

1 Zabbix агент

Обзор

Эти проверки используют общение с Zabbix агентом для сбора данных.

Существуют [пассивные](#) и [активные](#) проверки. При настройке элемента данных, вы можете выбрать требуемый тип:

- *Zabbix агент* - для пассивных проверок
- *Zabbix агент (активный)* - для активных проверок

Поддерживаемые ключи элементов данных

В таблице приводится подробная информация о ключах элементов данных, которые вы можете использовать элементами данных Zabbix агента.

Смотрите также:

- [Элементы данных поддерживаемые по платформам](#)
- [Специфичные ключи элементов данных для Windows агента](#)

Обязательные и необязательные параметры

Параметры без угловых скобок обязательны. Параметры, отмеченные угловыми скобками < >, необязательны.

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
agent.hostname			
Имя хоста агента.	Строка		Возвращает действительное значение hostname агента из файла конфигурации.
agent.ping			
Проверка доступности агента.	Ничего - недоступен 1 - доступен		Используйте nodata() функцию триггера для проверки недоступности узла сети.
agent.version			
Версия Zabbix агента.	Строка		Пример возвращаемого значения: 1.8.2
kernel.maxfiles			
Максимальное количество открытых файлов поддерживаемое ОС.	Целое число		
kernel.maxproc			
Максимальное количество процессов поддерживаемое ОС.	Целое число		
log[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Мониторинг файлов журналов (логов).	Журнал (лог)	<p>файл - абсолютный путь и имя файла журнала</p> <p>регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - идентификатор кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет отправлять Zabbix серверу или прокси. Этот параметр заменяет значение 'MaxLinesPerSecond' в zabbix_agentd.conf</p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только на недавно созданные элементы данных, которые еще не получили никаких данных).</p> <p>вывод - дополнительный шаблон форматирования вывода.</p> <p>\0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-ной совпадающей группой (или пустой строкой, если N превышает количество найденных групп).</p> <p>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 течении "максзадержка" секунд. Перед использованием прочитайте заметки по максзадержке!</p>	<p>Элемент данных должен быть настроен активной проверкой. Если файл не существует или доступ к нему не разрешен, элемент данных переходит в состояние неподдерживается.</p> <p>Если параметр вывод оставить пустым - будет возвращена вся строка соответствующая регулярному выражению. Заметьте, что все глобальные регулярные выражения исключая 'Результат ИСТИНА' всегда возвращают всю строку соответствующую выражению и параметр вывод игнорируется.</p> <p>Извлечение содержания согласно параметра вывод производится на стороне агента.</p> <p><i>Примеры:</i> ⇒ <code>log[/var/log/syslog]</code> ⇒ <code>log[/var/log/syslog.error]</code> ⇒ <code>log[/home/zabbix/logs/logfile,,100]</code></p> <p><i>Пример использования параметра вывод для извлечения числа из записи в журнале:</i> <code>log[/app1/app.log,"task run [0-9]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,1]→</code> будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится только число 6080. Так как отправляется число, у этого элемента данных журнала можно изменить "Тип информации" с "Журнал (лог)" на "Числовой (целое положительное)" и это значение можно будет использовать на на графиках, в триггерах и т.д.</p> <p><i>Пример использования параметра вывод для изменения вывода записи журнала перед отправкой на сервер:</i> <code>log[/app1/app.log,"([0-9]+) task run ([0-9]+) sec, processed ([0-9]+) records, ([0-9]+) errors",,,,"1 RECORDS: \3, ERRORS: \4, DURATION: \2"]→</code> будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится измененная запись "2015-11-13 10:08:26 RECORDS: 6080, ERRORS: 0, DURATION: 6.08".</p> <p>Параметр режим поддерживается начиная с Zabbix 2.0. Параметр вывод поддерживается начиная с Zabbix 2.2. Параметр максзадержка поддерживается начиная с Zabbix 3.2.</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p>
log.count[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>]			
Подсчёт количества совпадающих строк в мониторинге файла журнала (логов).	Целое число	<p>файл - абсолютный путь и имя файла журнала</p> <p>регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - идентификатор кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет анализировать. Значением по умолчанию является <code>10*MaxLinesPerSecond</code> в zabbix_agentd.conf.</p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только на недавно созданные элементы данных).</p> <p>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 течении "максзадержка" секунд. Перед использованием прочитайте заметки по максзадержке!</p>	<p>Элемент данных должен быть настроен активной проверкой. Если файл не существует или доступ к нему не разрешен, элемент данных переходит в состояние неподдерживается.</p> <p>Также смотрите дополнительную информацию о мониторинге файлов журналов.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p>
logrt[файл_regexp,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>,<опции>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Мониторинг файлов журналов (логов) с поддержкой ротации логов.	Журнал (лог)	<p>файл_regex - абсолютный путь к файлу и регулярное выражение описывающее его имя</p> <p>регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - идентификатор кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет отправлять Zabbix серверу или прокси. Этот параметр заменяет значение 'MaxLinesPerSecond' в <code>zabbix_agentd.conf</code></p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только на недавно созданные элементы данных, которые еще не получили никаких данных). вывод - дополнительный шаблон форматирования вывода: <i>\0</i> - управляющая последовательность заменяется найденным текстом, тогда <i>\N</i> (где N=1..9) - управляющая последовательность заменяется N-ной 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 течении "максзадержка" секунд. Перед использованием прочитайте заметки по максзадержка! опции - тип ротации файлов журналов. Возможные значения: <i>rotate</i> (по умолчанию), <i>copytruncate</i>. Обратите внимание, что <i>copytruncate</i> нельзя использовать вместе с <i>maxdelay</i>. В этом случае <i>maxdelay</i> должен быть равен 0 или не задан. Смотрите заметки по <i>copytruncate</i>.</p>	<p>Элемент данных должен быть настроен активной проверкой. Ротация журналов основывается на времени последнего изменения файлов.</p> <p>Если параметр вывод оставить пустым - будет возвращена вся строка соответствующая регулярному выражению. Заметьте, что все глобальные регулярные выражения исключая 'Результат ИСТИНА' всегда возвращают всю строку соответствующую выражению и параметр вывод игнорируется.</p> <p>Извлечение содержимого согласно параметра вывод производится на стороне агента.</p> <p><i>Примеры:</i> ⇒ <code>logrt["/home/zabbix/logs/^logfile[0-9]{1,3}\$",,,,100]</code> → совпадает с файлом наподобии "logfile1" (но не совпадает с ".logfile1") ⇒ <code>logrt["/home/user/logfile_*_[0-9]{1,3}",,"pattern_to_match","UTF-8",100]</code> → будет собирать данные с файлов таких как "logfile_abc_1" или "logfile_001".</p> <p><i>Пример использования параметра вывод для извлечения числа из записи в журнале:</i> <code>logrt[/app1/^test.*log\$,,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,1]</code> → будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится только число 6080. Так как отправляется число, у этого элемента данных журнала можно изменить "Тип информации" с "Журнал (лог)" на "Числовой (целое положительное)" и это значение можно будет использовать на на графиках, в триггерах и т.д.</p> <p><i>Пример использования параметра вывод для изменения вывода записи журнала перед отправкой на сервер:</i> <code>logrt[/app1/^test.*log\$,,"([0-9 -:]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",,,,,"1 RECORDS: \3, ERRORS: \4, DURATION: \2"]</code> → будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится измененная запись "2015-11-13 10:08:26 RECORDS: 6080, ERRORS: 0, DURATION: 6.08".</p> <p>Параметр режим поддерживается начиная с Zabbix 2.0. Параметр вывод поддерживается начиная с Zabbix 2.2. Параметр максзадержка поддерживается начиная с Zabbix 3.2. Параметр опции поддерживается начиная с Zabbix 4.0.</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p>
logrt.count[файл_regex,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>,<опции>]			
Подсчёт количества совпадающих строк в мониторинге файла журнала (логов) с поддержкой ротации логов.	Целое число	<p>файл_regex - абсолютный путь к файлу и регулярное выражение описывающее его имя</p> <p>регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - идентификатор кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет анализировать. Значением по умолчанию является <code>10*MaxLinesPerSecond</code> в <code>zabbix_agentd.conf</code>.</p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - 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 течении "максзадержка" секунд. Перед использованием прочитайте заметки по максзадержка! опции - тип ротации файлов журналов. Возможные значения: <i>rotate</i> (по умолчанию), <i>copytruncate</i>. Обратите внимание, что <i>copytruncate</i> нельзя использовать вместе с <i>maxdelay</i>. В этом случае <i>maxdelay</i> должен быть равен 0 или не задан. Смотрите заметки по <i>copytruncate</i>.</p>	<p>Элемент данных должен быть настроен активной проверкой. Ротация журналов основывается на времени последнего изменения файлов.</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p> <p>Параметр опции поддерживается начиная с Zabbix 4.0.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p>
net.dns[<ip>,<имя>,<тип>,<время ожидания>,<количество>,<протокол>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Проверка, работает ли DNS сервис.	0 - DNS не работает (сервер не ответил или ответ DNS некорректен) 1 - DNS работает	ip - IP адрес DNS сервера (оставьте пустым, чтобы использовать DNS сервера по умолчанию, игнорируется в Windows) имя - проверяемое имя DNS тип - запрашиваемый тип записи (по умолчанию SOA) время ожидания (игнорируется в Windows) - время ожидания ответа в секундах (по умолчанию 1 секунда) количество (игнорируется в Windows) - количество попыток запросов (по умолчанию 2) протокол - используемый протокол при выполнении DNS запросов: <i>udp</i> (по умолчанию) или <i>tcp</i>	<i>Пример ключа:</i> ⇒ net.dns[8.8.8.8,zabbix.com,MX,2,1] Возможные значения для тип : ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS (исключая Windows), HINFO, MINFO, TXT, SRV Национальные доменные имена не поддерживаются вместо них, пожалуйста, используйте имена в IDNA кодировке. Параметр протокол поддерживается начиная с Zabbix 3.0. Тип записи SRV поддерживается Zabbix агентом начиная с версий 1.8.6 (Unix) и 2.0.0 (Windows). Название до Zabbix 2.0 (еще поддерживается): <i>net.tcp.dns</i>
net.dns.record[<ip>, <имя>, <тип>, <время ожидания>, <количество>, <протокол>]			
Выполнение DNS запроса.	Строка символов с требуемым типом информации	ip - IP адрес DNS сервера (оставьте пустым, чтобы использовать DNS сервера по умолчанию, игнорируется в Windows) имя - проверяемое имя DNS тип - запрашиваемый тип записи (по умолчанию SOA) время ожидания (игнорируется в Windows) - время ожидания ответа в секундах (по умолчанию 1 секунда) количество (игнорируется в Windows) - количество попыток запросов (по умолчанию 2) протокол - используемый протокол при выполнении DNS запросов: <i>udp</i> (по умолчанию) или <i>tcp</i>	<i>Пример ключа:</i> ⇒ net.dns.record[8.8.8.8,zabbix.com,MX,2,1] Возможные значения для тип : ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS (исключая Windows), HINFO, MINFO, TXT, SRV Национальные доменные имена не поддерживаются вместо них, пожалуйста, используйте имена в IDNA кодировке. Параметр протокол поддерживается начиная с Zabbix 3.0. Тип записи SRV поддерживается в Zabbix агенте начиная с версий 1.8.6 (Unix) и 2.0.0 (Windows). Название до Zabbix 2.0 (еще поддерживается): <i>net.tcp.dns.query</i>
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		Поддерживается Zabbix агентом начиная с версии 2.0. На FreeBSD, OpenBSD и NetBSD поддерживается начиная с версии Zabbix агента 2.2 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 (Windows) режим - возможные значения: <i>bytes</i> - количество байт (по умолчанию) <i>packets</i> - количество пакетов <i>errors</i> - количество ошибок <i>dropped</i> - количество отброшенных пакетов <i>overruns (fifo)</i> - количество ошибок FIFO буфера <i>frame</i> - количество ошибок кадрирования пакетов <i>compressed</i> - количество сжатых пакетов, полученных драйвером устройства <i>multicast</i> - количество кадров многоадресного вещания, которые получены драйвером устройства	В Windows элемент данных берет значения с 64-битный счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битный счетчики недоступны, агент будет использовать 32-битные счетчики. Начиная с Zabbix агента 1.8.6 на Windows поддерживаются мультибайтные имена интерфейса. <i>Примеры:</i> ⇒ net.if.in[eth0,errors] ⇒ net.if.in[eth0] Вы можете получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list. Вы можете использовать этот ключ с шагом преобработки <i>Изменение в секунду</i> для получения статистики байт 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 (Windows) режим - возможные значения: <i>bytes</i> - количество байт (по умолчанию) <i>packets</i> - количество пакетов <i>errors</i> - количество ошибок <i>dropped</i> - количество отброшенных пакетов <i>overruns (fifo)</i> - количество ошибок FIFO буфера <i>collisions (cols)</i> - количество коллизий обнаруженных на интерфейсе <i>carrier</i> - количество потерь несущей, обнаруженных драйвером устройства <i>compressed</i> - количество сжатых пакетов, переданных драйвером устройства	В Windows элемент данных берет значения с 64-битный счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битный счетчики недоступны, агент будет использовать 32-битные счетчики. Начиная с Zabbix агента 1.8.6 на Windows поддерживаются мультибайтные имена интерфейса. <i>Примеры:</i> ⇒ net.if.out[eth0,errors] ⇒ net.if.out[eth0] Вы можете получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list. Вы можете использовать этот ключ с шагом преобработки <i>Изменение в секунду</i> для получения статистики байт в секунду.
net.if.total[if, <режим>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Суммарная статистика входящего и исходящего трафика на сетевом интерфейсе.	Целое число	if - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4 адрес (Windows) режим - возможные значения: bytes - количество байт (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 с 64-битный счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битный счетчики недоступны, агент будет использовать 32-битные счетчики. <i>Примеры:</i> ⇒ net.if.total[eth0_errors] ⇒ net.if.total[eth0] Вы можете получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list. Вы можете использовать этот ключ с шагом преобразования <i>Изменение в секунду</i> для получения статистики байт в секунду. Обратите 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	Пример: ⇒ net.tcp.listen[80] В Linux поддерживается Zabbix агентом начиная с версии 1.8.4 Начиная с Zabbix 3.0.0, в ядрах Linux 2.6.14 и более новых, информация о TCP сокетах в состоянии listen берется из интерфейса ядра NETLINK, если возможно. В противном случае, информация берется из файлов /proc/net/tcp и /proc/net/tcp6.
net.tcp.port[<ip>,порт]			
Проверка возможности создания TCP подключения на указанный номер порта.	0 - не удалось подключиться 1 - удалось подключиться	ip - IP адрес (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 используйте net.tcp.service.perf[tcp,<ip>,<порт>] Обратите внимание, что эти проверки могут привести к дополнительным записям в системных файлах журналов (обычно сессии SMTP и SSH журналируются). Старое название: check_port[*]
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: <i>ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet</i> (смотри детали) ip - IP адрес (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. Обратите 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. Как решение, пожалуйста, для подобных проверок используйте net.tcp.port. Проверка LDAP и HTTPS через Windows агента в настоящее время не поддерживается. Обратите внимание, что telnet проверка ищет запрос на вход (с ':' в конце). Смотрите также известные проблемы по проверке HTTPS сервиса. Сервисы <i>https</i> и <i>telnet</i> поддерживается Zabbix с версии 2.0. Старое название: check_service[*]
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: <i>ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet</i> (смотри детали) ip - IP адрес (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. Как решение, пожалуйста, для подобных проверок используйте net.tcp.service.perf[tcp,<ip>,<порт>]. Проверка LDAP и HTTPS через Windows агента в настоящее время не поддерживается. Обратите внимание, что telnet проверка ищет запрос на вход (с ':' в конце). Смотрите также известные проблемы по проверке HTTPS сервиса. Сервисы <i>https</i> и <i>telnet</i> поддерживается Zabbix с версии 2.0. Старое название: check_service_perf[*]
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] В Linux поддерживается Zabbix агентом начиная с версии 1.8.4
net.udp.service[сервис,<ip>,<порт>]			
Проверка, запущен ли сервис и принимает ли он UDP подключения.	0 - сервис недоступен 1 - сервис запущен	сервис - <i>ntp</i> (смотри детали) ip - IP адрес (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. Этот элемент данных поддерживается начиная с Zabbix 3.0.0, но <i>ntp</i> сервис был доступен в net.tcp.service[] элементе данных и в предыдущих версиях.
net.udp.service.perf[сервис,<ip>,<порт>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Проверка производительности UDP сервиса.	0 - сервис недоступен секунды - количество секунд потраченное на подключение к сервису	сервис - <i>ntp</i> (смотри детали) ip - IP адрес (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 ответа от NTP сервиса. Этот элемент данных поддерживается начиная с Zabbix 3.0.0, но <i>ntp</i> сервис был доступен в net.tcp.service[] элементе данных и в предыдущих версиях.
proc.cpu.util[<имя>,<пользователь>,<тип>,<cmdline>,<режим>,<зона>]			
Использование CPU процесса в процентах.	Число с плавающей точкой	имя - имя процесса (по умолчанию <i>все процессы</i>) пользователь - имя пользователя (по умолчанию <i>все пользователи</i>) тип - тип использования CPU: <i>total</i> (по умолчанию), <i>user</i> , <i>system</i> cmdline - фильтр по командной строке (является регулярным выражением) режим - режим сбора данных: <i>avg1</i> (по умолчанию), <i>avg5</i> , <i>avg15</i> зона - целевая зона: <i>current</i> (по умолчанию), <i>all</i> . Этот параметр поддерживается только на Solaris платформе.	Примеры: ⇒ proc.cpu.util[<i>root</i>] → Использование CPU по всем процессам работающими под " <i>root</i> " пользователем ⇒ 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 полного использования двух ядер равна 200%. Данные использования CPU собираются коллекторами, которые поддерживают максимально до 1024 уникальных (по имени, пользователю и командной строке) запросов. Запросы, к которым не было обращения в течении последних 24 часов, удаляются из коллектора. <i>Обратите внимание</i> , что в случае когда агент скомпилирован на Solaris без поддержки зон, если задать параметр зона значением <i>current</i> (или по умолчанию), и запустить его на более новом Solaris, где зоны поддерживаются, тогда агент вернёт NOTSUPPORTED (агент не может ограничить результаты в пределах только текущей зоны). Однако, значение <i>all</i> в этом случае поддерживается. Этот ключ поддерживается начиная с Zabbix 3.0.0 и также доступен на нескольких платформах (смотри Поддерживаемые элементы данных по платформам).
proc.mem[<имя>,<пользователь>,<режим>,<cmdline>,<тип памяти>]			
Количество памяти используемое процессом в байтах.	Целое число	имя - имя процесса (по умолчанию <i>все процессы</i>) пользователь - имя пользователя (по умолчанию <i>все пользователи</i>) режим - возможные значения: <i>avg</i> , <i>max</i> , <i>min</i> , <i>sum</i> (по умолчанию) cmdline - фильтр по командной строке (является регулярным выражением) тип памяти - тип памяти используемый процессом.	Примеры: ⇒ proc.mem[<i>root</i>] - память используемая всеми процессами запущенными под пользователем " <i>root</i> " ⇒ proc.mem[zabbix_server,zabbix] - память используемая всеми процессами zabbix_server запущенными под пользователем zabbix ⇒ proc.mem[oracle,max,oracleZABBIX] - максимальное значение используемой памяти процессами запущенными под пользователем oracle и имеющими oracleZABBIX в содержимом командной строки <i>Обратите внимание</i> : Когда несколько процессов используют разделяемую память, сумма памяти используемой процессами в результате может быть большой, нереальной величиной. Смотрите заметки по выбору процессов с параметрами имя и cmdline (специфика для Linux). Когда 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. Параметр тип памяти поддерживается на нескольких платформах начиная с Zabbix 3.0.0.
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: <i>all</i> (по умолчанию), <i>disk</i> - непрерывный сон, <i>run</i> - в процессе работы, <i>sleep</i> - прерываемый сон, <i>trace</i> - остановлен, <i>zomb</i> - зомби cmdline - фильтр по командной строке (является регулярным выражением) зона - целевая зона: <i>current</i> (по умолчанию), <i>all</i> . Этот параметр поддерживается только на Solaris платформе.	Примеры ключей: ⇒ proc.num[mysql] - количество процессов выполняемых под пользователем mysql ⇒ proc.num[apache2,www-data] - количество процессов apache2 выполняемых под пользователем www-data ⇒ proc.num[oracle,sleep,oracleZABBIX] - количество процессов в спящем состоянии выполняемых под oracle и имеющих oracleZABBIX в содержимом командной строки Смотрите заметки по выбору процессов с параметрами имя и cmdline (специфика для Linux). В Windows, поддерживаются только параметры имя и пользователь. Когда 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. <i>Обратите внимание</i> , что в случае когда агент скомпилирован на Solaris без поддержки зон, если задать параметр зона значением <i>current</i> (или по умолчанию), и запустить его на более новом Solaris, где зоны поддерживаются, тогда агент вернёт NOTSUPPORTED (агент не может ограничить результаты в пределах только текущей зоны). Однако, значение <i>all</i> в этом случае поддерживается. Значения <i>disk</i> и <i>trace</i> для параметра state поддерживаются начиная с Zabbix 3.4.0.
sensor[устройство,сенсор,<режим>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Чтение аппаратного сенсора.	Число с плавающей точкой	устройство - имя устройства сенсор - имя сенсора режим - возможные значения: <i>avg, max, min</i> (если этот параметр не указан, то устройство и сенсор обрабатываются дословно).	В Linux 2.4, читается <code>/proc/sys/dev/sensors</code> . Пример: ⇒ <code>sensor[w83781d-i2c-0-2d,temp1]</code> До Zabbix 1.8.4, использовался формат <code>sensor[temp1]</code> . В Linux 2.6+, читается <code>/sys/class/hwmon</code> . Более подробное описание смотрите у элемента данных сенсора в Linux. В OpenBSD, читается MIB <code>hw.sensors</code> . Примеры: ⇒ <code>sensor[cpu0,temp0]</code> - температура одного CPU ⇒ <code>sensor[cpu[0-2]\$,temp,avg]</code> - средняя температура первых трех CPU Поддерживается в OpenBSD Zabbix агентом начиная с версии 1.8.4.
system.boottime			
Штамп времени (timestamp) загрузки системы.	Целое число (UNIX timestamp)		
system.cpu.discovery			
Список найденных CPU/CPU ядер. Используется низкоуровневым обнаружением.	Объект JSON		Поддерживается на всех платформах с 2.4.0.
system.cpu.intr			
Прерывания устройств.	Целое число		
system.cpu.load[<cpu>,<режим>]			
Загрузка CPU .	Число с плавающей точкой	cpu - возможные значения: <i>all</i> (по умолчанию), <i>percpu</i> (общая загрузка делится на количество CPU онлайн) режим - возможные значения: <i>avg1</i> (усреднение за одну минуту, по умолчанию), <i>avg5, avg15</i>	Пример: ⇒ <code>system.cpu.load[avg5]</code> Параметр <i>percpu</i> поддерживается начиная с Zabbix 2.0.0. Старое название: <code>system.cpu.loadX</code>
system.cpu.num[<тип>]			
Количество CPU.	Целое число	тип - возможные значения: <i>online</i> (по умолчанию), <i>max</i>	Пример: ⇒ <code>system.cpu.num</code>
system.cpu.switches			
Количество переключений контекста.	Целое число		Старое название: <code>system[switches]</code>
system.cpu.util[<cpu>,<тип>,<режим>]			
Утилизация CPU в процентах.	Число с плавающей точкой	cpu - номер CPU (по умолчанию все CPU) тип - возможные значения: <i>idle, nice, user</i> (по умолчанию), <i>system</i> (по умолчанию для Windows), <i>iowait, interrupt, softirq, steal, guest, nice</i> (на ядре Linux 2.6.24 и выше), <i>guest_nice</i> (на ядре Linux 2.6.33 и выше). Параметры времени <i>user</i> и <i>nice</i> более не включают время <i>guest</i> и время <i>guest_nice</i> начиная с Zabbix 3.0.14, 3.4.5 и 4.0.0. режим - возможные значения: <i>avg1</i> (усреднение за одну минуту, по умолчанию), <i>avg5, avg15</i>	Пример: ⇒ <code>system.cpu.util[0,user,avg5]</code> Старое название: <code>system.cpu.idleX, system.cpu.niceX, system.cpu.systemX, system.cpu.userX</code>
system.hostname[<тип>]			
Системное имя хоста.	Строка	тип (только для Windows, не должен использоваться на остальных системах) - возможные значения: <i>netbios</i> (по умолчанию) или <i>host</i>	Получение значения либо функцией <code>GetComputerName()</code> (для netbios), либо функцией <code>gethostname()</code> (для host) в Windows и с помощью команды "hostname" на других системах. Примеры возвращаемых значений: в Linux: ⇒ <code>system.hostname</code> → <code>linux-w7x1</code> ⇒ <code>system.hostname</code> → <code>www.zabbix.com</code> в Windows: ⇒ <code>system.hostname</code> → <code>WIN-SERV2008-I6</code> ⇒ <code>system.hostname[host]</code> → <code>Win-Serv2008-I6LonG</code> Параметр <code>тип</code> поддерживается этим элементом данных начиная с версии 1.8.6 . Смотрите также более детальную информацию .
system.hw.chassis[<информация>]			
Информация о шасси.	Строка	информация - одно из <code>full</code> (по умолчанию), <code>model</code> , <code>serial</code> , <code>type</code> или <code>vendor</code>	Пример: <code>system.hw.chassis[full]</code> <code>Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop</code> Этот ключ зависит от наличия SMBIOS таблицы. Будет выполнена попытка чтения из DMI таблицы с <code>sysfs</code> , если доступ к <code>sysfs</code> будет неудачным, тогда будет попытка чтения напрямую из памяти. Требуются Права root , потому что значение читается из <code>sysfs</code> или памяти. Поддерживается Zabbix агентом начиная с версии 2.0.
system.hw.cpu[<cpu>,<информация>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Информация о CPU.	Строка или целое число	cpu - номер CPU или all (по умолчанию) информация - возможные значения: full (по умолчанию), currfreq, 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 <i>currfreq</i> или <i>maxfreq</i> , то возвращается числовое значение (Гц). Поддерживается Zabbix агентом начиная с версии 2.0.
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) Поддерживается Zabbix агентом начиная с версии 2.0.
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 чьи имена интерфейсов совпадают с заданным в интерфейс регулярным выражением (<i>all</i> список всех интерфейсов). Пример: ⇒ system.hw.macaddr["eth0\$",full] ⇒ [eth0] 00:11:22:33:44:55 Если формат задан как <i>short</i> , имена интерфейсов и одинаковые MAC адреса не попадают в список. Поддерживается Zabbix агентом начиная с версии 2.0.
system.localtime[<тип>]			
Системное время.	Целое число - с тип как <i>utc</i> Строка - с тип как <i>local</i> .	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'	Параметры для этого элемента данных поддерживаются начиная с версии 2.0. Например: ⇒ system.localtime[local] → создайте элемент данных, используя этот ключ, и далее используйте его для отображения времени узла сети в элемента комплексного экрана Часы .
system.run[команда,<режим>]			
Выполнение указанной команды на узле сети.	Текст с результатом выполнения команды 1 - в режиме <i>lowait</i> (независимо от результата выполнения команды)	команда - выполняемая команда режим - один из wait (по умолчанию), ожидая завершения выполнения), <i>lowait</i> (не ожидать)	Возвращается до 512КБ (64КБ до Zabbix 2.0.5), включая пробелы в конце, которые отрезаются. Для корректной обработки, возвращаемые данные должны быть текстом. Пример: ⇒ system.run[ls -l /] → подробный список файлов в папке root. Обратите внимание: Для того чтобы включить этот функционал, в файле конфигурации агента должна быть указана опция <code>EnableRemoteCommands=1</code> . Возвращаемое значение элемента данных является стандартным выводом совместно со стандартным выводом ошибок, которые получены от команды. Код выхода не проверяется. Пустой результат разрешается начиная с 2.4.0. Смотрите также: Выполнение команд .
system.stat[источник,<тип>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Статистика системы.	Целое число или число с плавающей точкой	<p>ent - количество процессорных единиц выделенных на ресурс (с плавающей точкой)</p> <p>kthr, <тип> - информация о состояниях потоков ядра: <i>r</i> - среднее количество запускаемых потоков ядра (дробное) <i>b</i> - среднее количество потоков ядра помещенных в очередь ожидания Менеджера Виртуальной Памяти (с плавающей точкой)</p> <p>memory, <тип> - информация о использовании виртуальной и реальной памяти: <i>avm</i> - активные виртуальные страницы (целое) <i>fre</i> - размер свободного списка (целое)</p> <p>page, <тип> - информация об ошибках страниц и активности страниц: <i>fi</i> - операции страничного ввода файлов в секунду (с плавающей точкой) <i>fo</i> - операции страничного вывода файлов в секунду (с плавающей точкой) <i>pi</i> - pages paged in from paging space (с плавающей точкой) <i>po</i> - pages paged out to paging space (с плавающей точкой) <i>fr</i> - pages freed (page replacement) (с плавающей точкой) <i>sr</i> - количество сканированных страниц алгоритмом замещения страниц (с плавающей точкой)</p> <p>faults, <тип> - соотношение ловушек и прерываний: <i>in</i> - прерывания устройства (с плавающей точкой) <i>su</i> - системные вызовы (с плавающей точкой) <i>cs</i> - переключения контекста потока ядра (с плавающей точкой)</p> <p>cpu, <тип> - раздельное использование процессорного времени в процентах: <i>us</i> - пользовательское время (с плавающей точкой) <i>sy</i> - системное время (с плавающей точкой) <i>id</i> - время простоя (с плавающей точкой) <i>wa</i> - время простоя в ходе которого система обрабатывала запросы I/O диска/NFS (с плавающей точкой) <i>pc</i> - количество использованных физических процессоров (с плавающей точкой) <i>ec</i> - процентное соотношение размеченного занятого пространства (с плавающей точкой) <i>lbusy</i> - показывает утилизация в процентах для логических процессор(ов), которая произошла при выполнении уровней пользователя и системы (с плавающей точкой) <i>app</i> - отображает количество доступных физических процессоров в распределенном пуле (с плавающей точкой)</p> <p>disk, <тип> - статистика о дисках: <i>bps</i> - отображает суммарное количество данных переданных (чтением или записью) на диск в байтах в секунду (целое) <i>tps</i> - показывает количество передач за секунду, которые произошли на физическом диске/ленте (с плавающей точкой)</p> <p>Комментарии Этот элемент данных поддерживается только на AIX, начиная с Zabbix 1.8.1. Следующие элементы данных поддерживаются только на AIX LPAR с типом "Shared": ⇒ system.stat[cpu,app] ⇒ system.stat[cpu,ec] (также на "Dedicated" начиная с Zabbix 4.0.1; всегда возвращает 100 (процентов)) ⇒ system.stat[cpu,lbusy] ⇒ system.stat[cpu,pc] (также на "Dedicated" начиная с Zabbix 4.0.1) ⇒ system.stat[ent] (также на "Dedicated" начиная с Zabbix 4.0.1)</p>	
system.sw.arch			
Информация о архитектуре программного обеспечения.	Строка		<p>Пример: ⇒ system.sw.arch → i686</p> <p>Для получения информации используется функция uname().</p> <p>Поддерживается Zabbix агентом начиная с версии 2.0.</p>
system.sw.os[<информация>]			
Информация об операционной системе.	Строка	информация - возможные значения: full (по умолчанию), short или name	<p>Пример: ⇒ system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11</p> <p>Информация получается с (обратите внимание, на то 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)</p> <p>Поддерживается Zabbix агентом начиная с версии 2.0.</p>
system.sw.packages[<пакет>,<менеджер>,<формат>]			
Список установленных пакетов.	Текст	<p>пакет - <i>all</i> (по умолчанию) или регулярное выражение менеджер - <i>all</i> (по умолчанию) или конкретный менеджер пакетов формат - full (по умолчанию) или short</p>	<p>Список (в алфавитном порядке) установленных пакетов имена которых совпадают с регулярным выражением заданным в пакет (<i>all</i> списки всех).</p> <p>Пример: ⇒ system.sw.packages[mini,dpkg,short] ⇒ python-minimal, python2.6-minimal, ubuntu-minimal</p> <p>Поддерживаемые менеджеры пакетов (выполняемые команды): dpkg (dpkg --get-selections) pkgtool (ls /var/log/packages) rpm (rpm -qa) pacman (pacman -Q)</p> <p>Если указан формат равный <i>full</i>, пакеты группируются по менеджерам пакетов (каждый менеджер с отдельной строки, которая начинается с его имени заключенном в квадратные скобки). Если указан формат равный <i>short</i>, пакеты не группируются и отображаются одной строкой.</p> <p>Поддерживается Zabbix агентом начиная с версии 2.0.</p>
system.swap.in[<устройство>,<тип>]			
Статистика файла подкачки в (с устройства в память).	Целое число	<p>устройство - устройство используемое для файла подкачки (по умолчанию <i>all</i>) тип - возможные значения: <i>count</i> (количество swarps), <i>sectors</i> (секторов swapped in), <i>pages</i> (страниц swapped in). Смотрите поддержку по платформам для получения деталей по умолчанию.</p>	<p>Пример: ⇒ system.swap.in[,pages]</p> <p>Источником этой информации является: /proc/swaps, /proc/partitions, /proc/stat (Linux 2.4) /proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)</p>
system.swap.out[<устройство>,<тип>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Статистика файла подкачки из (из памяти в устройство).	Целое число.	устройство - устройство используемое для файла подкачки (по умолчанию <i>all</i>) тип - возможные значения: <i>count</i> (количество swapouts), <i>sectors</i> (секторов swapped out), <i>pages</i> (страниц swapped out). Смотрите поддержку по платформам для получения деталей по умолчанию.	Пример: ⇒ <code>system.swap.out[,pages]</code> Источником этой информации является: <code>/proc/swaps</code> , <code>/proc/partitions</code> , <code>/proc/stat</code> (Linux 2.4) <code>/proc/swaps</code> , <code>/proc/diskstats</code> , <code>/proc/vmstat</code> (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 <i>all</i>) тип - возможные значения: <i>free</i> (свободно в файле подкачки, по умолчанию), <i>pfree</i> (свободно в файле подкачки, в процентах), <i>used</i> (используемое место в файле подкачки, в процентах), <i>total</i> (полный размер файла подкачки), <i>used</i> (используется места в файле подкачки)	Пример: ⇒ <code>system.swap.size[,pfree]</code> → свободно места в файле подкачки в процентах Если <i>устройство</i> не указано, Zabbix агент будет брать во внимание только swap устройства (файлы), физическая память будет игнорироваться. Например, на Solaris системах команда <code>swap -s</code> включает часть физической памяти и swap устройства (в отличие от <code>swap -l</code>). Обратите внимание, этот ключ может возвращать некорректные данные на виртуализированных (VMware ESXi, VirtualBox) Windows платформах. В этом случае используйте ключ <code>perf_counter[\700(Total)\702]</code> для получения корректного использования файла подкачки в процентах. Старое название: <code>system.swap.free</code> , <code>system.swap.total</code>
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 начиная с Zabbix 2.2.0 значение этого элемента извлекается при помощи системного вызова <code>uname()</code> . Ранее эта информация запрашивалась с помощью системного вызова <code>"uname -a"</code> . Значение этого элемента данных может отличаться от вывода <code>"uname -a"</code> и может не включать дополнительную информацию, так как <code>"uname -a"</code> выводит информацию основываясь на других источниках. В Windows начиная с Zabbix 3.0 значение этого элемента данных извлекается при помощи <code>Win32_OperatingSystem</code> и <code>Win32_Processor</code> классов WMI. 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. Обратите внимание, что этот элемент данных в Windows возвращает архитектуру ОС, тогда как в Unix он возвращает архитектуру CPU.
system.uptime			
Время работы в секундах.	Целое число		При настройке элемент данных , используйте единицы измерения s или uptime для получения читаемых значений.
system.users.num			
Количество пользователей находящихся в системе.	Целое число		Для получения значений используется команда <code>who</code> на стороне агента.
vfs.dev.read[<устройство>,<тип>,<режим>]			
Статистика чтения диска.	Целое число при тип равном: <i>sectors</i> , <i>operations</i> , <i>bytes</i> Число с плавающей точкой при тип равном: <i>sps</i> , <i>ops</i> , <i>bps</i>	устройство - дисковое устройство (по умолчанию <i>all</i>) тип - возможные значения: <i>sectors</i> , <i>operations</i> , <i>bytes</i> , <i>sps</i> , <i>ops</i> , <i>bps</i> Этот параметр необходимо указывать, т.к. умолчания отличаются для разных ОС. <i>sps</i> , <i>ops</i> , <i>bps</i> соответствуют: секторам, операциям, байтам в секунду соответственно режим - возможные значения: <i>avg1</i> (усреднение за минуту, по умолчанию), <i>avg5</i> (усреднение за 5 минут), <i>avg15</i> (усреднение за 15 минут). Третий параметр поддерживается только, если тип один из: <i>sps</i> , <i>ops</i> , <i>bps</i> .	Значения по умолчанию параметра 'Тип' для различных ОС: AIX - operations FreeBSD - bps Linux - sps OpenBSD - operations Solaris - bytes Пример: ⇒ <code>vfs.dev.read[,operations]</code> <i>ops</i> , <i>bps</i> и <i>sps</i> на поддерживаемых платформах ограничено 8 устройствами (7 отдельных устройств и одно <i>all</i>). Начиная с Zabbix 2.0.1 этот лимит увеличен до 1024 (1023 отдельных устройств и один для <i>all</i>). По умолчанию используется <i>all</i> первым параметром, который возвращает суммарную статистику, включая: все блочные устройства такие как <i>sda</i> , <i>sdb</i> и их разделы <i>sda1</i> , <i>sda2</i> , <i>sdb3</i> ... и несколько устройств (MD raid) на основе этих блочных устройств/разделов и логические разделы (LVM) на основе этих блочных устройств/разделов. В некоторых случаях возвращаемые значения следует рассматривать как относительные значения (изменяемые во времени), но не как абсолютные значения. LVM поддерживается начиная с Zabbix 1.8.6. До Zabbix 1.8.6, можно было использовать только относительные имена устройств (например, <i>sda</i>), начиная с 1.8.6 можно использовать опциональный префикс <i>/dev/</i> (например, <i>/dev/sda</i>) Старое название: <code>i[*]</code>
vfs.dev.write[<устройство>,<тип>,<режим>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Статистика записи на диск.	Целое число при тип равном: <i>sectors, operations, bytes</i> Число с плавающей точкой при тип равном: <i>sps, ops, bps</i>	устройство - дисковое устройство (по умолчанию <i>all</i>) тип - возможные значения: <i>sectors, operations, bytes, sps, ops, bps</i> Этот параметр необходимо указывать, т.к. умолчания отличаются для разных ОС. <i>sps, ops, bps</i> соответствуют: секторам, операциям, байтам в секунду соответственно режим - возможные значения: <i>avg1</i> (усреднение за минуту, по умолчанию), <i>avg5</i> (усреднение за 5 минут), <i>avg15</i> (усреднение за 15 минут). Третий параметр поддерживается только если тип один из: <i>sps, ops, bps</i> .	Значения по умолчанию параметра 'Тип' для различных ОС: AIX - operations FreeBSD - bps Linux - sps OpenBSD - operations Solaris - bytes Пример: ⇒ <code>vfs.dev.write[,operations]</code> <i>ops, bps</i> и <i>sps</i> на поддерживаемых платформах ограничено 8 устройствами (7 отдельных устройств и одно <i>all</i>). Начиная с Zabbix 2.0.1 этот лимит увеличен до 1024 (1023 отдельных устройств и один для <i>all</i>). По умолчанию используется <i>all</i> первым параметром, который возвращает суммарную статистику, включая: все блочные устройства такие как <i>sda, sdb</i> и их разделы <i>sda1, sda2, sdb3</i> ... и несколько устройств (MD raid) на основе этих блочных устройств/разделов и логические разделы (LVM) на основе этих блочных устройств/разделов. В некоторых случаях возвращаемые значения следует рассматривать как относительные значения (изменяемые во времени), но не как абсолютные значения. LVM поддерживается начиная с Zabbix 1.8.6. До Zabbix 1.8.6, можно было использовать только относительные имена устройств (например, sda), начиная с 1.8.6 можно использовать опциональный префикс /dev/ (например, /dev/sda) Старое название: <code>io[*]</code>

vfs.dir.count[директория,<regex_вкл>,<regex_искл>,<типы_вкл>,<типы_искл>,<макс_глубина>,<мин_размер>,<макс_размер>,<мин_возраст>,<макс_возраст>]

Количество записей в директории.	Целое число	директория - абсолютный путь к директории regex_вкл - регулярное выражение описывающее включаемые файл, директорию и символическую ссылку (если пусто, включаются все файлы, директории и символические ссылки; пустая строка является значением по умолчанию) regex_искл - регулярное выражение описывающее исключаемые файл, директорию и символическую ссылку (если пусто, ничего не исключать; пустая строка является значением по умолчанию) типы_вкл - перечисление типов записей в директории, которые будут подсчитываться, возможные значения: <i>file</i> - обычный файл, <i>dir</i> - поддиректория, <i>sym</i> - символическая ссылка, <i>sock</i> - сокет, <i>bdev</i> - блочное устройство, <i>cdev</i> - устройство посимвольного ввода-вывода, <i>fifo</i> - FIFO, <i>dev</i> - синонимы к " <i>bdev,cdev</i> ", <i>all</i> - все вышеупомянутые типы, т.е. " <i>file,dir,sym,sock,bdev,cdev,fifo</i> ". Это значение по умолчанию, если параметр оставлен пустым. Несколько типов необходимо разделять запятой и весь список нужно заключать в кавычки "" типы_искл - перечисление типов записей в директории, которые НЕ будут подсчитываться, те же самые значения и синтаксис что и для <типы_вкл>. Если некоторые типы записей указаны как в <типы_вкл>, так и в <типы_искл>, записи этих типов в директории НЕ будут подсчитываться. макс_глубина - максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) - без ограничения, 0 - без проверки нижестоящих поддиректорий. мин_размер - минимальный размер подсчитываемых файлов. Файлы, которые меньше этого значения не будут подсчитаны. Значение в байтах. Можно использовать суффиксы памяти . макс_размер - максимальный размер подсчитываемых файлов. Файлы, которые больше этого значения не будут подсчитаны. Значение в байтах. Можно использовать суффиксы памяти . мин_возраст - минимальный возраст подсчитываемой записи директории. Записи, которые изменены раньше не будут подсчитаны. Целочисленное значение в виде секунд. Можно использовать суффиксы времени . макс_возраст - максимальный возраст подсчитываемой записи директории. Записи, которые настолько старые не будут подсчитаны (время модификации). Целочисленное значение в виде секунд. Можно использовать суффиксы времени .	Переменные сред, такие как <code>%APP_HOME%</code> , <code>\$HOME</code> и <code>%TEMP%</code> , не поддерживаются. Псевдо-директории "." и ".." никогда не учитываются. Символические ссылки никогда проверяются для обхода директорий. В Windows символические ссылки директорий пропускаются и жесткие ссылки учитываются лишь один раз. <i>regex_incl</i> и <i>regex_excl</i> являются Perl совместимыми регулярными выражениями (PCRE). Оба параметра <i>regex_вкл</i> и <i>regex_искл</i> применяются к файлам и директориям при вычислении общего размера, но игнорируются при выборе поддиректорий (если <i>regex_вкл</i> равен "(?i)^\.zip\$" и <i>макс_глубина</i> не задана, тогда проход будет выполнен по всем поддиректориям, но учтены будут только файлы с типом zip). Если имя файла совпадает как с <i>regex_вкл</i> , так и с <i>regex_искл</i> , такой файл не будет учитываться. Время выполнения ограничено значением времени ожидания по умолчанию, которое равно 3 секундам (параметр "Timeout" в файле конфигурации агента). Та как обход большой директории может занять более длительное время чем время ожидания, тогда никакие данные не возвратятся и элемент данных отметится как "Не поддерживается". Частичный подсчет не возвращается. При фильтрации по размеру только обычные файлы имеют осмысленные размеры. В Linux и BSD директории также имеют не нулевые размеры (в основном несколько Кб). Устройства имеют нулевые размеры, например, размер /dev/sda1 не влияет на соответствующий размер раздела. Поэтому, при использовании параметров <мин_размер> и <макс_размер> рекомендуется указать <типы_вкл> равным "file", чтобы избежать сюрпризов. Примеры: ⇒ <code>vfs.dir.count[/dev]</code> - мониторинг количества устройств в /dev (Linux) ⇒ <code>vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"]</code> - мониторинг количества файлов во временной директории (Windows) Поддерживается начина с Zabbix 4.0.0.
----------------------------------	-------------	--	---

vfs.dir.size[директория,<regex_вкл>,<regex_искл>,<режим>,<макс_глубина>]

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Размер директории (в байтах).	Целое число	<p>директория - абсолютный путь к директории</p> <p>gegex_вкл - регулярное выражение описывающее включаемые файл, директорию и символическую ссылку (если пусто, включаются все файлы, директорию и символические ссылки; пустая строка является значением по умолчанию)</p> <p>gegex_искл - регулярное выражение описывающее исключаемые файл, директорию и символическую ссылку (если пусто, ничего не исключать; пустая строка является значением по умолчанию)</p> <p>режим - возможные значения: <i>apparent</i> (по умолчанию) - получение реальных размеров файлов вместо использования диска (работает как <code>du -sb</code> директория), <i>disk</i> - получение использования диска (работает как <code>du -s -B1</code> директория). В отличие от <code>du</code> команды, элемент данных <code>vfs.dir.size</code> принимает во внимание скрытые файлы при вычислении размера директории (работает как <code>du -sb .[^.]* *</code> в пределах директории).</p> <p>макс_глубина - максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) - без ограничения, 0 - без проверки нижестоящих поддиректорий.</p>	<p>Подсчитываются только те директории к которым <i>zabbix</i> пользователь имеет по крайней мере права чтения.</p> <p>В Windows любая символическая ссылка пропускается и жесткие ссылки принимаются во внимание только один раз.</p> <p>При наличии больших директорий или медленных дисков этот элемент данных может превысить время ожидания из-за настройки <code>Timeout</code> в файлах конфигурации агента и сервера/прокси. При необходимости увеличьте эти значения времени ожидания.</p> <p>Примеры: <code>⇒ vfs.dir.size[/tmp,log]</code> - вычисление размеров всех файлов в /tmp, которые содержат в своём имени 'log' <code>⇒ vfs.dir.size[/tmp,log,^.\.old\$]</code> - вычисление размеров всех файлов в /tmp, которые содержат в своём имени 'log', исключая файлы содержащие в своём имени '.old'</p> <p>Ограничение размера файлов зависит от наличия поддержки больших файлов.</p> <p>Поддерживается начиная с Zabbix 3.4.0.</p>
vfs.file.cksum[файл]			
Контрольная сумма файла, вычисленная по алгоритму используемому в UNIX cksum.	Целое число	файл - абсолютный путь к файлу	<p>Пример: <code>⇒ vfs.file.cksum[/etc/passwd]</code></p> <p>Пример возвращаемого значения: 1938292000</p> <p>Старое название: cksum</p> <p>Ограничение размера файла зависит от поддержки больших файлов.</p>
vfs.file.contents[файл,<кодировка>]			
Получение содержимого файла.	Текст	файл - абсолютный путь к файлу кодировка - идентификатор кодовой страницы	<p>Возвращает пустую строку, если файл пустой или содержит только LF/CR символы.</p> <p>Пример: <code>⇒ vfs.file.contents[/etc/passwd]</code></p> <p>Этот элемент данных ограничен файлами не превышающими 64 КБайт.</p> <p>Поддерживается Zabbix агентом начиная с версии 2.0.</p>
vfs.file.exists[файл]			
Проверка существования файла	0 - файл не найден 1 - обычный файл или ссылка (символическая или жесткая) на обычный файл существует.	файл - абсолютный путь к файлу	<p>Пример: <code>⇒ vfs.file.exists[/tmp/application.pid]</code></p> <p>Возвращаемое значение зависит от того, что вернет <code>S_ISREG</code> POSIX макрос.</p> <p>Ограничение размера файла зависит от поддержки больших файлов.</p>
vfs.file.md5sum[файл]			
MD5 контрольная сумма файла	Строка (MD5 хэш файла)	файл - абсолютный путь к файлу	<p>Пример: <code>⇒ vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]</code></p> <p>Пример возвращаемого значения: b5052decb577e0fffd622d6ddc017e82</p> <p>Ограничение (64МБ) на размер файла для этого элемента данных удалено в версии 1.8.6.</p> <p>Ограничение размера файла зависит от поддержки больших файлов.</p>
vfs.file.regexp[файл,регулярное выражение,<кодировка>,<начальная строка>,<конечная строка>,<вывод>]			
Поиск строки в файле.	Строка содержащая подстроку или то, что определено в дополнительном параметре вывод.	<p>файл - абсолютный путь к файлу</p> <p>регулярное выражение - <i>Perl</i> совместимое регулярное выражение (PCRE) или расширенное POSIX регулярное выражение до Zabbix 3.4</p> <p>кодировка - идентификатор кодовой страницы</p> <p>начальная строка - номер первой строки для поиска (по умолчанию, первая строка файла).</p> <p>конечная строка - номер последней строки для поиска (по умолчанию, последняя строка файла).</p> <p>вывод - дополнительный шаблон форматирования вывода.</p> <p>\0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-ной совпадающей группой (или пустой строкой, если N превышает количество найденных групп).</p>	<p>Возвращается только первая совпавшая строка. Будет возвращена пустая строка, если не найдено совпадений с выражением.</p> <p>Извлечение содержимого при использовании параметра вывод выполняется агентом.</p> <p>Параметры начальная строка, конечная строка и вывод поддерживаются начиная с версии 2.2.</p> <p>Примеры: <code>⇒ vfs.file.regexp[/etc/passwd,zabbix]</code> <code>⇒ vfs.file.regexp[/path/to/some/file,"([0-9]+)"",3,5,1]</code> <code>⇒ vfs.file.regexp[/etc/passwd,"^zabbix:([0-9]+)";,1]</code> → получение ID пользователя <i>zabbix</i></p>

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
vfs.file.regmatch[<i>файл</i>,<i>регулярное выражение</i>,<<i>кодировка</i>>,<<i>начальная строка</i>>,<<i>конечная строка</i>>]			
Поиск строки в файле.	0 - совпадение не найдено 1 - найдено	файл - абсолютный путь к файлу регулярное выражение - Perl совместимое регулярное выражение (PCRE) или расширенное POSIX регулярное выражение до Zabbix 3.4 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).	Параметры начальная строка, конечная строка поддерживаются с Zabbix 2.2. Например: ⇒ <code>vfs.file.regmatch[/var/log/app.log,error]</code>
vfs.file.size[<i>файл</i>]			
Размер файла (в байтах).	Целое число	файл - абсолютный путь к файлу	Пользователь zabbix должен иметь права на чтение файла Например: ⇒ <code>vfs.file.size[/var/log/syslog]</code> Ограничение размера файла зависит от поддержки больших файлов .
vfs.file.time[<i>файл</i>,<<i>режим</i>>]			
Информация о времени файла.	Целое число (Unix timestamp)	файл - полный путь к файлу режим - возможные значения: <i>modify</i> (по умолчанию) - время последней модификации содержимого файла, <i>access</i> - время последнего чтения файла, <i>change</i> - время последнего изменения свойств файла	Пример: ⇒ <code>vfs.file.time[/etc/passwd,modify]</code> Ограничение размера файла зависит от поддержки больших файлов .
vfs.fs.discovery			
Список примонтированных файловых систем. Используется низкоуровневым обнаружением.	JSON объект		Поддерживается Zabbix агентом начиная с версии 2.0. Макрос <code>{#FSDRIVETYPE}</code> поддерживается в Windows начиная с Zabbix агента версии 3.0.
vfs.fs.inode[<i>fs</i>,<<i>режим</i>>]			
Количество или процент inodes.	Целое число - для количества Число с плавающей точкой - для процентов	fs - файловая система режим - возможные значения: <i>total</i> (по умолчанию), <i>free</i> , <i>used</i> , <i>pfree</i> (свободно, в процентах), <i>used</i> (использовано, в процентах)	Пример: ⇒ <code>vfs.fs.inode[/,pfree]</code> Старое название: <code>vfs.fs.inode.free[*]</code> , <code>vfs.fs.inode.pfree[*]</code> , <code>vfs.fs.inode.total[*]</code>
vfs.fs.size[<i>fs</i>,<<i>режим</i>>]			
Размер диска в байтах или процентах от общего размера.	Целое число - для байт Число с плавающей точкой - для процентов.	fs - файловая система режим - возможные значения: <i>total</i> (по умолчанию), <i>free</i> , <i>used</i> , <i>pfree</i> (доступно, в процентах), <i>used</i> (использовано, в процентах)	В случае примонтированного раздела, будет возвращен размер диска локального файла системы. Пример: ⇒ <code>vfs.fs.size[/tmp,free]</code> Зарезервированное место на файловой системе принимается во внимание и не включено при использовании режима <i>free</i> . Старые названия: <code>vfs.fs.free[*]</code> , <code>vfs.fs.total[*]</code> , <code>vfs.fs.used[*]</code> , <code>vfs.fs.pfree[*]</code> , <code>vfs.fs.used[*]</code>
vm.memory.size[<<i>режим</i>>]			
Размер памяти в байтах или в процентах от общего количества.	Целое число - для байт Число с плавающей точкой - для процентов.	режим - возможные значения: <i>total</i> (по умолчанию), <i>active</i> , <i>anon</i> , <i>buffers</i> , <i>cached</i> , <i>exec</i> , <i>file</i> , <i>free</i> , <i>inactive</i> , <i>pinned</i> , <i>shared</i> , <i>slab</i> , <i>wired</i> , <i>used</i> , <i>used</i> , <i>used</i> (использовано, в процентах), <i>available</i> , <i>available</i> (доступно, в процентах)	Этот элемент данных принимает три категории параметров: 1) <i>total</i> - общего количества памяти. 2) специфичные для платформ типы памяти: <i>active</i> , <i>anon</i> , <i>buffers</i> , <i>cached</i> , <i>exec</i> , <i>file</i> , <i>free</i> , <i>inactive</i> , <i>pinned</i> , <i>shared</i> , <i>slab</i> , <i>wired</i> . 3) оценка на уровне пользователя как много памяти используется и доступно: <i>used</i> , <i>used</i> , <i>available</i> , <i>available</i> . Смотрите более подробное описание параметров vm.memory.size . Старые названия: <code>vm.memory.buffers</code> , <code>vm.memory.cached</code> , <code>vm.memory.free</code> , <code>vm.memory.shared</code> , <code>vm.memory.total</code>
web.page.get[<i>хост</i>,<<i>путь</i>>,<<i>порт</i>>]			
Получение содержимого веб-страницы.	Исходная веб-страница как текст (включая заголовки)	хост - имя хоста путь - путь к HTML документу (по умолчанию /) порт - номер порта (по умолчанию 80)	Возвращает пустую строку при ошибке. Например: ⇒ <code>web.page.get[www.zabbix.com,index.php,80]</code>
web.page.perf[<i>хост</i>,<<i>путь</i>>,<<i>порт</i>>]			
Время полной загрузки веб-страницы (в секундах).	Число с плавающей точкой	хост - имя хоста путь - путь к HTML документу (по умолчанию /) порт - номер порта (по умолчанию 80)	Возвращает 0 при ошибке. Например: ⇒ <code>web.page.perf[www.zabbix.com,index.php,80]</code>
web.page.regexpr[<i>хост</i>,<<i>путь</i>>,<<i>порт</i>>,<<i>регулярное выражение</i>>,<<i>длина</i>>,<<i>вывод</i>>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Поиск строки на веб-странице.	Совпадающая строка, или как указан опциональный параметр вывод	хост - имя хоста путь - путь к HTML документу (по умолчанию - /) порт - номер порта (по умолчанию - 80) регулярное выражение - Perl совместимое регулярное выражение (PCRE) или расширенное POSIX регулярное выражение до Zabbix 3.4 длина - максимальное количество возвращаемых символов вывод - дополнительный шаблон форматирования вывода. \0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-ной совпадающей группой (или пустой строкой, если N превышает количество найденных групп).	Возвращается пустая строка, если нет совпадений или при ошибке. Извлечение содержимого согласно дополнительного параметра вывод осуществляется агентом. Параметр вывод поддерживается начиная с версии 2.2. Пример: ⇒ web.page.regexp[www.zabbix.com,index.php,80,OK,2]

Заметка специфики для Linux. Zabbix агент должен иметь только права чтения файловой системы `/proc`. Патчи к ядру для ограничения прав непривилегированных пользователей вы можете найти на странице www.grsecurity.org.

Доступные кодировки

Параметр кодировка используется для того, чтобы указать кодировку при обработке соответствующих проверок элементов данных, так чтобы полученные данные не были повреждены. Для получения списка поддерживаемых кодировок (идентификаторов кодовых страниц), пожалуйста, обратитесь к соответствующей документации, такой как [libiconv](#) (GNU Project) или Microsoft Windows SDK документации по "Идентификаторам кодовых страницы".

Если задана пустая кодировка, тогда по умолчанию используются UTF-8 (языковой стандарт по умолчанию для новых Unix/Linux дистрибутивов, смотрите настройки вашей системы) или ANSI с определенным расширением в системе (Windows).

Поиск проблем с элементами данных агента

1. Если используются с пассивным агентов, значение *Timeout* конфигурации сервера, возможно, потребуется выше, чем *Timeout* в файле конфигурации агента. В противном случае элемент данных может не возвращать никаких значений по причине того, что запрос сервера к агенту превысит время ожидания раньше агента.

From: <https://www.zabbix.com/documentation/4.0/> - Zabbix Documentation 4.0

Permanent link: https://www.zabbix.com/documentation/4.0/ru/manual/config/items/itemtypes/zabbix_agent

Last update: 2018/10/19 10:11

