

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>]			

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 <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.</p> <p>\O - 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>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>Параметр вывод поддерживается начиная с Zabbix 2.2. Параметр режим поддерживается начиная с Zabbix 2.0.</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p>
logrt[файл_регехр,<регулярное выражение>,<кодировка>,<макс. кол-вл строк>,<журнал>,<вывод>]			
Мониторинг файлов журналов (логов) с поддержкой ротации логов.	Журнал (лог)	<p>файл_регехр - 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:</p> <p>\O - 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>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>Параметр вывод поддерживается начиная с Zabbix 2.2 Параметр режим поддерживается начиная с Zabbix 2.0</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p>
net.dns[<ip>,<зона>,<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) зона - тестируемая зона 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>,зона,<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) зона - тестируемая зона 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	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 мультитайтные 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 параметром хранения значений <i>Дельта (скорость в секунду)</i> для получения статистики байт в секунду.
net.if.out[if,<режим>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
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	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 параметром хранения значений <i>Дельта (скорость в секунду)</i> для получения статистики байт в секунду.
net.if.total[if,<режим>]			
Суммарная статистика входящего и исходящего трафика на сетевом интерфейсе.	Целое число	if - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4 адрес (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	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 параметром хранения значений <i>Дельта (скорость 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	<i>Пример:</i> ⇒ 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	<i>Пример:</i> ⇒ 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>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
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: ⇒ <code>net.tcp.service[ftp,,45]</code> - 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 <code>net.tcp.port</code> . 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: <code>check_service[*]</code>
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: ⇒ <code>net.tcp.service.perf[ssh]</code> - 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 <code>net.tcp.service.perf[tcp,<ip>,<порт>]</code> . 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: <code>check_service_perf[*]</code>
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: ⇒ <code>net.udp.listen[68]</code> 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 TCP 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: ⇒ <code>net.udp.service[ntp,,45]</code> → можно использовать для тестирования доступности NTP сервиса на 45 порту UDP. Этот элемент данных поддерживается начиная с Zabbix 3.0, но <i>ntp</i> сервис был доступен в <code>net.tcp.service[]</code> элементе данных и в предыдущих версиях.
net.udp.service.perf[сервис,<ip>,<порт>]			
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: ⇒ <code>net.udp.service.perf[ntp]</code> → можно использовать для тестирования времени ответа от NTP сервиса. Этот элемент данных поддерживается начиная с Zabbix 3.0, но <i>ntp</i> сервис был доступен в <code>net.tcp.service[]</code> элементе данных и в предыдущих версиях.
proc.cpu.util[<имя>,<пользователь>,<тип>,<cmdline>,<режим>,<зона>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Использование CPU процесса в процентах.	Число с плавающей точкой	<p>имя - имя процесса (по умолчанию <i>все процессы</i>)</p> <p>пользователь - имя пользователя (по умолчанию <i>все пользователи</i>)</p> <p>тип - тип использования CPU: <i>total</i> (по умолчанию), <i>user</i>, <i>system</i></p> <p>cmdline - фильтр по командной строке (является регулярным выражением)</p> <p>режим - режим сбора данных: <i>avg1</i> (по умолчанию), <i>avg5</i>, <i>avg15</i></p> <p>зона - целевая зона: <i>current</i> (по умолчанию), <i>all</i>. Этот параметр поддерживается только на Solaris платформе.</p>	<p>Примеры: ⇒ <code>proc.cpu.util[,root]</code> → Использование CPU по всем процессам работающими под "root" пользователем ⇒ <code>proc.cpu.util[zabbix_server,zabbix]</code> → Использование CPU по всем процессам <code>zabbix_server</code> работающими под пользователем <code>zabbix</code></p> <p>Возвращаемое значение основывается на использовании одного ядра CPU, в процентах. Например, утилизация CPU процесса полного использования двух ядер равна 200%.</p> <p>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.</p> <p>Этот ключ поддерживается начиная с Zabbix 3.0.0 и также доступен на нескольких платформах (смотри Поддерживаемые элементы данных по платформам).</p>
proc.mem[<имя>,<пользователь>,<режим>,<cmdline>,<тип памяти>]			
Количество памяти используемое процессом в байтах.	Целое число	<p>имя - имя процесса (по умолчанию <i>все процессы</i>)</p> <p>пользователь - имя пользователя (по умолчанию <i>все пользователи</i>)</p> <p>режим - возможные значения: <i>avg</i>, <i>max</i>, <i>min</i>, <i>sum</i> (по умолчанию)</p> <p>cmdline - фильтр по командной строке (является регулярным выражением)</p> <p>тип памяти - тип памяти используемое процессом.</p>	<p>Примеры: ⇒ <code>proc.mem[,root]</code> - память используемая всеми процессами запущенными под пользователем "root" ⇒ <code>proc.mem[zabbix_server,zabbix]</code> - память используемая всеми процессами <code>zabbix_server</code> запущенными под пользователем <code>zabbix</code> ⇒ <code>proc.mem[,oracle,max,oracleZABBIX]</code> - максимальное значение используемой памяти процессами запущенными под пользователем <code>oracle</code> и имеющими <code>oracleZABBIX</code> в содержимом командной строки</p> <p>Смотрите заметки по выбору процессов с параметрами <code>имя</code> и <code>cmdline</code>.</p> <p>Параметр тип памяти поддерживается начиная с Zabbix 3.0.0. Он поддерживается на нескольких платформах (смотри Поддерживаемые элементы данных по платформам). Также смотрите специальные заметки по параметру тип памяти.</p>
proc.num[<имя>,<пользователь>,<состояние>,<cmdline>]			
Количество процессов.	Целое число	<p>имя - имя процесса (по умолчанию "все процессы")</p> <p>пользователь - имя пользователя (по умолчанию "все пользователи")</p> <p>состояние - возможные значения: <i>all</i> (по умолчанию), <i>run</i>, <i>sleep</i>, <i>zomb</i></p> <p>cmdline - фильтр по командной строке (является регулярным выражением)</p>	<p>Примеры ключей: ⇒ <code>proc.num[,mysql]</code> - количество процессов выполняемых под пользователем <code>mysql</code> ⇒ <code>proc.num[apache2,www-data]</code> - количество процессов <code>apache2</code> выполняемых под пользователем <code>www-data</code> ⇒ <code>proc.num[,oracle,sleep,oracleZABBIX]</code> - количество процессов в спящем состоянии выполняемых под <code>oracle</code> и имеющих <code>oracleZABBIX</code> в содержимом командной строки</p> <p>Смотрите заметки по выбору процессов с параметрами <code>имя</code> и <code>cmdline</code>.</p> <p>В Windows, поддерживаются только параметры <code>имя</code> и <code>пользователь</code>.</p>
sensor[устройство,сенсор,<режим>]			
Чтение аппаратного сенсора.	Число с плавающей точкой	<p>устройство - имя устройства</p> <p>сенсор - имя сенсора</p> <p>режим - возможные значения: <i>avg</i>, <i>max</i>, <i>min</i> (если этот параметр не указан, то устройство и сенсор обрабатываются дословно).</p>	<p>В Linux 2.4, читается <code>/proc/sys/dev/sensors</code>.</p> <p>Пример: ⇒ <code>sensor[w83781d-i2c-0-2d,temp1]</code></p> <p>До Zabbix 1.8.4, использовался формат <code>sensor[temp1]</code>.</p> <p>В Linux 2.6+, читается <code>/sys/class/hwmon</code>.</p> <p>Более подробное описание смотрите у элемента данных сенсора в Linux.</p> <p>В OpenBSD, читается MIB <code>hw.sensors</code>.</p> <p>Примеры: ⇒ <code>sensor[cpu0,temp0]</code> - температура одного CPU ⇒ <code>sensor[cpu[0-2]\$,temp,avg]</code> - средняя температура первых трех CPU</p> <p>Поддерживается в OpenBSD Zabbix агентом начиная с версии 1.8.4.</p>
system.boottime			
Штамп времени (timestamp) загрузки системы.	Целое число (UNIX timestamp)		

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
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</i> , <i>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</i> , <i>nice</i> , <i>user</i> (по умолчанию), <i>system</i> (по умолчанию для Windows), <i>iowait</i> , <i>interrupt</i> , <i>softirq</i> , <i>steal</i> , <i>guest</i> (на ядре Linux 2.6.24 и выше), <i>guest_nice</i> (на ядре Linux 2.6.33 и выше) режим - возможные значения: <i>avg1</i> (усреднение за одну минуту, по умолчанию), <i>avg5</i> , <i>avg15</i>	Пример: ⇒ <code>system.cpu.util[0,user,avg5]</code> Старое название: <code>system.cpu.idleX</code> , <code>system.cpu.niceX</code> , <code>system.cpu.systemX</code> , <code>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 - возможные значения: <i>full</i> (по умолчанию), <i>model</i> , <i>serial</i> , <i>type</i> или <i>vendor</i>	Пример: ⇒ <code>system.hw.chassis[full]</code> → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop] Этот ключ зависит от наличия SMBIOS таблицы в памяти. Требуются Права root , потому что значение читается из 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), curfreq, maxfreq, model или vendor	Пример: ⇒ system.hw.cpu[0,vendor] → AuthenticAMD Информация собирается из /proc/cpuinfo и из /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq. Если указаны номер CPU и <i>curfreq</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.
system.run[команда,<режим>]			
Выполнение указанной команды на узле сети.	Текст с результатом выполнения команды 1 - в режиме <i>powait</i> (независимо от результата выполнения команды)	команда - выполняемая команда режим - один из wait (по умолчанию, ожидать завершения выполнения), powait (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. Обратите внимание: Возвращаемое значение элемента данных является стандартным выводом совместно со стандартным выводом ошибок, которые получены от команды. Обратите внимание: 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:</p> <p>r - среднее количество запускаемых потоков ядра (дробное)</p> <p>b - 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:</p> <p>avm - активные виртуальные страницы (целое)</p> <p>fre - размер свободного списка (целое)</p> <p>page, <тип> - информация об ошибках страниц и активности страниц:</p> <p>fi - операции страничного ввода файлов в секунду (с плавающей точкой)</p> <p>fo - операции страничного вывода файлов в секунду (с плавающей точкой)</p> <p>pi - pages paged in from paging space (с плавающей точкой)</p> <p>po - pages paged out to paging space (с плавающей точкой)</p> <p>fr - pages freed (page replacement) (с плавающей точкой)</p> <p>sr - количество сканированных страниц алгоритмом замещения страниц (с плавающей точкой)</p> <p>faults, <тип> - соотношение ловушек и прерываний:</p> <p>in - прерывания устройства (с плавающей точкой)</p> <p>sy - системные вызовы (с плавающей точкой)</p> <p>cs - переключения контекста потока ядра (с плавающей точкой)</p> <p>cru, <тип> - раздельное использование процессорного времени в процентах:</p> <p>us - пользовательское время (с плавающей точкой)</p> <p>sy - системное время (с плавающей точкой)</p> <p>id - время простоя (с плавающей точкой)</p> <p>wa - время простоя в ходе которого система обрабатывала запросы I/O диска/NFS (с плавающей точкой)</p> <p>pc - количество использованных физических процессоров (с плавающей точкой)</p> <p>ec - процентное соотношение размеченного занятого пространства (с плавающей точкой)</p> <p>lbusy - показывает утилизация в процентах для логических процессор(ов), которая произошла при выполнении уровней пользователя и системы (с плавающей точкой)</p> <p>app - отображает количество доступных физических процессоров в распределенном пуле (с плавающей точкой)</p> <p>disk, <тип> - статистика о дисках:</p> <p>bps - отображает суммарное количество данных переданных (чтением или записью) на диск в байтах в секунду (целое)</p> <p>tps - показывает количество передач за секунду, которые произошли на физическом диске/ленте (с плавающей точкой)</p> <p>Этот элемент данных поддерживается начиная с версии 1.8.1.</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	<p>информация - возможные значения: full (по умолчанию), short или name</p>	<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 представлены во всех дистрибутивах): /proc/version (full) /proc/version_signature (short) /etc/issue.net (name)</p> <p>Поддерживается Zabbix агентом начиная с версии 2.0.</p>
system.sw.packages[<пакет>, <менеджер>, <формат>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Список установленных пакетов.	Текст	пакет - <i>all</i> (по умолчанию) или регулярное выражение менеджер - <i>all</i> (по умолчанию) или конкретный менеджер пакетов формат - <i>full</i> (по умолчанию) или <i>short</i>	Список (в алфавитном порядке) установленных пакетов имена которых совпадают с регулярным выражением заданным в пакет (<i>all</i> списки всех). Пример: ⇒ <code>system.sw.packages[mini,dpkg,short]</code> → <code>python-minimal, python2.6-minimal, ubuntu-minimal</code> Поддерживаемые менеджеры пакетов (выполняемые команды): <code>dpkg (dpkg --get-selections)</code> <code>pkgtool (ls /var/log/packages)</code> <code>rpm (rpm -qa)</code> <code>расman (расman -Q)</code> 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. 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 <i>all</i>) тип - возможные значения: <i>count</i> (количество swarpins), <i>sectors</i> (секторов swarpped in), <i>pages</i> (страниц swarpped in). Смотрите поддержку по платформам для получения деталей по умолчанию.	Пример: ⇒ <code>system.swap.in[,pages]</code> Источником этой информации является: <code>/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4)</code> <code>/proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)</code>
system.swap.out[<устройство>,<тип>]			
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 swarouts), <i>sectors</i> (секторов swarpped out), <i>pages</i> (страниц swarpped out). Смотрите поддержку по платформам для получения деталей по умолчанию.	Пример: ⇒ <code>system.swap.out[,pages]</code> Источником этой информации является: <code>/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4)</code> <code>/proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)</code>
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>pused</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 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, system.swap.total</code>
system.uptime			
Подробная информация о хосте.	Строка		Пример возвращаемого значения: <code>FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386</code> Начиная с Zabbix 2.2.0 значение этого элемента получается при помощи системного вызова <code>uname()</code> . Тогда как ранее информация запрашивалась с помощью системного вызова " <code>uname -a</code> " на Unix системах. Следовательно значение этого элемента данных может отличаться от вывода " <code>uname -a</code> " и не включать дополнительные сведения, которые " <code>uname -a</code> " отображает на основе дополнительных источников.
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			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
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> ²) 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.read[,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, sbd</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.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> ²) 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, sbd</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.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	<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 S_ISREG 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 совпадающую подстроку или то, что определено в дополнительном параметре вывод.	файл - абсолютный путь к файлу регулярное выражение - регулярное выражение GNU 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).	<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>
vfs.file.regmatch[файл,регулярное выражение,<кодировка>,<начальная строка>,<конечная строка>]			

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Поиск строки в файле.	0 - совпадение не найдено 1 - найдено	файл - абсолютный путь к файлу регулярное выражение - регулярное выражение GNU 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[файл]			
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[файл,<режим>]			
Информация о времени файла.	Целое число (Unix timestamp)	файл - полный путь к файлу режим - один из modify (по умолчанию, время модификации), access - время последнего доступа, change - время последнего изменения	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 {#FSDRIVETYPE} поддерживается в Windows начиная с Zabbix агента версии 3.0.
vfs.fs.inode[fs,<режим>]			
Количество или процент inodes.	Целое число - для количества Число с плавающей точкой - для процентов	fs - файловая система режим - один из total (по умолчанию), free, used, pfree (свободно, в процентах), pused (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[fs,<режим>]			
Размер диска в байтах или процентах от общего размера.	Целое число - для байт Число с плавающей точкой - для процентов.	fs - файловая система режим - один из total (по умолчанию), free, used, pfree (свободно, в процентах), pused (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 <code>free</code> . 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.pused[*]</code>
vm.memory.size[<режим>]			
Размер памяти в байтах или в процентах от общего количества.	Целое число - для байт Число с плавающей точкой - для процентов.	режим - один из total (по умолчанию), active, anon, buffers, cached, exec, file, free, inactive, pinned, shared, wired, used, pused, available, pavailable	Этот элемент данных принимает три категории параметров: 1) <code>total</code> - общего количества памяти. 2) специфичные для платформ типы памяти: <code>active</code> , <code>anon</code> , <code>buffers</code> , <code>cached</code> , <code>exec</code> , <code>file</code> , <code>free</code> , <code>inactive</code> , <code>pinned</code> , <code>shared</code> , <code>wired</code> . 3) оценка на уровне пользователя как много памяти используется и доступно: <code>used</code> , <code>pused</code> , <code>available</code> , <code>pavailable</code> . 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[хост,<путь>,<порт>]			
Получение содержимого веб-страницы.	Исходная веб-страница как текст	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>

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
web.page.perf[хост,<путь>,<порт>]			
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: ⇒ web.page.perf[www.zabbix.com,index.php,80]
web.page.regexр[хост,<путь>,<порт>,<регулярное выражение>,<длина>,<вывод>]			
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 - регулярное выражение GNU длина - максимальное количество возвращаемых символов вывод - дополнительный шаблон форматирования вывода. \O -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. Извлечение содержимого согласно дополнительного параметра вывод осуществляется агентом. Параметр вывод поддерживается начиная с версии 2.2. Пример: ⇒ web.page.regexр[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/4.2/> - **Zabbix Documentation 4.2**

Permanent link:

https://www.zabbix.com/documentation/4.2/ru/manual/config/items/itemtypes/zabbix_agent?rev=1450832747

Last update: **2018/10/01 09:42**

