

Chaves específicas para Windows

Chaves de item

A tabela a seguir apresenta detalhes das chaves de item que estão disponíveis somente no Zabbix Agent para Windows.

Chave			
Descrição	Retorno	Parâmetros	Comentários
eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]			
Monitoramento de eventos em log.	Log	<p>name - nome log de eventos</p> <p>regexp - expressão regular com o padrão desejado</p> <p>severity - expressão regular com a severidade desejada</p> <p>Este parâmetro aceita os seguintes valores: <i>"Information"</i>, <i>"Warning"</i>, <i>"Error"</i>, <i>"Critical"</i>, <i>"Verbose"</i> (desde o Zabbix 2.2.0 executando em Windows Vista ou superior)</p> <p>source - expressão regular descrevendo a fonte de identificação (o uso de expressão regular é suportado desde o Zabbix 2.2.0)</p> <p>eventid - expressão regular descrevendo os identificadores de evento</p> <p>maxlines - quantidade máxima de novas linhas por segundo que o Zabbix Agent enviará ao Zabbix Server ou Zabbix Proxy. Este parâmetro sobrescreve a definição 'MaxLinesPerSecond' que estiver definida no arquivo zabbix_agentd.win.conf</p> <p>mode - valores possíveis: <i>all</i> (padrão), <i>skip</i> - não processar os dados anteriores (afeta somente as linhas adicionadas após o início da coleta).</p>	<p>O item precisa ser configurado com o tipo Agente Zabbix (ativo).</p> <p>Exemplos: ⇒ eventlog[Application] ⇒ eventlog[Security,,"Failure Audit",,^(529 680)\$] ⇒ eventlog[System,,"Warning Error"] ⇒ eventlog[System,,,,^1\$] ⇒ eventlog[System,,,,@TWOSHORT] - aqui uma expressão regular customizada chamada TWOSHORT é referenciada (definida com o tipo Resultado VERDADEIRO, a expressão em sí seria ^1\$ ^70\$).</p> <p>O parâmetro mode é suportado desde o Zabbix 2.0.0. "Windows Eventing 6.0" é suportado desde o Zabbix 2.2.0.</p> <p>Observe que selecionando um tipo de dado diferente de Log para este item ocasionará a perda do carimbo de hora, gravidade de evento e sua origem.</p> <p>Consulte também informações adicionais em monitoramento de log.</p>
net.if.list			

Chave			
Descrição	Retorno	Parâmetros	Comentários
Lista de interfaces de rede (incluindo o tipo, status, endereço IPv4, descrição).	Texto		<p>Suportado desde o Zabbix 1.8.1. Interfaces com caracteres Multi-byte no nome são suportadas desde o Zabbix 1.8.6. Interfaces desabilitadas não são listadas.</p> <p>Observe que habilitando/desabilitando alguns componentes poderá alterar a sua ordem na lista de interfaces do Windows.</p> <p>Algumas versões do Windows (por exemplo, Windows Server 2008) poderão requerer os últimos updates para suportar caracteres não ASCII nos nomes de interface.</p>
perf_counter[counter,<interval>]			
Valor de qualquer contador do Windows.	Inteiro, numérico, string or texto (dependendo da requisição)	<p>counter - caminho para o contador</p> <p>interval - últimos N segundos de armazenamento do valor. O parâmetro <code>interval</code> precisa estar entre 1 e 900 segundos (inclusive), o padrão é 1.</p>	<p>O monitor de performance pode ser utilizado também para obter a lista de contadores disponível. Antes da versão 1.6 este parâmetro retornaria corretamente somente se fosse seguido um padrão (algo como: <code>\System\Threads</code>). Pode não funcionar corretamente para contadores que necessitam de mais de uma mostra (como a utilização de CPU). Desde o Zabbix 1.6, o parâmetro <code>interval</code> é utilizado para se obter o valor médio para o último intervalo de N segundos.</p> <p>Maix detalhes em: Contadores de performance Windows.</p>
proc_info[process,<attribute>,<type>]			

Chave			
Descrição	Retorno	Parâmetros	Comentários
<p>Informações sobre processos específicos.</p>	<p>Numérico</p>	<p>process - nome do processo attribute - atributo de processo necessário type - unidade de representação (significativo quando existe mais de um processo com o mesmo nome)</p>	<p>O parâmetro attributes pode ser: <i>vmsize</i> (padrão) - tamanho da memória virtual do processo (em Kbytes) <i>wkset</i> - tamanho do pacote de trabalho (quantidade de memória física utilizada pelo processo) em Kbytes <i>pf</i> - quantidade de falhas de página <i>ktime</i> - tempo de kernel em milisegundos <i>utime</i> - tempo de usuário em milisegundos <i>io_read_b</i> - quantidade de bytes lidos pelo processo em operações de I/O <i>io_read_op</i> - quantidade de operações de leitura feitas pelo processo <i>io_write_b</i> - quantidade de bytes gravados pelo processo durante operações de I/O <i>io_write_op</i> - quantidade de operações de escrita feitas pelo processo <i>io_other_b</i> - quantidade de bytes transferidos pelo processo durante operações diversas <i>io_other_op</i> - quantidade de operações de I/O executadas pelo processo, que não sejam de leitura e gravação <i>gdiobj</i> - quantidade de objetos GDI utilizados pelo processo <i>userobj</i> - quantidade de objetos de usuário utilizados pelo processo</p> <p>Valores possíveis para o parâmetro types: <i>avg</i> (padrão) - valor médio de todos os processos <process> <i>min</i> - valor mínimo de todos os processos <process> <i>max</i> - valor máximo de todos os processos <process> <i>sum</i> - valor total de todos os processos <process></p> <p>Exemplos: ⇒ <code>proc_info[iexplore.exe,wkset,sum]</code> - para obter o total de memória física por todos os processos do Internet Explorer ⇒ <code>proc_info[iexplore.exe,pf,avg]</code> - para obter a média de falhas de página do processo do Internet Explorer</p> <p>Observe que em um ambiente de 64-bit, uma versão do Zabbix Agent em 64-bit será necessária para o correto funcionamento.</p> <p>Nota: Os atributos <i>io_*</i>, <i>gdiobj</i> e <i>userobj</i> estão disponíveis somente a partir do Windows 2000.</p>

Chave			
Descrição	Retorno	Parâmetros	Comentários
service.discovery			
Lista os serviços do Windows. Usado pelo processo de autobusca.	Objeto JSON		Suportado desde o Zabbix 3.0.
service.info[service,<param>]			
Informação sobre um serviço.	<p>Inteiro - com o parâmetro param definido como: <i>state, startup</i></p> <p>String - com o parâmetro param definido como: <i>displayname, path, user</i></p> <p>Texto - com o parâmetro param definido como: <i>description</i></p> <p>Especificamente para <i>state</i>: 0 - em execução, 1 - pausado, 2 - iniciação pendente, 3 - pausa pendente, 4 - retorno pendente, 5 - finalização pendente, 6 - finalizado, 7 - desconhecido, 255 - serviço desconhecido</p> <p>Especialmente para <i>startup</i>: 0 - automático, 1 - automático com atraso, 2 - manual, 3 - desabilitado, 4 - desconhecido</p>	<p>service - o nome real do serviço ou o nome de visualização como visto no snap MMC Services</p> <p>param - <i>state</i> (padrão), <i>displayname, path, user, startup</i> ou <i>description</i></p>	<p>Exemplos: ⇒ <code>service.info[SNMPTRAP]</code> - estado do serviço SNMPTRAP ⇒ <code>service.info[SNMP Trap]</code> - estado do mesmo serviço, mas através do nome ⇒ <code>service.info[EventLog,startup]</code> - Eventos do serviço de EventLog com o tipo: inicialização</p> <p>Itens <code>service.info[service,state]</code> e <code>service.info[service]</code> retornarão a mesma informação.</p> <p>Note que apenas com o parâmetro param como <i>state</i> este item retornará o código de retorno para serviços inexistentes (255).</p> <p>Este item é suportado desde o Zabbix 3.0.0. Ele também pode ser utilizada a chave depreciada 'service_state[service]'.</p>
services[<type>,<state>,<exclude>]			

Chave			
Descrição	Retorno	Parâmetros	Comentários
Lista de serviços.	0 - se vazia Texto - lista dos serviços separados por quebra de linha	type - <i>all</i> (padrão), <i>automatic</i> , <i>manual</i> ou <i>disabled</i> state - <i>all</i> (padrão), <i>stopped</i> , <i>started</i> , <i>start_pending</i> , <i>stop_pending</i> , <i>running</i> , <i>continue_pending</i> , <i>pause_pending</i> ou <i>paused</i> exclude - serviços a restringir no resultado. Os serviços a serem ignorados deverão estar entre aspas, separados por vírgulas e sem espaços.	Exemplos: ⇒ <code>services[,started]</code> - lista dos serviços iniciados ⇒ <code>services[automatic, stopped]</code> - lista dos serviços parados, mas que poderiam ser executados ⇒ <code>services[automatic, stopped, "service1,service2,service3"]</code> - lista dos serviços parados, que poderiam estar executando, excluindo os serviços: <code>service1</code> , <code>service2</code> e <code>service3</code> A restrição de serviços é suportado desde o Zabbix 1.8.1.
wmi.get[<namespace>,<query>]			
Executa uma consulta WMI e retorna o primeiro objeto encontrado.	Inteiro, numérico, string or texto (dependendo da requisição)	namespace - nome de espaço WMI query - consulta WMI retornando um objeto simples	Example: ⇒ <code>wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%']</code> - retorna o status do primeiro disco físico. Esta chave é suportada desde o Zabbix 2.2.0.

Monitorando serviços do Windows

Este tutorial provê um passo-a-passo de como se configurar a monitoração de serviços do Windows. Partimos do princípio que o Zabbix Agent e o Zabbix Server estão configurados e operacionais.

Passo 1

Obter o nome do serviço.

Você pode obter o nome do serviço ao acessar o snap-in “MMC Services” e analisar as propriedades do serviço desejado. Na aba “Geral” existirá o campo 'Service name'. Este valor deverá ser obtido para configurar o item de monitoração.

Por exemplo, se você precisa monitorar o serviço “workstation” então o nome do serviço seria: **lanmanworkstation**.

Passo 2

[Configurar um item](#) para monitorar o serviço.

A chave 'service.info[service,<param>]' recupera informações sobre um serviço. Dependendo do que você precisar, defina o parâmetro *param* para um destes valores: *displayname*, *state*, *path*, *user*, *startup* ou *description*. O valor padrão é *state* se *param* não for definido (service.info[service]).

O tipo do valor de retorno depende do que for definido no parâmetro *param*: inteiro para *state* e *startup*; string para *displayname*, *path* e *user*; texto para *description*.

Exemplo:

- *Key*: service.info[lanmanworkstation]
- *Type of information*: Numeric (unsigned)
- *Show value*: select the *Windows service state* value mapping

Por padrão estão disponíveis no Zabbix dois mapeamentos de valores: *Windows service state* e *Windows service startup type* para mapear o valor numérico obtido para uma representação textual na interface web do Zabbix.

Descoberta de serviços do Windows

O [processo de autobusca](#) provê um caminho para a criação automática de itens, triggers, e graphs para diferentes entidades no mesmo computador. O Zabbix pode monitorar automaticamente os serviços Windows de sua máquina, sem que você precise saber exatamente o nome do serviço para criar os itens manualmente. Um filtro poderá ser utilizado para criar itens, triggers, e graphs apenas para os serviços de interesse.

From: <https://www.zabbix.com/documentation/3.2/> - **Zabbix Documentation 3.2**

Permanent link: https://www.zabbix.com/documentation/3.2/pt/manual/config/items/itemtypes/zabbix_agent/win_keys

Last update: **2016/02/16 15:25**

