брокером, и ожидание публикаций. | Зависит от содержимого темы.  Если использовался шаблон, возвращает содержимое темы в виде JSON. | url\_брокера – URL брокера MQTT (если пустой, используется localhost с портом 1883). тема – тема MQTT (обязательно). Поддерживаются шаблоны (+,#). имя\_пользователя, пароль – учетные данные для аутентификации (если требуется) | Элемент данных необходимо настроить активной проверкой (тип элемента данных "Агент (активный)"). Можно использовать сертификаты TLS-шифрования, сохранив их в место по умолчанию (например, в Ubuntu директория /etc/ssl/certs/). Для TLS использовать схему tls://. |

Ключ MySQL

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| mysql.custom.query[строка\_подкл,<имя\_пользователя>,<пароль>,имяЗапроса,<аргументы...>] | | | |
| Возвращает результат пользовательского запроса. | Объект JSON | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. имяЗапроса – имя пользовательского запроса, должно совпадать с именем SQL-файла без расширения. аргументы – один или несколько перечисленных через запятую аргументов, которые будут переданы запросу. |  |
| mysql.db.discovery[строка\_подкл, <имя\_пользователя>, <пароль>] | | | |
| Список баз данных MySQL. Используется низкоуровневым обнаружением. | Результат SQL запроса "show databases" в формате LLD JSON. | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. |  |
| mysql.db.size[строка\_подкл, <имя\_пользователя>, <пароль>,имя\_базы] | | | |
| Размер базы данных в байтах. | Результат SQL запроса "select coalesce(sum(data\_length + index\_length),0) as size from information\_schema.tables where table\_schema=?" для указанной базы данных в байтах. | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. имя\_базы – имя базы данных. |  |
| mysql.get\_status\_variables[строка\_подкл, <имя\_пользователя>, <пароль>] | | | |
| Значения переменных глобального состояния. | Результат SQL запроса "show global status" в формате JSON. | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. |  |
| mysql.ping[строка\_подкл, <имя\_пользователя>, <пароль>] | | | |
| Проверка, можно ли установить подключение. | 1 – подключение успешно  0 – подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками). | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. |  |
| mysql.replication.discovery[строка\_подкл, <имя\_пользователя>, <пароль>] | | | |
| Список репликаций MySQL. Используется низкоуровневым обнаружением. | Результат SQL запроса "show slave status" в формате LLD JSON. | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. |  |
| mysql.replication.get\_slave\_status[строка\_подкл, <имя\_пользователя>, <пароль>, <хост\_мастера>] | | | |
| Состояние репликации. | Результат SQL запроса "show slave status" в формате JSON. | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. хост\_мастера – Имя хоста мастера репликации. Если не найдено, возвращается ошибка. Если данный параметр не указан, возвращаются все хосты. |  |
| mysql.version[строка\_подкл, <имя\_пользователя>, <пароль>] | | | |
| Версия MySQL. | Строка с версией MySQL экземпляра. | строка\_подкл – URI или имя сессии. имя\_пользователя, пароль – учетные данные для входа в MySQL. |  |

Ключи PostgreSQL

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| pgsql.autovacuum.count[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Количество autovacuum worker. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.archive[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Информация об архивных файлах. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.archive.count\_archived\_files – количество успешно заархивированных WAL-файлов. pgsql.archive.failed\_trying\_to\_archive – количество неудачных попыток архивирования WAL-файлов. pgsql.archive.count\_files\_to\_archive – количество файлов для архивирования. pgsql.archive.size\_files\_to\_archive – размер файлов для архивирования. |
| pgsql.bgwriter[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Общее количество контрольных точек (checkpoint) по кластеру базы данных, с разбивкой по типу контрольной точки. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.bgwriter.buffers\_alloc – количество выделенных буферов. pgsql.bgwriter.buffers\_backend – количество буферов, записанных самим серверным процессом. pgsql.bgwriter.maxwritten\_clean – сколько раз фоновый процесс записи останавливал сброс грязных страниц на диск из-за того, что записал слишком много буферов. pgsql.bgwriter.buffers\_backend\_fsync – сколько раз серверному процессу пришлось выполнить fsync самостоятельно, вместо фонового процесса. pgsql.bgwriter.buffers\_clean – количество буферов, записанных фоновым процессом записи. pgsql.bgwriter.buffers\_checkpoint – количество буферов, записанных при выполнении контрольных точек. pgsql.bgwriter.checkpoints\_timed – количество запланированных контрольных точек, которые уже были выполнены. pgsql.bgwriter.checkpoints\_req – количество запрошенных контрольных точек, которые уже были выполнены. pgsql.bgwriter.checkpoint\_write\_time – общее время, которое было затрачено на этап обработки контрольной точки, в котором файлы записываются на диск, в миллисекундах. pgsql.bgwriter.sync\_time – общее время, которое было затрачено на этап обработки контрольной точки, в котором файлы синхронизируются с диском, в миллисекундах. |
| pgsql.cache.hit[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Коэффициент успешности кэша буфера PostgreSQL. | Число с плавающей точкой | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.connections[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Подключения по типу. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.connections.active – серверный процесс выполняет запрос. pgsql.connections.fastpath\_function\_call – серверный процесс выполняет fast-path функцию. pgsql.connections.idle – серверный процесс ожидает новой команды от клиента. pgsql.connections.idle\_in\_transaction – серверный процесс находится внутри транзакции, но в настоящее время не выполняет никакой запрос. pgsql.connections.prepared – общее количество prepared подключений. pgsql.connections.total – общее количество подключений. pgsql.connections.total\_pct – процентное отношение общего количества подключений к "max\_connections" настройке сервера PostgreSQL. pgsql.connections.waiting – количество подключений в состоянии в запросе. pgsql.connections.idle\_in\_transaction\_aborted – это состояние подобно idle in transaction, за исключением того, что один из операторов в транзакции вызывал ошибку. |
| pgsql.custom.query[uri,<имя\_пользователя>,<пароль>,имя\_запроса,<аргументы...>] | | | |
| Получение результата пользовательского запроса. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_запроса – имя пользовательского запроса, должно совпадать с именем sql- файла без расширения. аргументы – один или несколько аргументов (через запятую), которые будут переданы запросу. |  |
| pgsql.dbstat[uri,<имя\_пользователя>,<пароль>, имя\_бд] | | | |
| Сбор статистики по каждой базе данных. Используется низкоуровневым обнаружением. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.dbstat.numbackends["{#DBNAME}"] – количество обслуживающих процессов, в настоящее время подключенных к этой базе данных, либо NULL для общих объектов. pgsql.dbstat.sum.blk\_read\_time["{#DBNAME}"] – время, которое затратили обслуживающие процессы в этой базе на чтение блоков из файлов данных, в миллисекундах. pgsql.dbstat.sum.blk\_write\_time["{#DBNAME}"] – время, которое затратили обслуживающие процессы в этой базе на запись блоков в файлы данных, в миллисекундах. pgsql.dbstat.sum.checksum\_failures["{#DBNAME}"] – количество ошибок контрольных сумм в страницах данных этой базы (или общего объекта) либо NULL, если контрольные суммы не проверяются.(только PostgreSQL версии 12) pgsql.dbstat.blks\_read.rate["{#DBNAME}"] – количество прочитанных дисковых блоков в этой базе данных. pgsql.dbstat.deadlocks.rate["{#DBNAME}"] – количество взаимных блокировок, зафиксированное в этой базе данных. pgsql.dbstat.blks\_hit.rate["{#DBNAME}"] – сколько раз дисковые блоки обнаруживались в буферном кэше, так что чтение с диска не потребовалось (в значение входят только случаи обнаружения в буферном кэше PostgreSQL Pro, а не в кэше файловой системы ОС). pgsql.dbstat.xact\_rollback.rate["{#DBNAME}"] – количество транзакций в этой базе данных, для которых был выполнен откат транзакции. pgsql.dbstat.xact\_commit.rate["{#DBNAME}"] – количество зафиксированных транзакций в этой базе данных. pgsql.dbstat.tup\_updated.rate["{#DBNAME}"] – количество строк, измененных запросами в этой базе данных. pgsql.dbstat.tup\_returned.rate["{#DBNAME}"] – количество строк, возвращенных запросами в этой базе данных. pgsql.dbstat.tup\_inserted.rate["{#DBNAME}"] – количество строк, вставленных запросами в этой базе данных. pgsql.dbstat.tup\_fetched.rate["{#DBNAME}"] – количество строк, извлеченных запросами в этой базе данных. pgsql.dbstat.tup\_deleted.rate["{#DBNAME}"] – количество строк, удаленных запросами в этой базе данных. pgsql.dbstat.conflicts.rate["{#DBNAME}"] – количество запросов, отмененных из-за конфликта с восстановлением в этой базе данных (конфликты происходят только на ведомых серверах). pgsql.dbstat.temp\_files.rate["{#DBNAME}"] – количество временных файлов, созданных запросами в этой базе данных. Подсчитываются все временные файлы независимо от причины их создания (например, для сортировки или для хеширования) и независимо от установленного значения log\_temp"\_files. pgsql.dbstat.temp\_Bytes.rate["{#DBNAME}"] – общий объем данных, записанных во временные файлы запросами в этой базе данных. Учитываются все временные файлы, вне зависимости от того, по какой причине они созданы и вне зависимости от значения log\_temp\_files. |
| pgsql.dbstat.sum[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Обобщенные данные по всем базам данных в кластере. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.dbstat.numbackends – количество обслуживающих процессов, в настоящее время подключенных к этой базе данных. pgsql.dbstat.sum.blk\_read\_time – время, которое затратили обслуживающие процессы в этой базе на чтение блоков из файлов данных, в миллисекундах. pgsql.dbstat.sum.blk\_write\_time – время, которое затратили обслуживающие процессы в этой базе на запись блоков в файлы данных, в миллисекундах. pgsql.dbstat.sum.checksum\_failures – количество ошибок контрольных сумм в страницах данных этой базы (или общего объекта) либо NULL, если контрольные суммы не проверяются (только PostgreSQL версия 12). pgsql.dbstat.sum.xact\_commit – количество зафиксированных транзакций в этой базе данных. pgsql.dbstat.sum.conflicts – количество запросов, отмененных из-за конфликта с восстановлением на ведомых серверах в этой базе данных. pgsql.dbstat.sum.deadlocks – количество взаимных блокировок, зафиксированное в этой базе данных. pgsql.dbstat.sum.blks\_read – количество прочитанных дисковых блоков в этой базе данных. pgsql.dbstat.sum.blks\_hit – сколько раз дисковые блоки обнаруживались в буферном кэше, так что чтение с диска не потребовалось (в значение входят только случаи обнаружения в буферном кэше PostgreSQL Pro, а не в кэше файловой системы ОС). pgsql.dbstat.sum.temp\_Bytes – общий объем данных, записанных во временные файлы запросами в этой базе данных. Учитываются все временные файлы, вне зависимости от того, по какой причине они созданы и вне зависимости от значения log\_temp\_files. pgsql.dbstat.sum.temp\_files – количество временных файлов, созданных запросами в этой базе данных. Подсчитываются все временные файлы независимо от причины их создания (например, для сортировки или для хеширования) и независимо от установленного значения log\_temp\_files. pgsql.dbstat.sum.xact\_rollback – количество транзакций в этой базе данных, для которых был выполнен откат транзакции. pgsql.dbstat.sum.tup\_deleted – количество строк, удаленных запросами в этой базе данных. pgsql.dbstat.sum.tup\_fetched – количество строк, извлеченных запросами в этой базе данных. pgsql.dbstat.sum.tup\_inserted – количество строк, вставленных запросами в этой базе данных. pgsql.dbstat.sum.tup\_returned – количество строк, возвращенных запросами в этой базе данных. pgsql.dbstat.sum.tup\_updated – количество строк, измененных запросами в этой базе данных. |
| pgsql.db.age[uri,<имя\_пользователя>,<пароль>, имя\_бд] | | | |
| Возраст самого старого FrozenXID в базе данных. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.db.bloating\_tables[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Количество "распухших" (bloating) таблиц по каждой базе данных. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.db.discovery[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Список баз данных PostgreSQL. Используется низкоуровневым обнаружением. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.db.size[uri,<имя\_пользователя>,<пароль>, имя\_бд] | | | |
| Размер базы данных в байтах. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.locks[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Информация о предоставленных блокировках по каждой базе данных. Используется низкоуровневым обнаружением. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.locks.shareupdateexclusive["{#DBNAME}"] – количество share update exclusive блокировок. pgsql.locks.accessexclusive["{#DBNAME}"] – количество access exclusive блокировок. pgsql.locks.accessshare["{#DBNAME}"] – количество access share блокировок. pgsql.locks.exclusive["{#DBNAME}"] – количество exclusive блокировок. pgsql.locks.rowexclusive["{#DBNAME}"] – количество row exclusive блокировок. pgsql.locks.rowshare["{#DBNAME}"] – количество row share блокировок. pgsql.locks.share["{#DBNAME}"] – количество shared блокировок. pgsql.locks.sharerowexclusive["{#DBNAME}"] – количество share row exclusive блокировок. |
| pgsql.oldest.xid[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Возраст самого старого XID. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.ping[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Проверка, можно ли установить подключение. | 1 – подключение успешно  0 – подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками). | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.queries[uri,<имя\_пользователя>,<пароль>, <имя\_бд>,период\_времени] | | | |
| Измерение времени выполнения запросов. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. период\_времени – ограничение времени выполнения для подсчета медленных запросов (должно быть целым положительным числом). | Полученные данные обрабатываются зависимыми элементами данных: pgsql.queries.mro.time\_max["{#DBNAME}"] – максимальное время запросов обслуживания. pgsql.queries.query.time\_max["{#DBNAME}"] – максимальное время выполнения запросов. pgsql.queries.tx.time\_max["{#DBNAME}"] – максимальное время запросов транзакций. pgsql.queries.mro.slow\_count["{#DBNAME}"] – количество медленных запросов обслуживания. pgsql.queries.query.slow\_count["{#DBNAME}"] – количество медленных запросов. pgsql.queries.tx.slow\_count["{#DBNAME}"] – количество медленных запросов транзакций. pgsql.queries.mro.time\_sum["{#DBNAME}"] – суммарное время запросов обслуживания. pgsql.queries.query.time\_sum["{#DBNAME}"] – суммарное время выполнения запросов. pgsql.queries.tx.time\_sum["{#DBNAME}"] – суммарное время запросов транзакций. |
| pgsql.replication.count[uri,<имя\_пользователя>,<пароль>] | | | |
| Количество серверов в режиме ожидания. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.replication.process[uri,<имя\_пользователя>,<пароль>] | | | |
| Отставание сброса, отставания записи и отставания replay по каждому процессу sender. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.replication.process.discovery[uri,<имя\_пользователя>,<пароль>] | | | |
| Обнаружение имени процесса репликации. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.replication.recovery\_role[uri,<имя\_пользователя>,<пароль>] | | | |
| Состояние восстановления. | 0 – режим мастера 1 – восстановление еще в процессе (режим ожидания) | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.replication.status[uri,<имя\_пользователя>,<пароль>] | | | |
| Состояние репликации. | 0 – передача потока не работает 1 – передача потока работает 2 – режим мастера | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.replication\_lag.b[uri,<имя\_пользователя>,<пароль>] | | | |
| Отставание репликации в байтах. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.replication\_lag.sec[uri,<имя\_пользователя>,<пароль>] | | | |
| Отставание репликации в секундах. | Целое число | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. |  |
| pgsql.uptime[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Время работы PostgreSQL в миллисекундах. | Число с плавающей точкой | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.version[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Вернуть версию PostgreSQL. | Строка | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. |  |
| pgsql.wal.stat[uri,<имя\_пользователя>,<пароль>, <имя\_бд>] | | | |
| Статистика WAL. | Объект JSON | uri – URI или имя сессии. имя\_пользователя, пароль – учетные данные PostgreSQL. имя\_бд – Имя базы данных. | Полученные данные обрабатываются зависимыми элементами данных: pgsql.wal.count — количество WAL файлов. pgsql.wal.write – использовано WAL lsn (в байтах). |

Ключи Redis

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| redis.config[строка\_подкл,<пароль>,<шаблон>] | | | |
| Получение параметров конфигурации экземпляра Redis, которые соответствуют шаблону | JSON – если использовался шаблон в стиле glob  одно значение – если шаблон не содержит какого-либо символа подстановки | строка\_подкл – URI или имя сессии. пароль – пароль к Redis. шаблон – шаблон в стиле glob (по умолчанию \*). |  |
| redis.info[строка\_подкл,<пароль>,<раздел>] | | | |
| Получение вывода команды INFO. | JSON – вывод преобразован в JSON | строка\_подкл – URI или имя сессии. пароль – пароль к Redis. раздел – раздел информации (по умолчанию default). |  |
| redis.ping[строка\_подкл,<пароль>] | | | |
| Проверка, можно ли установить подключение. | 1 – подключение успешно 0 – подключение разорвано (если имеется какая-либо ошибка, включая проблемы с аутентификацией или проблемы с настройками) | строка\_подкл – URI или имя сессии. пароль – пароль к Redis. |  |
| redis.slowlog.count[строка\_подкл,<пароль>] | | | |
| Количество медленных записей в журнале с момента запуска Redis. | Целое число | строка\_подкл – URI или имя сессии. пароль – пароль к Redis. |  |

Ключи S.M.A.R.T.

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| smart.attribute.discovery | | | |
| Получение списка S.M.A.R.T. параметров устройств. | Объект JSON |  | Возвращаются следующие макросы и их значения: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH}. Поддерживаются типы дисков HDD, SSD и NVME. Диски могут быть одиночными или объединенными в RAID. {#NAME} будет иметь дополнение в случае RAID, например: {"{#NAME}": "/dev/sda cciss,2"} |
| smart.disk.discovery | | | |
| Получение списка S.M.A.R.T. устройств. | Объект JSON |  | Возвращаются следующие макросы и их значения: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE}. Поддерживаются типы дисков HDD, SSD и NVME. Если диск не объединен в RAID, {#RAIDTYPE} будет пустым. {#NAME} будет иметь дополнение в случае RAID, например: {"{#NAME}": "/dev/sda cciss,2"} |
| smart.disk.get[<путь>,<тип\_raid>] | | | |
| Получение всех доступных свойств S.M.A.R.T. устройств. | Объект JSON | путь – путь к диску, макрос {#PATH} можно использовать значением тип\_raid – тип RAID, макрос {#RAID} можно использовать значением. | Поддерживаются типы дисков HDD, SSD и NVME. Диски могут быть одиночными или объединенными в RAID. Данные включают версию smartctl и аргументы вызова, а также дополнительные поля: disk\_name – содержит имя с требуемым дополнением для обнаружения RAID, например: {"disk\_name": "/dev/sda cciss,2"} disk\_type – содержит тип диска HDD, SSD или NVME, например: {"disk\_type": "ssd"}) Если параметры не указаны, элемент данных вернет информацию обо всех дисках. |

Ключи Systemd

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| systemd.unit.get[имя модуля,<интерфейс>] | | | |
| Получение всех свойств модуля systemd. | Объект JSON | имя модуля – имя модуля (возможно потребуется использовать {#UNIT.NAME} макрос в прототипе элементов данных для обнаружения имени) интерфейс – тип интерфейса модуля, возможные значения: Unit (по умолчанию), Service, Socket, Device, Mount, Automount, Swap, Target, Path | Этот элемент данных поддерживается только на Linux-платформе.  LoadState, ActiveState и UnitFileState для интерфейса модуля возвращаются в виде текста и целого числа: "ActiveState":{"state":1,"text":"active"} |
| systemd.unit.info[имя модуля,<свойство>,<интерфейс>] | | | |
| Информация о systemd модуле. | Строка | unit name – имя модуля (возможно потребуется использовать {#UNIT.NAME} макрос в прототипе элементов данных для обнаружения имени) свойство – имя свойства (например, ActiveState (по умолчанию), LoadState, Description) интерфейс – тип интерфейса (например, Unit (по умолчанию), Socket, Service) | Этот элемент данных позволяет получать определенные свойства с заданного типа интерфейса, которые описаны в [dbus API](https://www.freedesktop.org/wiki/Software/systemd/dbus/). Этот элемент данных поддерживается только на Linux-платформе. |
| systemd.unit.discovery[<тип>] | | | |
| Список модулей systemd и их деталей. Используется низкоуровневым обнаружением. | Объект JSON | тип – возможные значения: all, automount, device, mount, path, service (по умолчанию), socket, swap, target | Этот элемент данных поддерживается только на Linux-платформе. |

Ключ веб-сертификата

| Описание | Возвращаемое значение | Параметры | Описание |
| --- | --- | --- | --- |
| web.certificate.get[имя\_хоста,<порт>,<адрес>] | | | |
| Проверка достоверности сертификатов и получение деталей сертификатов. | Объект JSON | имя\_хоста – может быть как IP, так и DNS. Может содержать схему URL (только https), путь (будет проигнорирован) и порт. Если порт указывается в первом и втором параметрах, значения портов должны совпадать. Если указан адрес ( 3-ий параметр), имя хоста используется только для удостоверения SNI и имени хоста. порт – номер порта (по умолчанию 443 для HTTPS). адрес – может быть как IP, так и DNS. Если указан, будет использоваться для подключения и имя хоста (1-ый параметр) будет использоваться для удостоверения SNI и хоста. В случае, если 1-ый параметр IP-адрес и третий параметр DNS, то 1-ый параметр будет использоваться для подключения и третий параметр будет использоваться для удостоверения SNI и хоста. | Этот элемент данных становится неподдерживаемым, если ресурс, указанный в имя\_хоста не существует или, если TLS-квитирование завершится с любой ошибкой, за исключением ошибочного сертификата.  В настоящее время расширение AIA (Authority Information Access) X.509, CRL и OCSP (включая OCSP сшивание), Certificate Transparency и пользовательские хранилища CA не поддерживаются. |

1. Специфичные ключи элементов данных для Windows

В таблице 70 приводится подробная информация о ключах элементов данных, которые можно использовать только с Windows-Агентом.

Ключи элементов данных для Windows

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| eventlog[имя,<регулярное выражение>,<важность>,<источник>,<eventid>,<макс. кол-во строк>,<режим>] | | | |
| Мониторинг журналов событий. | Журнал (лог) | имя – имя журнала событий. регулярное выражение – регулярное выражение, описывающее требуемый шаблон содержимого. важность – регулярное выражение, описывающее важность (без учета регистра) Параметр принимает регулярное выражение на основе следующих значений: Information, Warning, Error, Critical, Verbose). источник – регулярное выражение, описывающее идентификатор источника. eventid – регулярное выражение, описывающее идентификатор(ы) событий. макс. кол-во строк – максимальное количество новых строк в секунду, которое Агент будет отправлять Серверу или Прокси. Этот параметр переопределяет значение "MaxLinesPerSecond" в zabbix\_agentd.win.conf. режим – возможные значения: all (по умолчанию), skip – пропустить обработку старых данных (влияет только на недавно созданные элементы данных). | Элемент данных должен быть настроен активной проверкой. Обратить внимание: Агент не может отправлять события из журнала "Пересланные события". Обратить внимание, что выбор не журнального типа информации для этого элемента данных приведет к потере локального штампа времени, а также важности журнала и информации об источнике. |
| net.if.list | | | |
| Список сетевых интерфейсов (включая тип, состояние, IPv4-адрес, описание интерфейса). | Текст |  | Обратить внимание, что включение/отключение некоторых компонентов Windows могут изменить порядок имен интерфейсов в Windows. В некоторых версиях Windows (к примеру, Server 2008) может потребоваться установка последних обновлений для поддержки не-ASCII символов в именах интерфейсов. |
| perf\_counter[счетчик,<период>] | | | |
| Значение любого счетчика производительности Windows. | Целое число, число с плавающей точкой, строка или текст (в зависимости от запроса) | счетчик – путь к счетчику период – последние N секунд для сохранения усредненного значения. Значение период должно быть от 1 до 900 секунд (включительно), значение по умолчанию 1. | Можно использовать Мониторинг производительности для получения списка счетчиков. |
| perf\_counter\_en[счетчик,<период>] | | | |
| Значение любого счетчика производительности Windows на английском языке. | Целое число, число с плавающей точкой, строка или текст (в зависимости от запроса) | счетчик – путь к счетчику на английском языке период – последние N секунд для сохранения усредненного значения. Значение период должно быть от 1 до 900 секунд (включительно), значение по умолчанию 1. | Этот элемент данных поддерживается только на Windows Server 2008/Vista и более новых версиях. Можно найти список строк на английском языке, заглянув в следующую ветку реестра: HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009. |
| perf\_instance.discovery[объект] | | | |
| Список экземпляров объекта счетчиков производительности Windows. Используется для низкоуровневого обнаружения. | Объект JSON | объект – имя объекта (локализованное) |  |
| perf\_instance\_en.discovery[объект] | | | |
| Список экземпляров объекта счетчиков производительности Windows, обнаруженных с использованием имен объектов на английском языке. Используется для низкоуровневого обнаружения. | Объект JSON | объект – имя объекта (на английском языке) |  |
| proc\_info[процесс,<параметр>,<тип>] | | | |
| Различная информация об указанном процессе(-ах). | Число с плавающей точкой | процесс – имя процесса параметр – запрашиваемый параметр процесса. тип – тип представления (имеет смысл, когда есть более одного процесса с одинаковым именем) | Поддерживаются следующие параметры: vmsize (по умолчанию) – размер виртуальной памяти процесса в Кбайтах wkset – размер working set процесса (количество используемой процессом физической памяти) в Кбайтах pf – количество ошибок на страницах ktime – время ядра процесса в миллисекундах utime – пользовательское время процесса в миллисекундах io\_read\_b – количество байт, прочитанных процессом во время операций ввода/вывода io\_read\_op – количество операций чтения, выполненных процессом io\_write\_b – количество байт, записанных процессом во время операций ввода/вывода io\_write\_op – количество операций записи, выполненных процессом io\_other\_b – количество байт, пересланных процессом во время операций, отличных от чтения и записи io\_other\_op – количество выполненных процессом операций ввода/вывода, отличных от операций чтения и записи gdiobj – количество используемых процессом объектов GDI userobj – количество используемых процессом объектов USER Допустимые типы: avg (по умолчанию) – среднее значение среди всех процессов с именем <процесс> min – минимальное значение среди всех процессов с именем <процесс> max – максимальное значение среди всех процессов с именем <процесс> sum – сумма значений для всех процессов с именем <процесс> Обратить внимание, что для корректной работы этого элемента данных на 64-битной системе потребуется 64-битный Агент. Обратить внимание: Все параметры io\_\*, gdiobj и userobj доступны только в Windows 2000 и более поздних версиях Windows, не в Windows NT 4.0. |
| service.discovery | | | |
| Список служб Windows. Используется низкоуровневым обнаружением. | Объект JSON |  |  |
| service.info[служба,<парам>] | | | |
| Информация о службе. | Целое число – с парам равным state, Startup Строка – с парам равным displayname, path, user Текст – с парам равным description В частности при state: 0 – запущена, 1 – пауза, 2 – ожидание старта, 3 – ожидание паузы, 4 – ожидание продолжения, 5 – ожидание остановки, 6 – остановлена, 7 – неизвестно, 255 – такой службы не существует  В частности при Startup: 0 – автоматически, 1 – автоматически (отложенный запуск), 2 – вручную, 3 – отключена, 4 – неизвестно, 5 – автоматический запуск по триггеру, 6 – автоматический отложенный запуск по триггеру, 7 – ручной запуск по триггеру | служба – действительное имя службы или ее отображаемое имя как в оснастке MMC Службы парам – state (по умолчанию), displayname, path, user, Startup или description | Примеры: => service.info[SNMPTRAP] – состояние службы SNMPTRAP => service.info[SNMP Trap] – состояние этой же службы, но указано отображаемое имя => service.info[EventLog,Startup] – состояние запуска при загрузке службы Журнала событий Элементы данных service.info[служба,state] и service.info[служба] вернут одинаковую информацию. Обратить внимание, что только парам равный state у этого элемента данных возвращает значение по несуществующим службам (255). |
| services[<тип>,<состояние>,<исключения>] | | | |
| Список служб. | 0 – если список служб пуст.  Текст – список служб, разделенных новой строкой. | тип – all (по умолчанию), automatic, manual, disabled состояние – all (по умолчанию), stopped, Started, Start\_pending, stop\_pending, running, continue\_pending, pause\_pending или paused исключения – список служб, исключаемых из результата. Исключаемые службы должны быть указаны в двойных кавычках, разделены запятой, без пробелов. | Примеры: => services[,Started] – список запущенных служб => services[automatic, stopped] – список остановленных служб, которые должны быть запущены => services[automatic, stopped, "service1,service2,service3"] – список остановленных служб, которые должны быть запущены, за исключением служб с именами service1,service2 и service3 |
| wmi.get[<пространство\_имен>,<запрос>] | | | |
| Выполнение WMI запроса и получение первого выбранного объекта. | Целое число, число с плавающей точкой, строка или текст (в зависимости от запроса.) | пространство\_имен – название пространства имен WMI запрос – WMI запрос, возвращающий один объект | WMI запросы выполняются с использованием [WQL](https://en.wikipedia.org/wiki/WQL) [en]. Пример: => wmi.get[root\cimv2,select status from Win32\_DiskDrive where Name like "%PHYSICALDRIVE0%"] – получение состояния первого физического диска |
| wmi.getall[<пространство\_имен>,<запрос>] | | | |
| Выполнение WMI запроса и получение полного ответа.  Можно использовать для низкоуровневого обнаружения. | Объект JSON | пространство\_имен – название пространства имен WMI запрос – WMI-запрос | Запросы WMI выполняются при помощи [WQL](https://en.wikipedia.org/wiki/WQL) [en]. Пример: => wmi.getall[root\cimv2,select \* from Win32\_DiskDrive where Name like "%PHYSICALDRIVE%"] – получение информации о состоянии физических дисков Можно использовать JSONPath предобработку при обращении к более конкретным значениям в полученном JSON. |
| vm.vmemory.size[<тип>] | | | |
| Размер виртуального пространства в байтах или в процентах от общего размера. | Целое число – для байтов  Число с плавающей точкой – для процентов | тип – возможные значения: available (доступно виртуальной памяти), pavailable (доступно виртуальной памяти, в процентах), pused (использовано виртуальной памяти, в процентах), total (всего виртуальной памяти, по умолчанию), used (использовано виртуальной памяти) | Пример: => vm.vmemory.size[pavailable]– доступно виртуальной памяти, в процентах  Мониторинг статистики виртуальной памяти основывается на: \* Общее количество виртуальной памяти в Windows (всего физической + размер файла подкачки); \* Максимального количества памяти, которое может занять Агент; \* Текущий лимит выделенной памяти системе или Агенту, смотря что меньше. |

**Мониторинг служб Windows**

Пошаговые инструкции по настройке мониторинга служб Windows. Предполагается, что Сервер и Агент уже настроены и работают.

**Шаг 1.** Узнать имя службы.

Можно получить имя, перейдя в оснастку MMC-службы и открыв свойства службы. На вкладке Общие в значении поля, называемого "Имя службы", будет имя, которое используется при настройке элемента данных для наблюдения.

Например, если требуется наблюдать службу workstation, то служба, вероятно, будет называться lanmanworkstation.

Шаг 2. Настроить элемент данных для наблюдения за службой.

Элемент данных service.info[служба,<парам>] возвращает информацию о конкретной службе. В зависимости от требуемой информации, нужно указать опцию парам, которая принимает следующие значения: displayname, state, path, user, Startup или description. Значением по умолчанию является state, если парам не указан (service.info[служба]).

Тип возвращаемого значения зависит от выбранного парам: целое число при state и Startup; строка символов при displayname, path и user; текст при description.

Пример:

* Ключ service.info[lanmanworkstation];
* Тип информации Числовой (целое положительное).

Элемент данных service.info[lanmanworkstation] будет извлекать информацию о состоянии службы как числовое значение. Чтобы преобразовать числовое значение в текстовое представление в веб-интерфейсе ("0" как "Running", "1" как "Paused" и т.д.), можно настроить преобразование значений на узле сети, на котором сконфигурирован элемент данных. Чтобы это сделать, нужно либо присоединить шаблон "Windows services by Zabbix agent" или "Windows services by Zabbix agent active" к узлу сети, либо настроить на узле сети новое преобразование значений, основанное на преобразованиях значений "Windows service state", уже настроенных в упомянутых шаблонах.

Следует обратить внимание, что оба упомянутых шаблона имеют правило обнаружения, настроенное для автоматического обнаружения служб. Если не требуется этого, можно деактивировать правило обнаружения на уровне узла сети после присоединения к нему шаблона.

Низкоуровневое обнаружение дает возможность автоматического создания элементов данных, триггеров и графиков по различных объектам на компьютере. Подсистема может автоматически начать наблюдение за службами Windows на узле без необходимости знания точного имени службы или создания элементов данных по каждой службе вручную. Можно использовать фильтр для генерирования реальных элементов данных, триггеров и графиков только по интересующим службам.

1. SNMP-агент

SNMP используется для мониторинга таких устройств, как: принтеры, сетевые коммутаторы, маршрутизаторы или ИБП, которые, как правило, поддерживают SNMP и на которых было бы непрактично пытаться устанавливать полноценные ОС и Агенты.

Чтобы была возможность получать данные, переданные SNMP-агентами с этих устройств, Сервер должен быть изначально сконфигурирован с поддержкой SNMP.

SNMP-проверки выполняются только через UDP-протокол.

Серверы и Прокси опрашивают устройства SNMP по несколько значений за один запрос. Такое поведение влияет на все виды элементов данных SNMP (обычные элементы данных SNMP, элементы данных с динамическими индексами, а также низкоуровневые SNMP обнаружения), и делает работу SNMP более эффективной. Также массовые запросы можно отключить у устройств, которые не способны обработать их должным образом, используя настройку "Использовать массовые запросы", доступную в каждом интерфейсе.

Процессы Сервера и Прокси запишут в журнал строки наподобие следующих в случае получения неправильного/искаженного SNMP-ответа:

SNMP response from host "gateway" does not contain all of the requested variable bindings

Пока они не покрывают все возможные проблемные случаи, но они полезны для идентификации отдельных SNMP-устройств, на которых массовую обработку нужно отключить.

Сервер/Прокси всегда повторят запрос минимум один раз после неуспешной попытки: либо через механизм библиотеки SNMP, либо через внутренний механизм сбора множества значений за один запрос (bulk).

При мониторинге устройств по SNMPv3 следует убедиться, что msgAuthoritativeEngineID (также известное как snmpEngineID или "Engine ID") никогда не будет общим для двух и более устройств. Согласно RFC 2571 оно должно быть уникальным для каждого устройства.

RFC3414 требует, чтобы SNMPv3-устройства сохраняли свои значения engineBoots. Некоторые устройства не выполняют этого требования, что приводит к тому, что после перезагрузки их SNMP-сообщения отбрасываются как устаревшие. В этой ситуации необходимо вручную очистить кэш SNMP на Сервере/Прокси (используя "-R snmp\_cache\_reload") или же перезапустить Сервер/Прокси.

Для настройки мониторинга устройства по SNMP нужно выполнить следующие шаги:

1. Шаг 1. Узнать строку SNMP (или OID) элемента данных, который требуется мониторить.

Для получения списка строк SNMP использовать команду snmpwalk или эквивалентную утилиту:

snmpwalk -v 2c -c public <IP хоста>

где "2c" означает версию SNMP (можно заменить его на "1", чтобы использовать на устройстве SNMPv1).

Эта команда должна показать список SNMP-строк и их последние значения. Если это не произойдет, то возможно, что SNMP-"community" отличается от стандартного "public". В этом случае необходимо узнать это имя.

Для поиска конкретной строки, которую нужно мониторить, используют список; например, если требуется мониторить количество входящих байтов на третьем порту коммутатора, можно использовать IF-MIB::ifInOctets.3 из этой строки:

IF-MIB::ifInOctets.3 = Counter32: 3409739121

Также можно воспользоваться командой snmpget, чтобы определить цифровой OID для IF-MIB::ifInOctets.3:

snmpget -v 2c -c public -On 10.62.1.22 IF-MIB::ifInOctets.3

Следует обратить внимание, что последнее число в строке – это номер порта, который ищется для мониторинга.

Вывод команды покажет примерно следующее:

.1.3.6.1.2.1.2.2.1.10.3 = Counter32: 3472126941

Последнее число в OID является номером порта.

Некоторые из наиболее часто используемых SNMP OID автоматически конвертируются Подсистемой в числовое представление.

В последнем примере тип значения Counter32 (32-битный счетчик) внутренне соответствует типу ASN\_COUNTER. Полный список поддерживаемых типов: ASN\_COUNTER, ASN\_COUNTER64, ASN\_UINTEGER, ASN\_UNSIGNED64, ASN\_INTEGER, ASN\_INTEGER64, ASN\_FLOAT, ASN\_DOUBLE, ASN\_TIMETICKS, ASN\_GAUGE, ASN\_IPADDRESS, ASN\_OCTET\_STR и ASN\_OBJECT\_ID. Приведенные типы грубо соответствуют "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" в выводе утилиты snmpget, но могут также отображаться как "STRING", "Hex-STRING", "OID" и другими в зависимости от наличия подсказки.

1. Шаг 2. Создать узел сети, соответствующий устройству.

Добавить к узлу сети SNMP-интерфейс:

* + 1. ввести IP-адрес/DNS-имя и номер порта;
    2. выбрать "Версия SNMP" из выпадающего списка;
    3. добавить учетные данные к интерфейсу в зависимости от выбранной версии SNMP:
* SNMPv1, v2 требуют только community (обычно "public");
* SNMPv3 требует более специфичные опции (таблица 71);
  + 1. оставить выбранным "Использовать массовые запросы", чтобы разрешить массовую обработку SNMP-запросов.

Параметр SNMPv3

| Параметр SNMPv3 | Описание |
| --- | --- |
| Имя контекста | Ввести контекстное имя для определения элемента данных в SNMP подсети. В данном поле раскрываются пользовательские макросы. |
| Имя безопасности | Ввести имя безопасности. В данном поле раскрываются пользовательские макросы. |
| Уровень безопасности | Выбрать уровень безопасности: noAuthNoPriv – ни аутентификация, ни протокол безопасности не используются; AuthNoPriv – используется протокол аутентификации, протокол безопасности – нет. AuthPriv – используются и протокол аутентификации, и протокол безопасности |
| Протокол аутентификации | Выбрать протокол аутентификации – MD5, SHA1, SHA224, SHA256, SHA384 или SHA512. |
| Пароль аутентификации | Ввести фразу-пароль для аутентификации. В данном поле раскрываются пользовательские макросы. |
| Протокол безопасности | Ввести протокол безопасности – DES, AES128, AES192, AES256, AES192C (Cisco) или AES256C (Cisco). Следует обратить внимание, что: - на некоторых более старых системах net-snmp может не поддерживать AES256. - на некоторых более новых системах (например, RHEL9) поддержка DES в пакете net-snmp может быть убрана. |
| Ключевая фраза безопасности | Ввести фразу-пароль безопасности. В данном поле раскрываются пользовательские макросы. |

В случае некорректных учетных данных SNMPv3 (имя безопасности, протокол/фраза-пароль аутентификации, протокол безопасности):

* Подсистема получит ERROR от net-snmp, за исключением ошибочного значения "Ключевая фраза безопасности", в этом случае – ошибку превышения "ВРЕМЕНИ ОЖИДАНИЯ" от net-snmp;
* доступность интерфейса SNMP переключится на красный цвет (недоступно).

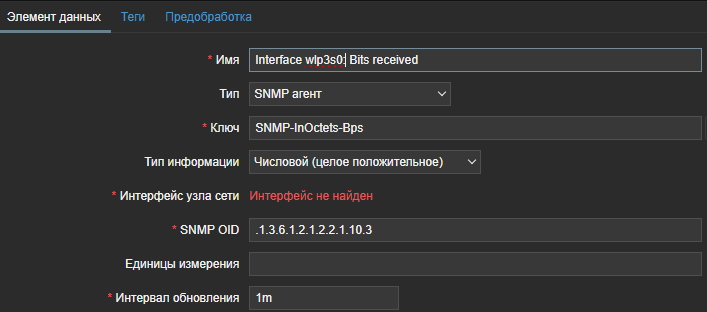
Изменения в "Протокол аутентификации", "Пароль аутентификации", "Протокол безопасности" или "Ключевая фраза безопасности" без изменения "Имя безопасности" вступят в силу только после ручной очистки кэша на Сервере/Прокси (используя "-R snmp\_cache\_reload") или при перезапуске Сервера/Прокси. В случае если "Имя безопасности" также меняется, все параметры будут обновлены немедленно.

Можно использовать один из поставляемых шаблонов SNMP ("Template SNMP Device" и другие), которые автоматически добавят набор элементов данных. Но шаблон может быть несовместим с узлом сети. Далее нажать на кнопку Добавить для сохранения узла сети.

1. Шаг 3. Создать элемент данных для мониторинга.

В меню Подсистемы нажать на Элементы данных у ранее созданного SNMP узла сети. В зависимости от того, использовался ли шаблон при создании узла сети или нет, можно увидеть или список элементов данных SNMP, связанных с узлом сети, или пустой список. Для самостоятельного создания элемента данных с помощью собранной информации, используя snmpwalk или snmpget, нажать на Создать элемент данных. В окне нового элемента данных (рисунок 3):

* + 1. ввести имя элемента данных;
    2. изменить поле Тип на "SNMP Агент";
    3. ввести в поле Ключ, например: SNMP-InOctets-Bps;
    4. убедиться, что в поле "Интерфейс узла сети" указан нужный коммутатор/роутер;
    5. ввести в поле "SNMP OID" текстовый или числовой OID, который получен ранее, например: .1.3.6.1.2.1.2.2.1.10.3;
    6. установить "Тип информации" в значение равное "Числовой (с плавающей точкой)";
    7. ввести "Интервал обновления" и период "Хранения истории", если нужно, чтобы значения параметров отличались от значений по умолчанию;
    8. на вкладке Предобработка добавить шаг "Изменение в секунду" (это важно, так как иначе с SNMP-устройства будут получены накопленные значения вместо последнего изменения); выбрать пользовательский множитель при необходимости.



Окно ввода

Теперь нужно сохранить элемент данных и перейти в "Мониторинг → Последние данные", чтобы увидеть данные SNMP.

1. Обработка массовых SNMP-запросов

Сервер и Прокси одним запросом опрашивают множество SNMP-элементов данных. Такое поведение затрагивает следующие типы SNMP-элементов данных:

* обычные SNMP-элементы данных;
* SNMP-элементы данных с динамическими индексами;
* правила низкоуровневого SNMP-обнаружения.

Все элементы данных SNMP с одного интерфейса планируются на опрос в одно время. Первые два типа элементов данных собираются поллерами порциями не более чем по 128 элементов данных, в то время как правила низкоуровневого обнаружения обрабатываются индивидуально, как и ранее.

На низком уровне есть два вида операций, выполняемых при опросе значений: получение нескольких заданных объектов и обход дерева OID.

Для "получения" используется GetRequest-PDU c не более чем 128 привязанных переменных. Для "обхода" используется GetNextRequest-PDU для SNMPv1 и GetBulkRequest с полем "max-repetitions" с наибольшим количеством в 128 полученных значений для SNMPv2 и SNMPv3.

Таким образом, преимущества массовой обработки для каждого типа SNMP элемента данных описаны ниже:

* обычные SNMP элементы данных получают преимущество от улучшенного получения;
* SNMP-элементы данных с динамическими индексами получают преимущество и от улучшенного "получения "и "обхода": "получение" используется для проверки индексов, а "обход" – для построения кэша значений;
* правила низкоуровневого SNMP обнаружения получают преимущество от улучшенного "обхода".

Тем не менее есть техническая проблема: не все устройства способны вернуть 128 значений за один запрос. Некоторые всегда возвращают корректный ответ, но другие либо отвечают с ошибкой "tooBig(1)", либо не отвечают вообще, если потенциальный запрос превышает определенный лимит.

Для вычисления оптимального количества объектов, запрашиваемых с данного устройства, Подсистема использует итерационную стратегию. Начинается с осторожного запроса одного значения. Если запрос выполнен успешно, запрашивается 2 значения за один запрос. Если запрос снова выполнен успешно, запрашивается 3 значения за запрос и продолжается аналогично путем умножения количества запрашиваемых значений на 1.5, в результате получается следующая последовательность размера запросов: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Как только устройство отказывается давать корректный ответ (к примеру, на 42 переменных), Подсистема выполняет два действия.

Во-первых, для текущей серии элементов данных Подсистема делит пополам количество объектов на один запрос и запрашивает 21 переменную. Если устройство доступно, то запрос должен работать в большинстве случаев, потому что было известно, что с 28 переменными работало, а 21 значительно меньше. Если проблема продолжается, Подсистема возвращается к опросу значений по одному. Если проблемы есть и в этом случае, значит устройство определенно не отвечает и проблема не в размере запроса.

Во-вторых, для дальнейших порций элементов данных Подсистема начинает с последнего удачного количества переменных (28 в нашем случае) и продолжает, увеличивая количество переменных за запрос на 1 до достижения лимита. Например, если наибольший размер ответа – это 32 переменных, то последующие запросы будут размерами 29, 30, 31, 32 и 33. Последний запрос будет неудачным, и Подсистема никогда более не запросит 33 значения за один запрос. С этого момента Подсистема всегда будет опрашивать максимум по 32 переменных для этого устройства.

Если большие запросы неудачно завершаются с этим количеством переменных, это может означать одно из двух. Точный критерий, используемый устройством для ограничения размера ответа, неизвестен, но пытаются приблизительно оценить это, используя количество переменных. Поэтому первая возможность – в общем случае это количество переменных около реального ограничения размера ответа для данного устройства: иногда ответ меньше этого предела, иногда больше. Вторая возможность – что UDP-пакет (в любом направлении) просто был потерян. По этим причинам, если Подсистема сталкивается с неудачным запросом, то он уменьшает максимальное количество переменных, чтобы попытаться перейти в приемлемый для устройства диапазон, но только до 2 раз.

В примере выше, если запрос с 32 переменными будет неудачен, Подсистема уменьшит количество до 31. Если неудача случится снова, Подсистема уменьшит количество до 30. Тем не менее Подсистема не будет уменьшать количество ниже 30, потому что он предположит, что дальнейшие проблемы по причине потерянных UDP-пакетов, нежели из-за ограничения устройства.

Если устройство не умеет обрабатывать массовые запросы корректно и описанная выше эвристика не работает, то у каждого интерфейса имеется настройка "Использовать массовые запросы", позволяющая отключить массовые запросы у этого устройства.

1. Динамические индексы

Не всегда требуемый номер индекса (например, сетевого интерфейса) из всех SNMP OID остается таким же.

Индексы могут быть динамическими – они могут изменяться время от времени и, как следствие, элемент данных может перестать работать.

Чтобы избежать этого, имеется возможность указать OID, который принимает во внимание возможность изменения номера индекса.

Например, если необходимо получить значение индекса, который относится к ifInOctets, которое соответствует интерфейсу GigabitEthernet0/1 на устройстве Cisco, используют следующий OID:

ifInOctets["index","ifDescr","GigabitEthernet0/1"]

Для OID используется специальный синтаксис (параметры – в таблице 72).

<OID данных>["index","<базовый OID индекса>","<искомая строка>"]

Синтаксис OID

| Параметр | Описание |
| --- | --- |
| OID данных | Основной OID, который используется для получения данных по элементу данных. |
| index | Метод обработки. |
| базовый OID индекса | Этот OID будет просматриваться для получения значения индекса, соответствующего заданной строке. |
| искомая строка | Строка, используемая для точного совпадения со значением при выполнении поиска, зависит от регистра. |

Например, для получения использования памяти процессом apache, если используется этот синтаксис OID:

HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]

номер индекса будет искаться здесь:

...

HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"

HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"

HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"

HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"

...

Теперь при наличии индекса, равного 5388, он будет присоединен к OID данных, чтобы получить требуемое значение:

HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes

При запросе динамического индекса элемента данных Подсистема извлекает и кэширует всю таблицу SNMP под базовым OID, даже если совпадение будет найдено ранее. Это делается на тот случай, если другой элемент данных в дальнейшем ссылается на тот же самый базовый OID – Подсистема будет искать индекс в кэше вместо еще одного запроса к наблюдаемому устройству. Следует обратить внимание, что каждый процесс поллера использует свой собственный кэш.

Во всех последующих операциях получения значений проверяется только найденный индекс. Если он не изменился, значение запрашивается. Если он изменился, кэш перестраивается – каждый поллер, который встречает измененный индекс, снова обходит и кэширует таблицу индексов SNMP.

1. Специальные OID

Некоторые OID, наиболее часто используемые в SNMP, автоматически преобразовываются в числовые представления самим Подсистемой. Например, ifIndex преобразовывается в 1.3.6.1.2.1.2.2.1.1, ifIndex.0преобразовывается в 1.3.6.1.2.1.2.2.1.1.0.

В таблице 73 содержится перечень специальных OID.

Перечень специальных OID

| Специальный OID | Идентификатор | Описание |
| --- | --- | --- |
| ifIndex | 1.3.6.1.2.1.2.2.1.1 | Уникальное значение каждого интерфейса. |
| ifDescr | 1.3.6.1.2.1.2.2.1.2 | Текстовая строка, которая содержит информацию о интерфейсе. Эта строка может включать в себя название компании производителя, имя продукта, аппаратную версию интерфейса. |
| ifType | 1.3.6.1.2.1.2.2.1.3 | Тип интерфейса выделяется в соответствии с физическим/канальным протоколом(-ами) непосредственно "под" сетевым уровнем стека сетевых протоколов. |
| ifMtu | 1.3.6.1.2.1.2.2.1.4 | Размер наибольшей дейтаграммы, которую может отправить/получить интерфейс, указывается в байтах |
| ifSpeed | 1.3.6.1.2.1.2.2.1.5 | Оценочная скорость интерфейса текущей пропускной способности в битах за секунду. |
| ifPhysAddress | 1.3.6.1.2.1.2.2.1.6 | Физический адрес интерфейса на канальном уровне непосредственно "под" сетевым уровнем в стеке сетевых протоколов. |
| ifAdminStatus | 1.3.6.1.2.1.2.2.1.7 | Текущее административное состояние интерфейса. |
| ifOperStatus | 1.3.6.1.2.1.2.2.1.8 | Текущее рабочее состояние интерфейса. |
| ifInOctets | 1.3.6.1.2.1.2.2.1.10 | Полное число полученных байтов, включая символы заголовков. |
| ifInUcastPkts | 1.3.6.1.2.1.2.2.1.11 | Количество пакетов одноадресной рассылки, доставленных на верхний уровень стека сетевых протоколов. |
| ifInNUcastPkts | 1.3.6.1.2.1.2.2.1.12 | Количество пакетов НЕ одноадресной рассылки (broadcast и multicast), доставленных на верхний уровень стека сетевых протоколов. |
| ifInDiscards | 1.3.6.1.2.1.2.2.1.13 | Количество входящих, но отвергнутых пакетов, даже если не было обнаружено ошибок, мешающих доставке пакетов на верхний уровень стека сетевых протоколов. Одной из возможных причин для отвержения пакета могло быть освобождение места в буфере. |
| ifInErrors | 1.3.6.1.2.1.2.2.1.14 | Количество входящих пакетов, полученных с ошибкой, из-за которой пакеты не были доставлены на верхний уровень стека сетевых протоколов. |
| ifInUnknownProtos | 1.3.6.1.2.1.2.2.1.15 | Количество пакетов, полученных через интерфейс, но отвергнутых из-за неизвестного или не поддерживаемого протокола. |
| ifOutOctets | 1.3.6.1.2.1.2.2.1.16 | Полное количество отправленных октетов с интерфейса, включая символы заголовков. |
| ifOutUcastPkts | 1.3.6.1.2.1.2.2.1.17 | Полное количество пакетов, которые пытался отправить верхний уровень стека сетевых протоколов, и которые адресованы не на broadcast или multicast-адреса на этом подуровне, включая те, которые были отвергнуты или не отправлены. |
| ifOutNUcastPkts | 1.3.6.1.2.1.2.2.1.18 | Полное количество пакетов, которые пытался отправить верхний уровень стека сетевых протоколов, и которые адресованы на broadcast или multicast-адреса на этом подуровне, включая те, которые были отвергнуты или не отправлены. |
| ifOutDiscards | 1.3.6.1.2.1.2.2.1.19 | Количество исходящих пакетов, которые были отвергнуты даже если не было обнаружено ошибок, мешающих отправке. Одной из возможных причин отвержения пакета могло быть освобождение места в буфере. |
| ifOutErrors | 1.3.6.1.2.1.2.2.1.20 | Количество исходящих пакетов, которые не могут быть отправлены из-за ошибок. |
| ifOutQLen | 1.3.6.1.2.1.2.2.1.21 | Длина очереди исходящих пакетов (в пакетах). |

1. MIB-файлы

MIB (база управляющей информации) – это виртуальная база данных, используемая для управления объектами в сети связи. MIB-файлы позволяют использовать текстовое представление OID. При мониторинге Подсистемой устройств SNMP можно использовать первичные OID, но для комфортного использования текстовых представлений потребуется установить MIB-файлы.

Например, ifHCOutOctets является текстовым представлением OID 1.3.6.1.2.1.31.1.1.1.10.

Установка MIB-файлов на ОС на основе Debian:

apt install snmp-mibs-downloader

download-mibs

Установка MIB-файлов на ОС на основе RedHat:

dnf install net-snmp-libs

На ОС на основе RedHat MIB-файлы должны быть подключены по умолчанию. На ОС на основе Debian нужно отредактировать файл /etc/snmp/snmp.conf и закомментировать строку, которая содержит mibs:

# As the snmp packages come without MIB files due to license reasons, loading# of MIBs is disabled by default. If you added the MIBs you can re-enable# loading them by commenting out the following line.

mibs :

Тестирование snmp MIB-файлов можно выполнить с использованием утилиты snmpwalk. Если эта утилита не установлена, можно воспользоваться следующими инструкциями:

1. для ОС на основе Debian:

apt install snmp

1. для ОС на основе RedHat:

dnf install net-snmp-utils

После этого следующая команда должна не выдавать ошибку при выполнении запроса к сетевому устройству:

$ snmpwalk -v 2c -c public <IP СЕТЕВОГО УСТРОЙСТВА> ifInOctets

IF-MIB::ifInOctets.1 = Counter32: 176137634

IF-MIB::ifInOctets.2 = Counter32: 0

IF-MIB::ifInOctets.3 = Counter32: 240375057

IF-MIB::ifInOctets.4 = Counter32: 220893420

[...]

Следует обратить внимание, что процессы Подсистемы не определяют изменения, выполненные с MIB-файлами, поэтому после каждого изменения необходимо перезапускать Сервер или Прокси, например:

service zabbix-server reStart

После выполнения этой команды изменения, сделанные в MIB-файлах, вступят в силу.

1. SNMP-трапы

Получение SNMP-трапов является полной противоположностью запросам к SNMP-устройствам.

В этом случае информация отправляется с SNMP-устройства и собирается или определяется Подсистемой.

Обычно трапы отправляются при наступлении некоторых условий, и Агент подключается на 162-ый порт Сервера (в отличие от 161-ого порта на стороне Агента, который используется для запросов). Использование трапов может помочь обнаружить некоторые кратковременные проблемы, которые происходят между интервалами опроса и могут быть пропущены при запросах данных.

Получение SNMP-трапов в Подсистеме рассчитано на работу с snmptrapd и с одним из встроенных механизмов передачи трапов в Подсистема – либо bash/perl- скрипты, либо SNMPTT.

Самый простой способ настройки мониторинга трапов после настройки Подсистемы – это использовать решение на основе Bash-скрипта, так как Perl и SNMPTT зачастую отсутствуют в современных дистрибутивах и требуют более сложной настройки. Данное решение использует скрипт, который настраивается в виде traphandle. Для повышения производительности в производственных системах используют встроенное решение Perl (либо скрипт с "do perl"-опцией, либо SNMPTT).

Последовательность действий при получении трапа следующая:

1. snmptrapd получает трап;
2. snmptrapd передает трап скрипту получателю (Bash, Perl) или SNMPTT;
3. получатель разбирает, форматирует и записывает трап в файл;
4. SNMP-траппер читает и анализирует файл с трапами;
5. для каждого трапа Подсистема находит все элементы данных с типом "SNMP трап" и интерфейсом узла сети, совпадающим с полученным адресом из трапа;

Примечание – В процессе поиска соответствия используется только выбранный "IP" или "DNS" у интерфейса узла сети.

1. у каждого найденного элемента данных трап сравнивается с регулярным выражением "snmptrap\[регулярное выражение\]". Трап записывается в качестве значения для всех совпавших элементов данных. Если совпадений не найдено, но существует элемент данных snmptrap.fallback, то трап записывается в качестве значения для этого элемента данных;
2. если трап не был записан в качестве значения для какого-либо элемента данных, Подсистема по умолчанию журналирует не совпавшие трапы. Это поведение настраивается в "Журналировать не совпадающие SNMP трапы" в меню "Администрирование → Общие → Прочие").
3. Настройка SNMP-трапов

Настройка следующих полей в веб-интерфейсе является специфичной для этого типа элементов данных:

1. Узел сети должен иметь SNMP-интерфейс. В "Настройка → Узлы сети", в поле "Интерфейсы узла сети" добавляют SNMP-интерфейс с корректными IP- или DNS-адресами. Адрес из каждого полученного трапа сравнивается с IP- и DNS-адресами всех SNMP-интерфейсов для поиска подходящих узлов сети.
2. Настройка элемента данных. В поле Ключ используют один из ключей SNMP-трапов (таблица 74).

Ключи SNMP-трапов

|  |  |  |
| --- | --- | --- |
| Описание | Возвращаемое значение | Комментарии |
| snmptrap[регулярное выражение] | | |
| Отлов всех SNMP-трапов, которые соответствуют регулярному выражению, указанному в поле "регулярное выражение". Если регулярное выражение не указано, принимаются все трапы. | SNMP-трап | Этот элемент данных можно привязать только к SNMP-интерфейсам. В параметре этого ключа элемента данных поддерживаются пользовательские макросы и глобальные регулярные выражения. |
| snmptrap.fallback | | |
| Отлов всех SNMP-трапов, которые не были перехвачены ни одним из элементов данных snmptrap[] для этого интерфейса. | SNMP-трап | Такой элемент данных можно привязать только к SNMP- интерфейсам. |

Многострочное совпадение по регулярному выражению времени не поддерживается.

В поле "Тип информации" указывают значение "Журнал (лог)" для обработки штампов времени. Следует обратить внимание, что другие форматы, такие как "Числовой", также приемлемы, но для этого может потребоваться пользовательский обработчик трапов.

Для того чтобы мониторинг SNMP-трапов работал, он должен быть сначала корректно настроен.

1. Настройка мониторинга SNMP-трапов

Для чтения трапов Сервер или Прокси должны быть настроены на запуск процесса SNMP-траппера, а также должны знать абсолютный путь к файлу с трапами, который заполняется при помощи SNMPTT или получателя трапов Bash/Perl. Чтобы это сделать, изменяют файл конфигурации (zabbix\_server.conf или zabbix\_proxy.conf):

StartSNMPTrapper=1

SNMPTrapperFile=[TRAP FILE]

Если используется system-параметр PrivateTmp, то этот файл не заработает в /tmp.

Для настройки получателя трапов Bash используется только snmptrapd. Можно использовать Bash-скрипт получателя трапов для передачи трапов Серверу напрямую с snmptrapd. Чтобы его настроить, добавляют опцию traphandle в файл конфигурации snmptrapd.conf.

Для настройки SNMPTT snmptrapd необходимо настроить на использование SNMPTT.

Для лучшей производительности SNMPTT нужно сконфигурировать как демон, используя snmptthandler-embedded для передачи ему трапов.

После настройки SNMPTT на получение трапов, конфигурируют snmptt.ini:

1. включить использование модуля Perl из пакета NET-SNMP:

net\_snmp\_perl\_enable = 1

1. включить журналирование трапов в файл с трапами, который Подсистема будет читать:

log\_enable = 1

log\_file = [TRAP FILE]

1. задать формат даты/времени:

date\_time\_format = %H:%M:%S %Y/%m/%d

Пакет net-snmp-perl удален в RHEL 8.0-8.2; добавлен заново в RHEL 8.3.

Далее задают формат трапов, чтобы они распознавались Подсистемой (редактируют snmptt.conf):

1. каждая инструкция FORMAT должна начинаться с "ZBXTRAP [адрес]", где [адрес] будет сравниваться с IP- и DNS-адресами у SNMP-интерфейсов в Подсистеме, например:

EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal

FORMAT ZBXTRAP $aA Device reinitialized (coldStart)

1. не используют неизвестные трапы, так как Подсистема может их не распознать. Неизвестные трапы могут быть обработаны, если задать общее событие в snmptt.conf:

EVENT general .\* "General event" Normal

Для настройки получателя Perl-трапов используются Perl, Net-SNMP, скомпилированный с "--enable-embedded-perl".

Получатель трапов Perl (в misc/snmptrap/zabbix\_trap\_receiver.pl) можно использовать для передачи трапов Серверу напрямую с snmptrapd. Для его настройки нужно:

1. добавить Perl-скрипт в файл конфигурации snmptrapd (snmptrapd.conf), например:

perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";

1. настроить сам получатель, например:

$SNMPTrapperFile = "[TRAP FILE]";

$DateTimeFormat = "[DATE TIME FORMAT]";

Если имя скрипта не заключено в кавычки, snmptrapd не будет запускаться с подобными сообщениями:

Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line

Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line

Все заданные получатели трапов perl и конфигурация SNMPTT-трапов должны форматировать трап следующим образом:

[штамп времени] [трап, часть 1] ZBXTRAP [адрес] [трап, часть 2]

где:

* [штамп времени] – штамп времени, используемый в элементах данных типа "Журнал (лог)";
* ZBXTRAP – заголовок, который указывает, что с этой строки начался новый трап;
* [адрес] – IP-адрес, используемый для поиска узла сети для этого трапа.

Следует обратить внимание, что ZBXTRAP и [адрес] при обработке отрезаются из сообщения. Если трап форматируется как-то иначе, Подсистема может разобрать такие трапы непредсказуемым образом.

Пример трапа:

11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost – ZBXTRAP 192.168.1.1 Link down on interface 2. Admin state: 1. Operational state: 2

приведет к следующему виду трапа для SNMP-интерфейса с IP=192.168.1.1:

11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"

localhost – Link down on interface 2. Admin state: 1. Operational state: 2

1. Требования к системе

В Подсистеме имеется "поддержка больших файлов" при работе с файлами SNMP-трапов. Максимальный размер файла, который Подсистема может прочитать, это 263 (8 эксабайт). Следует обратить внимание, что файловая система может иметь меньшее ограничение на максимальный размер файлов.

Подсистема не предоставляет какой-либо механизм ротации журналов – это должно быть обеспечено пользователем. Ротация журналов должна начинаться с переименования старого файла и только после этого удаления, чтобы никакие трапы не пропали:

1. Подсистема открывает файл с трапами с последней известной позиции и переходит к шагу в);
2. Подсистема проверяет, была ли выполнена ротация открытого в данный момент файла, сравнивая номера inode с заданным у файла трапов номером inode; если открытого файла нет, Подсистема сбрасывает последнюю позицию и переходит к шагу а);
3. Подсистема читает данные из открытого в данный момент файла и устанавливает новую позицию;
4. обрабатываются новые данные; если этот файл был ротирован, то он закрывается, и Подсистема переходит назад к шагу б);
5. если не было новых данных, Подсистема приостанавливается на 1 секунду и возвращается к шагу б).

Из-за реализации файла с трапами для Подсистемы требуется файловая система с поддержкой inode для того, чтобы различать файлы (эта информация берется из вызова stat()).

1. Примеры с использованием разных версий SNMP-протоколов

Этот пример использует snmptrapd и Bash-скрипт получателя для передачи трапов Серверу.

Настройка выполняется с помощью следующих действий:

1. настроить Подсистему, чтобы запускался SNMP-траппер, и указать файл с трапами; добавить в zabbix\_server.conf:

StartSNMPTrapper=1

SNMPTrapperFile=/tmp/my\_zabbix\_traps.tmp

1. загрузить Bash-скрипт в /usr/sbin/zabbix\_trap\_handler.sh:

curl -o /usr/sbin/zabbix\_trap\_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/6.0/Dockerfiles/snmptraps/alpine/conf/usr/sbin/zabbix\_trap\_handler.sh

При необходимости следует подправить в скрипте переменную ZABBIX\_TRAPS\_FILE. Для использования значения по умолчанию создать сначала родительскую директорию:

mkdir -p /var/lib/zabbix/snmptraps

1. добавить в snmtrapd.conf:

traphandle default /bin/bash /usr/sbin/zabbix\_trap\_handler.sh

1. создать SNMP-элемент данных TEST:

SNMP интерфейс узла сети с IP: 127.0.0.1

Ключ: snmptrap["linkup"]

Формат времени в журнале (логе): yyyyMMdd.hhmmss

1. настроить snmptrapd на выбранную версию SNMP-протокола и отправить тестовые трапы, используя утилиту snmptrap.

Протоколы SNMPv1 и SNMPv2 основываются на аутентификации по community-строке. В примере ниже используется "secret" как community- строка. У отправителей SNMP-трапов необходимо указать такое же значение.

Следует обратить внимание, что хотя SNMPv2 все еще широко используется в производственных средах, он не предлагает какое-либо шифрование и реальной аутентификации отправителя. Так как данные отправляются в виде обычного текста, то эти версии протоколов можно использовать только в безопасных средах, таких как приватные частные сети, и никогда не использовать в какой-либо общедоступной или в сторонних сетях.

SNMPv1 в настоящее время практически не используется, так как не поддерживает 64-битные счетчики, и считается устаревшим протоколом.

Чтобы включить прием SNMPv1- или SNMPv2-трапов, необходимо добавить следующую строку в snmptrapd.conf, заменив "secret" на значение строки SNMP community, указанное в отправителях SNMP-трапов:

authCommunity log,execute,net secret

Далее можно отправить тестовый трап, используя команду snmptrap. В этом примере используется типовой OID "link up":

snmptrap -v 2c -c secret localhost 0 linkUp.0

Протокол SNMPv3 решает проблемы безопасности SNMPv1/v2 и обеспечивает аутентификацию и шифрование. Можно использовать методы аутентификации MD5 или несколько вариантов SHA и шифры DES или несколько вариантов AES.

Чтобы включить прием SNMPv3, добавляют следующие строки в snmptrapd.conf:

createUser -e 0x8000000001020304 traptest SHA mypassword AES

authuser log,execute traptest

Следует обратить внимание на ключевое слово "execute", которое позволяет выполнять скрипты для этой модели безопасности пользователя.

snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x8000000001020304 localhost 0 linkUp.0

Примечание – Если требуется использовать надежные методы шифрования, такие как AES192 или AES256, рекомендуется net-snmp, начиная с версии 5.8. Возможно, потребуется пересобрать его с опцией "configure: --enable-blumenthal-aes". Более старые версии net-snmp не поддерживают AES192/AES256.

В обоих примерах в /var/lib/zabbix/snmptraps/snmptraps.log есть похожие строки:

20220805.102235 ZBXTRAP 127.0.0.1

UDP: [127.0.0.1]:35736->[127.0.0.1]:162

DISMAN-EVENT-MIB::sysUpTimeInstance = 0:0:00:00.00

SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0

Значение элемента данных в Подсистеме будет таким:

2022-08-05 10:22:352022-08-05 10:22:33

20220805.102233 UDP: [127.0.0.1]:35736->[127.0.0.1]:162

DISMAN-EVENT-MIB::sysUpTimeInstance = 0:0:00:00.00

SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0

1. Проверки IPMI

В Подсистеме можно наблюдать за состоянием и доступностью устройств Intelligent Platform Management Interface (IPMI). Для выполнения проверок по IPMI Сервер должен быть изначально сконфигурирован с поддержкой IPMI.

IPMI – стандартизованный интерфейс для удаленного управления "lights-out" или "out-of-band" компьютерными инфраструктурами. Он позволяет наблюдать за состоянием аппаратного обеспечения напрямую с так называемых карт управления "out-of-band" независимо от ОС или же от наличия питания на узле.

IPMI-мониторинг работает только с устройствами, имеющими поддержку IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP и т.п.).

Процесс IPMI-менеджер выполняет распределение проверок IPMI между IPMI-поллерами. Теперь узел сети всегда опрашивается только одним IPMI-поллером одновременно, уменьшая количество открытых подключений к BMC-контроллерам. Благодаря этим изменениям можно безопасно увеличивать количество IPMI-поллеров, не беспокоясь о перегрузке BMC-контроллеров. Процесс IPMI-менеджер автоматически запускается, если запускается хотя бы один IPMI-поллер.

1. Настройка

Узел сети необходимо настроить для обработки проверок IPMI. Нужно добавить интерфейс IPMI с соответствующими IP-адресом и номером порта, а также задать параметры аутентификации IPMI.

По умолчанию Сервер не запускает IPMI-поллеры, поэтому любые добавленные элементы данных IPMI не будут работать. Чтобы изменить это, нужно открыть файл конфигурации (zabbix\_server.conf) Сервера из-под root и найти следующую строку:

# StartIPMIPollers=0

Далее раскомментировать эту строку и задать количество поллеров, например, равное 3:

StartIPMIPollers=3

Затем сохранить файл и перезапустить zabbix\_server.

Для настройки элемента данных на уровне узла сети:

1. в поле Тип выбрать "IPMI Агент";
2. ввести ключ элемента данных, уникальный в пределах узла сети (например, ipmi.fan.rpm);
3. в поле "Интерфейс узла сети" выбрать подходящий IPMI-интерфейс (IP и порт). Следует обратить внимание, что IPMI-интерфейс должен уже существовать на узле сети;
4. указать "IPMI датчик", с которого нужно забирать метрику (например, "FAN MOD 1A RPM" на Dell Poweredge). По умолчанию необходимо указать ID датчика. Также имеется возможность использования префиксов до самого значения:
   * 1. id: – чтобы указать ID датчика;
     2. name: – чтобы указать полное имя датчика. Эта опция может быть полезна в ситуациях, когда датчики можно отличить, только указав полное имя;
5. выбрать соответствующий тип информации ("Числовой (с плавающей точкой)" – в данном случае; для дискретных датчиков – "Числовой (целое положительное)"), единицы измерения (например, "rpm") и любые другие требуемые параметры элемента данных.

Таблица 75 описывает встроенные элементы данных, которые поддерживаются в проверках IPMI-агента.

Ключ в проверках IPMI-агента

|  |  |  |  |
| --- | --- | --- | --- |
| ▲ | Описание | Возвращаемое значение | Комментарии |
| ipmi.get | | | |
|  | Информация, связанная с IPMI-датчиком. | Объект JSON | Этот элемент данных можно использовать для обнаружения IPMI-датчиков. |

1. Время ожидания и завершение сессии

Время ожидания IPMI-сообщений и количество попыток определены в библиотеке OpenIPMI. В связи с текущим дизайном OpenIPMI невозможно сделать эти значения настраиваемыми из Подсистемы ни на уровне интерфейса, ни на уровне элемента данных.

Время ожидания неактивности IPMI-сессии для LAN равняется 60±3 секунды. В Подсистеме невозможно реализовать периодическую отправку команды активации сессии в OpenIPMI. Если проверки IPMI-элементов данных от Подсистемы к конкретному BMC не выполняются в течение времени большего, чем время ожидания сессии, настроенное в BMC, то следующая проверка IPMI после истечения времени ожидания приведет к ошибкам из-за превышения времени ожидания отдельного сообщения, повторных попыток или к ошибке при получении. После этого открывается новая сессия и инициируется полное повторное сканирование BMC. Если требуется избежать лишнего сканирования BMC, рекомендуется установить интервал опроса IPMI-элементов данных ниже времени ожидания неактивности IPMI-сессии, настроенного в BMC.

1. Дискретные датчики IPMI

Для поиска датчиков на узле сети запускают Сервер с включенным DebugLevel=4. После ожидания в течение двух минут можно поискать записи об обнаруженных датчиках в журнале Сервера:

$ grep "Added sensor" zabbix\_server.log

8358:20130318:111122.170 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:7 id:"CATERR" reading\_type:0x3 ("discrete\_state") type:0x7 ("processor") full\_name:"(r0.32.3.0).CATERR"

8358:20130318:111122.170 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:15 id:"CPU Therm Trip" reading\_type:0x3 ("discrete\_state") type:0x1 ("temperature") full\_name:"(7.1).CPU Therm Trip"

8358:20130318:111122.171 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:17 id:"System Event Log" reading\_type:0x6f ("sensor specific") type:0x10 ("event\_logging\_disabled") full\_name:"(7.1).System Event Log"

8358:20130318:111122.171 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:17 id:"PhysicalSecurity" reading\_type:0x6f ("sensor specific") type:0x5 ("physical\_security") full\_name:"(23.1).PhysicalSecurity"

8358:20130318:111122.171 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:14 id:"IPMI Watchdog" reading\_type:0x6f ("sensor specific") type:0x23 ("watchdog\_2") full\_name:"(7.7).IPMI Watchdog"

8358:20130318:111122.171 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:16 id:"Power Unit Stat" reading\_type:0x6f ("sensor specific") type:0x9 ("power\_unit") full\_name:"(21.1).Power Unit Stat"

8358:20130318:111122.171 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:16 id:"P1 Therm Ctrl %" reading\_type:0x1 ("threshold") type:0x1 ("temperature") full\_name:"(3.1).P1 Therm Ctrl %"

8358:20130318:111122.172 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:16 id:"P1 Therm Margin" reading\_type:0x1 ("threshold") type:0x1 ("temperature") full\_name:"(3.2).P1 Therm Margin"

8358:20130318:111122.172 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:13 id:"System Fan 2" reading\_type:0x1 ("threshold") type:0x4 ("fan") full\_name:"(29.1).System Fan 2"

8358:20130318:111122.172 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:13 id:"System Fan 3" reading\_type:0x1 ("threshold") type:0x4 ("fan") full\_name:"(29.1).System Fan 3"

8358:20130318:111122.172 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:14 id:"P1 Mem Margin" reading\_type:0x1 ("threshold") type:0x1 ("temperature") full\_name:"(7.6).P1 Mem Margin"

8358:20130318:111122.172 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:17 id:"Front Panel Temp" reading\_type:0x1 ("threshold") type:0x1 ("temperature") full\_name:"(7.6).Front Panel Temp"

8358:20130318:111122.173 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:15 id:"Baseboard Temp" reading\_type:0x1 ("threshold") type:0x1 ("temperature") full\_name:"(7.6).Baseboard Temp"

8358:20130318:111122.173 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:9 id:"BB +5.0V" reading\_type:0x1 ("threshold") type:0x2 ("voltage") full\_name:"(7.1).BB +5.0V"

8358:20130318:111122.173 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:14 id:"BB +3.3V STBY" reading\_type:0x1 ("threshold") type:0x2 ("voltage") full\_name:"(7.1).BB +3.3V STBY"

8358:20130318:111122.173 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:9 id:"BB +3.3V" reading\_type:0x1 ("threshold") type:0x2 ("voltage") full\_name:"(7.1).BB +3.3V"

8358:20130318:111122.173 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:17 id:"BB +1.5V P1 DDR3" reading\_type:0x1 ("threshold") type:0x2 ("voltage") full\_name:"(7.1).BB +1.5V P1 DDR3"

8358:20130318:111122.173 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:17 id:"BB +1.1V P1 Vccp" reading\_type:0x1 ("threshold") type:0x2 ("voltage") full\_name:"(7.1).BB +1.1V P1 Vccp"

8358:20130318:111122.174 Added sensor: host:"192.168.1.12:623" id\_type:0 id\_sz:14 id:"BB +1.05V PCH" reading\_type:0x1 ("threshold") type:0x2 ("voltage") full\_name:"(7.1).BB +1.05V PCH"

Для расшифровки типов датчиков IPMI и их состояний следует скачать копию спецификаций IPMI 2.0.

Для расшифровки кода "тип\_чтения"("reading\_type") используют раздел "Table 42-1, Event/Reading Type Code Ranges" из спецификации. Большинство датчиков из приведенного примера имеют "reading\_type:0x1", означающих "порог" датчика. "Table 42-3, Sensor Type Codes" показывает, что "type:0x1" – датчик температуры, "type:0x2" – датчик напряжения, "type:0x4" – датчик частоты вращения вентилятора системы охлаждения и так далее. Пороговые датчики иногда называют "аналоговыми" датчиками, так как они измеряют непрерывные параметры, такие как температуру, напряжение, частоту вращения в минуту.

Другой пример – датчик с "reading\_type:0x3". В "Table 42-1, Event/Reading Type Code Ranges" поясняется, что коды типов чтения 02h-0Ch означают "Общий Дискретный" датчик. Дискретные датчики имеют до 15 возможных состояний (другими словами, до 15 значащих бит). Например, для датчика "CATERR" с "type:0x7" "Table 42-3, Sensor Type Codes" показывает, что этот тип обозначает "Процессор" и значение отдельных битов: 00h (наименьший значащий бит) – IERR (внутренняя ошибка процессора), 01h – перегрев процессора и т.д.

В примере есть несколько датчиков с "reading\_type:0x6f". Для этих датчиков "Table 42-1, Event/Reading Type Code Ranges" советуют использовать "Table 42-3, Sensor Type Codes" для расшифровки значений битов. Например, датчик "Power Unit Stat" имеет тип "type:0x9", который означает "Блок питания". Смещение 00h означает "Выключено/Обесточено". Другими словами, если младший значащий бит равен 1, то Сервер выключен. Для проверки этого бита можно воспользоваться функцией band с маской "1". Выражение триггера для предупреждения о выключенном Сервере может выглядеть следующим образом:

bitand(last(/www.example.com/Power Unit Stat,#1),1)=1

Имена дискретных датчиков в OpenIPMI-2.0.16, 2.0.17 и 2.0.18 зачастую имеют дополнительный символ "0" (или какую-то другую цифру или символ), добавленный в конце имени. Например, если ipmitool и OpenIPMI-2.0.19 отображают имена датчиков как "PhysicalSecurity" или "CATERR", то в OpenIPMI-2.0.16, 2.0.17 и 2.0.18 эти имена – "PhysicalSecurity0" или "CATERR0", соответственно.

При настройке элемента данных IPMI для Сервера, использующего OpenIPMI-2.0.16, 2.0.17 и 2.0.18, нужно добавить к их именам "0" в поле "IPMI датчик" для элементов данных IPMI-агента. Когда Сервер будет обновлен в новом Linux-дистрибутиве, использующем OpenIPMI-2.0.19 (или более позднюю), элементы данных с такими IPMI-дискретными датчиками перейдут в состояние "НЕ ПОДДЕРЖИВАЕТСЯ", и потребуется изменить их имена "IPMI датчик" (удалить "0" в конце) и подождать некоторое время, пока они станут "Активированными" снова.

Некоторые IPMI-агенты предоставляют одновременно пороговые и дискретные датчики под одним именем. Предпочтение всегда отдается пороговому датчику.

Если IPMI-проверки не выполняются (по любой из причин: все элементы данных IPMI деактивированы/не поддерживаются на узле сети; сам узел сети деактивирован/удален; узел сети находится в режиме обслуживания и так далее), то соединение будет разорвано со стороны Сервера и Прокси через 3-4 часа в зависимости от времени запуска Сервера/Прокси.

1. Простые проверки

Простые проверки в основном используются для удаленных безагентных проверок сервисов.

Следует обратить внимание, что для простых проверок Агент не требуется. За обработку простых проверок (создание внешних подключений и т.д.) отвечает Сервер/Прокси.

Примеры использования простых проверок:

net.tcp.service[ftp,,155]

net.tcp.service[http]

net.tcp.service.perf[http,,8080]

net.udp.service.perf[ntp]

Список поддерживаемых простых проверок приведен в таблице 76.

Ключи элемента данных простых проверок

| Описание | Возвращаемое значение | Параметры | Комментарии |
| --- | --- | --- | --- |
| icmpping[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>] | | | |
| Доступность хоста через пинг по ICMP. | 0 – ошибка при пинге по ICMP 1 – успешный пинг по ICMP | цель – IP хоста или DNS-имя пакеты – количество пакетов интервал – время между последовательными пакетами в миллисекундах размер – размер пакета в байтах время ожидания – время ожидания в миллисекундах | Пример: => icmpping[,4] – если хотя бы один пакет из четырех вернется, элемент данных возвратит 1. |
| icmppingloss[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>] | | | |
| Процентное отношение потерянных пакетов. | Число с плавающей точкой. | цель – IP хоста или DNS-имя пакеты – количество пакетов интервал – время между последовательными пакетами в миллисекундах размер – размер пакета в байтах время ожидания – время ожидания в миллисекундах |  |
| icmppingsec[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>,<режим>] | | | |
| Время ответа на пинг по ICMP (в секундах). | Число с плавающей точкой | цель – IP хоста или DNS-имя пакеты – количество пакетов интервал – время между последовательными пакетами в миллисекундах размер – размер пакета в байтах время ожидания – время ожидания в миллисекундах режим – один из min, max, avg (по умолчанию) | Потерянные пакеты либо пакеты с превышенным временем ожидания при вычислениях не учитываются. Если хост недоступен (превышено время ожидания), элемент данных вернет 0. Если возвращаемое значение меньше 0.0001 секунд, то оно будет выставлено в 0.0001 секунд. |
| net.tcp.service[сервис,<ip>,<порт>] | | | |
| Проверка запущен ли сервис и отвечает ли на TCP подключения. | 0 – сервис недоступен 1 – сервис работает | сервис – возможные значения: ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet  ip – IP-адрес или DNS-имя (по умолчанию, используется IP/DNS узла сети) порт – номер порта (по умолчанию используется стандартный для сервиса номер порта). | Пример: => net.tcp.service[ftp,,45] – можно использовать для проверки доступности FTP-сервера на 45 порту TCP. Следует обратить внимание, что для сервиса tcp обязательно нужно указывать порт. Эти проверки могут привести к дополнительным записям в системных лог файлах (обычно сессии SMTP и SSH журналируются). Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как обходной вариант решения, использовать net.tcp.service[tcp,<ip>,порт] для подобных проверок. |
| net.tcp.service.perf[сервис,<ip>,<порт>] | | | |
| Проверка производительности TCP-сервиса. | Число с плавающей точкой. 0.000000 – сервис недоступен; секунды – количество секунд, потребовавшихся для подключения к сервису | сервис – возможные значения: ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet  ip – IP-адрес или DNS-имя (по умолчанию, используется IP/DNS узла сети) порт – номер порта (по умолчанию используется стандартный для сервиса номер порта). | Пример ключа: => net.tcp.service.perf[ssh] – можно использовать для проверки скорости начального ответа от SSH-сервера. Следует обратить внимание, что для сервиса tcp обязательно нужно указывать порт. Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как обходной вариант решения, использовать net.tcp.service.perf[tcp,<ip>,порт] для подобных проверок. |
| net.udp.service[сервис,<ip>,<порт>] | | | |
| Проверка запущен ли сервис и отвечает ли на UDP-запросы. | 0 – сервис недоступен  1 – сервис работает | сервис – возможные значения: ntp  ip – IP-адрес или DNS-имя (по умолчанию, используется IP/DNS узла сети) порт – номер порта (по умолчанию используется стандартный для сервиса номер порта). | Пример: => net.udp.service[ntp,,45] – можно использовать для тестирования доступности NTP-сервиса на 45 порту UDP. |
| net.udp.service.perf[сервис,<ip>,<порт>] | | | |
| Проверка производительности UDP-сервиса. | Число с плавающей точкой. 0.000000 – сервис недоступен секунды – количество секунд, потребовавшихся на ожидание ответа от сервиса | сервис – возможные значения: ntp  ip – IP-адрес или DNS-имя (по умолчанию, используется IP/DNS узла сети) порт – номер порта (по умолчанию используется стандартный для сервиса номер порта). | Пример: => net.udp.service.perf[ntp] – можно использовать для тестирования времени ответа от NTP-сервиса. |

Для поддержки SourceIP в простых проверках LDAP (например в net.tcp.service[ldap]) требуется OpenLDAP версии 2.6.1 или новее.

Подсистема не будет обрабатывать простую проверку дольше Timeout (времени ожидания) секунд, заданном в файле конфигурации Сервера/Прокси.

1. ICMP-пинг

Для обработки ICMP-пингов Подсистема использует внешнюю утилиту fping.

Эта утилита не является частью дистрибутива Подсистемы и должна быть установлена дополнительно. Если утилиты нет, у нее выставлены неверные разрешения или ее размещение не совпадает с размещением, заданным в файле конфигурации Сервера/Прокси (параметры "FpingLocation"), ICMP-пинги (icmpping, icmppingloss, icmppingsec) не будут обрабатываться.

fping должен быть выполняемым под пользователем демонов и должен иметь "setuid root". Выполняют такие команды из-под root для выставления корректных разрешений:

chown root:zabbix /usr/sbin/fping

chmod 4710 /usr/sbin/fping

После выполнения этих двух команд проверяют владельца исполняемого файла fping. В некоторых случаях владелец может сброситься при выполнении команды chmod.

Также проверяют, принадлежит ли пользователь zabbix к группе zabbix, запустив команду:

groups zabbix

и, если нет, добавляют следующей командой:

usermod -a -G zabbix

Значения по умолчанию, ограничения и описания значений для параметров ICMP-проверок приведены в таблице 77.

Значения для параметров ICMP-проверок

| Параметр | Ед. изм. | Описание | Флаг у fping | Значения по умолчанию у | | Разрешенные ограничения в Подсистеме | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | fping | Подсисема | мин | макс |
| пакеты | количество | количество запросов пакетов к цели | -C |  | 3 | 1 | 10000 |
| интервал | миллисекунды | время ожидания между последовательными пакетами | -p | 1000 |  | 20 | без ограничений |
| размер | байты | размер пакета в байтах 56 байтов на x86, 68 байтов на x86\_64 | -b | 56 или 68 |  | 24 | 65507 |
| время ожидания | миллисекунды | fping v3.x – время ожидания после отправки последнего пакета, на которое влияет флаг -C fping v4.x – отдельное время ожидания по каждому пакету | -t | fping v3.x – 500 fping v4.x – наследуется от -p флага, но не более 2000 |  | 50 | без ограничений |

Кроме того, Подсистема использует опции "fping -i интервал мс" (не путать с параметром интервал элемента данных, упомянутым выше в таблице, который соответствует опции -p) и "-S исходящий IP-адрес" (или -I в более старых версиях fping). Эти параметры автоматически определяются при выполнении проверок с различными комбинациями опций. Подсистема пытается определить минимальное значение в миллисекундах, которое fping позволяет использовать с "-i", перебирая три значения: 0, 1 и 10. Первое удачное значение затем используется для последующих ICMP-проверок. Этот процесс выполняется каждым процессом ICMP-пинг отдельно.

Автоматически определенные опции fping аннулируются каждый час, и процесс определения выполняется снова при следующей попытке выполнения ICMP-проверки. Следует задать DebugLevel>=4 для просмотра сведений об этом процессе в файлах журналов Сервера или Прокси.

Примечание – Значения по умолчанию для fping могут различаться в зависимости от платформы и версии. См. документацию по fping.

Подсистема записывает IP-адреса, проверяемые по всем трем ключам icmpping\*, во временный файл, который затем передается утилите fping. Если элементы данных имеют различные параметры ключа, то только элементы данных с идентичными параметрами ключа записываются в один файл.  
Все записанные в один файл IP-адреса проверяются утилитой fping в параллельном режиме, таким образом процесс "icmp pinger" тратит фиксированное время вне зависимости от количества IP-адресов в файле.

1. Мониторинг файлов журналов

Подсистему можно использовать для централизованного мониторинга и анализа файлов журналов с поддержкой или без поддержки ротации журналов.

Можно использовать оповещения для предупреждения пользователей, если файл журнала содержит конкретные строки или шаблоны строк.

Для наблюдения за файлом журнала необходимы:

* Агент, работающий на узле сети;
* Настроенный элемент данных для мониторинга журнала.

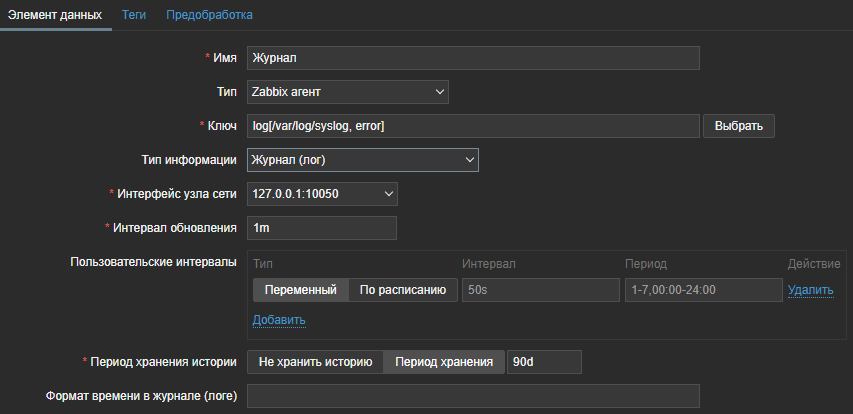
Максимальный размер наблюдаемого файла журнала зависит от поддержки больших файлов.

1. Настройка

Необходимо убедиться, что в файле конфигурации Агента:

* параметр "Hostname" совпадает с именем узла сети в веб-интерфейсе;
* в параметре "ServerActive" указаны Серверы для обработки активных проверок.

Далее задать параметры элемента данных для мониторинга журнала (рисунок 4).



Настройка элемента данных для мониторинга журнала

Примечание – Все обязательные поля во всех окнах ввода отмечены красной звездочкой.

Специально для элементов данных наблюдения за журналами необходимо указать параметры (таблица 78).

Параметры элементов данных

|  |  |
| --- | --- |
| Параметр | Описание |
| Тип | Выбрать Агент (активный). |
| Ключ | Использовать один из следующих ключей:  log[] или logrt[] – эти два ключа элементов данных позволяют выполнять мониторинг файлов журналов и фильтровать записи в журналах по их содержимому при помощи регулярного выражения, если оно присутствует. Например, log[/var/log/syslog,error]. Нужно убедиться, что у файла имеются права на чтение для пользователя zabbix, иначе состояние элемента данных будет "не поддерживается". log.count[] или logrt.count[] – эти два ключа элементов данных позволяют получить только количество соответствующих строк. |
| Тип информации | Заполняется автоматически: Для элементов данных log[] или logrt[] – Журнал (лог); для элементов данных log.count[] или logrt.count[] – Числовой (целое положительное). Если используется опциональный параметр вывод, можно вручную выбрать подходящий тип информации, отличный от "Журнал (лог)". Следует обратить внимание, что выбор не журнального типа информации приведет к потере локального штампа времени. |
| Интервал обновления (в сек) | Этот параметр определяет, как часто Агент будет проверять наличие любых изменений в файле журнала. Указав этот параметр равным 1 секунде, можно получить новые записи незамедлительно. |
| Формат времени журнала | В этом поле можно опционально задать шаблон для анализа штампа времени строки журнала. Если оставить пустым, штамп времени не будет анализироваться. Поддерживаемые заменители: \* y: Год (0001-9999) \* M: Месяц (01-12) \* d: День (01-31) \* h: Час (00-23) \* m: Минута (00-59) \* s: Секунда (00-59) Например, следующая строка из файла журнала Агента: " 23480:20100328:154718.045 Zabbix agent Started. Zabbix 1.8.2 (revision 11211)." Она начинается с шести символьных позиций для PID, далее следуют дата, время и остальная часть строки. Форматом времени журнала для такой строки могло бы являться "pppppp:yyyyMMdd:hhmmss". Следует обратить внимание, что символы "p" и ":" являются лишь заменителями и могут быть любыми символами, за исключением "yMdhms". |

Примечания:

* Сервер и Агент следят за размером наблюдаемого журнала и временем последней модификации (для logrt) при помощи двух счетчиков, кроме того:
* Агент также использует номера inode (на UNIX/GNU/Linux), индексы файлов (на Microsoft Windows) и MD5 суммы первых 512 байтов файла журнала для улучшения выбора в случае, когда файлы журнала усекаются и ротируются.
* На ОС UNIX/GNU/Linux предполагается, что файловые системы, где хранятся файлы журналов, сообщают номера inode-ов, которые могут быть использованы для отслеживания файлов.
* На ОС Microsoft Windows Агент определяет тип файловой системы, на которой находятся файлы журналов, и использует:
* на файловых системах NTFS – 64-битные файловые индексы;
* на файловых системах ReFS (только с Microsoft Windows Server 2012) – 128-битные ID файлов;
* на файловых системах, в которых файловые индексы меняются (т.е. FAT32, exFAT), используется запасной алгоритм для получения разумного подхода в неопределенных условиях, когда ротация файла журнала приводит в результате к множеству файлов журналов с одинаковым временем изменения.
* Номера inode, индексы файлов и суммы MD5 собираются Агентом. Они не передаются Серверу и теряются в случае остановки Агента.
* Не стоит менять время последней модификации файлов журналов утилитой touch, не копировать файл журнала с последующим восстановлением его имени (это изменит идентификатор inode-файла). В обоих случаях файл будет рассматриваться как другой и будет проанализирован с самого начала, что может привести к дубликатам оповещений.
* Если имеется несколько совпадающих файлов журналов для элемента данных logrt[], и Агент следит за наиболее новым из них, и этот более новый файл журнала удаляется, то будет записано сообщение с предупреждением: "There are no files matching '<шаблон регулярного выражения>' in '<директория>'". Агент игнорирует файлы журналов со временем модификации меньшим, чем последнее время модификации, полученное Агентом во время проверки элемента данных logrt[].
* Агент начинает читать файл журнала с той позиции, на которой он остановился последний раз.
* Количество уже проанализированных байтов (счетчик размера) и время последней модификации (счетчик времени) сохраняются в БД и отправляются Агенту, чтобы убедиться, что Агент начнет читать файл журнала с этой позиции в случаях, когда Агент только что был запущен или Агент получил элементы данных, которые были ранее деактивированы или не поддерживались. Если Агент получает ненулевой счетчик размера от Сервера, но элементу данных logrt[] или logrt.count[] не удается найти соответствующие файлы, то счетчик размера сбрасывается в 0, чтобы начать анализ с самого начала, если файлы появятся позже.
* Всякий раз, когда файл журнала становится меньше, чем известное Агенту значение счетчика размера, счетчик обнуляется, и Агент начинает читать файл журнала с самого начала, принимая во внимание счетчик времени.
* Если в директории есть несколько соответствующих файлов журналов с одинаковым временем последней модификации, Агент пытается корректно проанализировать все файлы журналов с одинаковым временем модификации и избежать пропуска данных или повторного анализа тех же данных, хотя этого и нельзя гарантировать во всех возможных ситуациях. Агент ни предполагает какую-либо определенную схему ротации файлов журналов, ни определяет ее. Когда есть несколько файлов журналов с одинаковым временем последнего изменения, Агент будет обрабатывать их лексикографически в порядке убывания. Таким образом, для некоторых схем ротации файлы журналов будут проанализированы в их оригинальном порядке. Для других же схем ротации журналов первоначальный порядок файла журнала не будет соблюдаться, что может привести к получению найденных по шаблону строк файла журнала в измененном порядке (проблема не возникает, если файлы журнала имеют разное время последней модификации).
* Агент обрабатывает новые записи файла журнала один раз за "Период обновления" секунд.
* Агент отправляет не более чем "макс. кол-во строк" записей из файла журнала за секунду. Это ограничение предотвращает перегрузку сети и ресурсов процессора и переопределяет значение по умолчанию, предусмотренное параметром MaxLinesPerSecond в файле конфигурации Агента.
* Для поиска необходимой строки Подсистема обработает в 10 раз больше строк, чем указано в параметре MaxLinesPerSecond. Таким образом, например, если элемент данных log[] или logrt[] имеет "Интервал обновления" в 1 секунду, то по умолчанию Агент за одну проверку проанализирует не более чем 200 строк файла журнала и отправит Серверу не более 20 совпавших записей. Увеличением параметра MaxLinesPerSecond в файле конфигурации Агента или указанием параметра "макс. кол-во строк" в ключе элемента данных лимит можно увеличить вплоть до 10000 проанализированных записей в журнале и 1000 совпадающих записей для отправки Серверу за одну проверку. Если для параметра "Интервал обновления" указано значение 2 секунды, лимиты для одной проверки могут быть установлены вдвое больше, чем для "Интервал обновления" в 1 секунду.
* Кроме того, значения log и log.count всегда ограничены 50% размера буфера отправки у Агента, даже если в буфере нет значений, не связанных с данными из файлов журналов. Таким образом, чтобы значения "макс. кол-во строк" были отправлены за одно подключение (а не за несколько подключений), параметр Агента BufferSize должен быть равен по крайней мере "макс. кол-во строк", умноженному на 2. Агент может отсылать данные во время сбора данных из журналов и таким образом освобождать буфер, в то время как Агент-2 остановит сбор данных из журналов до тех пор, пока данные не будут отосланы и буфер освобожден, что выполняется асинхронно.
* При отсутствии элементов данных журналов весь размер буфера используется для значений, не связанных с данными из журналов. Когда появляются значения от файлов журналов, они при необходимости заменяют более старые данные, не связанные с файлами журналов, до максимально предусмотренного уровня 50%.
* Если в файле журнала строка длиннее 256 КБ, то только первые 256 КБ сопоставляются с регулярным выражением, остальная часть игнорируется. Но если Агент был остановлен в процессе обработки длинной строки, внутреннее состояние Агента теряется и после перезапуска Агента длинная строчка может быть проанализирована заново и иначе.
* Специальное примечание для разделителей пути "\": если формат\_файла представлен как "file\.log", то там не должно быть директории "file", поскольку невозможно однозначно определить, экранируется ли это символ "." или же точка является первым символом в имени файла.
* Регулярные выражения для logrt поддерживаются только в именах файлов, совпадение регулярного выражения с директорией не поддерживается.
* На платформах UNIX элементы данных logrt[] становятся неподдерживаемыми в случае, если директория, где должен был бы находиться файл журнала, не существует.
* В Microsoft Windows, если директория не существует, то элемент данных не переводится в состояние "НЕ ПОДДЕРЖИВАЕТСЯ" (например, если в ключе элемента данных директория указана с ошибкой).
* Отсутствие файла журнала для элемента данных logrt[] не переводит его в состояние "НЕ ПОДДЕРЖИВАЕТСЯ". Ошибки чтения файлов журналов для элемента данных logrt[] записываются в журнал Агента как предупреждения, но не переводят элемент данных в состояние "НЕ ПОДДЕРЖИВАЕТСЯ".
* Журнал Агента может быть очень полезен для поиска причин, почему элементы данных log[] или logrt[] становятся неподдерживаемыми. Подсистема может мониторить свой файл журнала Агента, за исключением случая, когда он в режиме DebugLevel=4 или DebugLevel=5.
* Поиск знака вопроса при помощи регулярного выражения, например: "\?", может давать ложные срабатывания, если текстовый файл содержит символы NUL, поскольку они заменяются Подсистемой на "?", чтобы продолжать обрабатывать строку до символа перевода строки.

1. Извлечение совпадающей части регулярного выражения

Иногда при обнаружении совпадения с регулярным выражением требуется извлечь из требуемого файла только интересующие значения вместо получения всей строки.

Элементы данных файлов журналов имеют возможность извлекать из строк файла желаемые значения. Это достигается с помощью дополнительного параметра "вывод" у элементов данных log и logrt.

Использование параметра "вывод" позволяет обозначить рассматриваемую "подгруппу совпадения".

Например,

log[/путь/к/файлу,"large result buffer allocation.\*Entries: ([0-9]+)",,,,\1]

позволит получить количество элементов (Entries) из следующего содержимого:

Fr Feb 07 2014 11:07:36.6690 \*/ Thread Id 1400 (GLEWF) large result

buffer allocation – /Length: 437136/Entries: 5948/Client Ver: >=10/RPC

ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement

Будет возвращено только число, так как "\1" ссылается только на первую и единственную интересующую подгруппу ([0-9]+).

Вместе с возможностью извлечения и получения числа значение можно использовать в определениях триггеров.

1. Использование параметра "максзадержка"

Параметр максзадержка в элементах данных журналов позволяет игнорировать некоторые более старые строки с целью получения наиболее новых строк, проанализированных в течение максзадержка секунд.

Параметр maxdelay>0 может привести к игнорированию важных записей в файлах журналов и пропуску оповещений. Этот параметр используют только по необходимости.

По умолчанию элементы данных мониторинга журналов забирают все новые строки, появляющиеся в файлах журналов. Но имеются приложения, которые в некоторых ситуациях начинают записывать огромное количество сообщений в свои файлы журналов. Например, если БД или DNS-сервер недоступны, то такие приложения могут заполнять файлы журналов тысячами практически идентичных сообщений об ошибке до тех пор, пока не восстановится нормальный режим работы. По умолчанию все эти сообщения анализируются и совпадающие строки оправляются на Сервер, как это настроено в элементах данных log и logrt.

Встроенная защита от перегрузки состоит из настраиваемого параметра "макс. кол-во строк" (защищает Сервер от получения слишком большого количества совпадающих строк в журнале) и ограничения в 10\*"макс. кол-во строк" (защищает CPU и I/O хоста от перегрузки Агентом за одну проверку). Тем не менее имеется две проблемы со встроенным механизмом защиты. Во-первых, на Сервер будет отправлено большое количество потенциально не очень информативных сообщений, которые займут место в БД. Во-вторых, по причине ограниченного количества строк, анализируемых в секунду, Агент может отставать на несколько часов от самых новых записей в журнале. Вполне вероятно, что необходимо оперативно получать информацию о текущей ситуации в файлах журналов вместо длительного анализа старых записей.

Решением обеих проблем является использование параметра максзадержка. Если параметр максзадержка>0, во время каждой проверки измеряются количество обработанных байтов, количество оставшихся байтов и время обработки. Отталкиваясь от этих чисел, Агент вычисляет предполагаемую задержку в секундах на анализ всех оставшихся в файле журнала записей.

Если задержка не превышает максзадержка, то Агент поступает с анализом файла журнала как обычно.

Если задержка больше, чем максзадержка, то Агент игнорирует часть файла журнала, пропуская эту часть к новой оценочной позиции таким образом, чтобы оставшиеся строки можно было проанализировать за максзадержка секунд.

Следует обратить внимание, что Агент не читает проигнорированные строки в буфер, а вычисляет приблизительную позицию для перехода в файле.

Факт пропуска строк в файле журнала записывается в файл журнала Агента примерно следующим образом:

14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,,120.0]"

LogFile:"/home/zabbix32/test1.log" skipping 679858 Bytes

(from Byte 75653115 to Byte 76332973) to meet maxdelay

Количество "to Byte" является приблизительным, потому что после перехода Агент скорректирует позицию в файле на начало строки в журнале, которая может быть в файле чуть дальше или ближе.

В зависимости от того, как скорость роста соотносится со скоростью анализа файла журнала, можно не увидеть полных переходов, а лишь фиксировать редкие или частые переходы, большие или маленькие переходы, или даже маленькие переходы по каждой проверке. Колебания загрузки системы и сетевых задержек также влияют на вычисления задержки и, следовательно, переходов вперед, чтобы не отставать от параметра максзадержка.

Не рекомендуется указывать максзадержка<"интервал обновления" (это может привести к частым маленьким переходам).

1. Обработка ротации "copytruncate" файлов журналов

Элемент данных logrt с опцией copytruncate подразумевает, что разные файлы журналов имеют разные записи (по крайней мере в них отличаются штампы времени), поэтому MD5-суммы начальных блоков (до первых 512 байт) будут отличаться. Два файла с одинаковыми MD5-суммами начальных блоков означают, что один из них оригинал, а второй – копия.

Элемент данных logrt с опцией copytruncate делает попытку корректной обработки копий файлов журналов без дубликатов сообщений. Тем не менее такие варианты, как: создание нескольких копий файлов журналов с одинаковыми штампами времени, ротация файлов журналов чаще чем интервал обновления logrt[] элемента данных, частый перезапуск Агента – не рекомендуются. Агент пытается справиться со всеми этими ситуациями, но хорошие результаты не гарантируются при любых обстоятельствах.

1. Постоянные файлы у log\*[]-элементов данных при нагрузке на I/O

Постоянный файл элементов данных обновляется после успешной отправки каждой партии данных (содержащей данные по элементам данных) на Сервер. Например, по умолчанию BufferSize равен 100. Если элемент данных журнала нашел 70 совпадающих записей, то первые 50 записей будут отправлены одной партией, постоянный файл будет обновлен, затем оставшиеся 20 записей будут отправлены второй партией (возможно, с некоторой задержкой на буферизацию большего количества данных), и затем постоянный файл будет снова обновлен.

1. Действия при ошибке связи между Агентом и Сервером

Каждая совпадающая строка с элементов данных log[] и logrt[] и результат проверки каждого элемента данных log.count[] и logrt.count[] требует свободного слота в выделенных 50% области буфера отправки в Агенте. Элементы буфера периодически отправляются Серверу (или Прокси), и слоты буфера становятся снова пустыми.

Пока имеются свободные слоты в выделенной области для журналов в буфере отправки в Агенте, и связь между Агентом и Сервером (или Прокси) нарушена, результаты мониторинга журналов накапливаются в буфере отправки. Такое поведение позволяет смягчить кратковременные нарушения связи.

Во время длительных нарушений связи все слоты журналов становятся занятыми и выполняются следующие действия:

* Проверки элементов данных log[] и logrt[] останавливаются. Когда связь восстановится и появятся свободные слоты, проверки возобновятся с предыдущей позиции. Совпадающие строки не будут потеряны, они просто будут отправлены позже.
* Проверки log.count[] и logrt.count[] останавливаются, если maxdelay=0 (по умолчанию). Такое поведение похоже на поведение элементов данных log[] и logrt[], описанное выше. Следует обратить внимание, что потеря связи может повлиять на результаты log.count[] и logrt.count[]. Например, одна проверка насчитывает 100 совпадающих строк в файле журнала, но по причине отсутствия свободных слотов в буфере проверка остановлена. Когда связь восстановится, Агент насчитает те же 100 совпадающих строк, а также 70 новых совпадающих строк. После этого Агент отправит количество равное 170, как будто они были найдены за одну проверку.
* Проверки log.count[] и logrt.count[] при максзадержка>0: если не было перехода во время проверки, то поведение аналогично описанному выше. Если все же был переход через строки файла журнала, то позиция после перехода сохранится, а вычисленный результат будет отброшен. Таким образом, Агент пытается не отставать от увеличивающегося файла журнала даже в случае проблем со связью.

1. Обработка ошибок компиляции и времени выполнения для регулярных выражений

Если регулярное выражение, использованное в элементах данных log[], logrt[], log.count[] или logrt.count[], не может быть откомпилировано библиотекой PCRE или PCRE2, то элемент данных переходит в неподдерживаемое состояние с сообщением об ошибке. Для продолжения мониторинга журнального элемента данных регулярное выражение необходимо исправить.

Если регулярное выражение компилируется успешно, но дает сбой во время выполнения (на некоторых или на всех записях файла журнала), то журнальный элемент данных остается поддерживаемым и мониторинг продолжается. Ошибка времени выполнения добавляется в файл журнала Агента (без исходной журнальной записи).

Частота добавления в журнал ограничена одной ошибкой времени выполнения на одну проверку, чтобы дать возможность Агенту контролировать свой собственный файл журнала. Например, если анализируются 10 записей, и 3 из них вызывают ошибки времени выполнения, связанные с регулярным выражениями, то в файл журнала Агента будет добавлена одна запись.

Исключение: если MaxLinesPerSecond=1 и Интервал обновления=1 (на каждую проверку разрешено анализировать только одну запись), то ошибки времени выполнения при обработке регулярных выражений вообще не пишутся.

В случае ошибок времени выполнения zabbix\_agentd записывает в файл журнала ключ элемента данных, а zabbix\_agent2 записывает ID элемента данных, чтобы помочь идентифицировать, какой из элементов данных имеет ошибки времени выполнения. В случае ошибок времени выполнения рекомендуется переделать регулярное выражение.

* 1. **Назначение постоянных файлов**

При запуске Агент получает список активных проверок от Сервера или Прокси. Для log\*[]-метрик Агент получает размер обработанного журнала и время модификации, чтобы найти место начала мониторинга файла журнала. В зависимости от реального размера файла журнала и времени модификации, полученных от файловой системы, Агент решает: либо продолжить мониторинг файла журнала с обработанного размера журнала, либо проанализировать файл журнала с самого его начала.

Работающий Агент поддерживает большой набор параметров для отслеживания между проверками всех наблюдаемых файлов журналов. Это состояние теряется в памяти при остановке Агента.

Новый опциональный параметр "постоянное\_хранилище" указывает директорию для хранения этого состояния в файле по log[], log.count[], logrt[] или logrt.count[]-элементам данных. Состояние элемента данных журнала восстанавливается из постоянного файла после перезапуска Агента.

Основным сценарием применения такого подхода является мониторинг файла журнала, который расположен на зеркалируемой файловой системе. До некоторого момента файл журнала записывается в обоих зеркалах. Затем зеркала разделяются. На активном зеркале копия файла журнала продолжает расти, получая новые записи. Агент анализирует их и отправляет на Сервер размер обработанного журнала и время модификации. На стороне пассивного зеркала копия файла журнала остается той же самой, находясь далеко позади активной копии. Затем ОС и Агент перезагружаются с пассивной копии. Полученные от Сервера Агентом размер обработанного журнала и время модификации могут быть недостоверными для ситуации с пассивной копией. Для продолжения наблюдения за файлом журнала с того же места, на котором Агент остановил обработку файла при разделении зеркала, Агент восстанавливает свое состояние из постоянного файла.

* 1. **Работа Агента с постоянным файлом**

При запуске Агент ничего не знает о постоянных файлах. Только после получения списка активных проверок от Сервера (Прокси) Агент видит, что некоторые элементы данных журналов необходимо поддерживать при помощи постоянных файлов в указанных директориях.

В процессе работы Агента постоянные файлы открыты на запись (с использованием fopen(имя\_файла,"w")) и перезаписываются последними данными. Шанс потери данных постоянного файла, если перезапись и разделение зеркала файловой системы совпадут, очень мал, специальной обработки такого случая нет. После записи в постоянный файл не следует принудительная синхронизация с носителями информации (fsync() не вызывается).

Перезапись последними данными выполняется после успешного сообщения на Сервер о найденной строке файла журнала или о метаданных (размер обработанного журнала и время модификации). Это может происходить так же часто, как и проверки каждого элемента данных, если файл журнала продолжает изменяться.

При остановке Агента не выполняется никаких специальных действий.

После получения списка активных проверок Агент помечает устаревшие постоянные файлы для удаления. Постоянный файл становится устаревшим, если:

* + 1. соответствующий элемент данных журнала более не наблюдается;
    2. элемент данных журнала перенастраивается на другое расположение "постоянного\_хранилища", нежели ранее.

Удаление выполняется с задержкой в 24 часа, так как файлы журналов в неподдерживаемом состоянии не включаются в список активных проверок, но могут стать поддерживаемыми позже, и в этом случае их постоянные файлы будут полезны.

Если Агент остановлен до истечения 24 часов, то устаревшие файлы не будут удалены, так как Агент более не получает со стороны Сервера информацию об их расположении.

Перенастройка "постоянного\_хранилища" элементов данных назад к старому значению расположения "постоянного\_хранилища", пока Агент остановлен, без удаления старого постоянного файла пользователем приведет к восстановлению состояния Агента из старого постоянного файла, что приведет к пропущенным сообщениям или к ложным оповещениям.

* 1. **Именование и расположение постоянных файлов**

Агент различает активные проверки по их ключам. Например, logrt[/home/zabbix/test.log] и logrt[/home/zabbix/test.log,] – разные элементы данных. Изменение в веб-интерфейсе элемента данных logrt[/home/zabbix/test.log,,,10] на logrt[/home/zabbix/test.log,,,20] приведет к удалению элемента данных logrt[/home/zabbix/test.log,,,10] из списка активных проверок Агента и созданию элемента данных logrt[/home/zabbix/test.log,,,20] (некоторые параметры сохраняются при изменениях в веб-интерфейсе / на стороне Сервера, но не на стороне Агента).

Имя файла состоит из MD5-суммы ключа элемента данных с длиной ключа, добавляемой для уменьшения возможных пересечений. Например, состояние элемента данных

logrt[/home/zabbix50/test.log,,,,,,,,/home/zabbix50/agent\_private]

будет храниться в постоянном файле c963ade4008054813bbc0a650bb8e09266.

Несколько элементов данных могут использовать одно и то же значение "постоянного\_хранилища".

"постоянное\_хранилище" указывается с учетом структуры конкретной файловой системы, точек монтирования и опций монтирования, а также настроек зеркалирования хранилища: постоянный файл должен располагаться на том же зеркале файловой системы, что и наблюдаемый файл журнала.

Если директорию "постоянного\_хранилища" не удается создать, или она не существует, или права доступа Агента не позволяют создавать/записывать/читать/удалять файлы, то элемент данных журнала становится неподдерживаемым.

Если права доступа к файлам постоянного хранилища отозваны во время работы Агента или возникают другие ошибки (например, заполнился диск), то ошибки вносятся в файл журнала Агента, но элемент данных не становится неподдерживаемым.

1. Вычисляемые элементы данных

При помощи вычисляемых элементов данных имеется возможность выполнять подсчеты на основании значений других элементов данных.

Вычисления могут использовать:

* единичные значения отдельных элементов данных;
* сложные фильтры для выбора нескольких элементов данных для последующей агрегации.

Таким образом, вычисляемые элементы – это способ создания виртуальных источников данных. Все вычисления выполняются только на стороне Сервера. Значения периодически рассчитываются на основе использованного арифметического выражения.

Полученные данные записываются в БД Подсистемы так же, как и по любым другим элементам данных; хранятся значения истории и динамики изменений; можно сгенерировать графики.

Если результат вычисления – значение числа с плавающей точкой, то оно будет обрезаться до целого числа, если типом информации вычисляемого элемента данных выбрано "Числовой (целое положительное)".

Вычисляемые элементы данных имеют общий синтаксис с выражениями триггеров. В вычисляемых элементах данных допускается сравнение со строками. На вычисляемые элементы данных можно ссылаться макросами или другими такими же объектами, как и в случае любого другого типа элементов данных.

Для использования вычисляемых элементов данных выбирают Вычисляемый тип элементов данных.

1. Настраиваемые поля

Ключ – уникальный идентификатор элемента данных (в пределах узла сети). Можно создать любое имя ключа, используя поддерживаемые символы.

Формулу вычисления необходимо ввести в поле Формула. Не существует связи между формулой и ключом. Параметры ключа не используются в формуле ни коим образом.

Синтаксис простой формулы:

функция(/узел сети/ключ,<параметр1>,<параметр2>,...)

где:

* функция – одна из поддерживаемых функций: last, min, max, avg, count и т.д.;
* узел сети – узел сети элемента данных, который используется в вычислении; текущий узел сети может быть опущен (например, функция(//ключ,параметр,...));
* ключ – ключ элемента данных, который используется в вычислении;
* параметр(ы) – параметры функции, если требуются.

Пользовательские макросы в формуле будут раскрыты, если используется ссылка на параметр функции, параметр фильтра элементов данных или константу. Пользовательские макросы не будут раскрыты при использовании ссылки на функцию, имя узла сети, ключ элемента данных, параметры ключа элементов данных или оператор.

Для получения более сложной формулы можно использовать комбинацию функций, операторов и скобок. Можно использовать все функции и операторы, которые поддерживаются в выражениях триггеров. Логика и приоритет операторов точно такие же.

В отличие от выражений триггеров, Подсистема обрабатывает вычисляемые элементы данных в соответствии с интервалом обновления элемента данных, а не при получении нового значения.

Все элементы данных, на которые имеются ссылки из функций истории в формуле вычисляемого элемента данных, должны существовать и собирать данные. Также при изменении ключа элемента данных, на который ссылается вычисляемый элемент данных, нужно вручную обновить все формулы, в которых этот ключ используется.

Вычисляемый элемент данных может перейти в неподдерживаемое состояние в нескольких случаях:

* элемент(ы) данных, на которые имеется ссылка:
* не найден;
* деактивирован;
* принадлежит деактивированному узлу сети;
* не поддерживается (за исключением функции nodata() и операторов с неизвестными значениями);
* отсутствуют данные для вычисления функции;
* деление на ноль;
* использован некорректный синтаксис.

Примеры использования

**Пример 1.** Вычисление процентного отношения свободного места на диске для "/".

Использование функции last:

100\*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])

Подсистема будет брать последние значения полного и свободного объема диска и затем вычислять процентное отношение в соответствии с заданной формулой.

**Пример 2.** Вычисление 10-минутного усреднения количества значений, обрабатываемых Подсистемой.

Использование функции avg:

avg(/Zabbix Server/zabbix[wcache,values],10m)

Следует обратить внимание, что интенсивное использование вычисляемых элементов данных с большими периодами времени может повлиять на производительность Сервера.

**Пример 3.** Вычисление общей пропускной способности на eth0.

Сумма двух функций:

last(//net.if.in[eth0,Bytes])+last(//net.if.out[eth0,Bytes])

**Пример 4.** Вычисление процентного отношения входящего трафика.

Более сложное выражение:

100\*last(//net.if.in[eth0,Bytes])/(last(//net.if.in[eth0,Bytes])+last(//net.if.out[eth0,Bytes]))

1. Агрегированные вычисления

Агрегированные вычисления – это тип вычисляемых элементов данных, который позволяет собирать информацию с нескольких элементов данных Сервером и затем вычислять совокупность в зависимости от использованной функции агрегирования.

Для элементов данных агрегированного вычисления поддерживаются только целочисленные значения и значения с плавающей точкой (тип информации).

Агрегированные вычисления не требуют наличия какого-либо Агента, запущенного на наблюдаемом узле сети.

Для получения агрегирования можно:

* перечислить элементы данных для агрегации:

функция\_агрегации(функция(/узел\_сети/ключ,параметр),функция(/узел\_сети2/ключ2,параметр),...)

Следует обратить внимание, что здесь функция должна быть функцией истории/динамики изменений.

* использовать функцию foreach в качестве единственного параметра и ее фильтр элементов данных для выбора требуемых элементов данных:

функция\_агрегации(функция\_foreach(/узел\_сети/ключ?[group="узел сети"],период\_времени))

Функция агрегации – одна из поддерживаемых функций агрегации: avg, max, min, sum и т.д.

Функция foreach (такая как avg\_foreach, count\_foreach и т.п.) возвращает одно совокупное значение по каждому выбранному элементу данных. Элементы данных выбираются с использованием фильтра элементов данных (/узел\_сети/ключ?[group="группа узлов сети"]) из истории элементов данных.

Если некоторые элементы данных не имеют данных за требуемый период, они игнорируются в вычислении. Если никакие элементы данных не имеют данных, то функция вернет ошибку.

Для получения более подробных сведений о функции foreach (см. п.Ж.1).

Если в результате агрегации получится значение числа с плавающей точкой, оно будет обрезаться до целого числа, если типом информации вычисляемого элемента данных выбрано "Числовой (целое положительное)".

Агрегированное вычисление может перейти в неподдерживаемое состояние в нескольких случаях:

* ни один из указанных элементов данных не найден (что может произойти, если ключ элемента данных некорректный, элементы данных не существуют или все включенные в вычисление группы узлов сети некорректны);
* отсутствуют данные для вычисления функции.

**Примеры ключей для агрегированных вычислений**

Пример 1. Общий размер дискового пространства в группе узлов сети "MySQL Servers".

sum(last\_foreach(/\*/vfs.fs.size[/,total]?[group="MySQL Servers"]))

Пример 2. Сумма последних значений на узле сети всех элементов данных, совпадающих с net.if.in[\*].

sum(last\_foreach(/host/net.if.in[\*]))

Пример 3**.** Средняя загрузка процессора в группе узлов сети "MySQL Servers".

avg(last\_foreach(/\*/system.cpu.load[,avg1]?[group="MySQL Servers"]))

Пример 4. Усреднение за 5 минут количества запросов в секунду по группе узлов сети "MySQL Servers".

avg(avg\_foreach(/\*/mysql.qps?[group="MySQL Servers"],5m))

Пример 5.Средняя загрузка CPU по всем узлам сети в нескольких группах узлов сети, у которых имеются заданные теги.

avg(last\_foreach(/\*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (tag="Service:" or tag="Importance:High")]))

Пример 6.Вычисление с использованием суммы последних значений элементов данных по всей группе узлов сети.

sum(last\_foreach(/\*/net.if.out[eth0,Bytes]?[group="video"])) / sum(last\_foreach(/\*/nginx\_stat.sh[active]?[group="video"]))

Пример 7.Общее количество неподдерживаемых элементов данных в группе узлов сети "Zabbix servers".

sum(last\_foreach(/\*/zabbix[host,,items\_unsupported]?[group="Zabbix servers"]))

**Примеры корректного/некорректного синтаксиса.** Выражения (включая вызовы функций) нельзя использовать в качестве параметров к функциям истории, динамики изменений или foreach. Но сами эти функции могут быть использованы в параметрах других (не относящихся к истории) функций (таблица 79).

Примеры корректного/некорректного синтаксиса

| Выражение | Пример |
| --- | --- |
| Допустимое | avg(last(/host/key1),last(/host/key2)\*10,last(/host/key1)\*100) max(avg(avg\_foreach(/\*/system.cpu.load?[group="Servers A"],5m)),avg(avg\_foreach(/\*/system.cpu.load?[group="Servers B"],5m)),avg(avg\_foreach(/\*/system.cpu.load?[group="Servers C"],5m))) |
| Недопустимое | sum(/host/key,10+2) sum(/host/key, avg(10,2)) sum(/host/key,last(/host/key2)) |

Следует обратить внимание, что в таком выражении:

sum(sum\_foreach(//resptime[\*],5m))/sum(count\_foreach(//resptime[\*],5m))

нельзя гарантировать, что обе части всегда будут иметь одинаковый набор значений. Пока вычисляется одна часть выражения, может прийти новое значение за запрашиваемый период, и тогда вторая часть выражения будет иметь отличающийся набор значений.

1. Внутренние проверки

Внутренние проверки позволяют наблюдать за внутренним процессами Подсистемы. Другими словами, можно наблюдать, что происходит с Сервером или Прокси.

Внутренние проверки вычисляются:

* на стороне Сервера – если узел сети наблюдается через Сервер;
* на стороне Прокси – если узел сети наблюдается через Прокси.

Внутренние проверки обрабатываются Сервером или Прокси вне зависимости от состояния обслуживания узла сети.

Для использования этого элемента данных выбрать тип элемента данных Внутренний.

Внутренние проверки обрабатываются поллерами.

Использование некоторых внутренних элементов данных может отрицательно повлиять на производительность. Речь идет о следующих элементах данных:

* zabbix[host,,items];
* zabbix[host,,items\_unsupported];
* zabbix[hosts];
* zabbix[items];
* zabbix[items\_unsupported];
* zabbix[queue];
* zabbix[requiredperformance];
* zabbix[stats,,,queue];
* zabbix[triggers].

1. **Поддерживаемые проверки**

Параметры без угловых скобок являются константами, например, host и available в zabbix[host,<тип>,available]. Их используют в ключе элемента данных без изменения (как есть).

Значения элементов данных и параметров элементов данных, которые "не поддерживаются на Прокси" можно собирать только на узлах сети, которые наблюдаются через Сервер. И наоборот, значения "не поддерживается на Сервере" можно собирать, только если узел сети наблюдается через Прокси (таблица 80).

Описание поддерживаемых проверок

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ▲ | Описание | | | Возвращаемое значение | Комментарии |
| zabbix[boottime] | | | | | |
|  | Время запуска процесса Сервера или Прокси в секундах. | | | Целое число. | |
| zabbix[cluster,discovery,nodes] | | | | | |
|  | Обнаружение нод кластера высокой доступности | | | JSON. | Этот элемент данных можно использовать в низкоуровневом обнаружении. |
| zabbix[history] | | | | | |
|  | Количество значений, хранимых в таблице HISTORY | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL. (не поддерживается Прокси) |
| zabbix[history\_log] | | | | | |
|  | Количество значений, хранимых в таблице HISTORY\_LOG | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL. (не поддерживается Прокси) |
| zabbix[history\_str] | | | | | |
|  | Количество значений, хранимых в таблице HISTORY\_STR | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL. (не поддерживается Прокси) |
| zabbix[history\_text] | | | | | |
|  | Количество значений, хранимых в таблице HISTORY\_TEXT | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL. (не поддерживается Прокси) |
| zabbix[history\_uint] | | | | | |
|  | Количество значений, хранимых в таблице HISTORY\_UINT | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL. (не поддерживается Прокси) |
| zabbix[host,,items] | | | | | |
|  | Количество активированных элементов данных (поддерживаемых и неподдерживаемых) у узла сети. | | | Целое число. |  |
| zabbix[host,,items\_unsupported] | | | | | |
|  | Количество активированных неподдерживаемых элементов данных у узла сети. | | | Целое число. |  |
| zabbix[host,,maintenance] | | | | | |
|  | Возвращает текущее состояние обслуживания узла сети. | | | 0 – узел сети в нормальном состоянии, 1 – узел сети в обслуживании со сбором данных, 2 – узел сети в обслуживании без сбора данных. | Данный элемент данных всегда обрабатывается Сервером вне зависимости от настроек узла сети (мониторится через Сервер или Прокси). Прокси не получает этот элемент данных при получении своей конфигурации. Второй параметр должен быть пустым и зарезервирован для использования в будущем. |
| zabbix[host,discovery,interfaces] | | | | | |
|  | Детали по всем добавленным интерфейсам к узлу сети в веб-интерфейсе. | | | Объект JSON. | Этот элемент данных можно использовать в низкоуровневом обнаружении. (не поддерживается на Прокси) |
| zabbix[host,<тип>,available] | | | | | |
|  | Доступность основного интерфейса определенного типа проверок на узле сети. | | | 0 – недоступен, 1 – доступен, 2 – неизвестно. | Допустимые типы: agent, snmp, ipmi, jmx.  Значение элемента данных вычисляется согласно параметрам конфигурации доступности/недоступности касательно узлов сети. |
| zabbix[hosts] | | | | | |
|  | Количество наблюдаемых узлов сети | | | Целое число. | |
| zabbix[items] | | | | | |
|  | Количество активированных элементов данных (поддерживаемых и неподдерживаемых) | | | Целое число. | |
| zabbix[items\_unsupported] | | | | | |
|  | Количество неподдерживаемых элементов данных | | | Целое число. | |
| zabbix[java,,<параметр>] | | | | | |
|  | Получение информации, связанной с Java gateway. | | | Если <параметром> является ping, возвращается "1". Можно использовать для проверки доступности Java gateway, используя функцию триггера nodata(). Если <параметром> является version, возвращается версия Java gateway. | Допустимые значения для параметр: ping, version  Второй параметр должен быть пустым, т.к. зарезервирован для использования в будущем. |
| zabbix[lld\_queue] | | | | | |
|  | Количество значений, помещенных в очередь обработки низкоуровневых обнаружений. | | | Целое число. | Этот элемент данных можно использовать для мониторинга размера очереди обработки низкоуровневых обнаружений. |
| zabbix[preprocessing\_queue] | | | | | |
|  | Количество значений, помещенных в очередь предварительной обработки. | | | Целое число. | Этот элемент данных можно использовать для мониторинга размера очереди предварительной обработки. |
| zabbix[process,<тип>,<режим>,<состояние>] | | | | | |
|  | Время конкретного процесса или группы процессов (указываются с помощью <тип> и <режим>), потраченное на <состояние> в процентах. Расчет производится только за последнюю минуту.  Если в поле <режим> указан номер процесса Подсистемы, который не запущен (например, при 5 запущенных поллерах, в <режиме> указано 6), такой элемент данных перейдет в состояние "не поддерживается". Минимум и максимум относится к процентному использованию одним процессом. Таким образом, если в группе из 3 поллеров процентное использование составило 2, 18 и 66, то min должен вернуть 2, а max должен вернуть 66. Процессы сообщают то, что они делают, используя разделяемую память, и процесс само-диагностики обобщает эти данные каждую секунду. Изменения состояний (busy/idle) регистрируются при изменении – таким образом, процессы, которые становятся занятыми, регистрируют это состояние и не меняют, и не обновляют его, пока процесс не станет свободным. Такое поведение гарантирует, что даже полностью зависшие процессы будут корректно зарегистрированы как 100% занятые. В настоящее время, "busy" означает "не спит", но в будущем могут добавиться дополнительные состояния – ожидание блокировки, выполнение запросов в базу данных, и т.д. В Linux и в большинстве других систем, точность составляет 1/100 секунды. | | | Процент времени. Число с плавающей точкой. | Поддерживаемые типы процессов Сервера: alert manager, alert syncer, alerter, availability manager, configuration syncer, discoverer, escalator, ha manager, history poller, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, odbc poller, poller, preprocessing manager, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, trigger housekeeper, unreachable poller, vmware collector  Поддерживаемые типы процессов Прокси: availability manager, configuration syncer, data sender, discoverer, heartbeat sender, history poller, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, odbc poller, poller, preprocessing manager, preprocessing worker, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector  Допустимые режимы: avg – среднее значение по всем процессам указанного типа (по умолчанию) count – количество экземпляров указанного типа процесса, <состояние> не должно быть указано max – максимальное значение min – минимальное значение <номер процесса> – номер процесса (от 1 до количества экземпляров процессов). Например, если запущенно 4 траппера, то значение от 1 до 4.  Допустимые состояния: busy – процесс в занятом состоянии, например, обработка запроса (по умолчанию). idle – процесс в состоянии простоя, ничего не делающий.  Примеры: => zabbix[process,poller,avg,busy]– среднее время, потраченное процессами поллеров, которые что-либо делали за последнюю минуту => zabbix[process,"icmp pinger",max,busy] – максимальное время, потраченное любыми процессами ICMP pinger, которые что-либо делали за последнюю минуту => zabbix[process,"history syncer",2,busy]– время, потраченное на что-то процессом синхронизации истории номер 2 в течение последней минуты => zabbix[process,trapper,count] – общее количество запущенных процессов траппер |
| zabbix[proxy,<имя>,<параметр>] | | | | | |
|  | Информация доступности по Прокси. | | | Целое число. | имя: имя Прокси Допустимые значения для параметр: lastaccess – штамп времени последнего сообщения о доступности от Прокси delay – длительность задержки отправки собранных значений, рассчитывается как "задержка Прокси" (разница между текущим временем Прокси и отметкой времени самого старого неотправленного значения на Прокси) + ("текущее время сервера" – "последний доступ Прокси") Например, => zabbix[proxy,"Germany",lastaccess] Можно использовать функцию fuzzytime() для проверки доступности Прокси. Этот элемент данных всегда обрабатывается Сервером, независимо от расположения узла сети (на Сервере или Прокси). |
| zabbix[proxy\_history] | | | | | |
|  | Количество значений в таблице истории Прокси, ожидающих отправку на сервер | | | Целое число. | (не поддерживается сервером) |
| zabbix[queue,<от>,<до>] | | | | | |
|  | Количество наблюдаемых элементов данных в очереди, которые задерживаются минимум <от>, но менее <до> секунд. | | | Целое число. | от – по умолчанию: 6 секунд до – по умолчанию: бесконечно Для этих параметров поддерживаются суффиксы времени (s,m,h,d,w). |
| zabbix[rcache,<кэш>,<режим>] | | | | | |
|  | Статистика доступности кэша конфигурации | | | Целое число (для размера); число с плавающей точкой (для процентов). | кэш: buffer  Допустимые режимы: total – полный размер буфера free – размер свободного места в буфере pfree – процент свободного места в буфере used – размер использованного места в буфере pused – процент использования буфера |
| zabbix[requiredperformance] | | | | | |
|  | Требуемое быстродействие Сервера или Прокси, новых значений в секунду. | | | Число с плавающей точкой. | Приблизительно соответствует значению "Требуемое быстродействие Сервера, новые значения в секунду" со страницы "Отчеты → Информация о системе". |
| zabbix[stats,<ip>,<порт>] | | | | | |
|  | Внутренние метрики удаленного Сервера или Прокси. | | | Объект JSON. | ip – список IP/DNS/сетевой маски опрашиваемых удаленно Серверов / Прокси (по умолчанию 127.0.0.1) порт – порт опрашиваемого удаленно Сервера / Прокси (по умолчанию 10051) Следует обратить внимание, что запрос статистики будет приниматься только с адресов, которые перечислены в параметре "StatsAllowedIP" Сервера/Прокси на удаленном экземпляре. Этим элементом данных возвращается выбранный список внутренних метрик. |
| zabbix[stats,<ip>,<порт>,queue,<от>,<до>] | | | | | |
|  | Внутренние метрики очереди удаленного Сервера или Прокси. (см. zabbix[queue,<от>,<до>]). | | | Объект JSON. | ip – список IP/DNS/сетевой маски опрашиваемых удаленно Серверов / Прокси (по умолчанию 127.0.0.1) порт – порт опрашиваемого удаленно Сервера / Прокси (по умолчанию 10051) от – с задержкой по крайней мере (по умолчанию от 6 секунд) до – с задержкой не более (по умолчанию до бесконечности) Следует обратить внимание, что запрос статистики будет приниматься только с адресов, которые перечислены в параметре "StatsAllowedIP" Cервера / Прокси на удаленном экземпляре. |
| zabbix[tcache,cache,<параметр>] | | | | | |
|  | Статистика эффективности кэша функций динамики изменений. | | | Целое число (для размера); число с плавающей точкой (для процентов). | Допустимые значения для параметр: all – общее количество запросов кэша (по умолчанию) hits – количество попаданий в кэш phits – процент попаданий в кэш misses – количество непопаданий в кэш pmisses – процент непопаданий в кэш items – количество элементов данных в кэше requests – количество кэшируемых запросов pitems – процент элементов данных в кэше из кэшируемых элементов данных + запросов. Низкий процент зачастую означает, что размер кэша можно уменьшить. (не поддерживается Прокси) |
| zabbix[trends] | | | | | |
|  | Количество значений, хранимых в таблице TRENDS | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL! (не поддерживается Прокси) |
| zabbix[trends\_uint] | | | | | |
|  | Количество значений, хранимых в таблице TRENDS\_UINT | | | Целое число. | Не использовать с MySQL, InnoDB, Oracle или PostgreSQL. (не поддерживается Прокси) |
| zabbix[triggers] | | | | | |
|  | Количество активированных триггеров в базе данных Подсистемы со всеми активированными элементами данных, которые упомянуты в выражениях триггеров на активированных узлах сети. | | | Целое число. | (не поддерживается Прокси) |
| zabbix[uptime] | | | | | |
|  | Время непрерывной работы процесса Сервера или Прокси в секундах. | | | Целое число. | |
| zabbix[vcache,buffer,<режим>] | | | | | |
|  | Статистика доступности кэша значений. | | | Целое число (для размера); число с плавающей точкой (для процентов). | Допустимые значения режим: total – полный размер буфера free – размер свободного места в буфере pfree – процент свободного места в буфере used – размер использованного места в буфере pused – процент использованного места в буфере (не поддерживается Прокси) |
| zabbix[vcache,cache,<параметр>] | | | | | |
|  | Статистика эффективности кэша значений Подсистемы. | | | Целое число.  С параметром mode: 0 – нормальный режим, 1 – режим низкой памяти | Допустимые значения параметра: requests – общее количество запросов hits – количество попаданий в кэш (значения истории берутся из кэша) misses – количество непопаданий в кэш (значения истории берутся из базы данных) mode – режим работы кэша значений (не поддерживается Прокси) После переключения в режим низкой памяти, кэш значений будет оставаться в этом состоянии в течение 24 часов – даже если проблема, вызвавшая этот режим, будет решена раньше. Можно использовать этот ключ с шагом предобработки "Изменение в секунду" при необходимости получения статистики значений в секунду. |
| zabbix[version] | | | | | |
|  | Версия Сервера или Прокси. | | | Строка. | Пример возвращаемого значения: 1.0.0 |
| zabbix[vmware,buffer,<режим>] | | | | | |
|  | Статистика доступности кэша VMware. | | | Целое число (для размера); число с плавающей точкой (для процентов). | Допустимые значения для режим: total – полный размер буфера free – размер свободного места в буфере pfree – процент свободного места в буфере used – размер использованного места в буфере pused – процент использованного места в буфере |
| zabbix[wcache,<кэш>,<режим>] | | | | | |
|  | Статистика и доступность кэша записи Подсистемы. | | | | Требуется обязательно указывать <кэш>. |
|  | **Кэш** | **Режим** | | | |
| values | all (по умолчанию) | Общее количество значений, обрабатываемых Сервером или Прокси, исключая неподдерживаемые. | Целое число. | Счетчик. Можно использовать этот ключ с шагом предобработки "Изменение в секунду" при необходимости получения статистики значений в секунду. |
| float | Количество обработанных значений с плавающей точкой. | Целое число. | Счетчик. |
| uint | Количество обработанных целочисленных значений. | Целое число. | Счетчик. |
| str | Количество обработанных символьных/строковых значений. | Целое число. | Счетчик. |
| log | Количество обработанных значений из файлов журналов. | Целое число. | Счетчик. |
| text | Количество обработанных текстовых значений. | Целое число. | Счетчик. |
| not supported | Количество раз, когда обработка элементов данных приводила к тому, что элемент данных становился неподдерживаемым или сохранял это состояние. | Целое число. | Счетчик. |
| history | pfree (по умолчанию) | Свободное место в буфере истории в процентах. | Число с плавающей точкой. | Кэш истории используется для хранения значений элементов данных. Низкое количество свободного места отражает проблемы с производительностью на стороне базы данных. |
| free | Размер свободного места в буфере истории. | Целое число. | |
| total | Полный размер буфера. | Целое число. | |
| used | Размер занятого места в буфере истории. | Целое число. | |
| pused | Занятое место в буфере истории в процентах. | Число с плавающей точкой. |  |
| index | pfree (по умолчанию) | Свободное место в буфере индексов истории в процентах. | Число с плавающей точкой. | Кэш индексов истории используется для индексирования значений, записываемых в кэш истории. |
| free | Свободное место в буфере индексов истории. | Целое число. | |
| total | Полный размер буфера индексов истории. | Целое число. | |
| used | Размер занятого места в буфере индексов истории. | Целое число. | |
| pused | Занятое место в буфере индексов истории в процентах. | Число с плавающей точкой. |  |
| trend | pfree (по умолчанию) | Свободное место в буфере динамики изменений в процентах. | Число с плавающей точкой. | Кэш динамики изменений агрегируется за текущий час для всех элементов данных, которые получают данные. (не поддерживается Прокси) |
| free | Свободное место в буфере динамики изменений. | Целое число. | (не поддерживается Прокси) |
| total | Полный размер буфера динамики изменений. | Целое число. | (не поддерживается Прокси) |
| used | Размер занятого места в буфере динамики изменений. | Целое число | (не поддерживается Прокси) |
| pused | Занятое место в буфере динамики изменений в процентах. | Число с плавающей точкой. | (не поддерживается Прокси) |

1. Проверки через SSH

SSH-проверки выполняются без какого-либо Агента. Агент не требуется для проверок, выполняемых по SSH.

Для выполнения SSH-проверок Сервер должен быть изначально сконфигурирован с поддержкой SSH2 (libssh2 или libssh).

Начиная с RHEL 8, поддерживается только libssh. Для других дистрибутивов вместо libssh2 предлагается libssh.

Проверки SSH предоставляют два метода аутентификации: пара логин пользователя/пароль и на основе файла ключа.

Если не требуется использовать ключи, то (помимо привязки библиотеки libssh2/libssh к Подсистеме при компиляции из исходных кодов) никакой дополнительной настройки не требуется.

Чтобы использовать для элементов данных SSH-аутентификацию по ключу, необходимо произвести некоторые изменения в конфигурации Сервера:

1. открыть файл конфигурации Сервера zabbix\_server.conf из-под root и найти следующую строку:

# SSHKeyLocation=

1. раскомментировать ее и указать полный путь к директории, в которой размещены открытые и закрытые ключи (public and private keys):

SSHKeyLocation=/home/zabbix/.ssh

где:

* /home/zabbix – домашняя директория для учетной записи пользователя zabbix;
* .ssh – директория внутри нее, куда будут по умолчанию сгенерированы с помощью команды ssh-keygen открытые и закрытые ключи.

1. сохранить файл и перезапустить Сервер.

Обычно при установке Сервера из пакетов на разных дистрибутивах ОС создается аккаунт пользователя zabbix с домашней директорией в других местах (не как для системных аккаунтов), например /var/lib/zabbix.

До начала создания ключей следует рассмотреть вариант перемещения домашней директории в обычное место (интуитивно ожидаемое). Этот вариант будет соответствовать упомянутому выше параметру SSHKeyLocation конфигурации Сервера.

Эти шаги можно пропустить, если учетная запись zabbix добавлена вручную в соответствии с разделом установки, потому что в этом случае домашняя директория, скорее всего, уже расположена в /home/zabbix.

Для изменения этой настройки у учетной записи пользователя zabbix все работающие процессы, которые его используют, потребуется остановить:

service zabbix-agent stop

service zabbix-server stop

Чтобы изменить местоположение домашней директории с попыткой переместить ее (если директория существует), нужно выполнить команду:

usermod -m -d /home/zabbix

Возможно, что домашняя директория не существует в старом месте, поэтому в новом месте ее необходимо создать. Безопасно сделать это, выполнив:

test -d /home/zabbix || mkdir /home/zabbix

Для полной безопасности можно выполнить дополнительные команды для установки разрешений к домашней директории:

chown zabbix:zabbix /home/Zabbix

chmod 700 /home/zabbix

Далее можно снова запустить ранее остановленные процессы:

service zabbix-agent Start

service zabbix-server Start

Теперь шаги генерирования открытых и закрытых ключей можно выполнить следующими командами (для лучшей читабельности запросы команд закомментированы):

sudo -u zabbix ssh-keygen -t rsa

# Generating public/private rsa key pair.

# Enter file in which to save the key (/home/zabbix/.ssh/id\_rsa):

/home/zabbix/.ssh/id\_rsa

# Enter passphrase (empty for no passphrase):

<оставить пустым>

# Enter same passphrase again:

<оставить пустым>

# Your identification has been saved in /home/zabbix/.ssh/id\_rsa.

# Your public key has been saved in /home/zabbix/.ssh/id\_rsa.pub.

# The key fingerprint is:

# 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0

# The key"s randomart image is:

# +--[ RSA 2048]----+

# ||

# |.|

# |o|

# | .o|

# |+. S|

# |.+o =

|# |E .\* =|

# |=o . ..\* .|

# |... oo.o+|

# +-----------------+

Открытые и закрытые ключи (id\_rsa.pub и id\_rsa соответственно) будут сгенерированы по умолчанию в директории /home/zabbix/.ssh, которая соответствует параметру конфигурации SSHKeyLocation Сервера.

Утилитой ssh-keygen и Сервером SSH могут поддерживаться ключи и других типов, кроме rsa; но они могут не поддерживаться используемой Подсистемой библиотекой libssh2.

1. Диалог настройки командной строки

Этот шаг необходимо выполнить только один раз для каждого хоста, который будет наблюдаться с использованием SSH-проверок.

При использовании следующей команды файл открытого ключа может быть установлен на удаленный хост 10.10.10.10, для того чтобы потом можно было выполнять SSH-проверки при помощи аккаунта root (для лучшей читабельности подсказки команды закомментированы):

sudo -u zabbix ssh-copy-id [root@10.10.10.10](mailto:root@10.10.10.10)

# The authenticity of host "10.10.10.10 (10.10.10.10)" can"t be established.

# RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.

# Are you sure you want to continue connecting (yes/no)?

yes

# Warning: Permanently added "10.10.10.10" (RSA) to the list of known hosts.

# root@10.10.10.10"s password:

<Введите пароль пользователя root>

# Теперь попытайтесь зайти на машину с помощью "ssh "root@10.10.10.10"",

# и проверьте, чтобы убедиться, что не добавили лишние нежелательные ключи.

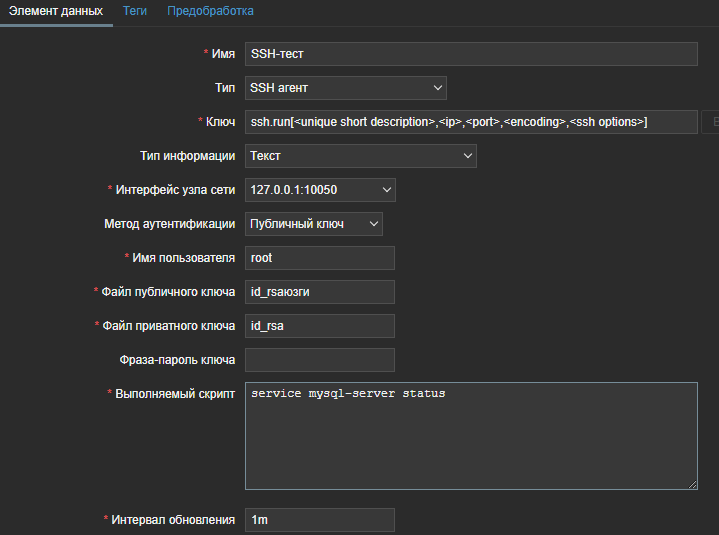
Теперь можно проверить вход по SSH с использованием закрытого ключа по умолчанию (/home/zabbix/.ssh/id\_rsa) учетной записи пользователя zabbix:

sudo -u zabbix ssh root@10.10.10.10

Если вход успешен, то часть настройки в командной строке завершена, и удаленную сессию по SSH можно закрыть.

1. Настройка элемента данных

Фактически выполняемые команды необходимо поместить в поле "Выполняемый скрипт" при настройке элемента данных. Можно выполнять несколько команд одну за другой, размещая каждую на новой строке. В этом случае возвращаемые значения также будут отформатированы как многострочные (рисунок 5).



Поле "Выполняемый скрипт"

Поля, которые требуют специфичную информацию для SSH-элементов данных, описаны в таблице 81.

Параметры для SSH-элементов данных

| Параметр | Описание | Комментарии |
| --- | --- | --- |
| Тип | Выбрать SSH-Агент |  |
| Ключ | Уникальный (в пределах узла сети) ключ элемента данных в формате ssh.run[уникальное короткое описание,<ip>,<порт>,<кодировка>,<опции\_ssh>] | "уникальное короткое описание" обязательно и должно быть уникальным у всех элементов данных типа SSH в рамках одного узла сети. По умолчанию порт 22, а не порт, указанный в интерфейсе узла сети, которому этот элемент данных назначен. опции\_ssh (требует libssh версии 0.9.0 или выше либо libssh2) позволяют передать дополнительные опции SSH в формате "ключ1=значение1;ключ2=значение2,значение3". Несколько значений для одного ключа можно передать через запятую (в этом случае параметр должен быть заключен в кавычки); несколько ключей опций можно передать через разделитель "точка с запятой". Поддерживаемые ключи и значения опций зависят от библиотеки SSH. Следует обратить внимание, что знак "+" для добавления настроек шифров и знак "!" для отключения конкретных настроек шифров (как в GnuTLS и OpenSSL) не поддерживаются. |
| Метод аутентификации | Один из "Пароль" или "Публичный ключ" |  |
| Имя пользователя | Имя пользователя для аутентификации на удаленном хосте. Требуется. |  |
| Файл публичного ключа | Имя файла открытого ключа, если "Метод аутентификации" задан значением "Публичный ключ". Требуется. | Например: id\_rsa.pub – имя по умолчанию файла открытого ключа, сгенерированного командой [ssh-keygen](http://en.wikipedia.org/wiki/Ssh-keygen) |
| Файл приватного ключа | Имя файла закрытого ключа, если "Метод аутентификации" задан значением "Публичный ключ". Требуется. | Например: id\_rsa – имя по умолчанию файла закрытого ключа |
| Пароль или Парольная фраза | Пароль для аутентификации или Парольная фраза, если она была использована для закрытого ключа | Оставить поле "Парольная фраза" пустым, если фраза не используется. |
| Выполняемый скрипт | Команда(-ы) оболочки, выполняемые с использованием удаленной сессии SSH | Значение, возвращаемое выполняемыми командами оболочки, ограничено 16 МБ (включая конечные пробелы, которые усекаются); также применяются ограничения базы данных. Следует обратить внимание, что библиотека libssh2 может усекать выполняемые скрипты до ~32 килобайт. Примеры: date +%s service mysql-server status ps auxww | grep httpd | wc -l |

1. Проверки через Telnet

Telnet-проверки выполняются без наличия какого-либо Агента. Агент не требуется для проверок через Telnet.

Фактические команды для выполнения необходимо поместить в поле "Выполняемый скрипт"при настройке элемента данных. Можно выполнять несколько команд одну за другой, размещая каждую на новой строке. В этом случае значение результата также будет отформатировано как многострочное.

Поддерживаемые символы, которыми может заканчиваться строка приглашения командной оболочки: $, #, >, %.

Строка telnet с запросом, которая заканчивается одним из этих символов, будет удалена из возвращаемого значения, но только для первой команды из списка команд, то есть только в начале telnet-сессии. Выполнение команды на удаленном устройстве с использованием telnet-соединения:

telnet.run[<уникальное короткое описание>,<ip>,<порт>,<кодировка>]

Если telnet-проверка возвращает значение с символами не-ASCII и не в кодировке UTF8, то необходимо указать корректный параметр ключа <кодировка>.

1. Внешние проверки

Внешняя проверка производится Сервером путем выполнения shell-скрипта или бинарного файла. Если узлы сети наблюдаются через Прокси, внешние проверки будут выполняться этим Прокси.

Внешние проверки не требуют на наблюдаемом узле сети какого-либо Агента.

Синтаксис ключа элемента данных:

скрипт[<параметр1>,<параметр2>,...]

где:

* скрипт – имя shell-скрипта или бинарного файла;
* параметр(ы) – опциональные параметры командной строки.

Если не требуется передавать какие-нибудь параметры скрипту, можно использовать:

script[]

или

script

Сервер проверит директорию, указанную для размещения внешних скриптов (параметр ExternalScripts в файле конфигурации Сервера), и выполнит заданную команду. Команда будет выполнена от имени пользователя, под которым запущен Сервер, так что любые права доступа или переменные среды должны быть обработаны в оболочке скрипта, если необходимо, и права доступа на команду должны разрешать этому пользователю выполнение скрипта. Для исполнения доступны только те команды, которые имеются в наличии в указанной директории.

Не следует часто использовать внешние проверки, так как каждый скрипт требует запуска Сервером дочернего процесса, а запуск множества скриптов может значительно снизить производительность Подсистемы.

Пример. Выполнение скрипта check\_oracle.sh с первым параметром "-h". Второй параметр будет заменен IP-адресом или DNS-именем узла сети в зависимости от выбранного в настройках узла сети:

check\_oracle.sh["-h","{HOST.CONN}"]

Если узел сети настроен на использование IP-адреса, то Подсистема выполнит:

check\_oracle.sh "-h" "192.168.1.4"

1. Результат внешней проверки

Возвращаемым значением внешней проверки является стандартный вывод вместе со стандартным выводом ошибок, производимые проверкой.

Элемент данных, возвращающий текст (символ,​ журнал или текстовый тип данных), не станет неподдерживаемым в случае появления чего-либо в стандартном выводе ошибок.

Возвращаемое значение ограничено 16 МБ (включая конечные пробелы, которые усекаются); также применяются ограничения БД.

В случае если выполняемый скрипт не найден или Сервер не имеет необходимых прав на его запуск, данный элемент станет неподдерживаемым, и отобразится соответствующее сообщение об ошибке.

В случае превышения времени ожидания элемент данных также станет неподдерживаемым, будет отображено соответствующее сообщение об ошибке, а отдельный процесс этого скрипта будет ликвидирован.

1. Траппер-элементы данных

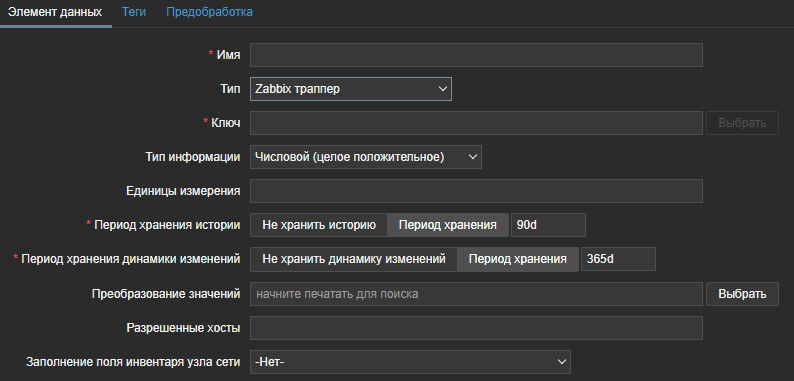
Траппер-элементы данных принимают входящие данные вместо опроса этих данных. Это полезно для любых данных, которые потребуется передать в Подсистему.

Для использования элемента данных траппера необходимо:

* настроить траппер-элемент данных;
* отправить данные в Подсистему.

Для настройки элемента данных траппер нужно:

1. перейти в "Настройка → Узлы сети***"***;
2. нажать на Элементы данных в строке с узлом сети;
3. нажать на Создать элемент данных;
4. ввести в окно параметры элемента данных (рисунок 6).



Настройка траппера

Поля, которые требуют указания специфичной информации для траппер-элемента данных:

* Тип – выбрать "Zabbix траппер**";**
* Ключ – ввести ключ, который будет использован для распознавания элемента данных при отправке данных.
* Тип информации – выбрать тип информации, который будет соответствовать формату отсылаемых данных.
* Разрешенные хосты – список разделенных запятыми IP-адресов, опционально в CIDR-представлении, или DNS-имен. Если задано, входящие подключения будут приниматься только с перечисленных здесь узлов сети. Если включена поддержка IPv6, то "127.0.0.1", "::127.0.0.1", "::ffff:127.0.0.1" обрабатываются одинаково и "::/0" будет разрешать любые IPv4 или IPv6-адреса. "0.0.0.0/0" можно использовать, чтобы разрешить любые IPv4-адреса.

Примечание – Возможно, нужно будет подождать до 60 секунд после сохранения элемента данных, пока Сервер подхватит изменения из обновленного кэша конфигурации, прежде чем получится посылать значения.

В самом простом случае можно использовать утилиту zabbix\_sender (см. п. Е.6) для отправки некоторого "тестового значения" траппер-элементу данных:

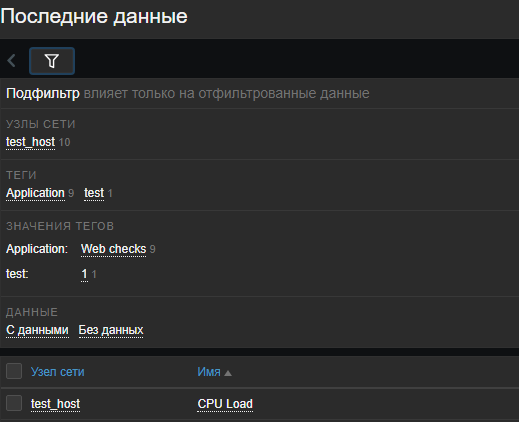
zabbix\_sender -z <IP-адрес сервера> -p 10051 -s "New host" -k trap -o "тестовое значение"

Для отправки значения используются следующие ключи, чтобы указать:

* -z – IP-адрес Сервера;
* -p – номер порта Сервера (10051 по умолчанию);
* -s – узел сети (следует убедиться, что используется "техническое" имя узла сети, вместо "видимого" имени);
* -k – ключ элемента данных, который добавляется;
* -o – фактически отправляемое значение.

​Процесс траппера не раскрывает макросы, которые используются в ключе элемента данных в попытке проверки наличия соответствующего ключа элемента данных на целевом узле сети.

Результат можно увидеть в "Мониторинг → Последние данные**"** (рисунок 7).



Последние данные

Следует обратить внимание, что, если отправлено одно числовое значение, на графике отобразится горизонтальная линия с левого до правого края от точки времени значения.

1. JMX-мониторинг

Мониторинг JMX можно использовать для мониторинга счетчиков JMX в Java-приложениях.

Когда Серверу нужно определить значение конкретного счетчика JMX у узла сети, он опрашивает Java gateway, который, в свою очередь, используя API- управление JMX, удаленно опрашивает интересующее приложение.

Подключения между Java gateway и наблюдаемым JMX-приложением не должны быть закрыты брандмауэром.

1. Включение удаленного JMX-мониторинга для Java-приложений

Приложению Java не требуется какое-либо дополнительно установленное ПО, но для поддержки удаленного мониторинга JMX-приложение необходимо запустить с указанными ниже параметрами командной строки.

Если требуется просто начать мониторинг приложения Java на локальном хосте без какой-либо защиты, его запускают со следующими опциями:

java \

-Dcom.sun.management.jmxremote \

-Dcom.sun.management.jmxremote.port=12345 \

-Dcom.sun.management.jmxremote.authenticate=false \

-Dcom.sun.management.jmxremote.ssl=false \

-Dcom.sun.management.jmxremote.registry.ssl=false \

-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar

С этими аргументами Java будет слушать входящие соединения JMX на порту 12345 только с локального хоста без необходимости в аутентификации или SSL.

Если нужно разрешить подключения с другого интерфейса, параметр "-Djava.rmi.server.Hostname" указывают равным IP-адресу этого интерфейса.

Если необходима более строгая проверка в плане безопасности, имеется много других доступных опций в Java. Следующий пример запускает приложение с более разносторонним набором опций и открывает это приложение для более широкой сети, а не только для локального компьютера:

java \

-Djava.rmi.server.Hostname=192.168.3.14 \

-Dcom.sun.management.jmxremote \

-Dcom.sun.management.jmxremote.port=12345 \

-Dcom.sun.management.jmxremote.authenticate=true \

-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \

-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \

-Dcom.sun.management.jmxremote.ssl=true \

-Dcom.sun.management.jmxremote.registry.ssl=true \

-Djavax.net.ssl.keyStore=$ВАШЕ\_ХРАНИЛИЩЕ\_КЛЮЧЕЙ \

-Djavax.net.ssl.keyStorePassword=$ВАШ\_ПАРОЛЬ\_К\_ХРАНИЛИЩУ\_КЛЮЧЕЙ \

-Djavax.net.ssl.trustStore=$ВАШЕ\_ДОВЕРЕННОЕ\_ХРАНИЛИЩЕ \

-Djavax.net.ssl.trustStorePassword=$ВАШ\_ПАРОЛЬ\_К\_ДОВЕРЕННОМУ\_ХРАНИЛИЩУ \

-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \

-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar

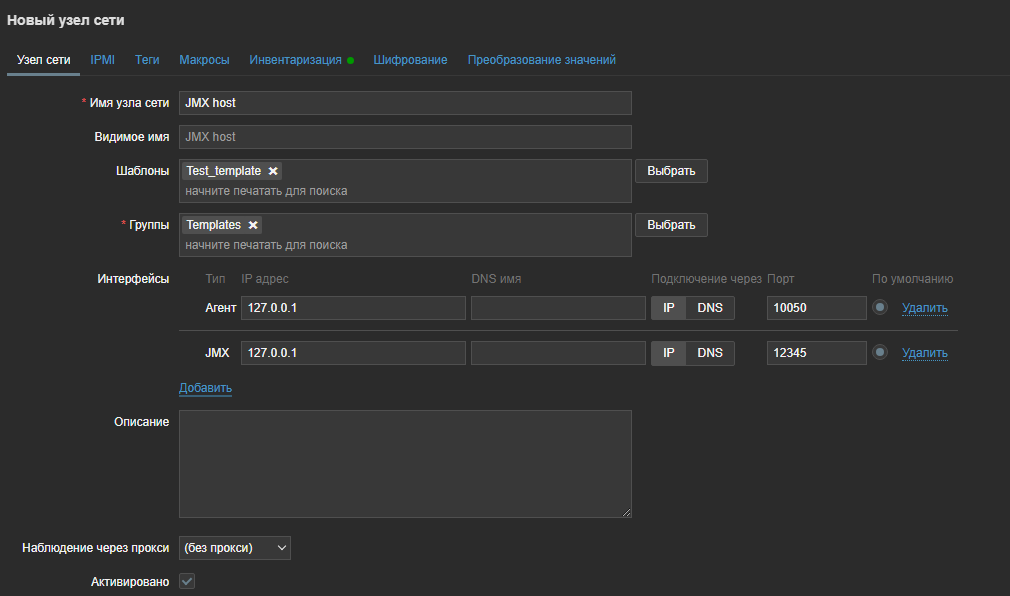
Значительное количество этих настроек можно указать в /etc/java-6-openjdk/management/management.properties (или в месте расположения файла на компьютере).

Следует обратить внимание, что, если требуется использовать SSL, нужно изменить скрипт Startup.sh Java gateway, добавив в него опции "-Djavax.net.ssl.\*" для обнаружения хранилища ключей и доверенного хранилища.

1. Настройка JMX-интерфейсов и элементов данных в веб-интерфейсе

Когда Java Gateway запущен, Сервер его определяет, и Java-приложение запущено с поддержкой удаленного JMX-мониторинга, настраивают интерфейсы и элементы данных в веб-интерфейсе Подсистемы.

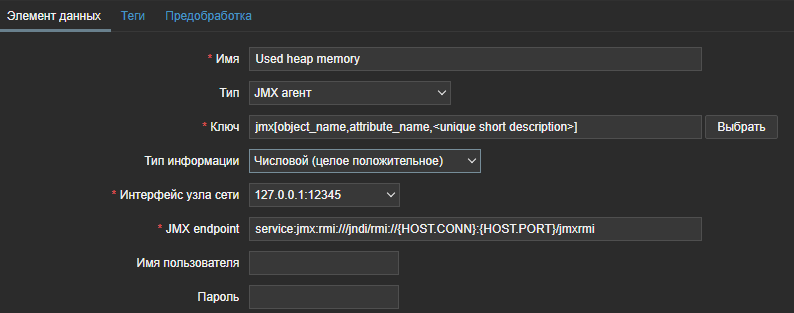
Начинают с создания интерфейса JMX-типа у требуемого узла сети (рисунок 8).



Настройка JMX-интерфейса

Для каждого требуемого счетчика JMX необходимо добавить элемент данных с типом "JMX Агент", присоединенный к этому интерфейсу.

Ключ на рисунке 9 имеет следующий вид: jmx["java.lang:type=Memory","HeapMemoryUsage.used"].



Добавление элемента данных

Поля, требующие специфичной информации для JMX-элементов данных, приведены в таблице 82.

Описание полей для JMX-элементов данных

|  |  |
| --- | --- |
| Параметр | Описание |
| Тип | Указать JMX-Агент. |
| Ключ | Ключ элемента данных jmx[] содержит три параметра: имя объекта – имя объекта MBean; имя параметра – имя параметра MBean с опциональными составными именами полей данных, разделенных точками. уникальное короткое описание – уникальное описание, которое позволит создать на одном узле сети несколько JMX элементов данных с одинаковыми именем объекта и именем параметра (опционально) |
| JMX endpoint | Можно указать пользовательский JMX endpoint. Убедитесь, что параметры подключения JMX endpoint совпадают с JMX-интерфейсом. Это можно сделать при помощи макросов {HOST.\*}, как уже сделано в JMX endpoint по умолчанию. |
| Имя пользователя | Указать имя пользователя, если настроена аутентификация Java-приложения. Поддерживаются пользовательские макросы. |
| Пароль | Указать пароль, если настроена аутентификация Java-приложения. Поддерживаются пользовательские макросы. |

Если нужно наблюдать за логическим счетчиком, который может быть true или false, следует указать тип информации "Числовой (целое положительное)" и добавить шаг предварительной обработки "Логический в десятичный" на вкладке Предобработка. Сервер будет сохранять логические значения как 1 или 0 соответственно.

1. Детальная информация о ключах JMX-элементов данных

Имя объекта MBean – это строка, которую определили в Java-приложении. Имя параметра, с другой стороны, может быть более сложным. В случае если параметр возвращает простой тип данных (число, строку и т.п.), ключ будет выглядеть примерно так:

jmx[com.example:Type=Hello,weight]

В этом примере именем объекта является "com.example:Type=Hello", именем параметра будет "weight" и, скорее всего, тип возвращаемого значения должен быть "Числовой (с плавающей точкой)".

Ключ становится все более сложным, если параметр возвращает составные данные. Например, если именем параметра является "apple", и он возвращает хеш представляющих его параметров, таких как "weight", "color" и прочее, то ключ может выглядеть примерно так:

jmx[com.example:Type=Hello,apple.weight]

Этот пример показывает, как разделяются с помощью точки имя параметра и ключ хеша. Также, если параметр возвращает вложенные составные данных, их части снова нужно разделять точкой:

jmx[com.example:Type=Hello,fruits.apple.weight]

Параметры с табличными данными состоят из одного или нескольких составных параметров. Если такой параметр указан в параметре имени параметра, то значение этого элемента данных вернет всю структуру параметра в JSON-формате. Значения отдельных элементов из табличных данных можно извлечь с использованием предварительной обработки.

Пример параметра с табличными данными:

jmx[com.example:type=Hello,foodinfo]

Значение элемента данных:

[

{

"a": "apple",

"b": "banana",

"c": "cherry"

},

{

"a": "potato",

"b": "lettuce",

"c": "onion"

}

]

Если имя параметра или ключ хеша содержит символ точки, например:

jmx[com.example:Type=Hello,all.fruits.apple.weight]

это является проблемой. В данном случае требуется экранировать точки, которые являются частью имени, обратной косой чертой:

jmx[com.example:Type=Hello,all\.fruits.apple.weight]

Аналогично, если ключ хеша содержит точку, необходимо ее экранировать:

jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]

Символ обратной косой черты также необходимо экранировать:

jmx[com.example:type=Hello,c:\\documents]

Имеется возможность работы с пользовательскими MBean, которые возвращают не примитивные типы данных, которые переопределяют метод toString().

1. ODBC-мониторинг

ODBC-мониторинг соответствует типу элемента данных "Монитор баз данных" в веб-интерфейсе Подсистемы.

ODBC – это API на языке программирования C к промежуточной прослойке для доступа к системам управления базами данных (DBMS).

Подсистема может выполнять запросы к любой БД, которая поддерживается ODBC. Чтобы это сделать, Подсистема не подключается напрямую к БД, а использует интерфейс ODBC и драйверы, установленные в ODBC. Эта функция позволяет с большей эффективностью мониторить различные БД с разнообразными целями, например: проверка специфичных запросов к базе данных, статистика использования и прочее. Подсистема поддерживает unixODBC, которая является одной из наиболее часто используемых реализаций API ODBC с открытым исходным кодом.

1. Установка unixODBC

Рекомендуемый способ установки unixODBC состоит в использовании репозиториев пакетов по умолчанию в операционной системе Linux. В наиболее популярные дистрибутивы Linux unixODBC добавлен в репозитории пакетов по умолчанию. Если он недоступен, можно обратиться к домашней странице UNIXODBC: <http://www.unixodbc.org/download.html>.

Установка unixODBC на ОС на базе RedHat/Fedora с использованием менеджера пакетов dnf:

dnf -y install unixODBC unixODBC-devel

Установка unixODBC на ОС на базе SUSE с использованием менеджера пакетов zypper:

zypper in unixODBC-devel

Пакет unixODBC-devel требуется для компиляции Подсистемы с поддержкой unixODBC.

1. Установка драйверов unixODBC

Драйвер unixODBC базы данных должен быть установлен для БД, которая будет наблюдаться. unixODBC имеет список поддерживаемых БД и драйверов: <http://www.unixodbc.org/drivers.html>. В некоторых дистрибутивах Linux драйвера БД включены в репозитории пакетов. Драйвера БД MySQL на ОС на базе RedHat/Fedora можно установить с помощью менеджера пакетов dnf:

shell> dnf install mysql-connector-odbc

Установка MySQL драйвера на ОС на базе SUSE с использованием менеджера пакетов zypper:

# zypper in MyODBC-unixODBC

1. Настройка unixODBC

Настройка ODBC выполняется путем редактирования файлов odbcinst.ini и odbc.ini. Для проверки местоположения этих файлов нужно ввести:

odbcinst -j

odbcinst.ini используется для перечисления установленных драйверов БД ODBC:

[mysql]

Description = ODBC for MySQL

Driver = /usr/lib/libmyodbc5.so

где:

* mysql – имя драйвера БД;
* Description – описание драйвера БД;
* Driver – размещение библиотеки драйвера БД.

odbc.ini используется для определения источников данных:

[test]Description = MySQL test database

Driver = mysql

Server = 127.0.0.1

User = root

Password =

Port = 3306

Database = zabbix

где:

* test – имя источника данных (DSN);
* Description – описание источника данных;
* Driver – имя драйвера БД, как указано в odbcinst.ini;
* Server – IP/DNS сервера БД;
* User – пользователь БД для подключения;
* Password – пароль пользователя БД;
* Port – порт подключения к БД;
* Database – имя БД.

Для проверки успешного соединения ODBC подключение к БД необходимо протестировать. Для этого можно воспользоваться утилитой isql (включена в пакет unixODBC):

shell> isql test

+---------------------------------------+

**|** Connected! **|**

**|** **|**

**|** sql-statement **|**

**|** help [tablename] **|**

**|** quit **|**

**|** **|**

+---------------------------------------+

SQL>

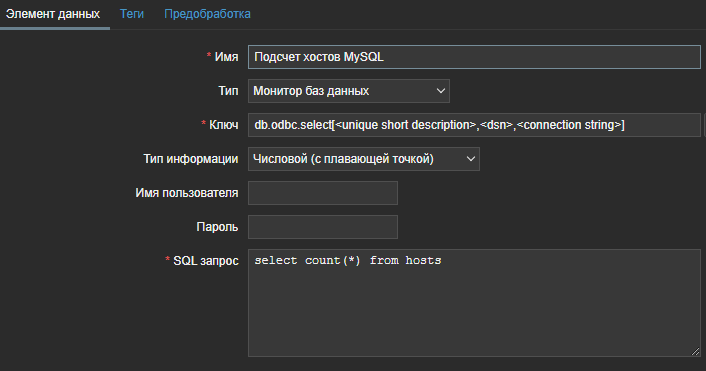
1. Компиляция с поддержкой ODBC

Для включения поддержки ODBC Подсистема должен быть скомпилирован со следующим флагом:

--with-unixodbc[=ARG] use odbc driver against unixODBC package

1. Настройка элемента данных в веб-интерфейсе

Настройка элемента данных для мониторинга БД показана на рисунке 10.



Настройка для мониторинга базы данных

Специально для элементов данных мониторинга БД должны быть указаны параметры из таблицы 83.

Описание параметров мониторинга БД

|  |  |
| --- | --- |
| Параметр | Описание |
| Тип | Выбрать "Монитор баз данных" |
| Ключ | Ввести один из двух поддерживаемых ключей элементов данных: db.odbc.select[<уникальное короткое описание>,<dsn>,<строка подключения>] – этот элемент данных предназначен для получения одного значения, то есть первого столбца первой строки из результата SQL-запроса. Если запрос возвращает более одного столбца, будет прочитан только первый столбец. Если запрос возвращает более одной строки, будет прочитана только первая строка. db.odbc.get[<уникальное короткое описание>,<dsn>,<строка подключения>] – этот элемент данных способен возвращать несколько строк / столбцов в формате JSON. Таким образом, его можно использовать основным элементом данных, который собирает все данные за один системный вызов, а предварительную обработку JSONPath можно использовать в зависимых элементах данных для извлечения отдельных значений. Уникальное описание будет служить для идентификации элемента данных в триггерах и т.п. Хотя dsn и "строка подключения" – опциональные параметры, по крайней мере один из них должен присутствовать. Если заданы оба параметра: имя источника данных (DSN) и строка подключения, то DSN будет игнорироваться. Имя источника данных, если используется, необходимо указать в точности как указано в odbc.ini. Строка подключения может содержать специфичные для драйверов аргументы. |
| Имя пользователя | Ввести имя пользователя для доступа к базе данных Этот параметр необязателен, если пользователь указан в odbc.ini. Если используется строка подключения и поле Имя пользователя непустое, то имя пользователя добавляется к строке подключения как UID=<пользователь> |
| Пароль | Ввести пароль пользователя для доступа к базе данных Этот параметр необязателен, если пароль указан в odbc.ini. Если используется строка подключения и поле Пароль непустое, то пароль добавляется к строке подключения как PWD=<пароль>. Если пароль содержит точку с запятой, его необходимо заключить в фигурные скобки, например: Password: {P?;)\*word} · (если фактически пароль P?;)\*word) Пароль будет добавлен к строке подключения после имени пользователя как: UID=<username>;PWD={P?;)\*word} Чтобы протестировать полученную строку, выполнить: isql -v -k "Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD={P?;)\*word}" |
| SQL запрос | Ввести необходимый SQL-запрос. Следует обратить внимание, что для элемента данных db.odbc.select[] запрос должен возвращать только одно значение. |
| Тип информации | Выбрать корректный тип информации. С некорректным типом информации элемент данных станет неподдерживаемым. |

Примечания:

* Элементы данных монитора БД станут неподдерживаемыми, если процессы "odbc poller" не запускаются в конфигурации Сервера или Прокси. Чтобы включить ODBC-поллеры, нужно задать параметр StartODBCPollers в файле конфигурации Сервера или, если проверки выполняются на стороне Прокси, в файле конфигурации Прокси.
* Подсистема не ограничивает время выполнения запроса. Пользователь сам выбирает запросы, которые выполняются за разумное время.
* Значение параметра Timeout с Сервера используется как время ожидания выполнения входа в ODBC (следует обратить внимание, что в зависимости от драйвера ODBC время ожидания выполнения входа может быть проигнорировано).
* SQL-команда должна вернуть набор данных, как и любой запрос с "SELECT…". Синтаксис запроса зависит от РСУБД, которая будет его обрабатывать. Синтаксис запроса к хранимой процедуре должен начинаться с ключевого слова CALL.

1. Сообщения об ошибках

Сообщения об ошибках ODBC структурированы по полям для предоставления подробной информации. Например:

Cannot execute ODBC query – Zabbix сообщение

[SQL\_ERROR] – ODBC возвращаемый код

[42601] – SQLState

[7] – Родной код ошибки

[ERROR: syntax error at or near ";"; Error while executing the query] – Родное сообщение об ошибке

Следует обратить внимание, что длина сообщения об ошибке ограничена 2048 байтами, поэтому сообщение может быть усечено. Если имеется более одной ODBC диагностической записи, Подсистема пытается их скомпоновать, насколько позволяет ограничение по длине.

1. Рекомендуемые настройки UNIXODBC для MySQL

Установка:

* в Red Hat Enterprise Linux:

dnf install mysql-connector-odbc

* Debian/Ubuntu:

См. материалы [документации MySQL](https://dev.mysql.com/downloads/connector/odbc/) для загрузки необходимого драйвера базы данных под соответствующую платформу.

Настройка ODBC выполняется изменением файлов odbcinst.ini и odbc.ini. Эти файлы конфигурации можно найти в директории /etc. Файл odbcinst.ini может отсутствовать, и в этом случае его необходимо создать вручную:

[mysql]

Description = General ODBC for MySQL

Driver = /usr/lib64/libmyodbc5.so

Setup = /usr/lib64/libodbcmyS.so

FileUsage = 1

1. Рекомендуемые настройки UNIXODBC для PostgreSQL

Установка:

* в Red Hat Enterprise Linux:

dnf install postgresql-odbc

* в Debian/Ubuntu:

См. материалы [документации PostgreSQL](https://www.postgresql.org/download/linux/ubuntu/) для загрузки необходимого драйвера базы данных под соответствующую платформу.

Настройка ODBC выполняется путем редактирования файлов odbcinst.ini и odbc.ini. Эти файлы конфигурации можно найти в директории /etc. Файл odbcinst.ini может отсутствовать, и в этом случае его необходимо создать вручную.

1. Рекомендуемые настройки UNIXODBC для MSSQL

Установка:

* в Red Hat Enterprise Linux:

dnf -y install freetds unixODBC

* Debian/Ubuntu:

См. материалы [документации пользователя FreeTDS](http://www.freetds.org/userguide/) для загрузки необходимого драйвера базы данных под соответствующую платформу.

Настройка ODBC выполняется путем редактирования файлов odbcinst.ini и odbc.ini. Эти файлы конфигурации можно найти в директории /etc. Файл odbcinst.ini может отсутствовать, и в этом случае его необходимо создать вручную.

1. Зависимые элементы данных

Зависимые элементы данных используются в ситуациях, когда один элемент данных собирает несколько метрик одновременно или требуется одновременный сбор связанных метрик, например:

* утилизация CPU по отдельным ядрам;
* входящий/исходящий/общий сетевой трафик.

Для обеспечения массового сбора метрик и использования синхронности в нескольких связанных элементах данных Подсистема поддерживает зависимые элементы данных. Зависимые элементы данных используют основной элемент данных для сбора своих данных одновременно, одним запросом. Новое значение у основного элемента данных автоматически заполняет значения и зависимых элементов данных. Зависимые элементы данных не могут иметь интервал обновления, отличный от своего основного элемента данных.

Опции предварительной обработки в Подсистеме можно использовать для извлечения части, которая необходима зависимому элементу данных, из данных основного элемента данных.

Предварительная обработка управляется при помощи процесса "preprocessing manager (менеджер предобработки)" вместе с рабочими процессами, которые выполняют шаги предварительной обработки. Все значения (с предварительной обработкой и без предварительной обработки) от разных сборщиков данных проходят через менеджер предварительной обработки перед добавлением в кэш истории. Для связи между сборщиками данных (поллерами, трапперами и т.д.) и процессами предобработки используется взаимодействие между процессами (IPC) на основе сокета.

Сервер или Прокси (если узел сети наблюдается через Прокси) выполняют шаги предварительной обработки и обрабатывают зависимые элементы данных.

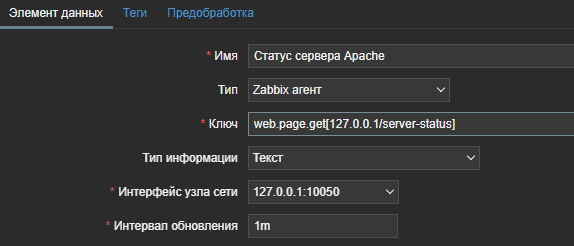
Элемент данных любого типа, даже зависимый элемент данных, может быть основным элементом данных. Дополнительные уровни зависимых элементов данных можно использовать для извлечения меньших частей значения уже существующего зависимого элемента данных.

Ограничения:

* Зависимости разрешены только в пределах одного узла сети (шаблона).
* Прототип элемента данных может зависеть от другого прототипа элемента данных или от обычного элемента данных с этого же узла сети.
* Максимальное количество зависимых элементов данных от одного основного элемента данных ограничено 29999 элементами данных (независимо от количества уровней зависимости).
* Разрешено максимум 3 уровня зависимости.
* Зависимый элемент данных с основным элементом данных из шаблона не экспортируется в XML.

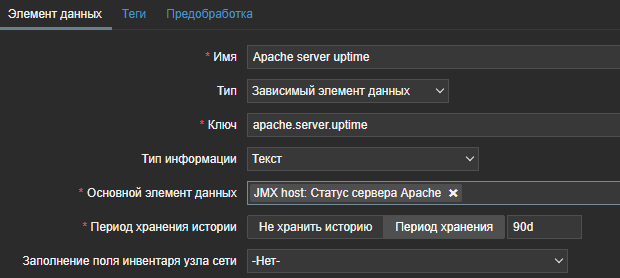
Зависимый элемент данных зависит от своего основного элемента данных, поэтому сначала необходимо настроить (или использовать существующий) основной элемент данных:

1. перейти в меню "Настройка → Узлы сети";
2. нажать на Элементы данных в строке с узлом сети;
3. нажать на Создать элемент данных;
4. в окне указать параметры элемента данных (рисунок 11);



Создание элемента данных

1. нажать на Добавить для сохранения основного элемента данных;
2. настроить зависимый элемент данных (рисунок 12).



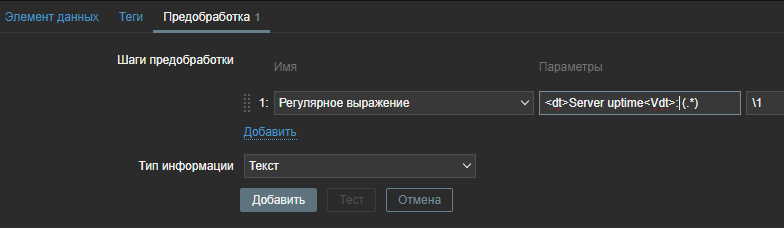
Зависимый элемент данных

Поля в таблице 84 требуют особые параметры по зависимым элементам данных.

Параметры зависимых элементов данных

|  |  |
| --- | --- |
| Параметр | Описание |
| Тип | Выбрать "Зависимый элемент данных" |
| Ключ | Указать ключ, который будет использоваться для распознавания этого элемента данных. |
| Основной элемент данных | Выбрать основной элемент данных. Значение основного элемента данных будет использовано для заполнения значения зависимого элемента данных. |
| Тип информации | Выбрать тип информации, который соответствует формату сохраняемых данных. |

Можно использовать предварительную обработку значений элементов данных для извлечения необходимой части значения с основного элемента данных (рисунок 13).

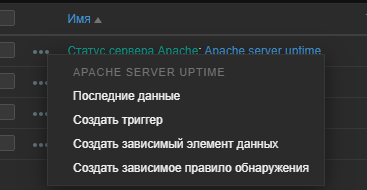


Предварительная обработка значений элементов данных

Без предварительной обработки значением зависимого элемента данных будет такое же значение, что и значение основного элемента данных.

1. нажать на Добавить, чтобы сохранить зависимый элемент данных.

Ссылка для более быстрого создания зависимого элемента данных может быть доступна после нажатия на пиктограмму  в списке элементов данных и выбора "Создать зависимый элемент данных" (рисунок 14).



Быстрое создание зависимого элемента данных

В списке элементов данных зависимые элементы данных отображаются с именем основного элемента данных в качестве префикса.

Если основной элемент данных удаляется, то будут удалены и все его зависимые элементы данных.

1. HTTP-Агент

Этот тип элемента данных позволяет получать данные с использованием HTTP/HTTPS-протокола. Трапы также возможны с использованием sender или протокола sender.

Проверка HTTP-элемента данных выполняется на стороне Сервера. Если узлы сети наблюдаются через Прокси, проверки HTTP-элементов данных выполняются этими Прокси.

Проверки HTTP-элементов данных не требуют наличия какого-либо Агента, запущенного на наблюдаемом узле сети.

HTTP-агент поддерживает как HTTP, так и HTTPS. Подсистема будет опционально следовать перенаправлениям. Максимальное количество перенаправлений вшито в код и равно 10 (используется cURL-опция CURLOPT\_MAXREDIRS).

Сервер/Прокси должны быть изначально сконфигурированы с поддержкой cURL (libcurl).

Для настройки HTTP-элемента данных нужно:

1. перейти в меню "Настройка → Узлы сети";
2. нажать на Элементы данных в строке с узлом сети;
3. нажать на Создать элемент данных;
4. ввести параметры элемента данных в окне.

Поля, которые требуют специфичную информацию HTTP-элементов данных, приведены в таблице 85.

Параметры HTTP-элементов данных

| Параметр | Описание |
| --- | --- |
| Тип | Выбрать HTTP-Агент. |
| Ключ | Ввести уникальный ключ элемента данных. |
| URL | URL для подключения и получения данных. Например: https://www.google.com Имена доменов можно указывать Юникод-символами. Они автоматически конвертируются методом punycode в ASCII при выполнении HTTP-проверки. Кнопку Анализ можно использовать, чтобы отделить из URL опциональные поля запроса (например, ?name=Admin&password=mypassword), переместив параметры и значения в "Поля запроса", чтобы URL-кодировка выполнялась автоматически. Ограничено 2048 символами. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_URL](https://curl.haxx.se/libcurl/c/CURLOPT_URL.html) [en] опцию cURL. |
| Поля запроса | Переменные для URL (см. выше). Задаются в виде пар параметров и значений. Значения автоматически URL-кодируются. Значения с макросов раскрываются и затем автоматически URL-кодируются. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_URL](https://curl.haxx.se/libcurl/c/CURLOPT_URL.html) [en] опцию cURL. |
| Тип запроса | Выбрать тип метода запроса: GET, POST, PUT или HEAD |
| Время ожидания | Подсистема не будет тратить более указанного времени при обработке URL (1-60 секунд). В действительности же этот параметр определяет максимальное время создания подключения к URL и максимальное время выполнения HTTP-запроса. Следовательно, Подсистема не будет тратить более 2-x "Время ожидания" секунд за одну проверку. Поддерживаются суффиксы времени, например, 30s, 1m. Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_TIMEOUT](https://curl.haxx.se/libcurl/c/CURLOPT_TIMEOUT.html) [en] опцию cURL. |
| Тип тела запроса | Выбрать тип запроса тела: Сырые данные – пользовательское тело HTTP-запроса, макросы заменяются, но кодирование не выполняется JSON данные – тело HTTP-запроса в формате JSON. Макросы можно использовать в виде строк, чисел, true или false; макросы, которые используются в виде строки, должны быть заключены в двойные кавычки. Значения из макросов раскрываются и затем экранируются автоматически. Если в заголовках "Content-Type" не задан, то будет указано значение по умолчанию "Content-Type: application/json" XML данные – тело HTTP-запроса в формате XML. Макросы можно использовать в виде нод текста, параметров или CDATA секции. Значения из макросов раскрываются и затем экранируются автоматически в нодах текста и в параметрах. Если в заголовках "Content-Type" не задан, то будет указано значение по умолчанию "Content-Type: application/xml" Следует обратить внимание, что выбор "XML данные" требует наличия поддержки libxml2. |
| Тело запроса | Ввести тело запроса. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. |
| Заголовки | Пользовательские HTTP-заголовки, которые будут отправлены при выполнении запроса. Задаются в виде пар параметров и значений. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_HTTPHEADER](https://curl.haxx.se/libcurl/c/CURLOPT_HTTPHEADER.html) [en] опцию cURL. |
| Требуемые коды состояния | Список ожидаемых кодов состояния HTTP. Если Подсистема получает код не из списка, то элемент данных станет неподдерживаемым. Если поле не заполнено, то проверка не производится. Например: 200,201,210-299 Поддерживаемые макросы в этом списке: пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLINFO\_RESPONSE\_CODE](https://curl.haxx.se/libcurl/c/CURLINFO_RESPONSE_CODE.html) [en] опцию cURL. |
| Следовать перенаправлениям | Отметить для следования HTTP перенаправлениям. Это поле задает [CURLOPT\_FOLLOWLOCATION](https://curl.haxx.se/libcurl/c/CURLOPT_FOLLOWLOCATION.html) [en] опцию cURL. |
| Режим получения | Выбрать часть ответа, которую необходимо получать: Тело – только тело Заголовки – только заголовки Тело и заголовки – тело и заголовки |
| Конвертация в JSON | Заголовки сохраняются в виде пар параметров и значений под ключом "header". Если встречается "Content-Type: application/json", то тело сохраняется в виде объекта, иначе оно сохраняется в виде строки, например: |
| HTTP Прокси | Можно указать необходимый HTTP-Прокси, следуя следующему формату: [протокол://][имя пользователя[:пароль]@]Прокси.mycompany.com[:порт] Можно использовать опциональный префикс протокол://, чтобы указать альтернативные протоколы Прокси (например: https, socks4, socks5; поддержка префиксов протоколов добавлена в cURL 7.21.7). Если протокол не указан, Прокси будет обрабатываться как HTTP-Прокси. Если указать ошибочный протокол, подключение завершится неудачей и элемент данных перейдет в неподдерживаемое состояние. По умолчанию будет использоваться порт 1080. Если этот параметр указан, Прокси заменит переменные окружения, связанные с Прокси, такие как http\_proxy, HTTPS\_PROXY. Если не указан, переменные окружения не будут заменены. Введенное значение передается "как есть", проверка корректности ввода не производится. Примечание – Для HTTP-Прокси поддерживается только простая аутентификация. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_PROXY](https://curl.haxx.se/libcurl/c/CURLOPT_PROXY.html) [en] опцию cURL. |
| HTTP Аутентификация | Тип аутентификации: Пусто – без использования аутентификации. Простая аутентификация – с использованием простой аутентификации. NTLM аутентификация – с использованием NTLM ([Windows NT LAN Manager)](http://en.wikipedia.org/wiki/NTLM)-аутентификации. Kerberos – с использованием Kerberos-аутентификации.  Digest – с использованием Digest-аутентификации. После выбора какого-либо метода аутентификации будут доступны два дополнительных поля для ввода имени пользователя и пароля, в которых поддерживаются пользовательские макросы и макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_HTTPAUTH](https://curl.haxx.se/libcurl/c/CURLOPT_HTTPAUTH.html) [en] опцию cURL. |
| Проверка SSL узла | Отметить для верификации SSL- сертификата веб-сервера. Сертификат Сервера будет автоматически взят из места центра сертификации (CA) всей системы. Можно переопределить расположение CA-файлов, воспользовавшись параметром конфигурации SSLCALocation на стороне Сервера или Прокси. Это поле задает [CURLOPT\_SSL\_VERIFYPEER](http://curl.haxx.se/libcurl/c/CURLOPT_SSL_VERIFYPEER.html) [en] опцию cURL. |
| Проверка SSL хоста | Отметить для удостоверения, что поле "Common Name" или поле "Subject Alternate Name" сертификата веб-сервера совпадают. Это поле задает [CURLOPT\_SSL\_VERIFYHOST](http://curl.haxx.se/libcurl/c/CURLOPT_SSL_VERIFYHOST.html) [en] опцию cURL. |
| Файл SSL сертификата | Имя файла SSL-сертификата для аутентификации клиента. Файл сертификата должен быть в формате PEM1. Если файл сертификата также содержит и закрытый ключ (private key), оставить поле "Файл SSL ключа" пустым. Если ключ зашифрован, указать пароль в поле Пароль к SSL-ключу. Директория, содержащая этот файл, указывается в параметре конфигурации SSLCertLocation Сервера или Прокси. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_SSLCERT](http://curl.haxx.se/libcurl/c/CURLOPT_SSLCERT.html) [en] опцию cURL. |
| Файл SSL ключа | Имя файла закрытого ключа SSL (private key), который используется для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM1. Директория, содержащая этот файл, указывается в параметре конфигурации SSLKeyLocation Сервера или Прокси. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_SSLKEY](http://curl.haxx.se/libcurl/c/CURLOPT_SSLKEY.html) [en] опцию cURL. |
| Пароль к SSL ключу | Пароль к файлу закрытого ключа. Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения. Это поле задает [CURLOPT\_KEYPASSWD](http://curl.haxx.se/libcurl/c/CURLOPT_KEYPASSWD.html) [en] опцию cURL. |
| Активировать трапы | При включении этой опции, элемент данных будет работать также и как траппер элемент данных и будет принимать данные, которые отправляются на этот элемент данных при помощи sender или с использованием протокола sender. |
| Разрешенные хосты | Доступно, только если выбрана опция "Активировать трапы". Список разделенных запятыми IP-адресов, опционально в CIDR-представлении, или DNS-имен. Если задано, входящие подключения будут приниматься только с перечисленных здесь узлов сети. Если включена поддержка IPv6, то "127.0.0.1", "::127.0.0.1", "::ffff:127.0.0.1" обрабатываются одинаково и "::/0" будет разрешать любые IPv4 или IPv6-адреса. "0.0.0.0/0" можно использовать, чтобы разрешить любые IPv4-адреса. Следует обратить внимание, что "IPv4-совместимые IPv6-адреса" (0000::/96 префикс) поддерживаются, но являются устаревшими согласно [RFC4291](https://tools.ietf.org/html/rfc4291#section-2.5.5). Пример: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, mon.rosa.dev, {HOST.HOST} В этом поле разрешены пробелы и пользовательские макросы. Макросы узлов сети: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} – поддерживаются в этом поле. |

Если поле HTTP-Прокси оставить пустым, можно воспользоваться другим способом указать HTTP-Прокси, для чего необходимо задать переменные окружения.

Для HTTP-проверок нужно указать переменную окружения http\_proxy для пользователя Сервера, например:

http\_proxy=http://proxy\_ip:proxy\_port.

Для HTTPS-проверок нужно указать переменную окружения HTTPS\_PROXY, например:

HTTPS\_PROXY=http://proxy\_ip:proxy\_port.

Подсистема поддерживает файлы сертификатов и закрытых ключей (private keys) только в формате PEM. В случае если имеются данные сертификата и закрытого ключа в формате файла PKCS #12 (обычно поставляются с расширениями \*.p12 или \*.pfx), можно сгенерировать из них PEM-файл, используя следующие команды:

openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem

openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes -out ssl-cert.key

1. Проверки Prometheus

Подсистема может опрашивать метрики, представленные в формате строки Prometheus.

Чтобы начать сбор данных Prometheus, необходимо выполнить два шага:

1. настроить основной элемент данных HTTP, указывающий на соответствующую конечную точку данных, например https://<хост prometheus>/metrics;
2. настроить зависимые элементы данных, использующие опцию предварительной обработки Prometheus для запроса необходимых данных из метрик, собранных основным элементом данных.

Существует два варианта предварительной обработки данных Prometheus:

* Шаблон Prometheus – используется в обычных элементах данных для запроса метрик Prometheus;
* Prometheus в JSON – используется в обычных элементах данных и для низкоуровневого обнаружения; в этом случае запрашиваемые данные Prometheus возвращаются в формате JSON.

Для зависимых метрик поддерживается массовая обработка. Чтобы включить кэширование и индексацию, первым шагом предварительной обработки должен быть "Шаблон Prometheus". Если "Шаблон Prometheus" является первым шагом предварительной обработки, то проанализированные данные Prometheus кэшируются и индексируются по первому условию <метка>==<значение> в шаге предварительной обработки "Шаблон Prometheus". Этот кэш используется повторно при обработке остальных зависимых элементов данных в этом пакете. Для достижения оптимальной производительности первая метка должна быть одной из тех, у которых наиболее разные значения.

Если другие операции предварительные обработки необходимо выполнить до первого шага, то их необходимо переместить либо в основной элемент данных, либо в новый зависимый элемент данных, который затем можно использовать в качестве основного элемента данных для этих зависимых элементов данных.

Если настроен основной элемент данных HTTP, нужно создать зависимый элемент данных, который использует шаг предварительной обработки Prometheus:

1. ввести общие параметры зависимого элемента данных в окне настройки;
2. перейти на вкладку Предобработка**;**
3. выбрать шаг предобработки Prometheus ("Шаблон Prometheus" или "Prometheus в JSON");

Параметры, указанные в таблице 86, специфичны для опции предобработки "Шаблон Prometheus".

Параметры для опции предобработки "Шаблон Prometheus"

| Параметр | Описание | Примеры |
| --- | --- | --- |
| Шаблон | Чтобы определить необходимый шаблон данных, можно использовать язык запросов, который похож на язык запросов Prometheus, например: <имя метрики> – выбор по названию метрики; {\_\_name\_\_="<имя метрики>"} – выбор по названию метрики; {\_\_name\_\_=~"<регулярное выражение>"} – выбор по названию метрики, совпадающему с регулярным выражением; {<имя метки>="<значение метки>",...} – выбор по имени метки; {<имя метки>=~"<регулярное выражение>",...} – выбор по имени метки, совпадающему с регулярным выражением; {\_\_name\_\_=~".\*"}==<значение> – выбор по значению метрики. Или сочетание вышеперечисленного: <имя метрики>{<имя метки1>="<значение метки1>",<имя метки2>=~"<регулярное выражение>",...}==<значение> Значением метки может быть любая последовательность символов UTF-8, но символы обратной косой черты, двойных кавычек и перевода строки необходимо экранированы как \\, \" и \n соответственно; другие символы не требуется экранировать. | wmi\_os\_physical\_memory\_free\_Bytes cpu\_usage\_system{cpu="cpu-total"} cpu\_usage\_system{cpu=~".\*"} cpu\_usage\_system{cpu="cpu-total",host=~".\*"} wmi\_service\_state{name="dhcp"}==1 wmi\_os\_timezone{timezone=~".\*"}==1 |
| Результат обработки | Указать, следует ли вернуть значение, метку или применить соответствующую функцию (если шаблон соответствует нескольким строкам и результат необходимо агрегировать): значение – получение значения метрики (ошибка, если совпадут несколько строк); метка – получение значения метки, указанной в поле Метка (ошибка, если совпадут несколько метрик); сумма – получение суммы значений; мин – получение минимального значения; макс – получение максимального значения; сред – получение среднего значения; кол-во – получение количества значений. Это поле доступно только для опции Шаблон Prometheus. |  |
| Вывод | Задать название метки (опционально). В этом случае вернется значение, соответствующее названию метки. Это поле доступно только для шага Шаблон Prometheus, если в поле "Результат обработки" выбрано "Метка". |  |

Данные из Prometheus можно использовать для низкоуровневого обнаружения. В этом случае данные необходимы в формате JSON, и опция предобработки "Prometheus в JSON" вернет их в нужном формате.

В таблице 87 перечислены различия и сходства между PromQL и языком запросов предобработки Prometheus в Подсистеме.

Сравнение PromQL и Prometheus

| Мгновенный векторный селектор PromQL | | Предобработка Prometheus в Подсистеме |
| --- | --- | --- |
| Различия | | |
| Цель запроса | Сервер Prometheus | Простой текст в формате представления Prometheus |
| Возвращает | Мгновенный вектор | Значение метрики или метки (Шаблон Prometheus) Массив метрик для одиночного значения в JSON (Prometheus в JSON) |
| Операторы сопоставления меток | =, !=, =~, !~ | =, !=, =~, !~ |
| Регулярное выражение, используемое при сопоставлении метки или имени метрики | RE2 | PCRE |
| Операторы сравнения | См. список | Только == (равно) поддерживается для фильтрации значений |
| Сходства | | |
| Выбор по названию метрики, равному строке | <имя метрики> или {\_\_name\_\_="<имя метрики>"} | <имя метрики> или {\_\_name\_\_="<имя метрики>"} |
| Выбор по имени метрики, которое соответствует регулярному выражению | {\_\_name\_\_=~"<регулярное выражение>"} | {\_\_name\_\_=~"<регулярное выражение>"} |
| Выбор по значению <имя метки>, равному строке | {<имя метки>="<значение метки>",...} | {<имя метки>="<значение метки>",...} |
| Выбор по значению <имя метки>, которое соответствует регулярному выражению | {<имя метки>=~"<регулярное выражение>",...} | {<имя метки>=~"<регулярное выражение>",...} |
| Выбор по значению, равному строке | {\_\_name\_\_=~".\*"} == <значение> | {\_\_name\_\_=~".\*"} == <значение> |

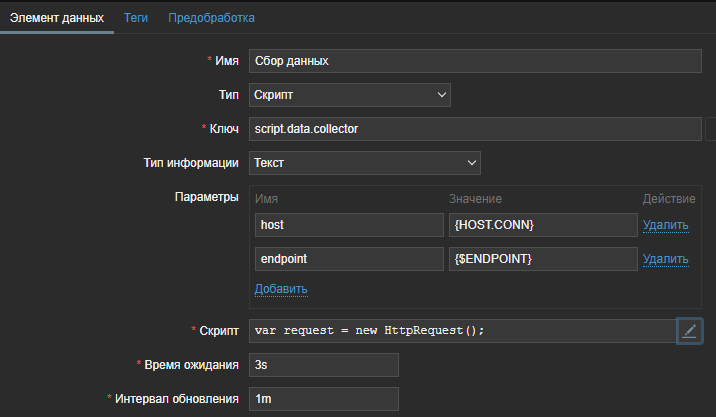
1. Скриптовые элементы данных

Скриптовые элементы данных можно использовать для сбора данных путем выполнения пользовательского JavaScript-кода с возможностью получения данных через HTTP/HTTPS. В дополнение к скрипту можно указать опциональный список параметров (пары из имени и значения) и время ожидания.

Этот тип элементов данных может быть полезен в сценариях сбора данных, требующих нескольких шагов или сложной логики. Например, скриптовый элемент данных можно настроить на выполнение HTTP-вызова, последующей обработки данных, полученных на первом шаге, и передачи преобразованного значения второму HTTP-вызову.

Скриптовые элементы данные обрабатываются Сервером или Прокси на уровне поллеров.

В поле Тип окна настройки элемента данных нужно выбрать Скрипт, затем заполнить необходимые поля (рисунок 15).



Скриптовый элемент данных

Поля, которые требуют информацию, специфичную для элементов данных Скрипт, перечислены в таблице 88.

Поля для элементов данных Скрипт

| Поле | Описание |
| --- | --- |
| Ключ | Ввести уникальный ключ, который будет использоваться для идентификации этого элемента данных. |
| Параметры | Указать переменные, которые будут переданы скрипту как пары параметров и значений. Поддерживаются встроенные макросы {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} и пользовательские макросы. |
| Скрипт | Ввести JavaScript-код в блоке, который появится при нажатии на поле параметров (или на кнопку просмотра/изменения сразу за полем параметров). Этот код должен обеспечивать логику получаемого значения метрики. Код имеет доступ ко всем параметрам, может выполнять запросы HTTP GET, POST, PUT и DELETE, а также имеет управление над HTTP-заголовками и телом запроса |
| Время ожидания | Время ожидания выполнения JavaScript (1-60s, по умолчанию 3s); превышение этого значения вернет ошибку. Поддерживаются суффиксы времени, например: 30s, 1m. В зависимости от скрипта срабатывание времени ожидания может занять больше времени. |

1. Браузер

Элементы данных "Браузер" позволяют осуществлять мониторинг сложных веб-сайтов и веб-приложений с использованием браузера.

Браузерные элементы данных собирают данные, выполняя определённый пользователем код JavaScript и извлекая данные по HTTP/HTTPS. Этот элемент данных может имитировать такие действия браузера, как нажатие кнопкой "мыши", ввод текста, навигация по веб-страницам и другие виды взаимодействия пользователя с веб-сайтами или веб-приложениями.

В дополнение к скрипту можно указать необязательный список параметров (пар из имени и значения) и время ожидания.

Проверки браузерных элементов данных выполняются и обрабатываются процессами-поллерами браузерных элементов данных (browser pollers) сервера Подсистемы или прокси. При необходимости можно настроить количество предварительно созданных экземпляров процессов browser pollers в параметре StartBrowserPollers файла конфигурации Подсистемы.

Для мониторинга сложных веб-сайтов и веб-приложений среди готовых шаблонов доступен шаблон "Website by Browser".

Для настройки элемента данных в диалоге в поле "Тип (Type)" нужно выбрать "Браузер" и заполнить обязательные поля.

Поля, требующие информации, специфичной для элемента данных "Браузер", приведены в таблице 89.

Поля элемента данных "Браузер"

| **Поле** | **Описание** |
| --- | --- |
| Ключ (Key) | Ввести уникальный ключ, который будет использоваться для идентификации элемента данных |
| Параметры (Parameters) | Указать переменные, которые будут переданы в скрипт как пары "атрибут- значение". Поддерживаются пользовательские макросы. Чтобы узнать, какие встроенные макросы поддерживаются, выполните поиск по запросу "Браузер" в таблице поддерживаемых макросов |
| Скрипт (Script) | Ввести код JavaScript в модальном редакторе, который открывается при нажатии на поле параметра или на значок карандаша рядом с ним. Этот код должен предоставлять логику для возврата значения метрики. Код имеет доступ ко всем параметрам, всем дополнительным объектам JavaScript и объектам JavaScript элемента данных "Браузер", добавленным Подсистемой |
| Время ожидания (Timeout) | Тайм-аут выполнения JavaScript (1-600s; при его превышении будет возвращена ошибка). Следует обратить внимание, что в зависимости от скрипта для срабатывания тайм-аута может потребоваться больше времени |

Приложение Е. Пользовательские команды

1. zabbix\_agent2

zabbix\_agent2 – демон Агента-2 для мониторинга параметров различных служб.

Синтаксис:

zabbix\_agent2 [-c config-file]

zabbix\_agent2 [-c config-file] -p

zabbix\_agent2 [-c config-file] -t item-key

zabbix\_agent2 [-c config-file] -R runtime-option

zabbix\_agent2 -h

zabbix\_agent2 -V

Параметры:

-c, --config config-file – использование альтернативного файла конфигурации config-file вместо файла по умолчанию;

-R, --runtime-control runtime-option – выполнение административных функций в соответствии с runtime-option.

Выполнение административных функций:

* UserParameter reload– повторная загрузка пользовательских параметров из файла конфигурации;
* loglevel increase – увеличение уровня журналирования;
* loglevel decrease – уменьшение уровня журналирования;
* help – Список доступных административных функций;
* metrics – список доступных метрик;
* version – вывод номера версии;

-p, --print – вывод известных элементов данных и выход. Для каждого элемента данных используются либо общие значения по умолчанию, либо предоставляются определенные значения по умолчанию для тестирования. Эти значения по умолчанию перечислены в квадратных скобках в качестве параметров ключей элементов данных. Возвращаемые значения заключены в квадратные скобки с префиксом типа данных возвращаемого значения, разделенные вертикальной чертой. Для пользовательских параметров всегда используется "t", так как Агент не может определить все возможные варианты возвращаемых значений. Элементы данных, которые отображаются как работающие, не обязательно будут работать при опросе запущенного демона Агента со стороны Сервера или утилиты zabbix\_get, так как доступы или среда могут различаться.

Типы возвращаемых значений:

* d – число с плавающей точкой;
* m – не поддерживается. Такое может произойти из-за запроса элемента данных, который работает только в активном режиме, таких как элемент данных наблюдения за журналом, элемент данных, требующий нескольких собранных значений. Проблемы прав доступа или некорректные пользовательские параметры могут также привести к неподдерживаемому состоянию;
* s – текст. Максимальная длина не ограничена;
* t – текст. По аналогии с s;
* u – беззнаковое целое число.
* -t, --test item-key – тестирование одного элемента данных и выход. См. --print для вывода описания.
* -h, --help – вывод справочной информации и выход;
* -V, --version – вывод информации о версии и выход.

Расположение файла конфигурации Агента-2 по умолчанию – /usr/local/etc/zabbix\_agent2.conf.

1. zabbix\_agentd

zabbix\_agentd – демон Агента (служба) для мониторинга различных параметров Сервера.

Синтаксис:

zabbix\_agentd [-c config-file]

zabbix\_agentd [-c config-file] -p

zabbix\_agentd [-c config-file] -t item-key

zabbix\_agentd [-c config-file] -R runtime-option

zabbix\_agentd -h

zabbix\_agentd -V

Параметры:

* -c, --config config-file – использование альтернативного файла конфигурации config-file вместо файла по умолчанию;
* -f, --foreground – запуск Агента на переднем плане (отключение перехода в фоновый режим);
* -R, --runtime-control runtime-option – выполнение административных функций в соответствии с runtime-option.

Возможные значения административных функций (runtime-option):

* UserParameter\_reload –загрузка заново пользовательских параметров из файла конфигурации;
* log\_level\_increase[=цель]– повышение уровня журналирования, влияющее на все процессы, если цель не указана;
* log\_level\_decrease[=цель] – понижение уровня журналирования, влияющее на все процессы, если цель не указана;

Цели для управления уровнем журналирования:

* тип-процесса – все процессы указанного типа (active checks, collector, LISTENer);
* тип-процесса,N – тип и номер процесса (например, "LISTENer,3");
* pid – идентификатор процесса, до 65535. Для больших значений указывают цель как "тип-процесса,N";
* version – вывод номера версии;
* -p, --print – вывод известных элементов данных и выход. Для каждого элемента данных используются либо общие значения по умолчанию, либо предоставляются определенные значения по умолчанию для тестирования. Эти значения по умолчанию перечислены в квадратных скобках в качестве параметров ключей элементов данных. Возвращаемые значения заключены в квадратные скобки с префиксом типа данных возвращаемого значения, разделенные вертикальной чертой. Для пользовательских параметров всегда используется "t", так как Агент не может определить все возможные варианты возвращаемых значений. Элементы данных, которые отображаются как работающие, не обязательно будут работать при опросе запущенного демона Агента со стороны Сервера или утилиты zabbix\_get, так как доступы или среда могут различаться.

Типы возвращаемых значений:

* d – число с плавающей точкой;
* m – не поддерживается. Такое может произойти из-за запроса элемента данных, который работает только в активном режиме, таких как элемент данных наблюдения за журналом, элемент данных, требующий нескольких собранных значений. Проблемы прав доступа или некорректные пользовательские параметры могут также привести к неподдерживаемому состоянию;
* s – текст. Максимальная длина не ограничена;
* t – текст. По аналогии с s;
* u – беззнаковое целое число.
* -t, --test item-key – тестирование одного элемента данных и выход. См. --print для вывода описания.
* -h, --help – вывод справочной информации и выход;
* -V, --version – вывод информации о версии и выход.

Расположение по умолчанию файла конфигурации Агента – /usr/local/etc/zabbix\_agentd.conf.

1. zabbix\_get

zabbix\_get – утилита командной строки для получения метрик, которая используется для получения данных от Агента Подсистемы.

Синтаксис:

zabbix\_get -s имя-узла-сети-или-IP [-p номер-порта] [-I IP-адрес] [-t тайм-аут] -k ключ-элемента-данных

zabbix\_get -s имя-узла-сети-или-IP [-p номер-порта] [-I IP-адрес] [-t тайм-аут] --tls-connect cert --tls-ca-file CA-файл [--tls-crl-file CRL-file] [--tls-agent-cert-issuer эмитент] [--tls-agent-cert-subject субъект] --tls-cert-file файл-сертификата --tls-key-file файл-ключа [--tls-cipher13 строка-шифра] [--tls-cipher строка-шифра] -k ключ-элемента-данных

zabbix\_get -s имя-узла-сети-или-IP [-p номер-порта] [-I IP-адрес] [-t тайм-аут] --tls-connect psk --tls-psk-identity идентификатор-PSK --tls-psk-file файл-PSK [--tls-cipher13 строка-шифра] [--tls-cipher строка-шифра] -k ключ-элемента-данных

zabbix\_get -h

zabbix\_get -V

Параметры:

* -s, --host имя-узла-сети-или-IP – имя узла сети или IP-адрес узла сети;
* -p, --port номер-порта – номер порта Агента, работающего на хосте. По умолчанию 10050;
* -I, --source-address IP-адрес – исходящий IP-адрес;
* -t, --Timeout секунды – время ожидания в допустимом диапазоне 1-30 секунд (по умолчанию: 30);
* -k, --key ключ-элемента-данных – ключ элемента данных, для которого необходимо получить значение;
* --tls-connect значение – способ подключения к Агенту. Значения:
* unencrypted – подключение без шифрования (по умолчанию);
* psk – подключение с использованием TLS и общего ключа (pre-shared key, PSK);
* cert – подключение с использованием TLS и сертификата;
* --tls-ca-file CA-file – полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата партнера;
* --tls-crl-file CRL-file – полный путь к файлу, содержащему отозванные сертификаты;
* --tls-agent-cert-issuer эмитент – разрешенный эмитент сертификата Агента;
* --tls-agent-cert-subject субъект – разрешенный субъект сертификата Агента;
* --tls-cert-file файл-сертификата – полный путь к файлу, содержащему сертификат или цепочку сертификатов;
* --tls-key-file файл-ключа – полный путь к файлу, содержащему закрытый ключ (private key);
* --tls-psk-identity идентификатор-PSK – строка идентификатора PSK;
* --tls-psk-file файл-PSK – полный путь к файлу, содержащему общий ключ (pre-shared key);
* --tls-cipher13 строка-шифра – строка шифра для OpenSSL 1.1.1 или новее для TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию. Эта опция недоступна, если версия OpenSSL ниже 1.1.1;
* --tls-cipher строка-шифра – строка приоритета GnuTLS (для TLS 1.2 и выше) или строка шифра OpenSSL (только для TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию;
* -h, --help – вывод справочной информации и выход;
* -V, --version – вывод информации о версии и выход.

1. zabbix\_js

zabbix\_js – утилита командной строки для выполнения скриптов JavaScript.

Синтаксис:

zabbix\_js -s файл-скрипта -p входной-параметр [-l log-level] [-t Timeout]

zabbix\_js -s файл-скрипта -i входной-файл [-l log-level] [-t тайм-аут]

zabbix\_js -h

zabbix\_js -V

Параметры:

* -s, --script файл-скрипта – имя файла скрипта для выполнения. Если в качестве имени файла указано "-", скрипт будет прочитан из стандартного ввода;
* -p, --param входной-параметр –входной параметр;
* -i, --input входной-файл – имя файла входного параметра. Если в качестве имени файла указано "-", входной-параметр будет считан из стандартного ввода;
* -l, --loglevel уровень-журналирования – уровень журналирования;
* -t, --Timeout тайм-аут – время ожидания в секундах. Допустимый диапазон: 1-60 секунд (по умолчанию: 10);
* -h, --help– вывод справочной информации и выход;
* -V, --version – вывод информации о версии и выход.

1. zabbix\_proxy

zabbix\_proxy – демон сервиса Прокси, который собирает данные мониторинга с устройств и отправляет их на Сервер.

Синтаксис:

zabbix\_proxy [-c config-file]

zabbix\_proxy [-c config-file] -R runtime-option

zabbix\_proxy -h

zabbix\_proxy -V

Параметры:

* -c, --config config-file – использование альтернативного файла конфигурации config-file вместо файла по умолчанию;
* -f, --foreground – запуск Прокси на переднем плане;
* -R, --runtime-control runtime-option – выполнение административных функций в соответствии с runtime-option. Параметры runtime-option:
* config\_cache\_reload – перезагрузка кэша конфигурации. Игнорируется, если кэш в данный момент загружается. Активный Прокси подключится к Серверу и запросит данные конфигурации. Сигнал будет отправлен процессу, указанному в файле PID, для поиска которого будет использован файл конфигурации по умолчанию (если не указана опция -c);
* snmp\_cache\_reload – перезагрузка кэш SNMP;
* housekeeper\_execute – запуск процесса housekeeper. Игнорируется, если housekeeper в данный момент выполняется;
* diaginfo[=раздел] – запись внутренней диагностической информации указанного раздела. Раздел может быть historycache, preprocessing. По умолчанию диагностическая информация всех разделов логируется;
* log\_level\_increase[=цель] – повышение уровня логирования; влияет на все процессы, если цель не указана;
* log\_level\_decrease[=цель] – понижение уровня логирования: влияет на все процессы, если цель не указана;

Цели для управления уровнем журналирования:

* тип-процесса – все процессы указанного типа (configuration syncer, data sender, discoverer, heartbeat sender, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector);
* тип-процесса,N – тип и номер процесса (например, "poller,3");
* pid – идентификатор процесса, до 65535. Для больших значений указать цель как "тип-процесса,N";
* -h, --help – вывод справочной информации и выход;
* -V, --version – вывод информацию о версии и выход.

Расположение по умолчанию файла конфигурации Прокси – /usr/local/etc/zabbix\_proxy.conf.

1. zabbix\_sender

zabbix\_sender – это утилита командной строки для отправки значений мониторинга на Сервер или Прокси. На Сервере должен быть создан элемент данных типа траппер с соответствующим ключом. Следует обратить внимание, что входящие данные будут приняты только от узлов сети, указанных в поле "Разрешенные хосты" для этого элемента данных.

Синтаксис:

**zabbix\_sender** [**-v**] **-z** сервер [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] **-s** узел-сети **-k** ключ **-o** значение

**zabbix\_sender** [**-v**] **-z** сервер [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] [**-T**] [**-N**] [**-r**] **-i** входной-файл

**zabbix\_sender** [**-v**] **-c** config-file [**-z** сервер] [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **-k** ключ **-o** значение

**zabbix\_sender** [**-v**] **-c** config-file [**-z** сервер] [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] [**-T**] [**-N**] [**-r**] **-i** входной-файл

**zabbix\_sender** [**-v**] **-z** сервер [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] **-s** узел-сети **--tls-connect** **cert** **--tls-ca-file** CA-file [**--tls-crl-file** CRL-file] [**--tls-server-cert-issuer** эмитент] [**--tls-server-cert-subject** субъект] **--tls-cert-file** cert-file **--tls-key-file** key-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] **-k** ключ **-o** значение

**zabbix\_sender** [**-v**] **-z** сервер [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **--tls-connect** **cert** **--tls-ca-file** CA-file [**--tls-crl-file** CRL-file] [**--tls-server-cert-issuer** эмитент] [**--tls-server-cert-subject** субъект] **--tls-cert-file** cert-file **--tls-key-file** key-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] [**-T**] [**-N**] [**-r**] **-i** входной-файл

**zabbix\_sender** [**-v**] **-c** config-file [**-z** сервер] [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **--tls-connect** **cert** **--tls-ca-file** CA-file [**--tls-crl-file** CRL-file] [**--tls-server-cert-issuer** эмитент] [**--tls-server-cert-subject** субъект] **--tls-cert-file** cert-file **--tls-key-file** key-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] **-k** ключ **-o** значение

**zabbix\_sender** [**-v**] **-c** config-file [**-z** сервер] [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **--tls-connect** **cert** **--tls-ca-file** CA-file [**--tls-crl-file** CRL-file] [**--tls-server-cert-issuer** эмитент] [**--tls-server-cert-subject** субъект] **--tls-cert-file** cert-file **--tls-key-file** key-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] [**-T**] [**-N**] [**-r**] **-i** входной-файл

**zabbix\_sender** [**-v**] **-z** сервер [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] **-s** узел-сети **--tls-connect** **psk** **--tls-psk-identity** PSK-identity **--tls-psk-file** PSK-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] **-k** ключ **-o** значение

**zabbix\_sender** [**-v**] **-z** сервер [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **--tls-connect** **psk** **--tls-psk-identity** PSK-identity **--tls-psk-file** PSK-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] [**-T**] [**-N**] [**-r**] **-i** входной-файл

**zabbix\_sender** [**-v**] **-c** config-file [**-z** сервер] [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **--tls-connect** **psk** **--tls-psk-identity** PSK-identity **--tls-psk-file** PSK-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] **-k** ключ **-o** значение

**zabbix\_sender** [**-v**] **-c** config-file [**-z** сервер] [**-p** порт] [**-I** IP-адрес] [**-t** тайм-аут] [**-s** узел-сети] **--tls-connect** **psk** **--tls-psk-identity** PSK-identity **--tls-psk-file** PSK-file [**--tls-cipher13** cipher-string] [**--tls-cipher** cipher-string] [**-T**] [**-N**] [**-r**] **-i** входной-файл

**zabbix\_sender -h**

**zabbix\_sender -V**

Параметры:

* -c, --config config-file – использование файла конфигурации config-file. **Sender** считывает сведения о Сервере из файла конфигурации agentd. По умолчанию **sender** не читает ни один конфигурационный файл. Поддерживаются только параметры **Hostname**, **ServerActive**, **SourceIP**, **TLSConnect**, **TLSCAFile**, **TLSCRLFile**, **TLSServerCertIssuer**, **TLSServerCertSubject**, **TLSCertFile**, **TLSKeyFile**, **TLSPSKIdentity** и **TLSPSKFile**. Все адреса, определенные в параметре конфигурации Агента **ServerActive**, используются для отправки данных. Если отправка пакетных данных на один адрес не удается, следующие пакеты на этот адрес не отправляются;
* -z, --zabbix-server сервер – имя хоста или IP-адрес Сервера. Если узел сети контролируется Прокси-сервером, вместо него следует использовать имя хоста или IP-адрес Прокси-сервера. При использовании вместе с **--config** переопределяет запись параметра **ServerActive**, указанного в файле конфигурации agentd;
* **-p**, **--port** порт – номер порта траппера Сервера, запущенного на Сервере. Значение по умолчанию – 10051. При использовании вместе с **--config** переопределяет значение порта параметра **ServerActive**, указанного в файле конфигурации agentd;
* **-I**, **--source-address** IP-адрес – исходящий IP-адрес. При использовании вместе с **--config**, переопределяет параметр **SourceIP**, указанный в файле конфигурации agentd;
* **-t**, **--Timeout** тайм-аут – время ожидания (тайм-аут). Допустимый диапазон: 1-300 секунд (по умолчанию: 60);
* **-s**, **--host** узел-сети – имя узла сети, которому принадлежит элемент данных (как зарегистрировано в веб-интерфейсе). IP-адрес хоста и DNS-имя работать не будут. При использовании вместе с **--config**, переопределяет параметр **Hostname**, указанный в файле конфигурации agentd;
* **-k**, **--key** ключ – ключ элемента данных для отправки значения;
* **-o**, **--value** значение – значение элемента данных;
* **-i**, **--input-file** входной-файл – загрузка значений из входного файла. Указать "**-**" (знак минуса) как **<input-file>**, чтобы считывать значения из стандартного ввода. Каждая строка файла содержит разделенные пробелом "**<узел-сети> <ключ> <значение>"**. Каждое значение должно быть указано в отдельной строке. Каждая строка должна содержать 3 разделенных пробелом поля "**<узел-сети> <ключ> <значение>"**, где "узел-сети" – это имя отслеживаемого узла сети, как зарегистрировано в веб-интерфейсе, "ключ" – это ключ целевого элемента, а "значение" – значение для отправки. Указать "**-**" (знак минуса) в качестве значения поля **<узел-сети>**, чтобы использовать имя узла сети из файла конфигурации Агента или из аргумента **--host**.

Пример строки входного файла:

**"Linux DB3" db.connections 43**

Тип значения должен быть правильно задан в настройках элемента данных в веб-интерфейсе. Sender отправит до 250 значений за одно соединение. Ограничение размера для отсылаемых данных из входного файла зависит от размера, описанного в протоколе коммуникации Подсистемы. Содержимое входного файла должно быть в кодировке UTF-8. Все значения из входного файла отправляются последовательно сверху вниз. Записи должен быть отформатированы, используя следующие правила:

* поддерживаются записи в кавычках и без кавычек;
* двойная кавычка – это символ, заключающий в кавычки;
* поля с пробелами должны быть заключены в кавычки;
* символы двойных кавычек и обратной косой черты внутри заключенной в кавычки записи должны быть экранированы обратной косой чертой (обратный слеш);
* экранирование не поддерживается в записях без кавычек;
* Escape-последовательности перевода строки (\n) поддерживаются в строках, заключенных в кавычки;
* экранирующие последовательности перевода строки обрезаются с конца записи.
* **-T**, **--with-timestamps** – этот параметр можно использовать только с параметром **--input-file**.

Каждая строка входного файла должна содержать 4 разделенных пробелами поля "**<узел-сети> <ключ> <метка-времени> <значение>"**. <Метка-времени> должна быть указана в формате UNIX timestamp. Если целевой элемент данных имеет триггеры, ссылающиеся на него, то все метки времени должны быть в порядке возрастания, иначе вычисление событий будет неправильным.

Пример строки входного файла:

**"Linux DB3" db.connections 1429533600 43**

Если значение с меткой времени отправляется для хоста, который находится в обслуживании с типом "без данных", то это значение будет отброшено, но возможно отправить значение с меткой времени для истекшего периода обслуживания, и оно будет принято.

* **-N**, **--with-ns** – этот параметр можно использовать только с параметром **--with-timestamps**.

Каждая строка входного файла должна содержать 5 разделенных пробелами полей "**<узел-сети> <ключ> <метка-времени> <наносекунды> <значение>"**.

Пример строки входного файла:

**"Linux DB3" db.connections 1429533600 7402561 43**

Для получения более подробной информации следует посмотреть опцию **--input-file**.

* **-r**, **--real-time** – отправка значений одно за другим, как только они будут получены. Это можно использовать при чтении из стандартного ввода;
* **--tls-connect** значение – способ подключения к Серверу или Прокси. Значения:
* **unencrypted** – подключение без шифрования (по умолчанию);
* **psk** – подключение с использованием TLS и общего ключа (pre-shared key);
* **cert** – подключение с использованием TLS и сертификата;
* **--tls-ca-file** CA-file – полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата партнера;
* **--tls-crl-file** CRL-file – полный путь к файлу, содержащему отозванные сертификаты;
* **--tls-server-cert-issuer** эмитент – разрешенный эмитент сертификата Сервера;
* **--tls-server-cert-subject** субъект – разрешенный субъект сертификата Сервера;
* **--tls-cert-file** cert-file – полный путь к файлу, содержащему сертификат или цепочку сертификатов;
* **--tls-key-file** key-file – полный путь к файлу, содержащему закрытый ключ (private key).
* **--tls-psk-identity** PSK-identity – строка идентификации PSK;
* **--tls-psk-file** PSK-file – полный путь к файлу, содержащему общий ключ (pre-shared key);
* **--tls-cipher13** cipher-string – строка шифра для OpenSSL 1.1.1 или новее для TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию. Эта опция недоступна, если версия OpenSSL ниже 1.1.1;
* **--tls-cipher** cipher-string – строка приоритета GnuTLS (для TLS 1.2 и выше) или строка шифра OpenSSL (только для TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию;
* **-v**, **--verbose** – подробный режим, **-vv** для еще более подробного;
* **-h**, **--help** – вывод справочной информации и выход;
* **-V**, **--version** – вывод информации о версии и выход.

Статус завершения равен 0, если значения были отправлены и все они были успешно обработаны Сервером. Если данные были отправлены, но обработка хотя бы одного из значений не удалась, статус завершения равен 2. Если отправка данных не удалось, статус завершения будет 1.

1. zabbix\_server

zabbix\_server – основной демон службы Сервера Подсистемы.

Синтаксис:

zabbix\_server [-c config-file]

zabbix\_server [-c config-file] -R runtime-option

zabbix\_server -h

zabbix\_server -V

Параметры:

* -c, --config config-file – использование альтернативного файла конфигурации config-file вместо файла по умолчанию;
* -f, --foreground – запуск Сервера на переднем плане;
* -R, --runtime-control runtime-option – выполнение административных функций в соответствии с runtime-option;
* -h, --help – вывод справочной информации и выход;
* -V, --version – вывод информации о версии и выход.

Опции выполнения административных функций:

* config\_cache\_reload – обновление кэша конфигурации. Игнорируется, если кэш уже в процессе загрузки. Для поиска PID-файла используется файл конфигурации по умолчанию (если опция "-c" не была указана), сигнал будет отправлен тому процессу, который указан в PID-файле;
* SNMP\_cache\_reload – перезагрузка кэша SNMP;
* housekeeper\_execute – выполнение процесса очистки истории. Игнорируется, если процесс очистки истории уже выполняется;
* trigger\_housekeeper\_execute – запуск процедуры очистки триггеров. Игнорируется, если процедура очистки триггеров уже работает;
* diaginfo[=раздел] – запись в журнал внутренней диагностической информации указанного раздела. Раздел может быть historycache, preprocessing, alerting, lld, valuecache. По умолчанию логируется диагностическая информация всех разделов;
* ha\_status – запись в журнал состояния кластера высокой доступности (HA);
* ha\_remove\_node[=цель] – удаление из кластера высокой доступности (HA) ноды, указанной по своему имени или ID. Следует обратить внимание, что активные/резервные ноды удалить нельзя;
* ha\_set\_failover\_delay[=delay] – изменение задержки автоматического переключения кластера высокой доступности (HA). Поддерживаются суффиксы времени, например: 10s, 1m;
* secrets\_reload – загрузка заново секретов из Хранилища;
* service\_cache\_reload – обновление кэша менеджера услуг;
* snmp\_cache\_reload – сброс кэша SNMP с очисткой свойства SNMP для всех узлов сети (engine time, engine boots, engine id, credentials);
* prof\_enable[=цель] – включение профилирования. Если цель не указана, то влияет на все процессы. Включенное профилирование предоставляет подробности всех блокировок на "чтение/запись" и мьютексов по имени функции;
* prof\_disable[=цель] – выключение профилирования. Если цель не указана, то влияет на все процессы;
* log\_level\_increase[=цель] – повышение уровня журналирования; если цель не указана, то влияет на все процессы;
* log\_level\_decrease[=цель] – понижение уровня журналирования; если цель не указана, то влияет на все процессы.

Цели управления уровнем журналирования:

* тип-процесса – все процессы указанного типа (alerter, alert manager, configuration syncer, discoverer, escalator, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, poller, preprocessing manager, preprocessing worker, proxy poller, self-monitoring, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector);
* тип-процесса,N – тип и номер процесса (например, "poller,3");
* pid – идентификатор процесса, до 65535. Для больших значений указать цель как "тип-процесса,N".

Расположение по умолчанию файла конфигурации Сервера – /usr/local/etc/zabbix\_server.conf.

1. zabbix\_web\_service

zabbix\_web\_service – веб-сервис для предоставления доступа к компонентам Подсистемы.

Синтаксис:

zabbix\_web\_service [-c config-file]

zabbix\_web\_service -h

zabbix\_web\_service -V

Параметры:

* -c, --config config-file – использование альтернативного файла конфигурации config-file вместо файла по умолчанию;
* -h, --help – вывод справочной информации и выход;
* -V, --version – вывод информации о версии и выход.

Расположение по умолчанию файла конфигурации веб-сервиса – /usr/local/etc/zabbix\_web\_service.conf.

Приложение Ж. Поддерживаемые функции

В таблице 90 перечислены поддерживаемые функции и группы функций.

Эти функции поддерживаются в триггерных выражениях и вычисляемых элементах данных.

Функции цикла поддерживаются только для агрегированных вычислений.

Группы и функции

| Группа функций | Функции |
| --- | --- |
| Функции агрегации | avg, bucket\_percentile, count, histogram\_quantile, item\_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp |
| Функции цикла (foreach) | avg\_foreach, bucket\_rate\_foreach, count\_foreach, exists\_foreach, last\_foreach, max\_foreach, min\_foreach, sum\_foreach |
| Побитовые функции | bitand, bitlshift, bitnot, bitor, bitrshift, bitxor |
| Функции даты и времени | date, dayofmonth, dayofweek, now, time |
| Функции истории | change, changecount, count, countunique, find, first, fuzzytime, last, logeventid, logseverity, logsource, monodec, monoinc, nodata, percentile, rate |
| Функции динамики изменений | baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum |
| Математические функции | abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate |
| Операторные функции | between, in |
| Функции прогнозирования | forecast, timeleft |
| Строковые функции | ascii, bitlength, Bytelength, char, concat, insert, left, length, ltrim, mid, repeat, replace, right, rtrim, trim |

1. Функции агрегации

Если не оговорено иначе, то все перечисленные в таблице 91 функции поддерживаются в:

* выражениях триггера;
* вычисляемых элементах данных.

Функции агрегации могут работать с:

* историей элементов данных, например: min(/узел\_сети/ключ,1h);
* функциями foreach в качестве единственного параметра, например: min(last\_foreach(/\*/ключ)) (только в вычисляемых элементах данных; нельзя использовать в триггерах).

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >";
* параметры, специфичные для функций, описываются с каждой функцией;
* параметры /узел\_сети/ключ и сек|#число)<:сдвиг\_времени> никогда не должны заключаться в кавычки.

Общие параметры:

* /узел\_сети/ключ – обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети;
* (сек|#число)<:сдвиг\_времени> – второй параметр для функций, ссылающихся на историю элемента данных узла сети,

где:

* сек – максимальный период вычисления в секундах (могут использоваться суффиксы времени), либо
* #число – максимальная область вычисления среди последних собранных значений (если начинается со знака решетки)
* сдвиг\_времени (опционально) – сдвиг точки вычислений по времени назад в прошлое.

Функции агрегации

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| avg (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Среднее значение элементов данных за указанный период вычислений. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаемы функции foreach: avg\_foreach, count\_foreach, exists\_foreach, last\_foreach, max\_foreach, min\_foreach, sum\_foreach Примеры: => avg(/узел\_сети/ключ,1h) – среднее значение за последний час до текущего момента => avg(/узел\_сети/ключ,1h:now-1d) – среднее значение за час с 25 часов назад до 24 часов назад от текущего момента => avg(/узел\_сети/ключ,#5)– среднее значение из пяти последних значений => avg(/узел\_сети/ключ,#5:now-1d)– среднее значение из пяти последних значений, исключая значения, полученные за последние 24 часа  Сдвиг времени полезен, когда нужно сравнить текущее среднее значение со средним значением некоторое время назад. |
| bucket\_percentile (фильтр,период\_времени,процент) | | |
| Вычисление процентиля из разрядов гистограммы. | фильтр – см. фильтр элементов данных "период\_времени" – см. период времени процент – процент (0-100) | Поддерживается только в вычисляемых элементах данных.  Эта функция является алиасом для histogram\_quantile(процент/100, bucket\_rate\_foreach(фильтр, период\_времени, 1)) |
| count (функция\_foreach(фильтр,<период\_времени>)) | | |
| Количество значений в массиве, возвращаемом функцией foreach. | функция\_foreach – функция foreach, для которой нужно подсчитать количество возвращаемых значений (с поддерживаемыми аргументами). | Поддерживаются типы значений: int Поддерживаются функции foreach: avg\_foreach, count\_foreach, exists\_foreach, last\_foreach, max\_foreach, min\_foreach, sum\_foreach Пример: => count(max\_foreach(/\*/net.if.in[\*],1h)) – количество элементов данных в net.if.in, которые получали данные в течение последнего часа до текущего момента  Следует обратить внимание, что использование count() с функциями foreach, относящимися к истории (max\_foreach, avg\_foreach и т.п.), может сказаться на производительности, в то время как использование функции exists\_foreach(), которая работает только с данными конфигурации, не окажет такого влияния. |
| histogram\_quantile (квантиль,разряд1,значение1,разряд2,значение2,...) | | |
| Вычисление φ-го квантиля из разрядов гистограммы. | квантиль – 0 ≤ φ ≤ 1 разрядN, значениеN – вручную введенные пары (>=2) параметров либо результат вызова bucket\_rate\_foreach | Поддерживается только в вычисляемых элементах данных. Поддерживаются функции foreach: bucket\_rate\_foreach По функционалу соответствует "histogram\_quantile [en]" из PromQL. Возвращает -1, если значения последнего разряда "Infinity" ("+inf") равны 0. Примеры: => histogram\_quantile(0.75,1.0,last(/узел\_сети/rate\_bucket[1.0]),"+Inf",last(/узел\_сети/rate\_bucket[Inf]) => histogram\_quantile(0.5,bucket\_rate\_foreach(//item\_key,30s)) |
| item\_count (фильтр) | | |
| Подсчет существующих в конфигурации элементов данных, соответствующих критериям фильтра. | фильтр – критерий для выбора элементов данных, позволяет ссылаться на группы узлов сети, узлы сети, ключи элементов данных и теги. Поддерживаются подстановочные символы. | Поддерживается только в вычисляемых элементах данных. Поддерживаются типы значений: int Работает как алиас для функции count(exists\_foreach(фильтр)). Пример: => item\_count(/\*/agent.ping?[group="Host group 1"]) – количество узлов сети с элементом данных agent.ping в группе узлов сети "Host group 1" |
| kurtosis (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Эксцесс ("островершинность") распределения вероятностей собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int  Поддерживаются функции foreach: last\_foreach  Пример: => kurtosis(/узел\_сети/ключ,1h) – эксцесс за последний час до текущего момента |
| mad (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Среднее абсолютное отклонение собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Пример: => mad(/узел\_сети/ключ,1h)– среднее абсолютное отклонение за последний час до текущего момента |
| max (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Наибольшее значение элемента данных в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: avg\_foreach, count\_foreach, exists\_foreach, last\_foreach, max\_foreach, min\_foreach, sum\_foreach Пример: => max(/узел\_сети/ключ,1h) – min(/узел\_сети/ключ,1h) – вычисление разницы между наибольшим и наименьшим значениями за последний час до текущего момента (дельта значений) |
| min (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Наименьшее значение элемента данных в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: avg\_foreach, count\_foreach, exists\_foreach, last\_foreach, max\_foreach, min\_foreach, sum\_foreach Пример: => max(/узел\_сети/ключ,1h) – min(/узел\_сети/ключ,1h)– вычисление разницы между наибольшим и наименьшим значениями за последний час до текущего момента (дельта значений) |
| skewness (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Асимметрия распределения вероятностей собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Пример: => skewness(/узел\_сети/ключ,1h) – коэффициент асимметрии за последний час до текущего момента |
| stddevpop (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Среднеквадратическое отклонение для генеральной совокупности собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Пример: => stddevpop(/узел\_сети/ключ,1h) – стандартное отклонение для генеральной совокупности за последний час до текущего момента |
| stddevsamp (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Среднеквадратическое отклонение для выборки собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Пример: => stddevsamp(/узел\_сети/ключ,1h) – стандартное отклонение для выборки за последний час до текущего момента |
| sum (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Сумма собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: avg\_foreach, count\_foreach, exists\_foreach, last\_foreach, max\_foreach, min\_foreach, sum\_foreach Пример: => sum(/узел\_сети/ключ,1h) – сумма значений за последний час до текущего момента |
| sumofsquares (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Сумма квадратов собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Пример: => sumofsquares(/узел\_сети/ключ,1h) – сумма квадратов за последний час до текущего момента |
| varpop (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Генеральная (совокупная) дисперсия собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Пример: => varpop(/узел\_сети/ключ,1h) – дисперсия случайной величины за последний час до текущего момента |
| varsamp (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>) | | |
| Выборочная дисперсия собранных значений в течение определенного периода оценки. | См. общие параметры. | Поддерживаются типы значений: float, int Поддерживаются функции foreach: last\_foreach Для работы этой функции требуется как минимум два значения данных. Пример: => varsamp(/узел\_сети/ключ,1h) – выборочная дисперсия за последний час до текущего момента |

1. Функции цикла

Функции foreach (таблица 92) используются в агрегированных вычислениях, чтобы вернуть одно агрегированное значение для каждого из элементов данных, выбранных используемым фильтром элементов данных; возвращают массив значений.

Например, функция avg\_foreach вернет массив значений, в котором каждое значение является средним значением истории выбранного элемента данных за указанный интервал времени.

Фильтр элементов данных является частью синтаксиса, используемого функциями foreach. В фильтре элементов данных поддерживаются подстановочные символы, поэтому нужные элементы данных могут быть выбраны достаточно гибко.

Поддерживаемые функции

| Функция | Описание |
| --- | --- |
| avg\_foreach | Возвращает среднее значение для каждого элемента данных. |
| bucket\_rate\_foreach | Возвращает пары (верхняя граница разряда, значение доли), подходящие для использования в функции histogram\_quantile(), где значение "верхняя граница разряда" – это значение параметра ключа элемента данных, определенного параметром <номер параметра>. |
| count\_foreach | Возвращает количество значений для каждого элемента данных. |
| exists\_foreach | Возвращает "1" для каждого активированного элемента данных. |
| last\_foreach | Возвращает последнее значение для каждого элемента данных. |
| max\_foreach | Возвращает наибольшее значение для каждого элемента данных. |
| min\_foreach | Возвращает наименьшее значение для каждого элемента данных. |
| sum\_foreach | Возвращает сумму значений для каждого элемента данных. |

Функции foreach поддерживают два общих параметра: фильтр\_элементов\_данных и период\_времени:

функция\_foreach(фильтр\_элементов\_данных,период\_времени)

Например:

avg\_foreach(/\*/mysql.qps?[group="MySQL Servers"],5m)

вернет среднее за пять минут значение для каждого из элементов данных mysql.qps в группе "MySQL server".

Следует обратить внимание, что некоторые функции поддерживают дополнительные параметры.

Фильтр элементов данных состоит из четырех частей:

/узел\_сети/ключ[параметры]?[условия]

где:

* узел\_сети – имя узла сети;
* ключ – ключ элемента данных (без параметров);
* параметры – параметры ключа элемента данных;
* условия – условия на основе группы узлов сети и/или тега элемента данных (как выражение).

Пробелы допустимы только внутри выражения условий.

Использование подстановочных символов:

* Подстановочный символ может быть использован для замены имени узла сети, ключа элемента данных или отдельного параметра ключа элемента данных.
* Либо узел сети, либо ключ элемента данных должны быть указаны без подстановочного символа. Так, /узел\_сети/\* и /\*/ключ являются корректными фильтрами, но /\*/\* некорректен.
* Подстановочный символ не может быть использован как часть имени узла сети, ключа элемента данных, параметра ключа элемента данных.
* Подстановочный символ соответствует не более чем одному параметру ключа элемента данных. Поэтому подстановочный символ должен быть указан для каждого параметра отдельно (т.е. ключ[abc,\*,\*]).

Выражение условий поддерживает:

* операнды:
* group – группа узлов сети;
* tag – тег элемента данных;
* "<текст>" – строковая константа, с использованием символа обратной косой черты "\" для экранирования символов (") и (\);
* операторы сравнения, чувствительного к регистру: "=", "<", ">";
* логические операторы: and, or, not;
* группировка при помощи круглых скобок: "( )".

Строковые константы обязательно должны заключаться в кавычки. Поддерживается только чувствительное к регистру сравнение полных строк.

При указании в фильтре тегов (т.е. tag="имя\_тега:значение"), двоеточие (":") используется как разделитель. Все после него рассматривается как значение тега. Поэтому в данный момент не поддерживается указание тегов, содержащих двоеточие (":") внутри имени.

Все упомянутые элементы данных должны существовать и собирать данные. В вычисления включаются только активированные элементы данных на активированных узлах сети.

Если ключ упомянутого элемента данных меняется, фильтр должен быть обновлен вручную.

Упоминание родительской группы узлов сети включает родительскую и все вложенные группы узлов сети с их элементами данных.

Второй параметр позволяет указать период времени для агрегирования. Период времени может быть указан только как время; количество значений (с префиксом #) не поддерживается.

Для удобства в этом параметре могут использоваться поддерживаемые символы единиц времени, например: "5m" (пять минут) вместо "300s" (300 секунд) или "1d" (одни сутки) вместо "86400" (86400 секунд).

Период времени игнорируется Сервером, если передан для функции last\_foreach, и может быть опущен:

last\_foreach(/\*/ключ?[group="группа узлов сети"])

Период времени не поддерживается для функции exists\_foreach.

Третий необязательный параметр поддерживается функцией bucket\_rate\_foreach:

bucket\_rate\_foreach(фильтр\_элементов\_данных,период\_времени,<номер\_параметра>)

где <номер\_параметра> – это положение значения разряд в ключе элемента данных. Например, если в myItem[aaa,0.2] значением разряд является "0.2", то его положение – это 2.

Значением по умолчанию для "<номера параметра>" является "1".

Таблица 93 иллюстрирует, как каждая функция ведет себя в случае ограниченной доступности узла сети/элемента данных и данных истории.

Поведение, зависящее от доступности

| Функция | Узел сети деактивирован | Узел сети с данными недоступен | Узел сети без данных недоступен | Элемент данных деактивирован | Элемент данных не поддерживается | Ошибка извлечения данных (SQL) |
| --- | --- | --- | --- | --- | --- | --- |
| avg\_foreach | игнорировать | вернуть среднее | игнорировать | игнорировать | игнорировать | игнорировать |
| bucket\_rate\_foreach | игнорировать | вернуть bucket rate | игнорировать | игнорировать | игнорировать | игнорировать |
| count\_foreach | игнорировать | вернуть количество | 0 | игнорировать | игнорировать | игнорировать |
| exists\_foreach | игнорировать | 1 | 1 | игнорировать | 1 | n/a |
| last\_foreach | игнорировать | вернуть последнее | игнорировать | игнорировать | игнорировать | игнорировать |
| max\_foreach | игнорировать | вернуть максимум | игнорировать | игнорировать | игнорировать | игнорировать |
| min\_foreach | игнорировать | вернуть минимум | игнорировать | игнорировать | игнорировать | игнорировать |
| sum\_foreach | игнорировать | вернуть сумму | игнорировать | игнорировать | игнорировать | игнорировать |

Если для элемента данных указано "игнорировать", к агрегированию ничего не добавляется.

1. Побитовые функции

Все перечисленные в таблице 94 функции поддерживаются в:

* триггерных выражениях;
* вычисляемых элементах данных.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* в качестве параметров допустимы выражения;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >".

Побитовые функции

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| bitand (значение,маска) | | |
| Результат операции "побитовое И" значения элемента данных и маски. | значение – проверяемое значение маска (обязательно) – 64-битное целое без знака (0 – 18446744073709551615) | Поддерживаются типы значений: int Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путем сравнения с числом 4, а не 100. Примеры: => bitand(last(/узел\_сети/ключ),12)=8 or bitand(last(/узел\_сети/ключ),12)=4 – выставлены третий или четвертый биты, но не оба сразу => bitand(last(/узел\_сети/ключ),20)=16 – третий бит не выставлен, а пятый – выставлен. |
| bitlshift (значение,сдвиг\_битов) | | |
| Побитовый сдвиг значения элемента данных влево. | значение – проверяемое значение сдвиг\_битов (обязательно) – количество битов для сдвига | Поддерживаются типы значений: int Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путем сравнения с числом 4, а не 100. |
| bitnot (значение) | | |
| Результат операции "побитовое НЕ" значения элемента данных. | значение – проверяемое значение | Поддерживаются типы значений: int Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путем сравнения с числом 4, а не 100. |
| bitor (значение,маска) | | |
| Результат операции "побитовое ИЛИ" значения элемента данных и маски. | значение – проверяемое значение маска (обязательно) – 64-битное целое без знака (0 – 18446744073709551615) | Поддерживаются типы значений: int Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путем сравнения с числом 4, а не 100. |
| bitrshift (значение,сдвиг\_битов) | | |
| Побитовый сдвиг значения элемента данных вправо. | значение – проверяемое значение сдвиг\_битов (обязательно) – количество битов для сдвига | Поддерживаются типы значений: int Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путем сравнения с числом 4, а не 100. |
| bitxor (значение,маска) | | |
| Результат операции "побитовое исключительное ИЛИ" значения элемента данных и маски. | значение – проверяемое значение маска (обязательно) – 64-битное целое без знака (0 – 18446744073709551615) | Поддерживаются типы значений: int Несмотря на то, что сравнение выполняется побитово, все значения должны быть представлены и затем возвращаются в десятичном формате. Например, проверка третьего бита выполняется путем сравнения с числом 4, а не 100. |

1. Функции даты и времени

Все перечисленные в таблице 95 функции поддерживаются в:

* триггерных выражениях;
* вычисляемых элементах данных.

Функции даты и времени не могут быть использованы в выражении сами по себе; в выражении должна присутствовать еще хотя бы одна не относящаяся к дате/времени функция, ссылающаяся на элемент данных узла сети.

Функции даты и времени

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| date | | |
| Текущая дата в формате ГГГГММДД (YYYYMMDD). |  | Пример: => date()<20240101 |
| dayofmonth | | |
| Число месяца в диапазоне от 1 до 31. |  | Пример: => dayofmonth()=1 |
| dayofweek | | |
| День недели в диапазоне от 1 до 7 (Пн – 1, Вс – 7). |  | Пример (только рабочие дни): => dayofweek()<6  Пример (только выходные): => dayofweek()>5 |
| now | | |
| Количество секунд с начала эпохи UNIX (00:00:00 1 января 1970 года по Гринвичу). |  | Пример: => now()<1640998800 |
| time | | |
| Текущее время в формате ЧЧММСС (HHMMSS). |  | Пример (только ночное время, 00:00-06:00): => time()<060000 |

1. Функции истории

Все перечисленные в таблице 96 функции поддерживаются в:

* триггерных выражениях;
* вычисляемых элементах данных.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >";
* параметры, специфичные для функций, описываются с каждой функцией;
* параметры /узел\_сети/ключ и (сек|#число)<:сдвиг\_времени> не должны заключаться в кавычки.

Общие параметры:

* /узел\_сети/ключ – обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети;
* (сек|#число)<:сдвиг\_времени> – второй параметр для функций, ссылающихся на историю элемента данных узла сети, где:
* сек – максимальный период вычисления в секундах (могут использоваться суффиксы времени), либо;
* #число – максимальная область вычисления среди последних собранных значений (если начинается со знака решетки);
* сдвиг\_времени (опционально) – сдвиг точки вычислений по времени назад в прошлое.

Функции истории

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| change (/узел\_сети/ключ) | | |
| Величина разницы между последним и предыдущим значениями. |  | Поддерживаемые типы значений: float, int, str, text, log Для строк возвращается: 0 – значения равны 1 – значения отличаются Пример: => change(/узел\_сети/ключ)>10 Разница между числовыми значениями будет вычисляться, как показано на примере следующих значений ("предыдущее" и "последнее" значение = разница): "1" and "5" = +4 "3" and "1" = -2 "0" and "-2.5" = -2.5 |
| changecount (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>,<режим>) | | |
| Количество изменений между соседними значениями за определенный период вычисления. | См. общие параметры. режим (опционально; должен быть в двойных кавычках) Поддерживаются режимы: all – считать все изменения (по умолчанию) dec – считать убывания значений inc – считать возрастания значений | Поддерживаемые типы значений: float, int, str, text, log Для нечисловых типов значений параметр режим игнорируется. Примеры: => changecount(/узел\_сети/ключ, 1w) – количество изменений значения за последнюю неделю до текущего момента => changecount(/узел\_сети/ключ,#10,"inc")– количество возрастаний значений (относительно соседнего значения) среди последних 10 значений => changecount(/узел\_сети/ключ,24h,"dec") – количество убываний значений (относительно соседнего значения) за последние 24 часа до текущего момента |
| count (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>,<оператор>,<шаблон>) | | |
| Количество значений за период вычисления. | оператор (опционально) должен быть в двойных кавычках) Поддерживаются операторы: eq – равно (по умолчанию для integer и float) ne – не равно gt – больше, чем ge – больше или равно lt – меньше, чем le – меньше или равно like (по умолчанию для string, text и log) – содержит подстроку шаблона (чувствительно к регистру) bitand – побитовое И regexp – чувствительное к регистру соответствие регулярному выражению, данному в шаблоне iregexp – нечувствительное к регистру соответствие регулярному выражению, данному в шаблоне Примечание: eq (по умолчанию), ne, gt, ge, lt, le, band, regexp, iregexp поддерживаются для целочисленных элементов данных eq (по умолчанию), ne, gt, ge, lt, le, regexp, iregexp поддерживаются для элементов данных с плавающей точкой like (по умолчанию), eq, ne, regexp, iregexp поддерживаются для элементов данных символ, текст и журнал (лог) шаблон (опционально) – искомый шаблон (строковые аргументы должны быть в двойных кавычках) | Поддерживаемые типы значений: float, integer, string, text, log Элементы данных с плавающей точкой совпадают с точностью 2.22e-16; если база данных не обновлена, то точность составляет 0.000001. Если третий параметр bitand, то четвертый параметр шаблон может быть указан как два числа, разделенных косой чертой ("/"): число\_для\_сравнения/маска. count() вычисляет "побитовое И" из значения и маски и сравнивает результат с числом\_для\_сравнения. Если результат "побитового И" равен числу\_для\_сравнения, то значение считается. Если число\_для\_сравнения и маска равны, требуется указывать только маску (без "/"). Если третий параметр regexp или iregexp, то четвертый параметр шаблон может быть обычным или глобальным (начинающимся с "@") регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. При проверке соответствия регулярному выражению, значения с плавающей точкой всегда будут представлены с 4 цифрами после десятичной точки ("."). Также обратить внимание, что для больших чисел различие между десятичным (хранящимся в базе данных) и двоичным (используемом сервером) представлениями может повлиять на 4-ю десятичную цифру. |
| countunique (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>,<оператор>,<шаблон>) | | |
| Количество уникальных значений за определенный период вычисления. | См. общие параметры. оператор (опционально) должен быть в двойных кавычках) Поддерживаются операторы: eq – равно (по умолчанию для integer и float) ne – не равно gt – больше, чем ge – больше или равно lt – меньше, чем le – меньше или равно like (по умолчанию для string, text и log) – значение содержит подстроку, заданную в шаблоне (с учетом регистра) bitand – побитовое И regexp – чувствительное к регистру соответствие регулярному выражению, данному в шаблоне iregexp – нечувствительное к регистру соответствие регулярному выражению, данному в шаблоне Примечание: eq (по умолчанию), ne, gt, ge, lt, le, band, regexp, iregexp поддерживаются для целочисленных элементов данных eq (по умолчанию)), ne, gt, ge, lt, le, regexp, iregexp поддерживаются для элементов данных с плавающей точкой like (по умолчанию)), eq, ne, regexp, iregexp поддерживаются для элементов данных символ, текст и журнал (лог) шаблон (опционально) – искомый шаблон (строковые аргументы должны быть в двойных кавычках) | Поддерживаемые типы значений: float, integer, string, text, log Элементы данных с плавающей точкой совпадают с точностью 2.22e-16; если база данных не обновлена, то точность составляет 0.000001. Если третий параметр bitand, то четвертый параметр шаблон может быть указан как два числа, разделенных косой чертой ("/"): число\_для\_сравнения/маска. countunique() вычисляет "побитовое И" из значения и маски и сравнивает результат с числом\_для\_сравнения. Если результат "побитового И" равен числу\_для\_сравнения, то значение считается. Если число\_для\_сравнения и маска равны, требуется указывать только маску (без "/"). Если третий параметр regexp или iregexp, то четвертый параметр шаблон может быть обычным или глобальным (начинающимся с "@") регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. При проверке соответствия регулярному выражению, значения с плавающей точкой всегда будут представлены с 4 цифрами после десятичной точки ("."). Также обратить внимание, что для больших чисел различие между десятичным (хранящимся в базе данных) и двоичным (используемом сервером) представлениями может повлиять на 4-ю десятичную цифру. |
| find (/узел\_сети/ключ,<(сек|#число)<:сдвиг\_времени>>,<оператор>,<шаблон>) | | |
| Поиск соответствующего значения. | См. общие параметры. сек или #число (опционально) – если не указано, то по умолчанию проверяется последнее значение оператор (опционально) должен быть заключен в двойные кавычки) Поддерживаются операторы: eq – равно (по умолчанию для integer и float) ne – не равно gt – больше, чем ge – больше или равно lt – меньше, чем le – меньше или равно like (по умолчанию для string, text и log) – значение содержит подстроку, заданную в шаблоне (с учетом регистра) bitand – побитовое И regexp – чувствительное к регистру соответствие регулярному выражению, данному в шаблоне iregexp – нечувствительное к регистру соответствие регулярному выражению, данному в шаблоне Примечание: eq (по умолчанию), ne, gt, ge, lt, le, band, regexp, iregexp поддерживаются для целочисленных элементов данных eq (по умолчанию), ne, gt, ge, lt, le, regexp, iregexp поддерживаются для элементов данных с плавающей точкой like (по умолчанию), eq, ne, regexp, iregexp поддерживаются для элементов данных символ, текст и журнал (лог) шаблон – искомый шаблон (строковые аргументы должны быть в двойных кавычках); регулярное выражение Perl Compatible Regular Expression (PCRE), если оператор – regexp или iregexp. | Поддерживаемые типы значений: float, int, str, text, log Возвращает: 1 – найдено 0 – иначе Если обрабатывается более одного значения, "1" возвращается, если имеется по крайней мере одно совпадающее значение. С третьим параметром regexp или iregexp, четвертый параметр шаблон может быть обычным или глобальным (начинающимся с "@") регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Пример: => find(/узел\_сети/ключ,10m,"like","error") – найти значение, содержащее "error" за последние 10 минут до текущего момента |
| first (/узел\_сети/ключ,сек<:сдвиг\_времени>) | | |
| Первое (самое старое) значение за указанный период вычисления. | См. общие параметры. | Поддерживаемые типы значений: float, int, str, text, log Пример: => first(/узел\_сети/ключ,1h) – извлечь самое старое значение за последний час вплоть до текущего момента |
| fuzzytime (/узел\_сети/ключ,сек) | | |
| Проверка того, насколько время пассивного Агента отличается от времени Сервера/Прокси. | См. общие параметры. | Поддерживаемые типы значений: float, int Возвращает: 1 – если разница между значением пассивного элемента данных (как штампа времени) и штампом времени Сервера/Прокси (временем получения значения) меньше или равна сек секунд 0 – иначе Обычно используется с "system.localtime" для проверки, что локальное время синхронизировано с локальным временем Сервера. Следует обратить внимание, что "system.localtime" должно быть настроено как пассивная проверка. Также можно использовать с ключом vfs.file.time[/путь/файл,modify] для проверки, что файл не обновлялся длительное время. Пример: => fuzzytime(/узел\_сети/ключ,60s)=0 – обнаружение проблемы, если разница во времени превышает 60 секунд Эту функцию не рекомендуется использовать в сложных триггерных выражениях (с участием нескольких элементов данных), поскольку это может привести к непредвиденным результатам (разница по времени будет считаться по наиболее свежей метрике), например, в таком: fuzzytime(/узел\_сети/system.localtime,60s)=0 or last(/узел\_сети/trap)<>0 |
| last (/узел\_сети/ключ,<#число<:сдвиг\_времени>>) | | |
| Самое новое значение. | См. общие параметры. #число (опционально) – N-ое значение из самых новых | Поддерживаемые типы значений: float, int, str, text, log  Следует обратить внимание, что #число (с решеткой, #N) здесь работает иначе, чем во многих других функциях. Например: last() всегда идентичен last(#1) last(#3) – третье из самых новых значение (не три последних значения) Подсистема не гарантирует точный порядок значений, если в истории существует более двух значений за одну секунду. |
| logeventid (/узел\_сети/ключ,<#число<:сдвиг\_времени>>,<шаблон>) | | |
| Проверка, соответствует ли указанному регулярному выражению идентификатор события последней записи из журнала. | См. общие параметры. #число (опционально) – N-ое из наиболее недавних значений шаблон (опционально) – регулярное выражение, описывающее нужный шаблон, в формате Perl Compatible Regular Expression (PCRE) (строковые аргументы должны быть заключены в двойные кавычки). | Поддерживаемые типы значений: log  Возвращает: 0 – нет соответствия 1 – соответствует |
| logseverity (/узел\_сети/ключ,<#число<:сдвиг\_времени>>) | | |
| Важность события последней записи в журнале. | См. общие параметры.  #число (опционально) – N-ое из наиболее недавних значений | Поддерживаемые типы значений: log Возвращает: 0 – важность по умолчанию N – важность (целое число, полезно для журналов событий Windows:  1 – Уведомление, 2– предупреждение,4 – Ошибка, 7 – Аудит отказов, 8 – Аудит успехов, 9 – Критическая ошибка, 10 – Детали). Подсистема берет важность журнала из поля Информация журнала событий Windows. |
| logsource (/узел\_сети/ключ,<#число<:сдвиг\_времени>>,<шаблон>) | | |
| Проверка, соответствует ли регулярному выражению источник последней записи в журнале. | См. общие параметры.  #число (опционально) – N-ое из наиболее недавних значений  шаблон (опционально) – регулярное выражение, описывающее нужный шаблон, в формате Perl Compatible Regular Expression (PCRE) (строковые аргументы должны быть заключены в двойные кавычки). | Поддерживаемые типы значений: log  Возвращает: 0 – нет соответствия 1 – соответствует  Обычно используется для журналов событий Windows. Например, logsource("VMware Server"). |
| monodec (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>,<режим>) | | |
| Проверка наличия монотонного убывания значений. | См. общие параметры.  режим (должен быть в двойных кавычках) – weak (каждое значение меньше или равно предыдущему; по умолчанию) или strict (каждое значение строго уменьшается) | Поддерживаемые типы значений: int Возвращает 1, если все элементы в течение указанного периода времени непрерывно уменьшаются, иначе – 0. Пример: => monodec(/Host1/system.swap.size[all,free],60s) + monodec(/Host2/system.swap.size[all,free],60s) + monodec(/Host3/system.swap.size[all,free],60s) – посчитать, на каком количестве узлов сети было уменьшение свободного места в области подкачки |
| monoinc (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>,<режим>) | | |
| Проверка наличия монотонного возрастания значений. | См. общие параметры.  режим (должен быть в двойных кавычках) – weak (каждое значение больше или равно предыдущему; по умолчанию) или strict (каждое значение строго возрастает) | Поддерживаемые типы значений: int Возвращает 1, если все элементы в течение указанного периода времени непрерывно возрастают, иначе 0. Пример: => monoinc(/Host1/system.localtime,#3,"strict")=0 – проверка того, постоянно ли увеличивается локальное время системы |
| nodata (/узел\_сети/ключ,сек,<режим>) | | |
| Проверка отсутствия полученных данных. | См. общие параметры.  сек – период должен быть не менее 30 секунд, так как процесс синхронизации истории (history syncer) вычисляет эту функцию раз в 30 секунд.  nodata(/узел\_сети/ключ,0) запрещено.  режим – при значении strict (в двойных кавычках), эта функция будет нечувствительна к доступности Прокси (см. комментарии для более подробной информации). | Поддерживаются все типы значений. Возвращает: 1 – если нет полученных данных за указанный период времени 0 – иначе Триггеры "nodata", контролируемые Прокси-серверами, по умолчанию чувствительны к доступности Прокси – если Прокси становится недоступным, триггеры "nodata" не будут срабатывать немедленно после восстановления соединения, а пропустят данные за задержанный период. Следует обратить внимание, что для пассивных Прокси подавление активируется если соединение восстанавливается более чем на 15 секунд, но не менее чем 2 & ProxyUpdateFrequency секунд позже. Для активных Прокси подавление активируется, если соединение восстанавливается более чем 15 секунд позже. Чтобы выключить чувствительность к доступности Прокси, использовать третий параметр, например: nodata(/узел\_сети/ключ,5m,"strict"). Следует обратить внимание, эта функция отобразит ошибку в случае, если за указанный в первом параметре период: - нет данных и Сервер был перезапущен - нет данных и было завершено обслуживание - нет данных и элемент данных был добавлен или заново активирован Ошибки отображаются в колонке Инфо в настройке триггеров. Эта функция может работать неправильно, если есть расхождения по времени между Сервером, Прокси и Агентом. См. также: требования по синхронизации времени. |
| percentile (/узел\_сети/ключ,(сек|#число)<:сдвиг\_времени>,процент) | | |
| P-й процентиль периода, где P (процент) указывается третьим параметром. | См. общие параметры.  процент – число с плавающей точкой от 0 до 100 (включительно), до 4 цифр после десятичной точки | Поддерживаемые типы значений: float, int |
| rate (/узел\_сети/ключ,сек<:сдвиг\_времени>) | | |
| Усредненная за секунду скорость увеличения монотонно возрастающего счетчика в течение указанного периода времени. | См. общие параметры. | Поддерживаемые типы значений: float, int Функционально соответствует функции "rate [en]" из PromQL. Пример: => rate(/узел\_сети/ключ,30s) – если за 30 секунд значение монотонно возросло на 20, эта функция вернет 0.67. |

1. Функции динамики изменений

Функции динамики изменений, в отличие от функций истории, используют для вычислений данные динамики изменений.

Динамика изменений сохраняется в виде агрегированных за час значений. Функции динамики изменений используют эти усредненные за час значения, и, таким образом, полезны для долгосрочного анализа.

Результаты функций динамики изменений кэшируются, поэтому несколько вызовов той же функции с теми же параметрами извлекут информацию из базы данных только однократно. Кэш функций динамики изменений управляется параметром Сервера TrendFunctionCacheSize.

Триггеры, которые ссылаются только на функции динамики изменений, пересчитываются один раз за наименьший период времени в выражении. Например, такой триггер:

trendavg(/узел\_сети/ключ,1d:now/d) > 1 or trendavg(/узел\_сети/ключ2,1w:now/w) > 2

будет пересчитываться раз в сутки. Если триггер содержит как функции динамики изменений, так и функции истории (либо относящиеся к времени), то он пересчитывается в соответствии с обычными принципами.

Все перечисленные в таблице 97 функции поддерживаются в:

* выражениях триггера;
* вычисляемых элементах данных.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >";
* параметры, специфичные для функций, описываются с каждой функцией;
* параметры /узел\_сети/ключ и период\_времени:сдвиг\_времени никогда не должны заключаться в кавычки.

Общие параметры:

* /узел\_сети/ключ – обязательный первый параметр;
* период\_времени:сдвиг\_времени – второй параметр,

где:

* период\_времени – период времени (минимум "1h"), определенный как <N><единица\_времени> где N – количество единиц времени, единица\_времени – h (час), d (сутки), w (неделя), M (месяц) или y (год);
* сдвиг\_времени – сдвиг периода времени (см. примеры функций).

Функции динамики изменений

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| baselinedev (/узел\_сети/ключ,период\_данных:сдвиг\_времени,сезон\_ед,сезон\_кол-во) | | |
| Возвращает количество стандартных отклонений (по алгоритму stddevpop, среднеквадратичное отклонение) между последним периодом данных и теми же периодами данных в предыдущие сезоны. | период\_данных – период сбора данных в течение сезона, определенный как <N><единица\_времени>, где N – количество единиц времени единица\_времени – h (час), d (сутки), w (неделя), M (месяц) или y (год), должно быть меньше или равно сезону – смещение периода времени (см. примеры) сезон\_ед – продолжительность одного сезона (h, d, w, M, y), не может быть меньше периода данных сезон\_кол-во – количество оцениваемых сезонов | Примеры: => baselinedev(/узел\_сети/ключ,1d:now/d,"M",6) – расчет количества стандартных отклонений (генеральная совокупность) между предыдущими сутками и теми же сутками в предыдущие 6 месяцев. Если дата в предыдущем месяце не существует, будет использован последний день месяца (31 июля будет проанализировано в сравнении с 31 января, 28 февраля, ..., 30 июня). => baselinedev(/узел\_сети/ключ,1h:now/h,"d",10) – расчет количества стандартных отклонений (генеральная совокупность) между предыдущим часом и тем же часом на протяжении декады до вчерашнего дня. |
| baselinewma (/узел\_сети/ключ,период\_данных:Сдвиг\_времени,сезон\_ед,сезон\_кол-во) | | |
| Вычисляет базовый уровень путем усреднения данных за один и тот же интервал времени в течение нескольких равных периодов времени ("сезонов") с использованием алгоритма взвешенного скользящего среднего. | период\_данных – период сбора данных в течение сезона, определенный как <N><единица\_времени>, где N – количество единиц времени единица\_времени – h (час), d (сутки), w (неделя), M (месяц) или y (год), должно быть меньше или равно сезону Сдвиг\_времени – смещение периода времени, определяет конец интервала сбора данных в сезонах (см. примеры)  сезон\_ед – продолжительность одного сезона (h, d, w, M, y), не может быть меньше периода данных  сезон\_кол-во – количество оцениваемых сезонов | Примеры: => baselinewma(/узел\_сети/ключ,1h:now/h,"d",3) – расчет базового уровня за последний полный час в течение 3-дневного периода, который закончился вчера. Если "now" (сейчас) – это 13:30 понедельника, то будут проанализированы данные по часу 12:00-12:59 за пятницу, субботу и воскресенье. => baselinewma(/узел\_сети/ключ,2h:now/h,"d",3) – расчет базового уровня на основе последних двух часов в течение 3-дневного периода, который закончился вчера. Если "now" (сейчас) – это 13:30 понедельника, то будут проанализированы данные для интервала 11:00-12:59 за пятницу, субботу и воскресенье. => baselinewma(/узел\_сети/ключ,1d:now/d,"M",4) – расчет базового уровня на основе того же числа месяца, что и "вчера", за период 4 месяца, предшествующих предыдущему полному месяцу. Если нужной даты не существует, берется последний день месяца. Если сегодня 1 сентября, то будут проанализированы данные за 31 июля, 30 июня, 31 мая и 30 апреля. |
| trendavg (/узел\_сети/ключ,период\_времени:Сдвиг\_времени) | | |
| Среднее из значений динамики изменений за определенный период времени. | См. общие параметры. | Примеры: => trendavg(/узел\_сети/ключ,1h:now/h) – среднее за предыдущий час (например, 12:00-13:00) => trendavg(/узел\_сети/ключ,1h:now/h-1h) – среднее за час два часа назад (11:00-12:00) => trendavg(/узел\_сети/ключ,1h:now/h-2h) – среднее за час три часа назад (10:00-11:00) => trendavg(/узел\_сети/ключ,1M:now/M-1y) – среднее за предыдущий месяц год назад |
| trendcount (/узел\_сети/ключ,период\_времени:Сдвиг\_времени) | | |
| Количество успешно извлеченных значений динамики изменений за определенный период времени. | См. общие параметры. | Примеры: => trendcount(/узел\_сети/ключ,1h:now/h) – количество за предыдущий час (например, 12:00-13:00) => trendcount(/узел\_сети/ключ,1h:now/h-1h) – количество за час два часа назад (11:00-12:00) => trendcount(/узел\_сети/ключ,1h:now/h-2h) – количество за час три часа назад (10:00-11:00) => trendcount(/узел\_сети/ключ,1M:now/M-1y) – количество за предыдущий месяц год назад |
| trendmax (/узел\_сети/ключ,период\_времени:Сдвиг\_времени) | | |
| Максимальное значение из динамики изменений за определенный период времени. | См. общие параметры. | Примеры: => trendmax(/узел\_сети/ключ,1h:now/h) – максимум за предыдущий час (например, 12:00-13:00) => trendmax(/узел\_сети/ключ,1h:now/h) – trendmin(/узел\_сети/ключ,1h:now/h) – расчет разницы между максимальным и минимальным значениями (дельта тренда) за предыдущий час (12:00-13:00) => trendmax(/узел\_сети/ключ,1h:now/h-1h) – максимум за час два часа назад (11:00-12:00) => trendmax(/узел\_сети/ключ,1h:now/h-2h) – максимум за час три часа назад (10:00-11:00) => trendmax(/узел\_сети/ключ,1M:now/M-1y) – максимум за предыдущий месяц год назад |
| trendmin (/узел\_сети/ключ,период\_времени:Сдвиг\_времени) | | |
| Минимальное значение из динамики изменений за определенный период времени. | См. общие параметры. | Примеры: => trendmin(/узел\_сети/ключ,1h:now/h) – минимум за предыдущий час (например, 12:00-13:00) => trendmin(/узел\_сети/ключ,1h:now/h) – trendmin(/узел\_сети/ключ,1h:now/h) – расчет разницы между максимальным и минимальным значениями (дельта тренда) за предыдущий час (12:00-13:00) => trendmin(/узел\_сети/ключ,1h:now/h-1h) – минимум за час два часа назад (11:00-12:00) => trendmin(/узел\_сети/ключ,1h:now/h-2h) – минимум за час три часа назад (10:00-11:00) => trendmin(/узел\_сети/ключ,1M:now/M-1y) – минимум за предыдущий месяц год назад |
| trendstl (/узел\_сети/ключ,период\_оценки:Сдвиг\_времени,период\_обнаружения,сезон,<отклонения>,<devalg>,<s\_window>) | | |
| Возвращает частоту аномалий за период обнаружения — десятичное значение от 0 до 1, которое определяется как ((количество аномальных значений)/(общее количество значений)). | период\_оценки – период времени, который необходимо разложить (минимум "1h"), определенный как <N><единица\_времени>, где N – количество единиц времени единица\_времени – h (час), d (сутки), w (неделя), M (месяц) или y (год). Сдвиг\_времени – смещение периода времени (см. примеры) период\_обнаружения – период времени до конца периода оценки, для которого вычисляются аномалии (минимум "1h", не может быть дольше периода оценки), определенный как <N><единица\_времени>, где N – количество единиц времени единица\_времени – h (час), d (сутки), w (неделя). сезон – самый короткий период времени, где ожидается повторяющийся шаблон ("сезон"), (минимум "2h", не может быть дольше периода оценки, количество записей в периоде оценки должно быть больше двукратной результирующей частоты (сезон/h)), определенной как <N><единица\_времени>, где N – количество единиц времени единица\_времени – h (час), d (сутки), w (неделя). отклонения – количество отклонений (вычисленных по алгоритму devalg), которые следует считать аномалией (может быть десятичным числом), (должно быть больше либо равно 1, по умолчанию: 3) devalg (должно быть в двойных кавычках) – алгоритм отклонений, может быть stddevpop, stddevsamp или mad (по умолчанию) s\_window – протяженность (в лагах) окна LOESS для извлечения сезонов (по умолчанию: 10 \* количество записей в периоде оценки + 1) | Примеры: => trendstl(/узел\_сети/ключ,100h:now/h,10h,2h) – анализировать последние 100 часов данных динамики изменений, найти частоту аномалий за последние 10 часов этого периода, ожидая, что будет периодичность 2 часа, значения компонентов остатка за период оценки рассматриваются как аномалии, если они достигают значения 3 отклонений по алгоритму MAD (среднее абсолютное отклонение) для этого компонента остатков => trendstl(/узел\_сети/ключ,100h:now/h-10h,100h,2h,2.1,"mad") – анализировать период 100 часов данных динамики изменений, до момента 10 часов назад, найти частоту аномалий за весь период при ожидании периодичности в 2 часа, значения компонентов остатка за период оценки рассматриваются как аномалии, если они достигают значения 2,1 отклонения по алгоритму MAD для этого компонента остатков => trendstl(/узел\_сети/ключ,100d:now/d-1d,10d,1d,4,,10) – анализировать период 100 дней данных динамики изменений до момента сутки назад, найти частоту аномалий за последние 10 дней этого периода, при ожидании, что периодичность составит 1 сутки, значения компонентов остатка за период оценки рассматриваются как аномалии, если они достигают значения 4 отклонений по алгоритму MAD для этого компонента остатков, переопределяя протяженность окна LOESS по умолчанию для извлечения сезонов "10 \* количество записей в периоде оценки + 1" на протяженность 10 лагов => trendstl(/узел\_сети/ключ,1M:now/M-1y,1d,2h,,"stddevsamp") – анализировать предыдущий месяц год назад, найти частоту аномалий за последний день этого периода, при ожидаемой периодичности в 2 часа, значения компонентов остатка за период оценки рассматриваются как аномалии, если они достигают значения 3 отклонений по алгоритму sample standard deviation (среднеквадратическое отклонение по выборке) для этого компонента остатков |
| trendsum (/узел\_сети/ключ,период\_времени:Сдвиг\_времени) | | |
| Сумма значений динамики изменений за определенный период времени. | См. общие параметры. | Примеры: => trendsum(/узел\_сети/ключ,1h:now/h) – сумма за предыдущий час (например, 12:00-13:00) => trendsum(/узел\_сети/ключ,1h:now/h-1h) – сумма за час два часа назад (11:00-12:00) => trendsum(/узел\_сети/ключ,1h:now/h-2h) – сумма за час три часа назад (10:00-11:00) => trendsum(/узел\_сети/ключ,1M:now/M-1y) – сумма за предыдущий месяц год назад |

1. Математические функции

Все перечисленные в таблице 98 функции поддерживаются в:

* выражениях триггера;
* вычисляемых элементах данных.

Математические функции поддерживаются с дробными и целочисленными типами значений, если не оговорено иначе.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* в качестве параметров допустимы выражения;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >".

Математические функции

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| abs (значение) | | |
| Абсолютное значение (по модулю) значения. | значение – проверяемое значение | Поддерживаются типы значений: float, int, str, text, log Для строк возвращается: 0 – значения равны 1 – значения отличаются Пример: => abs(last(/узел\_сети/ключ))>10 |
| acos (значение) | | |
| Арккосинус значения как угол, выраженный в радианах. | значение – проверяемое значение | Значение должно быть в диапазоне от -1 до 1. Например, арккосинус значения "0.5" будет "2.0943951". Пример: => acos(last(/узел\_сети/ключ)) |
| asin (значение) | | |
| Арксинус значения как угол, выраженный в радианах. | значение – проверяемое значение | Значение должно быть в диапазоне от -1 до 1. Например, арксинус значения "0.5" будет "-0.523598776". Пример: => asin(last(/узел\_сети/ключ)) |
| atan (значение) | | |
| Арктангенс значения как угол, выраженный в радианах. | значение – проверяемое значение | Например, арктангенс значения "1" будет "0.785398163". Пример: => atan(last(/узел\_сети/ключ)) |
| atan2 (значение,абсцисса) | | |
| Арктангенс указанных координат ординаты (значение) и абсцисса как угол, выраженный в радианах. | значение – проверяемое значение абсцисса – значение абсциссы | Например, арктангенс координат ордината и абсцисса "1" будет "2.21429744". Пример: => atan2(last(/узел\_сети/ключ),2) |
| avg (<значение1>,<значение2>,...) | | |
| Среднее значение перечисленных элементов данных. | значениеX – значение, возвращаемое одной из функций истории | Пример: => avg(avg(/узел\_сети/ключ),avg(/узел\_сети2/ключ2)) |
| cbrt (значение) | | |
| Кубический корень из значения. | значение – проверяемое значение | Например, кубический корень из "64" будет "4", а из "63" будет "3.97905721". Пример: => cbrt(last(/узел\_сети/ключ)) |
| ceil (значение) | | |
| Округлить значение вверх до ближайшего целого. | значение – проверяемое значение | Например, "2.4" будет округлено до "3". Пример: => ceil(last(/узел\_сети/ключ)) |
| cos (значение) | | |
| Косинус значения, где значение воспринимается как угол, выраженный в радианах. | значение – проверяемое значение | Например, косинус значения "1" будет "0.54030230586". Пример: => cos(last(/узел\_сети/ключ)) |
| cosh (значение) | | |
| Гиперболический косинус значения. | значение – проверяемое значение | Например, гиперболический косинус значения "1" будет "1.54308063482". Возвращает значение в виде действительного числа, а не в научном представлении. Пример: => cosh(last(/узел\_сети/ключ)) |
| cot (значение) | | |
| Котангенс значения, где значение воспринимается как угол, выраженный в радианах. | значение – проверяемое значение | Например, котангенс значения "1" будет "0.54030230586". Пример: => cot(last(/узел\_сети/ключ)) |
| degrees (значение) | | |
| Конвертация значения из радиан в градусы. | значение – проверяемое значение | Например, значение "1", сконвертированное в градусы, будет "57.2957795".  Пример: => degrees(last(/узел\_сети/ключ)) |
| e | | |
| Число Эйлера (2.718281828459045). | | Пример: => e() |
| exp (значение) | | |
| Число Эйлера (e) в степени значения. | значение – проверяемое значение | Например, число Эйлера, возведенное в степень "2", будет "7.38905609893065". Пример: => exp(last(/узел\_сети/ключ)) |
| expm1 (значение) | | |
| Число Эйлера (e) в степени значения, минус один. | значение – проверяемое значение | Например, число Эйлера, возведенное в степень "2", после вычитания единицы будет "6.38905609893065". Пример: => expm1(last(/узел\_сети/ключ)) |
| floor (значение) | | |
| Округление вниз до ближайшего целого. | значение – проверяемое значение | Например, "2.6" будет округлено вниз до "2". Пример: => floor(last(/узел\_сети/ключ)) |
| log (значение) | | |
| Натуральный логарифм. | значение – проверяемое значение | Например, натуральный логарифм значения "2" будет "0.69314718055994529". Пример: => log(last(/узел\_сети/ключ)) |
| log10 (значение) | | |
| Десятичный логарифм. | значение – проверяемое значение | Например, десятичный логарифм значения "5" будет равен "0.69897000433". Пример: => log10(last(/узел\_сети/ключ)) |
| max (<значение1>,<значение2>,...) | | |
| Наибольшее из значений, на которые ссылаются перечисленные элементы данных. | значениеX – значение, возвращаемое одной из функций истории | Пример: => max(avg(/узел\_сети/ключ),avg(//узел\_сети2/ключ2)) |
| min (<значение1>,<значение2>,...) | | |
| Наименьшее из значений, на которые ссылаются перечисленные элементы данных. | значениеX – значение, возвращаемое одной из функций истории | Пример: => min(avg(/узел\_сети/ключ),avg(/узел\_сети2/ключ2)) |
| mod (значение,делитель) | | |
| Остаток от деления. | значение – проверяемое значение делитель – знаменатель деления | Например, остаток от деления значения "5" с делителем "2" будет "1". Пример: => mod(last(/узел\_сети/ключ),2) |
| pi | | |
| Константа пи (3.14159265358979). | | Пример: => pi() |
| power (значение,степень) | | |
| Возведение в степень. | значение – проверяемое значение степень – используемый показатель степени | Например, третья степень значения "2" будет "8". Пример: => power(last(/узел\_сети/ключ),3) |
| radians (значение) | | |
| Конвертация значения из градусов в радианы. | значение – проверяемое значение | Например, значение "1", сконвертированное в радианы, будет "0.0174532925". Пример: => radians(last(/узел\_сети/ключ)) |
| rand | | |
| Вернуть случайное целое число. |  | Псевдослучайное число, сгенерированное с использованием времени в качестве seed (достаточно для математических целей, но не для криптографии). Пример: => rand() |
| round (значение,дес\_знаков) | | |
| Округлить значение до десятичных знаков. | значение – проверяемое значение дес\_знаков – количество десятичных знаков для округления (0 также допустим) | Например, значение "2.5482", округленное до 2 десятичных знаков, будет "2.55". Пример: => round(last(/узел\_сети/ключ),2) |
| signum (значение) | | |
| Возвращает "-1" для отрицательного значения, "0" для нуля, "1" для положительного значения. | значение – проверяемое значение | Пример: => signum(last(/узел\_сети/ключ)) |
| sin (значение) | | |
| Синус значения, воспринимаемого как угол, выраженный в радианах. | значение – проверяемое значение | Например, синус значения "1" будет "0.8414709848". Пример: => sin(last(/узел\_сети/ключ)) |
| sinh (значение) | | |
| Гиперболический синус значения. | значение – проверяемое значение | Например, гиперболический синус значения "1" будет "1.17520119364". Пример: => sinh(last(/узел\_сети/ключ)) |
| sqrt (значение) | | |
| Квадратный корень значения. | значение – проверяемое значение | Эта функция выдаст ошибку для отрицательного значения. Например, квадратный корень из значения "3.5" будет "1.87082869339". Пример: => sqrt(last(/узел\_сети/ключ)) |
| sum (<значение1>,<значение2>,...) | | |
| Сумма значений, на которые ссылаются перечисленные элементы данных. | значениеX – значение, возвращаемое одной из функций истории | Пример: => sum(avg(/узел\_сети/ключ),avg(/узел\_сети2/ключ2)) |
| tan (значение) | | |
| Тангенс значения. | значение – проверяемое значение | Например, тангенс значения "1" будет "1.55740772465". Пример: => tan(last(/узел\_сети/ключ)) |
| truncate (значение,дес\_знаков) | | |
| Усечение значения до указанного числа десятичных знаков. | значение – проверяемое значение дес\_знаков – количество десятичных знаков, оставляемых после усечения (0 также допустим) | Пример: => truncate(last(/узел\_сети/ключ),2) |

1. Операторные функции

Все перечисленные в таблице 99 функции поддерживаются в:

* триггерных выражениях;
* вычисляемых элементах данных.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* выражения принимаются в качестве параметров.

Операторные функции

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| between (значение,мин,макс) | | |
| Проверка того, попадает ли значение в заданный диапазон. | значение – проверяемое значение мин – минимальное значение макс – максимальное значение | Поддерживаемые типы значений: integer, float Возвращает: 1 – в диапазоне 0 – иначе Пример: => between(last(/узел\_сети/ключ),1,10)=1 – триггер сработает, если значение в пределах от 1 до 10. |
| in (значение,значение1,значение2,...значениеN) | | |
| Проверка того, что значение равно хотя бы какому-нибудь из перечисленных значений. | значение – проверяемое значение значение1,значение2,...значениеN – перечисленные значения (строковые значения должны быть в двойных кавычках) | Поддерживаемые типы значений: все Возвращает: 1 – если равно 0 – иначе Значение сравнивается с перечисленными значениями как число, если все эти значения могут быть преобразованы в число; иначе сравнивается как строка. Пример: => in(last(/узел\_сети/ключ),5,10)=1 – триггер сработает, если последнее значение равно либо 5, либо 10 => in("text", last(/узел\_сети/ключ),last(/узел\_сети/ключ,#2))=1 – триггер сработает, если любое из двух последних значений равно "text". |

1. Функции прогнозирования

Все перечисленные в таблице 100 функции поддерживаются в:

* выражениях триггера;
* вычисляемых элементах данных.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >";
* параметры, специфичные для функций, описываются с каждой функцией;
* параметры /узел\_сети/ключ и (сек|#число)<:Сдвиг\_времени> никогда не должны заключаться в кавычки.

Общие параметры:

* /узел\_сети/ключ – обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети
* (сек|#число)<:Сдвиг\_времени> – второй параметр для функций, ссылающихся на историю элемента данных узла сети,

где:

* сек – максимальный период вычисления в секундах (могут использоваться суффиксы времени), либо;
* #число – максимальная область вычисления среди последних собранных значений (если начинается со знака решетки);
* Сдвиг\_времени (опционально) – сдвиг точки вычислений по времени назад в прошлое.

Функции прогнозирования

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| forecast (/узел\_сети/ключ,(сек|#число)<:Сдвиг\_времени>,время,<аппроксимация>,<режим>) | | |
| Будущее значение, наибольшее, наименьшее, дельта или среднее элемента данных. | См. общие параметры.  время – горизонт предсказания в секундах (могут использоваться суффиксы времени); поддерживаются отрицательные значения  аппроксимация (опционально; должна быть в двойных кавычках) – функция, используемая для аппроксимации данных истории  Поддерживаются аппроксимации: linear – линейная функция polynomialN – полиномиальная степени N (1 <= N <= 6) exponential – экспоненциальная функция logarithmic – логарифмическая функция power – степенная функция Следует обратить внимание, что: по умолчанию – linear, а polynomial1 эквивалентно linear режим (опционально; должен быть в двойных кавычках) – требуемый вывод Поддерживаются режимы: value – значение (по умолчанию) max – максимум min – минимум delta – max-min avg – среднее Следует обратить внимание, что: value оценивает значение элемента данных на момент текущий\_момент + время max, min, delta и avg исследуют оценку значения элемента данных в интервале между текущий\_момент и текущий\_момент + время | Поддерживаются типы значений: float, int Если возвращаемое значение больше, чем 1.7976931348623157E+308, или меньше, чем -1.7976931348623157E+308, то возвращаемое значение усекается до 1.7976931348623157E+308 или -1.7976931348623157E+308 соответственно. Переходит в неподдерживаемое состояние только при некорректном использовании в выражении (ошибочный тип элемента данных, недопустимые параметры), иначе при ошибках возвращает -1. Примеры: => forecast(/узел\_сети/ключ,#10,1h) – предсказать значение элемента данных через час, основываясь на последних 10 значениях => forecast(/узел\_сети/ключ,1h,30m) – предсказать значение элемента данных через 30 минут на основании данных последнего часа => forecast(/узел\_сети/ключ,1h:now-1d,12h) – предсказать значение элемента данных через 12 часов на основании одного часа сутки назад => forecast(/узел\_сети/ключ,1h,10m,"exponential") – предсказать значение элемента данных через 10 минут, основываясь на данных последнего часа и экспоненциальной функции => forecast(/узел\_сети/ключ,1h,2h,"polynomial3","max") – предсказать максимальное значение, которое элемент данных может достичь через следующие два часа на основании данных последнего часа и кубическом (третьей степени) многочлене => forecast(/узел\_сети/ключ,#2,-20m) – ожидаемое значение элемента данных 20 минут назад на основании последних двух значений (это может оказаться более точным, чем использование last(), особенно если элемент данных обновляется редко – скажем, раз в час) |
| timeleft (/узел\_сети/ключ,(сек|#число)<:Сдвиг\_времени>,порог,<аппроксимация>) | | |
| Время в секундах, которое требуется для достижения элементом данных указанного порогового значения. | См. общие параметры.  порог – достигаемое значение (могут использоваться суффиксы единиц измерения)  аппроксимация (опционально; должна быть в двойных кавычках) – см. forecast() | Поддерживаются типы значений: float, int Если возвращаемое значение больше, чем 1.7976931348623157E+308, то возвращаемое значение усекается до 1.7976931348623157E+308. Возвращает 1.7976931348623157E+308, если пороговое значение недостижимо. Переходит в неподдерживаемое состояние только при некорректном использовании в выражении (ошибочный тип элемента данных, недопустимые параметры), иначе при ошибках возвращает -1. Примеры: => timeleft(/узел\_сети/ключ,#10,0) – время до тех пор, пока значение элемента данных не достигнет нуля, на основании последних 10 значений => timeleft(/узел\_сети/ключ,1h,100) – время до тех пор, пока значение элемента данных не достигнет 100, на основании данных последнего часа => timeleft(/узел\_сети/ключ,1h:now-1d,100) – время до тех пор, пока значение элемента данных не достигнет 100, на основании данных одного часа сутки назад => timeleft(/узел\_сети/ключ,1h,200,"polynomial2") – время до тех пор, пока значение элемента данных не достигнет 200, на основании данных последнего часа и предположении, что элемент данных ведет себя как квадратный (второй степени) двучлен |

1. Строковые функции

Все перечисленные в таблице 101 функции поддерживаются в:

* триггерных выражениях;
* вычисляемых элементах данных.

Некоторые общие замечания о параметрах функций:

* параметры функций разделяются запятыми;
* выражения принимаются в качестве параметров;
* строковые параметры должны заключаться в двойные кавычки; иначе они могут быть интерпретированы неверно;
* необязательные параметры функций (или части параметров) обозначаются угловыми скобками "< >".

Строковые функции

| Описание | Параметры, специфичные для функции | Комментарии |
| --- | --- | --- |
| ascii (значение) | | |
| Код ASCII самого левого символа значения. | значение – проверяемое значение | Поддерживаемые типы значений: string, text, log Например, значение "Abc" вернет "65" (код ASCII для буквы "A"). Пример: => ascii(last(/узел\_сети/ключ)) |
| bitlength (значение) | | |
| Длина значения в битах. | значение – проверяемое значение | Поддерживаемые типы значений: string, text, log, integer Пример: => bitlength(last(/узел\_сети/ключ)) |
| Bytelength (значение) | | |
| Длина значения в байтах. | значение – проверяемое значение | Поддерживаемые типы значений: string, text, log, integer Пример: => Bytelength(last(/узел\_сети/ключ)) |
| char (значение) | | |
| Вернуть символ, интерпретируя значение как код ASCII. | значение – проверяемое значение | Поддерживаемые типы значений: integer Значение должно быть в диапазоне 0-255. Например, значение "65" (интерпретируемое как код ASCII) вернет "A". Пример: => char(last(/узел\_сети/ключ)) |
| concat (<значение1>,<значение2>,...) | | |
| Строка, получаемая в результате конкатенации упоминаемых значений элементов данных либо констант. | значение – значение, возвращаемое одной из функций истории, либо константа (string, integer или float number) | Поддерживаемые типы значений: string, text, log, float, integer Например, значение вроде "Zab", объединенное с "bix" (строки-константы), вернет zabbix. Должно содержать минимум два параметра. Пример: => concat(last(/узел\_сети/ключ),"bix") => concat("1 min: ",last(/узел\_сети/system.cpu.load[all,avg1]),", 15 min: ",last(/узел\_сети/system.cpu.load[all,avg15])) |
| insert (значение,старт,длина,замена) | | |
| Вставить указанные символы или пробелы в символьную строку, начиная с указанной позиции в строке. | значение – проверяемое значение старт – начальная позиция длина – заменяемых позиций замена – подставляемая строка | Поддерживаемые типы значений: string, text, log Например, значение вроде "Rosssa" будет заменено на "Rosa", если "sss" (начиная с позиции 3, заменяемых позиций 3) заменяется строкой "s". Пример: => insert(last(/узел\_сети/ключ),3,2,"b") |
| left (значение,кол-во) | | |
| Самые левые символы значения. | значение – проверяемое значение кол-во – количество возвращаемых символов | Поддерживаемые типы значений: string, text, log Например, можно вернуть "Zab" из zabbix, указывая для возврата 3 крайних слева символа. Пример: => left(last(/узел\_сети/ключ),3) – возвращает три самых левых символа См. также right(). |
| length (значение) | | |
| Длина значения в символах. | значение – проверяемое значение | Поддерживаемые типы значений: string, text, log Пример: => length(last(/узел\_сети/ключ)) – длина последнего значения => length(last(/узел\_сети/ключ,#3)) – длина третьего из последних значения => length(last(/узел\_сети/ключ,#1:now-1d)) – длина самого свежего сутки назад значения |
| ltrim (значение,<символы>) | | |
| Удалить указанные символы с начала строки. | значение – проверяемое значение символы – (опционально) указать удаляемые символы. По умолчанию удаляются пробелы (если не указано опциональных символов). | Поддерживаемые типы значений: string, text, log Пример: => ltrim(last(/узел\_сети/ключ)) – удалить пробелы с начала строки => ltrim(last(/узел\_сети/ключ),"Z") – удалить любые символы "Z" с начала строки => ltrim(last(/узел\_сети/ключ)," Z") – удалить пробелы и символы "Z" с начала строки См. также: rtrim(), trim() |
| mid (значение,старт,длина) | | |
| Вернуть подстроку из N символов, начиная с символьной позиции, указанной параметром "старт". | значение – проверяемое значение старт – начальная позиция подстроки длина – количество позиций подстроки, которые нужно вернуть | Поддерживаемые типы значений: string, text, log Например, можно вернуть подстроку "abbi" из значения zabbix, если указать начальную позицию 2, а длину 4). Пример: => mid(last(/узел\_сети/ключ),2,4)="abbi" |
| repeat (значение,кол-во) | | |
| Повторить строку. | значение – проверяемое значение кол-во – количество повторений | Поддерживаемые типы значений: string, text, log Пример: => repeat(last(/узел\_сети/ключ),2) – повторить значение дважды |
| replace (значение,образец,замена) | | |
| Найти в значении образец и заменить его заменой. Все вхождения образца будут заменены. | значение – проверяемое значение образец – искомый образец замена – строка, которой заменяется образец | Поддерживаемые типы значений: string, text, log Пример: => replace(last(/узел\_сети/ключ),"ibb","abb") – заменить все "ibb" на "abb" |
| right (значение,кол-во) | | |
| Самые правые символы значения. | значение – проверяемое значение кол-во – количество возвращаемых символов | Поддерживаемые типы значений: string, text, log Например, можно вернуть "bix" из zabbix, если указать возвращать 3 символа справа. Пример: => right(last(/узел\_сети/ключ),3) – вернуть три символа справа См. также left(). |
| rtrim (значение,<символы>) | | |
| Удалить указанные символы с конца строки. | значение – проверяемое значение символы – (опционально) указать удаляемые символы. По умолчанию удаляются пробелы (если не указано опциональных символов). | Поддерживаемые типы значений: string, text, log Пример: => rtrim(last(/узел\_сети/ключ)) – удалить пробелы с конца строки => rtrim(last(/узел\_сети/ключ),"x") – удалить любые символы "x" с конца строки => rtrim(last(/узел\_сети/ключ),"x ") – удалить любые символы "x" или пробелы с конца строки См. также: ltrim(), trim() |
| trim (значение,<символы>) | | |
| Удалить указанные символы с начала и конца строки. | значение – проверяемое значение символы – (опционально) указать удаляемые символы. По умолчанию удаляются пробелы (если не указано опциональных символов). | Поддерживаемые типы значений: string, text, log Пример: => trim(last(/узел\_сети/ключ)) – удалить пробелы с начала и конца строки => trim(last(/узел\_сети/ключ),"\_") – удалить "\_" с начала и конца строки См. также: ltrim(), rtrim() |

Приложение И. Макросы

1. Поддерживаемые макросы

Таблица 102 содержит полный список макросов, поддерживаемых в Подсистеме по умолчанию.

Чтобы увидеть все макросы, поддерживаемые в каком-либо месте (например, в "URL карт сети"), можно вставить название места в поле поиска внизу окна браузера (доступное по нажатию CTRL+F) и искать следующее совпадение.

Поддерживаемые макросы

| Макрос | Поддерживается в | Описание |
| --- | --- | --- |
| {ACTION.ID} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения | Числовой идентификатор сработавшего действия. |
| {ACTION.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения | Имя сработавшего действия. |
| {ALERT.MESSAGE} | → Параметры скриптов оповещений | Значение "Сообщения по умолчанию" из настройки действия. |
| {ALERT.SENDTO} | → Параметры скриптов оповещений | Значение "Отправлять на" из настройки методов оповещений пользователя. |
| {ALERT.SUBJECT} | → Параметры скриптов оповещений | Значение "Темы по умолчанию" из настройки действия. |
| {DATE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Текущая дата в формате гггг.мм.дд. |
| {DISCOVERY.DEVICE.IPADDRESS} | → Оповещения и команды на события обнаружения | IP-адрес обнаруженного устройства. Доступно всегда, вне зависимости от того добавлен узел сети или нет. |
| {DISCOVERY.DEVICE.DNS} | → Оповещения и команды на события обнаружения | DNS-имя обнаруженного устройства. Доступно всегда, вне зависимости от того добавлен узел сети или нет. |
| {DISCOVERY.DEVICE.STATUS} | → Оповещения и команды на события обнаружения | Состояние обнаруженного устройства: может быть либо Доступен (UP), либо Недоступен (DOWN). |
| {DISCOVERY.DEVICE.UPTIME} | → Оповещения и команды на события обнаружения | Время с момента последнего изменения состояния обнаружения конкретного устройства, с точностью до секунды. Например: 1h 29m 01s. Для устройств с состоянием Недоступен, значением будет период недоступности. |
| {DISCOVERY.RULE.NAME} | → Оповещения и команды на события обнаружения | Имя правила обнаружения, обнаружившего наличие или отсутствие устройства или сервиса. |
| {DISCOVERY.SERVICE.NAME} | → Оповещения и команды на события обнаружения | Имя обнаруженного сервиса. Например: HTTP. |
| {DISCOVERY.SERVICE.PORT} | → Оповещения и команды на события обнаружения | Порт обнаруженного сервиса. Например: 80. |
| {DISCOVERY.SERVICE.STATUS} | → Оповещения и команды на события обнаружения | Состояние обнаруженного сервиса: может быть либо Доступен (UP), либо Недоступен (DOWN). |
| {DISCOVERY.SERVICE.UPTIME} | → Оповещения и команды на события обнаружения | Время с момента последнего изменения состояния обнаружения конкретного сервиса, с точностью до секунды. Например: 1h 29m 01s. Для сервисов с состоянием Недоступен, значением будет период недоступности. |
| {ESC.HISTORY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Внутренние оповещения | История эскалаций. Журнал ранее отправленных сообщений. Показывает ранее отправленные оповещения, на каком шаге эскалаций они были отправлены и их состояние (отправлено, в процессе или ошибочно). |
| {EVENT.ACK.STATUS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Состояние подтверждения события (Yes/No). |
| {EVENT.AGE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Возраст события, вызвавшего действие, с точностью до секунды. Полезно в эскалированных сообщениях. |
| {EVENT.DATE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Дата события, вызвавшего действие. |
| {EVENT.DURATION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Внутренние оповещения → Скрипты действий вручную над событиями | Продолжительность события (разница по времени между событиями проблемы и восстановления), с точностью до секунды. Полезно в сообщениях о восстановлении проблем. |
| {EVENT.ID} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Триггерные URL → Скрипты действий вручную над событиями | Числовой идентификатор события, вызвавшего действие. |
| {EVENT.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Внутренние оповещения → Скрипты действий вручную над событиями | Имя события проблемы, вызвавшего действие. |
| {EVENT.NSEVERITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями | Числовое значение важности события. Возможные значения: 0 – Не классифицировано, 1 – Информационный, 2 – Предупреждение, 3 – Средний, 4 – Высокий, 5 – Чрезвычайный. |
| {EVENT.OBJECT} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Числовое значение объекта события. Возможные значения: 0 – Триггер, 1 – Обнаруженный узел сети, 2 – Обнаруженный сервис, 3 – Авторегистрация, 4 – Элемент данных, 5 – Правило низкоуровневого обнаружения. |
| {EVENT.OPDATA} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Оперативные данные нижележащего триггера проблемы. |
| {EVENT.RECOVERY.DATE} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Дата события восстановления. |
| {EVENT.RECOVERY.ID} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Числовой идентификатор события восстановления. |
| {EVENT.RECOVERY.NAME} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Имя события восстановления. |
| {EVENT.RECOVERY.STATUS} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Словесное значение события восстановления. |
| {EVENT.RECOVERY.TAGS} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Список тегов события восстановления (через запятую). Раскрывается в пустую строку, если теги отсутствуют. |
| {EVENT.RECOVERY.TAGSJSON} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Массив JSON, содержащий объекты тегов события восстановления. Раскрывается в пустой массив, если теги отсутствуют. |
| {EVENT.RECOVERY.TIME} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Время события восстановления. |
| {EVENT.RECOVERY.VALUE} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем (если восстановление произошло) → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями (если восстановление произошло) | Числовое значение события восстановления. |
| {EVENT.SEVERITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями | Название важности события. |
| {EVENT.SOURCE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Числовое значение источника события. Возможные значения: 0 – Триггер, 1 – Обнаружение, 2 – Авторегистрация, 3 – Внутреннее событие, 4 – Услуга. |
| {EVENT.STATUS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Внутренние оповещения → Скрипты действий вручную над событиями | Словесное значение события, вызвавшего действие. |
| {EVENT.TAGS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями | Список тегов события (через запятую). Раскрывается в пустую строку, если теги отсутствуют. |
| {EVENT.TAGSJSON} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Скрипты действий вручную над событиями | Массив JSON, содержащий объекты тегов события). Раскрывается в пустой массив, если теги отсутствуют. |
| {EVENT.TAGS.<имя\_тега>} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения вебхуков типа URL и имен URL → Скрипты действий вручную над событиями | Значение тега события, относящееся к данному имени тега. Имя тега, содержащее не алфавитно-цифровые символы (в том числе неанглийские многобайтовые символы UTF) должно заключаться в кавычки. Кавычки и обратные слеши внутри заключенного в кавычки имени тега должны экранироваться обратным слешем. |
| {EVENT.TIME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Время события, вызвавшего действие. |
| {EVENT.UPDATE.ACTION} | → Оповещения и команды на обновление проблем | Удобочитаемое имя действия (действий), выполняемых при обновлении проблемы.  Раскрывается в следующие значения: acknowledged (подтверждено), commented (прокомментировано), changed severity from (original severity) to (updated severity) (важность изменена с (оригинальная важность) на (обновленная важность)) и closed (закрыто) (в зависимости от того, сколько действий выполняется за одно обновление). |
| {EVENT.UPDATE.DATE} | → Оповещения и команды на обновление проблем → Оповещения и команды на обновление услуг | Дата события обновления (подтверждение и т.п.). |
| {EVENT.UPDATE.HISTORY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Журнал обновлений проблемы (подтверждения и т.п.). |
| {EVENT.UPDATE.MESSAGE} | → Оповещения и команды на обновление проблем | Сообщение обновления проблемы. |
| {EVENT.UPDATE.NSEVERITY} | → Оповещения и команды на обновление услуг | Числовое значение новой важности события, заданное во время операции обновления проблемы. |
| {EVENT.UPDATE.SEVERITY} | → Оповещения и команды на обновление услуг | Название новой важности события, заданное во время операции обновления проблемы. |
| {EVENT.UPDATE.STATUS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Числовое значение статуса обновления проблемы. Возможные значения: 0 – Вебхук был вызван из-за события проблемы/восстановления, 1 – Операция обновления. |
| {EVENT.UPDATE.TIME} | → Оповещения и команды на обновление проблем → Оповещения и команды на обновление услуг | Время события обновления (подтверждения и т.п.). |
| {EVENT.VALUE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события восстановления услуг → Внутренние оповещения → Скрипты действий вручную над событиями | Числовое значение события, вызвавшего действие (1 для проблемы, 0 для восстановления). |
| {FUNCTION.VALUE<1-9>} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями → Имена событий | Результаты N-ой функции, относящейся к элементам данных, в триггерном выражении на момент события. Учитываются только функции с /узел\_сети/ключ в качестве первого параметра. См. нумерованные макросы. |
| {FUNCTION.RECOVERY.VALUE<1-9>} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Результаты N-ой функции, относящейся к элементам данных, в выражении восстановления на момент события. Учитываются только функции с /узел\_сети/ключ в качестве первого параметра. См. нумерованные макросы. |
| {HOST.CONN} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Подписи к элементам карты, имена и значения URL на картах сети → Параметры ключа элемента данных1 → IP/DNS интерфейса узла сети → Поле "Разрешенные хосты" элементов данных "траппер" → Дополнительные параметры мониторов баз данных → Скрипты SSH и Telnet → Поле JMX endpoint элементов данных → Веб-мониторинг4 → Регулярное выражение фильтра правила низкоуровневого обнаружения → Поле URL виджета панели "динамический URL" → Триггеры: имена, имена событий, оперативные данные и описания → Триггерные URL → Имена и значения тегов → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, Поля запроса, Тело запроса, Заголовки, Файл SSL сертификата, Файл SSL ключа, Разрешенные хосты. → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) → Описание виджета значения элемента данных | IP-адрес или DNS-имя узла сети, в зависимости от его настроек2.  Может использоваться с числовым индексом как {HOST.CONN<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.DESCRIPTION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Подписи к элементам карты → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Описание узла сети.  Этот макрос может использоваться с числовым индексом, т.е. {HOST.DESCRIPTION<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.DNS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Подписи к элементам карты, имена и значения URL на картах сети → Параметры ключа элемента данных1 → IP/DNS интерфейса узла сети → Поле "Разрешенные хосты" элементов данных "траппер" → Дополнительные параметры мониторов баз данных → Скрипты SSH и Telnet → Поле JMX endpoint элементов данных → Веб-мониторинг4 → Регулярное выражение фильтра правила низкоуровневого обнаружения → Поле URL виджета панели "динамический URL" → Триггеры: имена, имена событий, оперативные данные и описания → Триггерные URL → Имена и значения тегов → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, Поля запроса, Тело запроса, Заголовки, Файл SSL сертификата, Файл SSL ключа, Разрешенные хосты. → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) → Описание виджета значения элемента данных | DNS-имя узла сети2.  Этот макрос может использоваться с числовым индексом, т.е. {HOST.DNS<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.HOST} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события авторегистрации → Внутренние оповещения → Параметры ключа элемента данных → Подписи к элементам карты, имена и значения URL на картах сети → IP/DNS интерфейса узла сети → Поле "Разрешенные хосты" элементов данных "траппер" → Дополнительные параметры мониторов баз данных → Скрипты SSH и Telnet → Поле JMX endpoint элементов данных → Веб-мониторинг4 → Регулярное выражение фильтра правила низкоуровневого обнаружения → Поле URL виджета панели "динамический URL" → Триггеры: имена, имена событий, оперативные данные и описания → Триггерные URL → Имена и значения тегов → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, Поля запроса, Тело запроса, Заголовки, Файл SSL сертификата, Файл SSL ключа, Разрешенные хосты. → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) → Описание виджета значения элемента данных | Имя узла сети.  Этот макрос может использоваться с числовым индексом, т.е. {HOST.HOST<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.ID} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Подписи к элементам карты, имена и значения URL на картах сети → Поле URL виджета панели "динамический URL" → Триггерные URL → Имена и значения тегов → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Идентификатор узла сети.  Может использоваться с числовым индексом, т.е. {HOST.ID<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.IP} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события авторегистрации → Внутренние оповещения → Подписи к элементам карты, имена и значения URL на картах сети → Параметры ключа элемента данных1 → IP/DNS интерфейса узла сети → Поле "Разрешенные хосты" элементов данных "траппер" → Дополнительные параметры мониторов баз данных → Скрипты SSH и Telnet → Поле JMX endpoint элементов данных → Веб-мониторинг4 → Регулярное выражение фильтра правила низкоуровневого обнаружения → Поле URL виджета панели "динамический URL" → Триггеры: имена, имена событий, оперативные данные и описания → Триггерные URL → Имена и значения тегов → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, Поля запроса, Тело запроса, Заголовки, Файл SSL сертификата, Файл SSL ключа, Разрешенные хосты. → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) → Описание виджета значения элемента данных | IP-адрес узла сети2.  Этот макрос может использоваться с числовым индексом, т.е. {HOST.IP<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.METADATA} | → Оповещения и команды на события авторегистрации | Метаданные узла сети. Используется только для авторегистрации активного Агента. |
| {HOST.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Подписи к элементам карты, имена и значения URL на картах сети → Параметры ключа элемента данных → IP/DNS интерфейса узла сети → Поле "Разрешенные хосты" элементов данных "траппер" → Дополнительные параметры мониторов баз данных → Скрипты SSH и Telnet → Веб-мониторинг4 → Регулярное выражение фильтра правила низкоуровневого обнаружения → Поле URL виджета панели "динамический URL" → Триггеры: имена, имена событий, оперативные данные и описания → Триггерные URL → Имена и значения тегов → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, Поля запроса, Тело запроса, Заголовки, Файл SSL сертификата, Файл SSL ключа, Разрешенные хосты. → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) → Описание виджета значения элемента данных | Видимое имя узла сети.  Этот макрос может использоваться с числовым индексом, т.е. {HOST.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.PORT} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события авторегистрации → Внутренние оповещения → Триггеры: имена, имена событий, оперативные данные и описания → Триггерные URL → Поле JMX endpoint элементов данных → Имена и значения тегов → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Порт узла сети (Агента)2.  Этот макрос может использоваться с числовым индексом, т.е. {HOST.PORT<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {HOST.TARGET.CONN} | → Команды на события триггеров → Команды на обновление проблем → Команды на события обнаружения → Команды на события авторегистрации | IP-адрес или DNS-имя целевого узла сети в зависимости от его настроек. |
| {HOST.TARGET.DNS} | → Команды на события триггеров → Команды на обновление проблем → Команды на события обнаружения → Команды на события авторегистрации | DNS-имя целевого узла сети. |
| {HOST.TARGET.HOST} | → Команды на события триггеров → Команды на обновление проблем → Команды на события обнаружения → Команды на события авторегистрации | Техническое имя целевого узла сети. |
| {HOST.TARGET.IP} | → Команды на события триггеров → Команды на обновление проблем → Команды на события обнаружения → Команды на события авторегистрации | IP-адрес целевого узла сети |
| {HOST.TARGET.NAME} | → Команды на события триггеров → Команды на обновление проблем → Команды на события обнаружения → Команды на события авторегистрации | Видимое имя целевого узла сети |
| {HOSTGROUP.ID} | → Подписи к элементам карты, имена и значения URL на картах сети | Идентификатор группы узлов сети. |
| {INVENTORY.ALIAS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Псевдоним" (Alias) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.ALIAS<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.ASSET.TAG} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Этикетка владельца" (Asset tag) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.ASSET.TAG<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.CHASSIS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Шасси" (Chassis) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.CHASSIS<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.CONTACT} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Контактная информация" (Contact) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.CONTACT<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. {PROFILE.CONTACT<1-9>} является устаревшим. |
| {INVENTORY.CONTRACT.NUMBER} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Номер контракта" (Contract number) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.CONTRACT.NUMBER<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.DEPLOYMENT.STATUS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Состояние развертывания" (Deployment status) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.DEPLOYMENT.STATUS<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.HARDWARE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Аппаратные средства" (Hardware) в инвентарных данных узла сети.  Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HARDWARE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. |
| {INVENTORY.HARDWARE.FULL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Аппаратные средства (полная детализация)" (Hardware (Full details)) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HARDWARE.FULL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. |
| {INVENTORY.HOST.NETMASK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Маска подсети узла сети" (Host subnet mask) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HOST.NETMASK<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. |
| {INVENTORY.HOST.NETWORKS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Сети узла сети" (Host networks) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HOST.NETWORKS<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. |
| {INVENTORY.HOST.ROUTER} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Роутер узла сети" (Host router) в инвентарных данных узла сети.  Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HOST.ROUTER<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. |
| {INVENTORY.HW.ARCH} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Архитектура HW" (HW architecture) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HW.ARCH<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.HW.DATE.DECOMM} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Дата списания HW" (Date HW decommissioned) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HW.DATE.DECOMM<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.HW.DATE.EXPIRY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Дата окончания обслуживания HW" (Date HW maintenance expires) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HW.DATE.EXPIRY<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.HW.DATE.INSTALL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Дата установки HW" (Date HW installed) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HW.DATE.INSTALL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.HW.DATE.PURCHASE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Дата покупки HW" (Date HW purchased) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.HW.DATE.PURCHASE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.INSTALLER.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Имя установщика" (Installer name ) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.INSTALLER.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.LOCATION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Местоположение" (Location) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.LOCATION<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.LOCATION.LAT} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Размещение (широта)" (Location latitude) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.LOCATION.LAT<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.LOCATION.LON} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Размещение (долгота)" (Location longitude) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.LOCATION.LON<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.MACADDRESS.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "MAC адрес A" (MAC address A) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.MACADDRESS.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.MACADDRESS.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "MAC адрес B" (MAC address B) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.MACADDRESS.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.MODEL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Модель" (Model) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.MODEL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Имя" (Name) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.NOTES} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Примечания" (Notes) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.NOTES<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.OOB.IP} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "OOB IP-адрес" (OOB IP address) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.OOB.IP<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.OOB.NETMASK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "OOB маска подсети" (OOB subnet mask) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.OOB.NETMASK<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.OOB.ROUTER} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "OOB роутер" (OOB router) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.OOB.ROUTER<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.OS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "ОС" (OS) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.OS<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.OS.FULL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "ОС (полная детализация)" (OS (Full details)) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.OS.FULL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.OS.SHORT} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "ОС (короткое описание)" (OS (Short)) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.OS.SHORT<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.CELL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичный мобильный для контакта" (Primary POC cell) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.CELL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.EMAIL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичный email для контакта" (Primary POC email) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.EMAIL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичное имя для контакта" (Primary POC name) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.NOTES} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичные примечания для контакта" (Primary POC notes) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.NOTES<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.PHONE.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичный телефон A для контакта" (Primary POC phone A) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.PHONE.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.PHONE.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичный телефон B для контакта" (Primary POC phone B) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.PHONE.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.PRIMARY.SCREEN} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Первичное ник-имя для контакта" (Primary POC screen name) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.PRIMARY.SCREEN<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.CELL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичный мобильный для контакта" (Secondary POC cell) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.CELL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.EMAIL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичный email для контакта" (Secondary POC email) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.EMAIL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичное имя для контакта" (Secondary POC name) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.NOTES} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичные примечания для контакта" (Secondary POC notes) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.NOTES<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.PHONE.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичный телефон A для контакта" (Secondary POC phone A ) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.PHONE.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.PHONE.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичный телефон B для контакта" (Secondary POC phone B ) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.PHONE.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.POC.SECONDARY.SCREEN} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Вторичное ник-имя для контакта" (Secondary POC screen name ) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.POC.SECONDARY.SCREEN<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SERIALNO.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Серийный номер A" (Serial number A) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SERIALNO.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SERIALNO.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Серийный номер B" (Serial number B) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SERIALNO.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.ADDRESS.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Адрес A" (Site address A) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.ADDRESS.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.ADDRESS.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Адрес B" (Site address B) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.ADDRESS.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.ADDRESS.C} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Адрес C" (Site address C) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.ADDRESS.C<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.CITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Город" (Site city) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.CITY<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.COUNTRY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Страна" (Site country) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.COUNTRY<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.NOTES} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Заметки" (Site notes) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.NOTES<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.RACK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Размещение стойки" (Site rack location) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.RACK<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.STATE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Область / район" (Site state/province) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.STATE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SITE.ZIP} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Почтовый индекс" (Site ZIP/postal) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SITE.ZIP<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение" (Software) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE.APP.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение A" (Software application A) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE.APP.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE.APP.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение B" (Software application B) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE.APP.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE.APP.C} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение C" (Software application C) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE.APP.C<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE.APP.D} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение D" (Software application D) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE.APP.D<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE.APP.E} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение E" (Software application E) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE.APP.E<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.SOFTWARE.FULL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Программное обеспечение (полная детализация)" (Software (Full details)) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.SOFTWARE.FULL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.TAG} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Этикетка" (Tag) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.TAG<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.TYPE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Тип" (Type) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.TYPE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.TYPE.FULL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Тип (полная детализация)" (Type (Full details)) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.TYPE.FULL<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.URL.A} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "URL A" в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.URL.A<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.URL.B} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "URL B" в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.URL.B<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.URL.C} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "URL C" в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.URL.C<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {INVENTORY.VENDOR} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Имена и значения тегов → Подписи к элементам карты, имена и значения URL на картах сети → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Поле "Поставщик" (Vendor) в инвентарных данных узла сети. Этот макрос может использоваться с числовым индексом, т.е. {INVENTORY.VENDOR<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.DESCRIPTION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Описание N-го элемента данных в выражении триггера, являющегося причиной оповещения. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.DESCRIPTION<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.DESCRIPTION.ORIG} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Описание (с нераскрытыми макросами) N-го элемента данных в выражении триггера, являющегося причиной оповещения. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.DESCRIPTION.ORIG<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.ID} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров6 → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, поля запроса, тело запроса, заголовки, Прокси, файл SSL сертификата, файл SSL ключа → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Числовой идентификатор N-го элемента данных в выражении триггера, являющегося причиной оповещения. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.ID<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.KEY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров6 → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, поля запроса, тело запроса, заголовки, Прокси, файл SSL сертификата, файл SSL ключа → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Ключ N-го элемента данных в выражении триггера, являющегося причиной оповещения. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.KEY<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.KEY.ORIG} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скриптовые элемент данных, прототип элемента данных и правило низкоуровневого обнаружения: имена и значения параметров6 → Поля элементов данных, прототипов элементов данных и правил низкоуровневого обнаружения типа HTTP Агент: URL, Query fields, Request body, Headers, Proxy, SSL certificate file, SSL key file, Allowed hosts.6 → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Оригинальный ключ (с нераскрытыми макросами) N-го элемента данных в выражении триггера, являющегося причиной оповещения 4. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.KEY.ORIG<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LASTVALUE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, имена событий, оперативные данные и описания → Имена и значения тегов → Триггерные URL → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Последнее значение N-го элемента данных в выражении триггера, являющегося причиной оповещения. Макрос будет раскрываться в \*UNKNOWN\* в веб-интерфейсе, если последнее значение в истории было получено более чем Максимальный период отображения истории времени назад (заданное в разделе меню "Администрирование→ Общие"). Раскрытое значение для текстовых/журнальных элементов данных усекается веб-интерфейсом до 20 символов в следующих местах: - Оперативные данные; - Описание триггера; - Триггерные URL; - Описание виджета "значение элемента данных". Чтобы раскрыть в полное значение, можно использовать функции макросов. Сервером никаких усечений не производится. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LASTVALUE<1-9>} для ссылок на первый, второй, третий и т.д. элемент данных в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.AGE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Возраст события журнального элемента данных, с точностью до секунды. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.AGE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.DATE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Дата события журнального элемента данных. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.DATE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.EVENTID} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Идентификатор события журнального элемента данных. Только для мониторинга журнала событий Windows. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.EVENTID<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.NSEVERITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Числовое значение важности события журнального элемента данных. Только для мониторинга журнала событий Windows. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.NSEVERITY<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.SEVERITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Словесное значение важности события журнального элемента данных. Только для мониторинга журнала событий Windows. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.SEVERITY<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.SOURCE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Источник события журнального элемента данных. Только для мониторинга журнала событий Windows. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.SOURCE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.LOG.TIME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, оперативные данные и описания → Триггерные URL → Имена и значения тегов событий → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Время события журнального элемента данных. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.LOG.TIME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Имя N-го элемента данных в выражении триггера, являющегося причиной оповещения. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.NAME.ORIG} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Этот макрос может использоваться с числовым индексом, т.е. {ITEM.NAME.ORIG<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.STATE} | → Внутренние оповещения на события элементов данных → Описание виджета значения элемента данных | Последнее состояние N-го элемента данных в выражении триггера, являющегося причиной оповещения. Возможные значения: "Not supported" и Normal. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.STATE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {ITEM.STATE.ERROR} | → Внутренние оповещения на события элементов данных | Сообщение об ошибке с подробностями того, почему элемент данных стал неподдерживаемым. Если элемент данных переходит в неподдерживаемое состояние, а затем сразу же снова становится поддерживаемым, поле ошибки может быть пустым. |
| {ITEM.VALUE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Триггеры: имена, имена событий, оперативные данные и описания → Имена и значения тегов → Триггерные URL → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Раскрывается в одно из: 1) значение из истории (на момент события) N-го элемента данных из выражения триггера, если используется в контексте изменения состояния триггера; например, при отображении событий или при отсылке оповещений. 2) последнее значение N-го элемента данных из выражения триггера, если используется вне контекста изменения состояния триггера; например, при отображении списка триггеров во всплывающем окне выбора. В этом случае работает так же, как и {ITEM.LASTVALUE}. В первом случае макрос будет раскрываться в \*UNKNOWN\*, если значение истории уже удалено или никогда не хранилось. Во втором случае, и только в веб-интерфейсе, макрос будет раскрыт в \*UNKNOWN\*, если последнее значение истории собрано более чем Максимальный период отображения истории времени назад (задано в разделе меню "Администрирование→ Общие"). Раскрытое значение для текстовых/журнальных элементов данных усекается веб-интерфейсом до 20 символов в следующих местах: - Оперативные данные; - Описание триггера; - Триггерные URL; - Описание виджета "значение элемента данных". Чтобы раскрыть в полное значение, можно использовать функции макросов. Сервером никаких усечений не производится. Модификация значения макроса поддерживается для этого макроса. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.VALUE<1-9>} для ссылок на первый, второй, третий и т.д. элемент данных в триггерном выражении. См. нумерованные макросы. |
| {ITEM.VALUETYPE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения → Скрипты действий вручную над событиями → Описание виджета значения элемента данных | Тип значения N-го элемента данных в выражении триггера, являющегося причиной оповещения. Возможные значения: 0 – числовой с плавающей точкой, 1 – символ, 2 – журнал (лог), 3 – числовой целое неотрицательное, 4 – текст. Этот макрос может использоваться с числовым индексом, т.е. {ITEM.VALUETYPE<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {LLDRULE.DESCRIPTION} | → Внутренние оповещения на события низкоуровневого обнаружения | Описание правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.DESCRIPTION.ORIG} | → Внутренние оповещения на события низкоуровневого обнаружения | Описание (с нераскрытыми макросами) правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.ID} | → Внутренние оповещения на события низкоуровневого обнаружения | Числовой идентификатор правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.KEY} | → Внутренние оповещения на события низкоуровневого обнаружения | Ключ правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.KEY.ORIG} | → Внутренние оповещения на события низкоуровневого обнаружения | Оригинальный ключ (с нераскрытыми макросами) правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.NAME} | → Внутренние оповещения на события низкоуровневого обнаружения | Имя (с раскрытыми макросами) правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.NAME.ORIG} | → Внутренние оповещения на события низкоуровневого обнаружения | Оригинальное имя (с нераскрытыми макросами) правила низкоуровневого обнаружения, которое является причиной оповещения. |
| {LLDRULE.STATE} | → Внутренние оповещения на события низкоуровневого обнаружения | Последнее состояние правила низкоуровневого обнаружения. Возможные значения: "Not supported" и Normal. |
| {LLDRULE.STATE.ERROR} | → Внутренние оповещения на события низкоуровневого обнаружения | Сообщение об ошибке с подробностями того, почему правило низкоуровневого обнаружения стало неподдерживаемым. Если правило низкоуровневого обнаружения переходит в неподдерживаемое состояние и затем сразу же становится снова поддерживаемым, то поле ошибки может быть пустым. |
| {MAP.ID} | → Подписи к элементам карты, имена и значения URL на картах сети | Идентификатор карты сети. |
| {MAP.NAME} | → Подписи к элементам карты, имена и значения URL на картах сети → Текстовые поля в фигурах карт сети | Имя карты сети. |
| {PROXY.DESCRIPTION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Описание Прокси. Раскрывается в одно из: 1) Прокси N-го элемента данных из выражения триггера (в оповещениях на триггерные события). Здесь можно использовать нумерованные макросы. 2) Прокси, выполнявший обнаружение (в оповещениях на события обнаружения). Здесь использовать {PROXY.DESCRIPTION}, "без нумерации". 3) Прокси, на котором регистрировался активный Агент (в оповещениях на события авторегистрации). Здесь использовать {PROXY.DESCRIPTION}, "без нумерации".  Этот макрос может использоваться с числовым индексом, т.е. {PROXY.DESCRIPTION<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {PROXY.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Скрипты действий вручную над событиями | Имя Прокси. Раскрывается в одно из: 1) Прокси N-го элемента данных из выражения триггера (в оповещениях на триггерные события). Здесь можно использовать нумерованные макросы. 2) Прокси, выполнявший обнаружение (в оповещениях на события обнаружения). Здесь использовать {PROXY.NAME}, "без нумерации". 3) Прокси, на котором регистрировался активный Агент (в оповещениях на события авторегистрации). Здесь использовать {PROXY.DESCRIPTION}, "без нумерации". Этот макрос может использоваться с числовым индексом, т.е. {PROXY.NAME<1-9>} для ссылок на первый, второй, третий и т.д. узел сети в триггерном выражении. См. нумерованные макросы. |
| {SERVICE.DESCRIPTION} | → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг | Описание услуги (с раскрытыми макросами). |
| {SERVICE.NAME} | → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг | Имя услуги (с раскрытыми макросами). |
| {SERVICE.ROOTCAUSE} | → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг | Список событий триггерных проблем, являющихся причиной недоступности услуги, отсортированный по важности и имени узла сети. Включает следующие подробности: имя узла сети, имя события, важность, возраст, имена и значения тегов услуг. |
| {SERVICE.TAGS} | → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг | Список тегов событий на услуги (через запятую). Теги событий на услуги могут быть заданы в разделе "Теги" настроек услуг. Раскрывается в пустую строку, если теги отсутствуют. |
| {SERVICE.TAGSJSON} | → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг | Массив JSON, содержащий объекты тегов событий на услуги. Теги событий на услуги могут быть заданы в разделе "Теги" настроек услуг. Раскрывается в пустой массив, если теги отсутствуют. |
| {SERVICE.TAGS.<имя\_тега>} | → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг | Значение тега события на услуги, относящееся к данному имени тега. Теги событий на услуги могут быть заданы в разделе "Теги" настроек услуг. Имя тега, содержащее не алфавитно-цифровые символы (в том числе неанглийские многобайтовые символы UTF) должно заключаться в кавычки. Кавычки и обратные слеши внутри заключенного в кавычки имени тега должны экранироваться обратным слешем. |
| {TIME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Оповещения и команды на события услуг → Оповещения и команды на обновление услуг → Оповещения и команды на события обнаружения → Оповещения и команды на события авторегистрации → Внутренние оповещения → Имена событий триггеров → Скрипты действий вручную над событиями | Текущее время в формате чч:мм:сс. |
| {TRIGGER.DESCRIPTION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Описание триггера. Если {TRIGGER.DESCRIPTION} используется в тексте оповещения, то все макросы, поддерживаемые в описании триггера, будут раскрыты. {TRIGGER.COMMENT} является устаревшим. |
| {TRIGGER.EXPRESSION.EXPLAIN} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями → Имена событий | Частично вычисленное выражение триггера. Функции на основе элементов данных вычисляются и заменяются результатами на момент генерации события, в то время как все остальные функции отображаются в том виде, как написаны в выражении. Можно использовать для отладки триггерных выражений. |
| {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN} | → Оповещения и команды на события восстановления проблем → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Частично вычисленное выражение восстановления триггера. Функции на основе элементов данных вычисляются и заменяются результатами на момент генерации события, в то время как все остальные функции отображаются в том виде, как написаны в выражении. Можно использовать для отладки выражений восстановления триггеров. |
| {TRIGGER.EVENTS.ACK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Подписи к элементам карты → Скрипты действий вручную над событиями | Количество подтвержденных событий у элемента карты на картах сети, или у триггера, который сгенерировал текущее событие, в оповещениях. |
| {TRIGGER.EVENTS.PROBLEM.ACK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Подписи к элементам карты → Скрипты действий вручную над событиями | Количество подтвержденных событий ПРОБЛЕМА у всех триггеров, независимо от их состояния. |
| {TRIGGER.EVENTS.PROBLEM.UNACK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Подписи к элементам карты → Скрипты действий вручную над событиями | Количество неподтвержденных событий ПРОБЛЕМА у всех триггеров, независимо от их состояния. |
| {TRIGGER.EVENTS.UNACK} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Подписи к элементам карты → Скрипты действий вручную над событиями | Количество неподтвержденных событий у элемента карты на картах сети, или у триггера, который сгенерировал текущее событие, в оповещениях. |
| {TRIGGER.HOSTGROUP.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Отсортированный (запросом SQL) список групп узлов сети, в которых определен триггер (через запятую). |
| {TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK} | → Подписи к элементам карты | Количество подтвержденных событий ПРОБЛЕМА у триггеров в состоянии ПРОБЛЕМА. |
| {TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK} | → Подписи к элементам карты | Количество неподтвержденных событий ПРОБЛЕМА у триггеров в состоянии ПРОБЛЕМА. |
| {TRIGGER.EXPRESSION} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Выражение триггера. |
| {TRIGGER.EXPRESSION.RECOVERY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Выражение восстановления триггера, если параметр "Генерация ОК- событий" в настройках триггера задан в "Выражение восстановления"; иначе возвращается пустая строка. |
| {TRIGGER.ID} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Подписи к элементам карты, имена и значения URL на картах сети → Триггерные URL → Trigger tag value → Скрипты действий вручную над событиями | Числовой идентификатор триггера, который привел к срабатыванию этого действия. |
| {TRIGGER.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Имя триггера (с раскрытыми макросами). |
| {TRIGGER.NAME.ORIG} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Оригинальное имя триггера (т.е. без раскрытых макросов). |
| {TRIGGER.NSEVERITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Числовое значение важности триггера. Возможные значения: 0 – Не классифицировано, 1 – Информация, 2 – Предупреждение, 3 – Средняя, 4 – Высокая, 5 – Чрезвычайная. |
| {TRIGGER.SEVERITY} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Имя важности триггера. Может быть задано в разделе "Администрирование → Общие → Опции отображения триггеров". |
| {TRIGGER.STATE} | → Внутренние оповещения на события триггеров | Последнее состояние триггера. Возможные значения: Unknown и Normal. |
| {TRIGGER.STATE.ERROR} | → Внутренние оповещения на события триггеров | Сообщение об ошибке с подробностями того, почему триггер стал неподдерживаемым. Если триггер переходит в неподдерживаемое состояние, а затем сразу же снова становится поддерживаемым, поле ошибки может быть пустым. |
| {TRIGGER.STATUS} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Скрипты действий вручную над событиями | Значение триггера на момент выполнения шага операции. Может быть либо PROBLEM, либо OK. {STATUS} является устаревшим. |
| {TRIGGER.TEMPLATE.NAME} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | Отсортированный (запросом SQL) список шаблонов, в которых определен триггер (через запятую), или \*UNKNOWN\*, если триггер определен на узле сети. |
| {TRIGGER.URL} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Внутренние оповещения на события триггеров → Скрипты действий вручную над событиями | URL триггера. |
| {TRIGGER.VALUE} | → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Trigger expressions → Скрипты действий вручную над событиями | Текущее числовое значение триггера: 0 – триггер в состоянии OK, 1 – триггер в состоянии ПРОБЛЕМА. |
| {TRIGGERS.UNACK} | → Подписи к элементам карты | Количество неподтвержденных триггеров у элемента карты сети, независимо от состояния триггеров. Триггер считается неподтвержденным, если по крайней мере одно из всех его событий ПРОБЛЕМА не подтверждено. |
| {TRIGGERS.PROBLEM.UNACK} | → Подписи к элементам карты | Количество неподтвержденных триггеров в состоянии ПРОБЛЕМА у элемента карты сети. Триггер считается неподтвержденным, если по крайней мере одно из всех его событий ПРОБЛЕМА не подтверждено. |
| {TRIGGERS.ACK} | → Подписи к элементам карты | Количество подтвержденных триггеров у элемента карты сети, независимо от состояния триггеров. Триггер считается подтвержденным, если все его события ПРОБЛЕМА являются подтвержденными. |
| {TRIGGERS.PROBLEM.ACK} | → Подписи к элементам карты | Количество неподтвержденных триггеров в состоянии ПРОБЛЕМА у элемента карты сети. Триггер считается подтвержденным, если все его события ПРОБЛЕМА являются подтвержденными. |
| {USER.FULLNAME} | → Оповещения и команды на обновление проблем → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) | Имя, фамилия и имя пользователя того пользователя, который добавил подтверждение события или запустил скрипт. |
| {USER.NAME} | → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) | Имя того пользователя, который запустил скрипт. |
| {USER.SURNAME} | → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) | Фамилия того пользователя, который запустил скрипт. |
| {USER.USERNAME} | → Скрипты действий вручную над узлом сети (в том числе текст подтверждения) → Скрипты действий вручную над событиями (в том числе текст подтверждения) | Имя пользователя того пользователя, который запустил скрипт. |
| {$МАКРОС} | → Смотрите: Пользовательские макросы, поддерживаемые по местоположению | Макросы, определяемые пользователем. |
| {#МАКРОС} | → Смотрите: Макросы низкоуровневого обнаружения | Макросы низкоуровневого обнаружения.  Модификация значения макроса поддерживается для этих макросов. |
| {?ВЫРАЖЕНИЕ} | → Имена событий триггеров → Оповещения и команды на события триггеров → Оповещения и команды на обновление проблем → Команды скриптов и параметры их вебхуков → Подписи к элементам карты3 → Подписи к фигурам карты3 → Подписи к связям на карте3 → Имена графиков5 | См. макросы выражений. |
| $1...$9 | → Имена триггеров → Команды пользовательских параметров | Позиционные макросы/ссылки. |

Примечания:

1 Макросы {HOST.\*}, поддерживаемые в параметрах ключа элемента данных, будут раскрываться в интерфейс, выбранный для этого элемента данных. При использовании в элементах данных без интерфейсов они будут раскрываться в интерфейс Агента, SNMP, JMX или IPMI узла сети в этом порядке приоритетов либо в "UNKNOWN" при отсутствии у узла сети каких-либо интерфейсов.

2 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях макрос будет раскрываться в основной интерфейс Агента, но, если он отсутствует, будет использоваться основной интерфейс SNMP. Если SNMP также отсутствует, будет использоваться основной интерфейс JMX. Если же и JMX отсутствует, будет использоваться основной интерфейс IPMI. Если у узла сети вообще нет никаких интерфейсов, макрос будет раскрыт в значение "UNKNOWN".

3 В подписях к элементам карты в этом макросе поддерживаются только функции avg, last, max и min, с секундами в качестве параметра.

4 Макросы {HOST.\*} поддерживаются в полях веб-сценариев "Переменные", "Заголовки", "Файл SSL сертификата" и "Файл SSL ключа" и в полях шагов сценария "URL", "Post", "Заголовки" и "Требуемая строка". Макросы {HOST.\*} не поддерживаются в полях веб-сценария Имя и шага сценария Имя.

5 В именах графиков внутри этого макроса поддерживаются только функции avg, last, max и min, с секундами в качестве параметра. Макрос {HOST.HOST<1-9>} может использоваться в качестве узла сети внутри макроса. Например:

last(/Cisco switch/ifAlias[{#SNMPINDEX}])

last(/{HOST.HOST}/ifAlias[{#SNMPINDEX}])

1. Нумерованные макросы

Синтаксис нумерованных макросов {МАКРОС<1-9>} ограничен рамками выражений триггеров. Нумерованные макросы можно использовать, чтобы сослаться на узлы сети или функции в том порядке, в котором они указаны в выражении. Такие макросы как {HOST.IP1}, {HOST.IP2}, {HOST.IP3} будут раскрыты в IP первого, второго и третьего узла сети из выражения триггера (если только выражение триггера содержит эти узлы сети). Такие макросы как {FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3} будут раскрыты в значение первой, второй и третьей функции, относящейся к элементам данных, из выражения триггера на момент события (если только выражение триггера содержит эти функции).

Кроме того, макрос {HOST.HOST<1-9>} также поддерживается внутри макроса выражений {?функ(/узел\_сети/ключ,парам)} в именах графиков. Например, {?функ(/{HOST.HOST2}/ключ,парам)} в имени графика будет ссылаться на узел сети второго элемента данных графика.

Нумерованные макросы не раскрываются в любом другом контексте, кроме двух упомянутых здесь случаев. В остальных случаях следует использовать макросы без номера (например, {HOST.HOST}, {HOST.IP} и т.п.).

1. Пользовательские макросы, поддерживаемые по местоположению

В этом разделе представлен список мест, где поддерживаются определяемые пользователем макросы.

Для "Действий", "Сетевого обнаружения", "Прокси" и всех мест, перечисленных в разделе "Другие места" этой страницы, поддерживаются только пользовательские макросы, определенные на глобальном уровне. В упомянутых местах макросы, определенные на уровне узла сети и на уровне шаблона, раскрываться не будут.

При настройке действий пользовательские макросы могут быть использованы в полях, указанных в таблице 103.

Действия

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Оповещения и команды на события триггеров | | да |
| Внутренние оповещения на события триггеров | | да |
| Оповещения об обновлении проблем | | да |
| Оповещения и команды на события услуг | | да |
| Оповещения об обновлении услуг | | да |
| Условие периода времени | | нет |
| Операции | | |
|  | Длительность шага операции по умолчанию | нет |
| Длительность шага | нет |

При настройке узла сети и прототипа узла сети пользовательские макросы могут использоваться в полях, указанных в таблице 104.

Узлы сети / прототипы узлов сети

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| IP-адрес / DNS-имя интерфейса | | только DNS |
| Порт интерфейса | | нет |
| SNMP v1, v2 | | |
|  | SNMP community | да |
| SNMP v3 | | |
|  | Имя контекста | да |
| Имя безопасности | да |
| Пароль аутентификации | да |
| Ключевая фраза безопасности | да |
| IPMI | | |
|  | Имя пользователя | да |
| Пароль | да |
| Теги | | |
|  | Имена тегов | да |
| Значения тегов | да |

При настройке элементов данных или прототипов элементов данных пользовательские макросы могут использоваться в полях, указанных в таблице 105.

Элементы данных / прототипы элементов данных

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Параметры ключа элемента данных | | да |
| Интервал обновления | | нет |
| Пользовательские интервалы | | нет |
| Период хранения истории | | нет |
| Период хранения динамики изменений | | нет |
| Описание | | да |
| Вычисляемый элемент данных | | |
|  | Формула | да |
| Монитор баз данных | | |
|  | Имя пользователя | да |
| Пароль | да |
| SQL запрос | да |
| HTTP Агент | | |
|  | URL | да |
| Поля запроса | да |
| Время ожидания | нет |
| Тело запроса | да |
| Заголовки (имена и значения) | да |
| Требуемые коды состояния | да |
| HTTP Прокси | да |
| Имя пользователя для HTTP аутентификации | да |
| Пароль для HTTP аутентификации | да |
| Файл SSL сертификата | да |
| Файл SSL ключа | да |
| Пароль к SSL ключу | да |
| Разрешенные хосты | да |
| JMX Агент | | |
|  | JMX endpoint | да |
| Скриптовый элемент данных | | |
|  | Имена и значения параметров | да |
| SNMP Агент | | |
|  | SNMP OID | да |
| SSH Агент | | |
|  | Имя пользователя | да |
| Файл публичного ключа | да |
| Файл приватного ключа | да |
| Пароль | да |
| Выполняемый скрипт | да |
| TELNET Агент | | |
|  | Имя пользователя | да |
| Пароль | да |
| Выполняемый скрипт | да |
| Траппер | | |
|  | Разрешенные хосты | да |
| Теги | | |
|  | Имена тегов | да |
| Значения тегов | да |
| Предобработка | | |
|  | Параметры шагов (включая пользовательские скрипты) | да |

При настройке правил низкоуровневого обнаружения пользовательские макросы могут использоваться в полях, указанных в таблице 106.

Низкоуровневое обнаружение

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Параметры ключа | | да |
| Интервал обновления | | нет |
| Пользовательские интервалы | | нет |
| Период сохранения потерянных ресурсов | | нет |
| Описание | | да |
| SNMP Агент | | |
|  | SNMP OID | да |
| SSH Агент | | |
|  | Имя пользователя | да |
| Файл публичного ключа | да |
| Файл приватного ключа | да |
| Пароль | да |
| Выполняемый скрипт | да |
| TELNET Агент | | |
|  | Имя пользователя | да |
| Пароль | да |
| Выполняемый скрипт | да |
| траппер | | |
|  | Разрешенные хосты | да |
| Мониторинг баз данных | | |
|  | Имя пользователя | да |
| Пароль | да |
| SQL запрос | да |
| JMX Агент | | |
|  | JMX endpoint | да |
| HTTP Агент | | |
|  | URL | да |
| Поля запроса | да |
| Время ожидания | нет |
| Тело запроса | да |
| Заголовки (имена и значения) | да |
| Требуемые коды состояния | да |
| Имя пользователя для HTTP аутентификации | да |
| Пароль для HTTP аутентификации | да |
| Фильтры | | |
|  | Регулярное выражение | да |
| Замещения | | |
|  | Фильтры: регулярное выражение | да |
| Операции: интервал обновления (для прототипов элементов данных) | нет |
| Операции: период хранения истории (для прототипов элементов данных) | нет |
| Операции: период хранения динамики изменений (для прототипов элементов данных) | нет |

При настройке правил сетевого обнаружения пользовательские макросы могут использоваться в полях, указанных в таблице 107.

Сетевое обнаружение

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Интервал обновления | | нет |
| SNMP v1, v2 | | |
|  | SNMP community | да |
| SNMP OID | да |
| SNMP v3 | | |
|  | Имя контекста | да |
| Имя безопасности | да |
| Пароль аутентификации | да |
| Ключевая фраза безопасности | да |
| SNMP OID | да |

При настройке Прокси-серверов пользовательские макросы могут использоваться в полях, указанных в таблице 108.

Прокси-серверы

| Место | Несколько макросов/вместе с текстом |
| --- | --- |
| Порт интерфейса (для пассивных Прокси) | нет |

При настройке шаблонов пользовательские макросы могут использоваться в полях, указанных в таблице 109.

Шаблоны

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Теги | | |
|  | Имена тегов | да |
| Значения тегов | да |

При настройке триггеров пользовательские макросы могут использоваться в полях, указанных в таблице 110

Триггеры

| Место | | Несколько макросов/вперемешку с текстом |
| --- | --- | --- |
| Имя | | да |
| Оперативные данные | | да |
| Выражение (только в константах и параметрах функций; секретные макросы не поддерживаются) | | да |
| Описание | | да |
| URL | | да |
| Теги для совпадения | | да |
| Теги | | |
|  | Имена тегов | да |
| Значения тегов | да |

При настройке веб-сценариев пользовательские макросы могут использоваться в полях, указанных в таблице 111.

Веб-сценарии

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Имя | | да |
| Интервал обновления | | нет |
| Агент | | да |
| HTTP Прокси | | да |
| Переменные (только значения) | | да |
| Заголовки (имена и значения) | | да |
| Шаги | | |
|  | Имя | да |
| URL | да |
| Переменные (только значения) | да |
| Заголовки (имена и значения) | да |
| Время ожидания | нет |
| Требуемая строка | да |
| Требуемые коды состояния | нет |
| Аутентификация | | |
|  | Пользователь | да |
| Пароль | да |
| Файл SSL сертификата | да |
| Файл SSL ключа | да |
| Пароль к SSL ключу | да |
| Теги | | |
|  | Имена тегов | да |
| Значения тегов | да |

Помимо перечисленных здесь мест, пользовательские макросы могут использоваться в полях, указанных в таблице 112.

Другие места

| Место | | Несколько макросов/вместе с текстом |
| --- | --- | --- |
| Глобальные скрипты (скрипт, SSH, Telnet, IPMI), включая текст подтверждения | | да |
| Вебхуки | | |
|  | Скрипт JavaScript | нет |
| Имя параметра скрипта JavaScript | нет |
| Значение параметра скрипта JavaScript | да |
| Мониторинг → Панели | | |
|  | Поле описания виджета панели Значение элемента данных | да |
| Поле URL виджета панели динамический URL | да |
| Администрирование → Пользователи → Оповещения | | |
|  | Когда активен | нет |
| Администрирование → Общие → Веб-интерфейс | | |
|  | Рабочее время | нет |
| Администрирование → Способы оповещений → Шаблоны сообщений | | |
|  | Тема | да |
| Сообщение | да |

Примечания:

* Если несколько макросов в поле или макросы вместе с текстом не поддерживаются для данного места, то один макрос должен заполнять целиком все поле.
* Макросы, которые используются в именах и значениях тегов, раскрываются только во время процесса генерации событий.
* URL, которые содержат секретные макросы, работать не будут, поскольку макросы в них будут раскрыты как "\*\*\*\*\*\*".

Приложение К. Прочее

1. Символы единиц измерения

Использование каких-либо больших чисел, например "86400", для представления количества секунд в сутках бывает затруднительно и чревато ошибками. Поэтому можно использовать некоторые соответствующие символы единиц измерения (или суффиксы) для упрощения выражений триггеров и ключей элементов данных в Подсистеме.

Вместо "86400" можно ввести просто "1d". Функции суффиксов работают как множители.

Суффиксы времени можно использовать для указания времени:

* s – секунды (если используется, то работает так же, как и исходное значение);
* m – минуты;
* h – часы;
* d – сутки;
* w – недели;
* M – месяцы (только в функциях динамики изменений);
* y – годы (только в функциях динамики изменений).

Суффиксы времени поддерживают только целые числа (поэтому "1h" поддерживается, а "1,5h" или "1.5h" – нет; использовать вместо этого "90m").

Суффиксы времени поддерживаются в следующих местах:

* константы и параметры функций выражений триггеров;
* константы формул вычисляемых элементов данных;
* параметры внутренних элементов данных "zabbix[queue,<от>,<до>]";
* параметр период времени агрегированных вычислений;
* настройка элементов данных (поля "Интервал обновления", "Пользовательские интервалы", "Период хранения истории" и "Период хранения динамики изменений");
* настройка прототипов элементов данных (поля "Интервал обновления", "Пользовательские интервалы", "Период хранения истории" и "Период хранения динамики изменений");
* настройка правил низкоуровневого обнаружения (поля "Интервал обновления", "Пользовательские интервалы", "Период сохранения потерянных ресурсов");
* настройка сетевого обнаружения (поле "Интервал обновления");
* настройка веб-сценария (поля "Интервал обновления", "Время ожидания");
* настройка операции действия (поля "Длительность шага операции по умолчанию", "Длительность шага");
* настройки профиля пользователя (поля "Авто-выход", "Обновить", "Время отображения сообщения");
* виджет графика в разделе "Мониторинг → Панель" (поле "Сдвиг по времени");
* "Администрирование → Общие → Очистка истории" (поля периодов хранения);
* "Администрирование → Общие → Опции отображения триггеров" (поля "Отображать триггеры в состоянии ОК в течение", "Мигание триггеров при изменении состояния");
* "Администрирование → Общие → Другое" (поле "Интервал блокировки входа в систему" и поля, относящиеся к взаимодействию с Сервером);
* опция управления работой Сервера "ha\_set\_failover\_delay=задержка".

Суффиксы размера памяти поддерживаются в следующих местах:

* константы и параметры функций выражений триггера;
* константы формул вычисляемых элементов данных.

Для размера памяти можно использовать:

* К – килобайт;
* M – мегабайт;
* G – гигабайт;
* T – терабайт.

Символы единиц измерения также используются для более удобного представления данных в веб-интерфейсе.

Эти символы поддерживает как Сервер, так и веб-интерфейс:

* K – кило;
* M – мега;
* G – гига;
* T – тера.

Когда в веб-интерфейсе отображаются значения элементов данных в B, Bps, то применяется основание 2 (1K = 1024). Иначе используется основание 10 (1K = 1000).

Дополнительно веб-интерфейс также поддерживает отображение:

* P – пета;
* E – экса;
* Z – зетта;
* Y – иотта.

При использовании некоторых соответствующих суффиксов можно написать выражения триггеров, которые легче понимать и поддерживать, например, такие выражения:

last(/host/system.uptime[])<86400s

avg(/host/system.cpu.load,600s)<10

last(/host/vm.memory.size[available])<20971520

можно заменить на:

last(/host/system.uptime[])<1d

avg(/host/system.cpu.load,10m)<10

last(/host/vm.memory.size[available])<20M

1. Настройка периодов времени

Чтобы указать период времени, должен использоваться следующий формат:

d-d,hh:mm-hh:mm

где символы означают:

* d – день недели: 1 – понедельник, 2 – вторник, ... , 7 – воскресенье;
* hh – часы: 00-24;
* mm – минуты: 00-59.

Можно указать более одного периода времени, используя разделитель в виде точки с запятой (;):

d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...

Если оставить период времени пустым, это будет эквивалентно "1-7,00:00-24:00", что является значением по умолчанию.

Верхняя граница периодов времени не включается. Таким образом, если указать период 09:00-18:00, то последней секундой в периоде будет 17:59:59.

Например, рабочее время. С понедельника по пятницу с 9:00 до 18:00:

1-5,09:00-18:00

Рабочие часы плюс выходные. С понедельника по пятницу с 9:00 до 18:00, а также суббота и воскресенье с 10:00 до 16:00:

1-5,09:00-18:00**;**6-7,10:00-16:00

1. Выполнение команд

Подсистема использует единый функционал для внешних проверок, пользовательских параметров, элементов данных system.run, пользовательских скриптов оповещений, удаленных команд и глобальных скриптов.

По умолчанию все скрипты в Подсистеме выполняются с использованием оболочки sh, и оболочку по умолчанию изменить нельзя. Чтобы использовать другую оболочку, можно применить следующий прием: создать файл скрипта и вызвать этот скрипт при выполнении команды.

Команда/скрипт выполняется одинаково как на UNIX, так и на Windows-платформах:

1. Подсистема (родительский процесс) создает конвейер для связи;
2. Подсистема настраивает данный конвейер для вывода данных от создаваемого дочернего процесса;
3. Подсистема создает дочерний процесс (выполняет команду/скрипт);
4. создается новая группа процесса (в UNIX) или задача (в Windows) для дочернего процесса;
5. Подсистема считывает из конвейера данные до тех пор, пока не истечет установленное время ожидания или до момента, когда прекратится запись с другой стороны (закрыты все обработчики/файловые дескрипторы). Следует обратить внимание, что дочерний процесс может создать еще некоторое количество процессов и выйти до того, как эти процессы завершатся или будет закрыт обработчик/файловый дескриптор;
6. если установленное время ожидания не было достигнуто, Подсистема ждет завершения начального дочернего процесса или ждет достижения установленного времени ожидания;
7. если начальный дочерний процесс завершил свою работу и время ожидания не окончилось, то Подсистема проверяет код завершения начального дочернего процесса и сравнивает его с 0 (ненулевое значение считается ошибкой выполнения только для пользовательских скриптов оповещений, удаленных команд и пользовательских скриптов, выполняемых на Сервере и Прокси);
8. на этот момент подразумевается, что все выполнено, и все дерево процессов (т.е. группа процессов или задача) завершается.

Подсистема определяет, что команда/скрипт завершили обработку в тот момент, когда завершился изначальный дочерний процесс, и никакие другие процессы не держат открытым обработчик/файловый дескриптор вывода. Когда обработка завершена, все созданные процессы завершаются.

Все двойные кавычки и обратная косая черта в команде экранируются обратной косой чертой, и вся команда заключается в двойные кавычки.

Код завершения проверяется на соответствие следующим условиям:

* только для пользовательских скриптов оповещений, удаленных команд и пользовательских скриптов, выполняемых на Сервере и Прокси;
* любой код завершения, отличный от 0, считается ошибкой выполнения;
* содержимое стандартного вывода ошибки и стандартного вывода собирается при ошибочных выполнениях и доступно в веб-интерфейсе (где отображается результат выполнения);
* дополнительная запись добавляется в журнал удаленных команд на Сервере, чтобы сохранить вывод выполнения скрипта; эту возможность можно включить, используя параметр Агента LogRemoteCommands.

Возможные сообщения в веб-интерфейсе и записи в журналах при ошибочных командах/скриптах:

* содержимое стандартного вывода ошибки и стандартного вывода при ошибочных выполнениях (если имеется);
* "Process exited with code: N." (при пустом выводе и отличном от 0 коде выхода);
* "Process killed by signal: N." (при завершении процесса сигналом, только на Linux);
* "Process terminated unexpectedly." (при завершениях процесса по неизвестным причинам).

1. Обработка ошибок базы данных

Если Подсистема определяет, что его БД недоступна, он отправляет оповещение и продолжает попытки подключения к БД. Для некоторых механизмов БД распознаются определенные коды ошибок.

Например, для БД MySQL:

* CR\_CONN\_HOST\_ERROR;
* CR\_SERVER\_GONE\_ERROR;
* CR\_CONNECTION\_ERROR;
* CR\_SERVER\_LOST;
* CR\_UNKNOWN\_HOST;
* ER\_SERVER\_SHUTDOWN;
* ER\_ACCESS\_DENIED\_ERROR;
* ER\_ILLEGAL\_GRANT\_FOR\_TABLE;
* ER\_TABLEACCESS\_DENIED\_ERROR;
* ER\_UNKNOWN\_ERROR.

1. Динамическая библиотека sender для Windows

В ОС Windows приложения могут отправлять данные в Сервер/Прокси напрямую с помощью библиотеки динамической компоновки sender (zabbix\_sender.dll) вместо запуска внешнего процесса (zabbix\_sender.exe).

Динамическая библиотека с файлами разработки находится в директории bin. Для ее использования нужно включать заголовочный файл zabbix\_sender.h и связывать с библиотекой zabbix\_sender.lib. Пример файла с использованием API sender можно найти в директории build\_sender.

Динамическая библиотека sender предоставляет следующий функционал:

| int  zabbix\_sender\_send\_values(  const  char  \*address,  unsigned  short  port,  const  char  \*source,  const  zabbix\_sender\_value\_t \*values,  int  count,  char  \*\*result); |
| --- |
| char \*\*result);`{.c} |

Следующие структуры данных используются динамической библиотекой sender:

typedef struct

{

/\* имя узла сети, должно совпадать с именем целевого узла сети в Подсистеме \*/

char\*host;

/\* ключ элемента данных \*/

char\*key;

/\* значение элемента данных \*/

char\*value;

}

zabbix\_sender\_value\_t;

typedef struct

{

/\* общее количество обработанных значений \*/

int total;

/\* количество неуспешных значений \*/

int failed;

/\* время в секундах, потраченное сервером на обработку отосланных значений \*/

doubletime\_spent;

}

zabbix\_sender\_info\_t;

1. Библиотека Python для API

[zabbix\_utils [en]](https://github.com/zabbix/python-zabbix-utils/blob/main/README.md) – это библиотека Python для:

* работы с API;
* действий в качестве sender;
* действий в качестве утилиты get.

1. Вход в систему и systemd

Рекомендуется создать пользователя zabbix как системного, то есть без возможности входа в Подсистему. Некоторые пользователи игнорируют эту рекомендацию и используют ту же самую учетную запись для входа (например, используя SSH) на хост с работающим Подсистемой. Это может привести к аварийному останову демона Подсистемы при выходе. В этом случае в журнале Сервера получают примерно следующее:

zabbix\_server [27730]: [file:"selfmon.c",line:375] lock failed: [22] Invalid argument

zabbix\_server [27716]: [file:"dbconfig.c",line:5266] lock failed: [22] Invalid argument

zabbix\_server [27706]: [file:"log.c",line:238] lock failed: [22] Invalid argument

и в журнале Агента:

zabbix\_agentd [27796]: [file:"log.c",line:238] lock failed: [22] Invalid argument

Это происходит из-за того, что по умолчанию настройка systemd в файле /etc/systemd/logind.conf равна RemoveIPC=yes. Когда выполняется выход из Подсистемы, ранее созданные компонентами Подсистемы семафоры удаляются, что приводит к сбою.

В случае с systemd:

RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources after the last of the user"s sessions terminated. This covers System V semaphores, shared memory and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the root user and other system users are excluded from the effect of this setting. Defaults to "yes".

Имеется 2 способа решения этой проблемы:

* (рекомендуемый) Прекратить использование аккаунт zabbix для чего-либо, кроме процессов Подсистемы, выделить отдельную учетную запись для остальных потребностей.
* (не рекомендуемый) Задать RemoveIPC=no в /etc/systemd/logind.conf и перезагрузить ОС. Следует обратить внимание, что RemoveIPC является общесистемным параметром, его изменение повлияет на всю ОС.

Если веб-интерфейс работает за Прокси-сервером, путь cookie в файле конфигурации Прокси необходимо переписать, чтобы он соответствовал reverse-proxied path (обратному пути Прокси). Если путь cookie не переписан, пользователи могут столкнуться с проблемами авторизации при попытке войти в веб-интерфейс.

Пример конфигурации для Nginx:

# ..

location / {

# ..

proxy\_cookie\_path /zabbix /;

proxy\_pass http://192.168.0.94/zabbix/;

# ..

Пример конфигурации для Apache:

# ..

ProxyPass "/" http://host/zabbix/

ProxyPassReverse "/" http://host/zabbix/

ProxyPassReverseCookiePath /zabbix /

ProxyPassReverseCookieDomain host mon.rosa.dev

# ..

1. Отличия между Агентом-1 и Агентом-2

В таблице 113 описываются отличия между Агентом-1 и Агентом-2.

Отличия между Агентом-1 и Агентом-2

| Параметр | Агент-1 | Агент-2 |
| --- | --- | --- |
| Язык программирования | C | Go с некоторыми частями на C |
| Демонизация | да | только с помощью systemd (да на Windows) |
| Поддерживаемые расширения | Пользовательские загружаемые модули на C. | Пользовательские плагины на Go. |
| Требования | | |
| Поддерживаемые платформы | Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: все настольные и серверные версии, начиная с XP | Linux, Windows: все настольные и серверные версии, на которых может быть установлена актуальная поддерживаемая версия Go. |
| Поддерживаемые криптобиблиотеки | GnuTLS 3.1.18 и выше OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x.  LibreSSL – протестирована с версиями 2.7.4, 2.8.2 (применяются определенные ограничения, см. раздел Шифрование). | Linux: OpenSSL 1.0.1 и выше  MS Windows: OpenSSL 1.1.1 или выше. В библиотеке OpenSSL должны быть включена поддержка PSK. LibreSSL не поддерживается. |
| Процессы мониторинга | | |
| Процессы | Отдельный процесс активных проверок для каждой записи Сервера/Прокси. | Один процесс с автоматически созданными потоками. Максимальное количество потоков определяется переменной окружения GOMAXPROCS. |
| Метрики | **UNIX**: см. список поддерживаемых элементов данных.  **Windows**: см. список дополнительных элементов данных, специфичных для Windows. | **UNIX**: все метрики, поддерживаемые Агентом. Кроме того, Агент-2 предоставляет нативное решение для мониторинга с помощью Подсистемы: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd и других объектов мониторинга – см. полный список элементов данных, специфичных для Агента 2. **Windows**: все метрики, поддерживаемые Агентом, а также net.tcp.service\* проверки для HTTPS, LDAP. Кроме того, Агент-2 предоставляет нативное решение для мониторинга с помощью Подсистемы: PostgreSQL, Redis. |
| Параллелизм | Активные проверки для одного Сервера выполняются последовательно. | Проверки из разных плагинов или множественные проверки в одном плагине могут выполняться параллельно. |
| Пользовательские/гибкие интервалы | Поддерживаются только для пассивных проверок. | Поддерживаются для пассивных и активных проверок. |
| Сторонние трапы | нет | да |
| Дополнительный функционал | | |
| Постоянное хранилище | нет | да |
| Постоянные файлы для элементов данных log\*[] | да (только на UNIX) | нет |
| Передача данных журнала | Может выполняться во время сбора данных журнала, чтобы освободить буфер. | Сбор данных журнала останавливается при заполнении буфера, поэтому параметр BufferSize должен быть равен как минимум MaxLinesPerSecond\*2. |
| Настройки тайм-аутов | Задаются только на уровне Агента. | Тайм-аут плагина может переопределить таймаут, заданный на уровне Агента. |
| Смена пользователя во время выполнения | да (только UNIX-подобные системы) | нет (управляется systemd) |
| Настраиваемые пользователем шифры | да | нет |

1. Примеры экранирования

В этом разделе представлены примеры правильного экранирования при использовании регулярных выражений в различных контекстах.

При использовании конструктора триггерных выражений правильное экранирование в регулярных выражениях добавляется автоматически.

**Пользовательский макрос с контекстом:**

Регулярное выражение: \.+\"[a-z]+

Пользовательский макрос с контекстом: {$MACRO:regex:"\.+\\"[a-z]+"}

Замечания:

* обратные слеши не экранируются;
* кавычки экранируются.

**Функция макросов низкоуровневого обнаружения (LLD):**

Регулярное выражение: \.+\"[a-z]+

Макрос LLD: {{#MACRO}.iregsub("\.+\\"[a-z]+", \1)}

Замечания:

* обратные слеши не экранируются;
* кавычки экранируются.

**Функция макросов LLD внутри контекста пользовательского макроса:**

Регулярное выражение: \.+\"[a-z]+

Макрос LLD: {{#MACRO}.iregsub("\.+\\"[a-z]+", \1)}

Пользовательский макрос с контекстом: {$MACRO:"{{#MACRO}.iregsub(\".+\\\"[a-z]+\", \1)}"}

Замечания:

* экранирование обратных слешей для LLD не меняется;
* при вставке макроса LLD в контекст пользовательского макроса нужно поместить это в строку;
* выражение макроса заключается в кавычки;
* кавычки экранируются; всего добавляется 3 новых обратных слеша.

**Строковый параметр функции, не относящейся к истории**

Содержимое строки: \.+\"[a-z]+

Выражение: concat("abc", "\\.\\\"[a-z]+")

Замечание:

* строковый параметр требует экранирования как для обратных слешей, так и для кавычек.

**Строковый параметр функции истории**

Содержимое строки: \.+\"[a-z]+

Выражение: find(\_\_ITEM\_KEY\_\_,,"regexp","\.+\\"[a-z]+")

Замечания:

* обратные слеши не экранируются;
* кавычки экранируются.

**Функция макросов LLD внутри строкового параметра функции, не относящейся к истории**

Регулярное выражение: \.+\"[a-z]+

Макрос LLD: {{#MACRO}.iregsub("\.+\\"[a-z]+", \1)}

Выражение: concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"[a-z]+\", \\1)}")

Замечания:

* строковые параметры требуют экранирования как для обратных слешей, так и для кавычек;
* добавляется еще один уровень экранирования, так как макрос будет раскрыт только после раскрытия кавычек в строке.

**Функция макросов LLD внутри строкового параметра функции истории**

Регулярное выражение: \.+\"[a-z]+

Макрос LLD: {{#MACRO}.iregsub("\.+\\"[a-z]+", \1)}

Выражение: find(\_\_ITEM\_KEY\_\_,,"eq","{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}")

Замечания:

* обратные слеши не экранируются;
* кавычки экранируются.

**Пользовательский макрос с контекстом внутри строкового параметра функции, не относящейся к истории**

Регулярное выражение: \.+\"[a-z]+

Пользовательский макрос с контекстом: {$MACRO:regex:"\.+\\"[a-z]+"}

Выражение: concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")

Замечания:

* как и в предыдущем примере, необходим дополнительный уровень экранирования;
* обратные слеши и кавычки экранируются только для экранирования верхнего уровня (поскольку это строковый параметр).

**Пользовательский макрос с контекстом внутри строкового параметра функции истории**

Регулярное выражение: \.+\"[a-z]+

Пользовательский макрос с контекстом: {$MACRO:regex:"\.+\\"[a-z]+"}

Выражение: find(\_\_ITEM\_KEY\_\_,,"eq","{$MACRO:regex:\"\.+\\\"[a-z]+\"}")

Замечания:

* обратные слеши не экранируются;
* кавычки экранируются.

**Функция макросов LLD внутри контекста пользовательского макроса внутри функции, не относящейся к истории**

Регулярное выражение: \.+\"[a-z]+

Макрос LLD: {{#MACRO}.iregsub("\.+\\"[a-z]+", \1)}

Пользовательский макрос с контекстом: {$MACRO:"{{#MACRO}.iregsub(\".+\\\"[a-z]+\", \1)}"}

Выражение: concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\".+\\\\\\\"[a-z]+\\\", \\1)}\"}")

Замечания по трем уровням экранирования:

* для функции макросов LLD; без экранирования обратных слешей;
* для пользовательского макроса с контекстом; без экранирования обратных слешей;
* для строкового параметра функции; с экранированием обратных слешей.

**Функция макросов LLD внутри контекста пользовательского макроса внутри функции истории**

Регулярное выражение: \.+\"[a-z]+

Макрос LLD: {{#MACRO}.iregsub("\.+\\"[a-z]+", \1)}

Пользовательский макрос с контекстом: {$MACRO:"{{#MACRO}.iregsub(\".+\\\"[a-z]+\", \1)}"}

Выражение: find(\_\_ITEM\_KEY\_\_,,"eq","{$MACRO:\"{{#MACRO}.iregsub(\\".+\\\\"[a-z]+\\", \1)}\"}")

Замечания:

* обратные слеши не экранируются;
* кавычки экранируются.

**Пользовательский макрос с контекстом просто внутри строки**

Регулярное выражение: \.+\"[a-z]+

Пользовательский макрос с контекстом: {$MACRO:regex:"\.+\\"[a-z]+"}

Внутри строки некоторого выражения, например: func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"

Замечания:

* строки также требуют экранирования обратных слешей;
* строки также требуют экранирования кавычек;
* снова случай с двумя уровнями экранирования:
* экранирование для контекста пользовательского макроса; без экранирования обратных слешей;
* экранирование, поскольку это строка; с экранированием обратных слешей.

Перечень терминов и сокращений

| **Сокращение** | **Определение** |
| --- | --- |
| БД | База данных |
| ВМ | Виртуальная машина |
| ИТ | Информационные технологии |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| СИПА | Система идентификации, политик и аудита |
| ЦС | Центр сертификации |
| API | Application programming interface – программный интерфейс приложения |
| CA | Certification authority – центр сертификации (удостоверяющий центр) |
| CSV | Comma separated values – текстовый формат, предназначенный для представления табличных данных |
| DHCP | Dynamic host configuration protocol – протокол динамической настройки сетевой конфигурации хоста |
| DNS | Domain name system – система доменных имен |
| ENC | External node classifier – классификатор внешних хостов |
| ESR | Extended support release – релиз с расширенной (долговременной) поддержкой |
| HA | [High availability – высокая доступность](https://en.wikipedia.org/wiki/High_availability) |
| HSTS | HTTP Strict Transport Security – механизм, принудительно активирующий защищенное соединение через протокол HTTPS |
| HTTP | Hypertext transfer protocol – протокол передачи гипертекста |
| HTTPS | Hypertext transfer protocol secure – защищенная версия протокола передачи гипертекста |
| IP | Internet protocol – протокол межсетевого взаимодействия |
| IPA | Identity, policy and audit – система идентификации, политик и аудита (СИПА) |
| JSON | JavaScript Object Notation – формат обмена данными |
| LDAP | Lightweight directory access protocol – протокол доступа к каталогам |
| LDAPS | Lightweight directory access protocol secure – защищенная версия протокола доступа к каталогам |
| MAC | Media access control – уникальный идентификатор сетевого оборудования |
| MTA | Mail transfer agent – Агент передачи почтовых сообщений |
| NTP | Network time protocol – протокол сетевого времени |
| RHEL | Red Hat Enterprise Linux – наименование операционной системы |
| SMTP | Simple mail transfer protocol – протокол передачи почтовых сообщений |
| SSH | Secure shell – защищенная оболочка |
| SSL | Secure sockets layer – уровень защищенных сокетов |
| TCP | Transmission control protocol – протокол управления передачей данных |
| TFTP | Trivial file transfer protocol – протокол передачи файлов |
| UDP | User datagram protocol – протокол пользовательских датаграмм |
| URL | Uniform resource locator – сетевой адрес ресурса |
| YAML | Yet another markup language – язык разметки |

|  |  |
| --- | --- |
| Термины | Определение |
| Узел сети | Любое физическое или виртуальное устройство, приложение, служба или любой другой логически связанный набор параметров мониторинга |
| Группа узлов сети | Логическая группировка узлов сети; группа может содержать узлы сети и шаблоны. Узлы сети и шаблоны в группе узлов сети никаким образом не связаны друг с другом. Группы узлов сети используются при назначении прав доступа к узлам сети различным группам пользователей |
| Элемент данных | Конкретный фрагмент данных, который требуется получать от узла сети; метрика |
| Предобработка значений | Преобразование полученного метрикой значения перед сохранением этого значения в базу данных. |
| Триггер | Логическое выражение, которое определяет порог проблемы и используется для "оценки" данных, полученных элементами данных Если полученные данные превышают порог, триггер переходит из состояния "Ок" в состояние "Проблема". Когда полученные данные ниже порога, триггер остается/возвращается в состояние "Ок" |
| Событие | Одиночное возникновение ситуации, например изменение состояния триггера или обнаружение/авторегистрация Агента |
| Тег события | Предопределенный маркер для события. Его можно использовать в корреляции событий, в определении прав доступа и т.д. |
| Корреляция событий | Метод корреляции проблем с решением этих проблем Например, можно задать, что проблема, которую сообщил один триггер, может быть решена другим триггером, который может использовать даже совершенно другой метод сбора данных |
| Проблема | Триггер, который находится в состоянии "Проблема". |
| Обновление проблемы | Варианты управления проблемами в Подсистеме, такие как добавление комментария, подтверждение, изменение важности и закрытие вручную |
| Действие | Предопределенные средства реагирования на события. Действия состоят из операций (например, отправка оповещений) и условий осуществления операций |
| Эскалация | Пользовательский сценарий выполнения операций в рамках действия; последовательность отправки оповещений/выполнения удаленных команд |
| Способ оповещения | Средство доставки оповещений; канал доставки |
| Оповещение | Сообщение о некотором событии, отправленное пользователю через выбранный канал доставки |
| Удаленная команда | Предопределенная команда, которая автоматически выполняется на наблюдаемом узле сети при наступлении некоторых условий |
| Шаблон | Набор объектов (элементов данных, триггеров, графиков, правил низкоуровневого обнаружения, веб-сценариев), готовых к применению на одном или нескольких узлах сети. Задача шаблонов – ускорить развертывание задач мониторинга узлов сети, а также упростить применение массовых изменений к задачам наблюдения. Шаблоны соединяются напрямую с отдельными узлами сети |
| Веб-сценарий | Один или несколько HTTP-запросов для проверки доступности веб-сайта |
| Веб-Интерфейс | поставляемый с веб-интерфейс. |
| Панель | Настраиваемый раздел веб-интерфейса, который отображает сводку и визуализацию важной информации в визуальных блоках, называемых виджетами |
| Виджет | Визуальный блок, который отображает информацию определенного вида и источника (сводка, карта, график, часы и т.д.), используется на панели |
| API | API позволяет использовать протокол JSON RPC для создания, обновления и получения объектов Подсистемы (например, узлов сети, элементов данных, графиков и прочих) или для выполнения любых других оригинальных задач. |
| Сервер | Центральный процесс программного обеспечения Подсистемы, который выполняет мониторинг, взаимодействует с Прокси и Агентами, вычисляет триггеры, отправляет оповещения; центральное хранилище данных |
| Прокси | Процесс, который может собирать данные от имени Сервера, перенимая часть нагрузки Сервера |
| Агент-1 | Процесс, разворачиваемый на наблюдаемых целях для активного мониторинга локальных ресурсов и приложений |
| Агент-2 | Агент нового поколения для активного мониторинга локальных ресурсов и приложений, который позволяет использовать пользовательские плагины для выполнения функций мониторинга.  Поскольку Агент-2 имеет во многом схожий функционал с Агентом-1, термин "Агент" в документации означает как Агента-1, так и Агента-2, если функционал не отличается. Агент-2 указывается отдельно в случае наличия различий в функционале |
| Шифрование | Поддержка шифрованной связи между компонентами Подсистемы (сервером, Прокси, Агентом и утилитами zabbix\_sender и zabbix\_get) с использованием протокола защиты транспортного уровня (TLS) |
| Обнаружение сети | Автоматическое обнаружение сетевых устройств. |
| Низкоуровневое обнаружение | Автоматическое обнаружение низкоуровневых объектов на конкретном устройстве (например, файловые системы, сетевые интерфейсы и т.д.). |
| Правило низкоуровневого обнаружения | Набор определений для автоматического обнаружения низкоуровневых объектов на устройстве |
| Прототип элементов данных | Метрика, у которой некоторые параметры представлены в виде переменных, готовых для низкоуровневого обнаружения. После низкоуровневого обнаружения эти переменные автоматически заменяются реальными обнаруженными параметрами, и метрика автоматически начинает сбор данных |
| Прототип триггеров | Триггер, у которого некоторые параметры представлены в виде переменных, готовых для низкоуровневого обнаружения. После низкоуровневого обнаружения эти переменные автоматически заменяются реальными обнаруженными параметрами и триггер автоматически начинает оценку данных.  Прототипы некоторых других объектов Подсистемы также используются в низкоуровневом обнаружении: прототипы графиков, прототипы узлов сети, прототипы групп узлов сети |
| Авторегистрация Агентов | Автоматизированный процесс, при котором Агент самостоятельно регистрируется в качестве узла сети и начинает наблюдаться |