

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).</p> <p>вывод - дополнительный шаблон форматирования вывода: U - 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 максзадержка!</p> <p>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).</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 максзадержка!</p> <p>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: ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet (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 https и telnet 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: ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet (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 https и telnet 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 - ntp (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 ntp 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[, , ,arache2]), 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[, , ,arache2]), 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</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 (all 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 элементе комплексного экрана Часы.
system.run[команда,<режим>]			
Выполнение указанной команды на узле сети.	Текст с результатом выполнения команды 1 - в режиме <i>nowait</i> (независимо от результата выполнения команды)	команда - выполняемая команда режим - один из wait (по умолчанию, ожидать завершения выполнения), <i>nowait</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 EnableRemoteCommands=1. 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>]			
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: <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) memory,<тип> - информация о использовании виртуальной и реальной памяти: <i>avm</i> - активные виртуальные страницы (целое) <i>fre</i> - размер свободного списка (целое) 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) faults,<тип> - соотношение ловушек и прерываний: <i>in</i> - прерывания устройства (с плавающей точкой) <i>sy</i> - системные вызовы (с плавающей точкой) <i>cs</i> - переключения контекста потока ядра (с плавающей точкой) 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>es</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) 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) Этот элемент данных поддерживается начиная с версии 1.8.1.	
system.sw.arch			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Информация о архитектуре программного обеспечения.	Строка		Пример: ⇒ system.sw.arch → i686 Для получения информации используется функция uname(). Поддерживается Zabbix агентом начиная с версии 2.0.
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 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) Поддерживается Zabbix агентом начиная с версии 2.0.
system.sw.packages[<пакет>,<менеджер>,<формат>]			
Список установленных пакетов.	Текст	пакет - 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) rpm (rpm -Q) Если указан формат равный full, пакеты группируются по менеджером пакетов (каждый менеджер с отдельной строки, которая начинается с его имени заключенном в квадратные скобки). Если указан формат равный short, пакеты не группируются и отображаются одной строкой. Поддерживается Zabbix агентом начиная с версии 2.0.
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 swarins), sectors (секторов swapped in), pages (страниц swapped in). Смотрите поддержку по платформам для получения деталей по умолчанию.	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 swarouts), sectors (секторов swapped out), pages (страниц swapped out). Смотрите поддержку по платформам для получения деталей по умолчанию.	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)	Пример: ⇒ system.swap.size[,pfree] → 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 swap -s включает часть физической памяти и swap устройства (в отличии от swap -l). Обратите внимание, этот ключ может возвращать некорректные данные на виртуализированных (VMware ESXi, VirtualBox) Windows платформах. В этом случае используйте ключ perf_counter[700(_Total)]702] для получения корректного использования файла подкачки в процентах. Старое название: system.swap.free, system.swap.total
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 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, так как "uname -a" выводит информацию основываясь на других источниках. В Windows начиная с Zabbix 3.0 значение этого элемента данных извлекается при помощи Win32_OperatingSystem и Win32_Processor 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			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
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 who на стороне агента.
vfs.dev.read[<устройство>,<тип>,<режим>]			
Статистика чтения диска.	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: ⇒ <i>vfs.dev.read[,operations]</i> <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, <i>sda</i>), начиная с 1.8.6 можно использовать опциональный префикс /dev/ (например, /dev/sda) Старое название: <i>io[*]</i>
vfs.dev.write[<устройство>,<тип>,<режим>]			
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: ⇒ <i>vfs.dev.write[,operations]</i> <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, <i>sda</i>), начиная с 1.8.6 можно использовать опциональный префикс /dev/ (например, /dev/sda) Старое название: <i>io[*]</i>
vfs.dir.count[директория,<regex_вкл>,<regex_искл>,<типы_вкл>,<типы_искл>,<макс_глубина>,<мин_размер>,<макс_размер>,<мин_возраст>,<макс_возраст>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Количество записей в директории.	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, директории 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>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 "bdev,cdev", <i>all</i> - все вышеупомянутые типы, т.е. "file,dir,sym,sock,bdev,cdev,fifo". Это значение по умолчанию, если параметр оставлен пустым. Несколько типов необходимо разделять запятой и весь список нужно заключать в кавычки "".</p> <p>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.</p> <p>макс_глубина - максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) - без ограничения, 0 - без проверки нижестоящих поддиректорий.</p> <p>мин_размер - минимальный размер подсчитываемых файлов. Файлы, которые меньше этого значения не будут подсчитаны. Значение в байтах. Можно использовать суффиксы памяти.</p> <p>макс_размер - максимальный размер подсчитываемых файлов. Файлы, которые больше этого значения не будут подсчитаны. Значение в байтах. Можно использовать суффиксы памяти.</p> <p>мин_возраст - минимальный возраст подсчитываемой записи директории. Записи, которые изменены раньше не будут подсчитаны. Целочисленное значение в виде секунд. Можно использовать суффиксы времени.</p> <p>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.</p>	<p>Переменные сред, такие как %APP_HOME%, \$HOME и %TEMP%, не поддерживаются.</p> <p>Псевдо-директории "." и ".." никогда не учитываются.</p> <p>Символические ссылки никогда проверяются для обхода директорий.</p> <p>В Windows символические ссылки директорий пропускаются и жесткие ссылки учитываются лишь один раз.</p> <p>gegex_incl и gegex_excl являются Perl совместимыми регулярными выражениями (PCRE). Оба параметра gegex_вкл и gegex_искл применяются к файлам и директориям при вычислении общего размера, но игнорируются при выборе поддиректорий (если gegex_вкл равен "(?i)^\.\.zip\$" и макс_глубина не задана, тогда проход будет выполнен по всем поддиректориям, но учтены будут только файлы с типом zip). Если имя файла совпадает как с gegex_вкл, так и с gegex_искл, такой файл не будет учитываться.</p> <p>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.</p> <p>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.</p> <p>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)</p> <p>Поддерживается начиная с Zabbix 4.0.0.</p>
vfs.dir.size[директория,<gegex_вкл>,<gegex_искл>,<режим>,<макс_глубина>]			
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, директории 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 du -sb директория), <i>disk</i> - получение использования диска (работает как du -s -B1 директория). В отличие от du команды, элемент данных vfs.dir.size принимает во внимание скрытые файлы при вычислении размера директории (работает как du -sb .[^.]* * 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 Timeout в файлах конфигурации агента и сервера/прокси. При необходимости увеличьте эти значения времени ожидания.</p> <p>Примеры: ⇒ vfs.dir.size[/tmp,log] - вычисление размеров всех файлов в /tmp, которые содержат в своём имени 'log' ⇒ vfs.dir.size[/tmp,log,^\.\.old\$] - вычисление размеров всех файлов в /tmp, которые содержат в своём имени 'log', исключая файлы содержащие в своем 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]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Контрольная сумма файла, вычисленная по алгоритму используемому в UNIX cksum.	Целое число	файл - абсолютный путь к файлу	Пример: ⇒ vfs.file.cksum[/etc/passwd] Пример возвращаемого значения: 1938292000 Старое название: cksum 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. Пример: ⇒ vfs.file.contents[/etc/passwd] Этот элемент данных ограничен файлами не превышающими 64 КБайт. Поддерживается Zabbix агентом начиная с версии 2.0.
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	Пример: ⇒ vfs.file.exists[/tmp/application.pid] Возвращаемое значение зависит от того, что вернет S_ISREG POSIX макрос. Ограничение размера файла зависит от поддержки больших файлов .
vfs.file.md5sum[файл]			
MD5 контрольная сумма файла	Строка (MD5 хэш файла)	файл - абсолютный путь к файлу	Пример: ⇒ vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf] Пример возвращаемого значения: b5052decb577e0fffd622d6dc017e82 Ограничение (64МБ) на размер файла для этого элемента данных удалено в версии 1.8.6. 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 подстроку или то, что определено в дополнительном параметре вывод.	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). 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 только первая совпавшая строка. Будет возвращена пустая строка, если не найдено совпадений с выражением. Извлечение содержимого при использовании параметра вывод выполняется агентом. Параметры начальная строка, конечная строка и вывод поддерживаются начиная с версии 2.2. 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
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 - 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: ⇒ vfs.file.regmatch[/var/log/app.log,error]
vfs.file.size[файл]			
Размер файла (в байтах).	Целое число	файл - абсолютный путь к файлу	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: <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: ⇒ vfs.file.time[/etc/passwd,modify] Ограничение размера файла зависит от поддержки больших файлов .
vfs.fs.discovery			

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/4.4/> - **Zabbix Documentation 4.4**

Permanent link:
https://www.zabbix.com/documentation/4.4/ru/manual/config/items/itemtypes/zabbix_agent

Last update: **2019/04/02 06:04**

