

1 Creating an item

Overview

To create an item in Zabbix frontend, do the following:

- Go to: *Configuration* → *Hosts*
- Click on *Items* in the row of the host
- Click on *Create item* in the upper right corner of the screen
- Enter parameters of the item in the form

You can also create an item by opening an existing one, pressing the *Clone* button and then saving under a different name.

Configuration

The **Item** tab contains general item attributes:

Item
Preprocessing

Name

Type

Key

Host interface

Type of information

Units

Update interval

Custom intervals	Type	Interval	Period	Action
	<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>	Remove
	<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input style="background-color: #f0f0f0;" type="text" value="{FLEX_INTERVAL}"/>	<input style="background-color: #f0f0f0;" type="text" value="{FLEX_PERIOD}"/>	Remove
	<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input type="text" value="wd1-5h9-18"/>		Remove
	<input type="text" value="Flexible"/> <input type="text" value="Scheduling"/>	<input style="background-color: #f0f0f0;" type="text" value="{SSCHEDULING}"/>		Remove

[Add](#)

History storage period

Trend storage period

Show value [show value mappings](#)

New application

Applications

- None-
- CPU
- Filesystems
- General
- Memory
- Network interfaces
- OS
- Performance
- Processes
- Security

Populates host inventory field

Description

Enabled

Parameter	Description
<i>Name</i>	This is how the item will be named. The following macros can be used: \$1, \$2...\$9 - referring to the first, second... ninth parameter of the item key For example: Free disk space on \$1 If the item key is "vfs.fs.size[/,free]", the description will automatically change to "Free disk space on /"

Parameter	Description
<i>Type</i>	Item type. See individual item type sections.
<i>Key</i>	Item key. The supported item keys can be found in individual item type sections. The key must be unique within a single host. If key type is 'Zabbix agent', 'Zabbix agent (active)', 'Simple check' or 'Zabbix aggregate', the key value must be supported by Zabbix agent or Zabbix server. See also: the correct key format .
<i>Host interface</i>	Select the host interface. This field is available when editing an item on the host level.
<i>Type of information</i>	Type of data as stored in the database after performing conversions, if any. Numeric (unsigned) - 64bit unsigned integer Numeric (float) - floating point number Negative values can be stored. Allowed range: -999999999999.9999 to 999999999999.9999. Starting with Zabbix 2.2, receiving values in scientific notation is also supported. E.g. 1e+7, 1e-4. Character - short text data Log - long text data with optional log related properties (timestamp, source, severity, logeventid) Text - long text data <i>Limits of text data are described in the table below.</i>
<i>Units</i>	If a unit symbol is set, Zabbix will add post processing to the received value and display it with the set unit postfix. By default, if the raw value exceeds 1000, it is divided by 1000 and displayed accordingly. For example, if you set <i>bps</i> and receive a value of 881764, it will be displayed as 881.76 Kbps. Special processing is used for B (byte), Bps (bytes per second) units, which are divided by 1024. Thus, if units are set to B or Bps Zabbix will display: 1 as 1B/1Bps 1024 as 1KB/1KBps 1536 as 1.5KB/1.5KBps Special processing is used if the following time-related units are used: unixtime - translated to "yyyy.mm.dd hh:mm:ss". To translate correctly, the received value must be a <i>Numeric (unsigned)</i> type of information. uptime - translated to "hh:mm:ss" or "N days, hh:mm:ss" For example, if you receive the value as 881764 (seconds), it will be displayed as "10 days, 04:56:04" s - translated to "yyy mmm ddd hhh mmm sss ms"; parameter is treated as number of seconds. For example, if you receive the value as 881764 (seconds), it will be displayed as "10d 4h 56m" Only 3 upper major units are shown, like "1m 15d 5h" or "2h 4m 46s". If there are no days to display, only two levels are displayed - "1m 5h" (no minutes, seconds or milliseconds are shown). Will be translated to "< 1 ms" if the value is less than 0.001. See also the unit blacklist .
<i>Update interval</i>	Retrieve a new value for this item every N seconds. Maximum allowed update interval is 86400 seconds (1 day). Time suffixes are supported, e.g. 30s, 1m, 2h, 1d, since Zabbix 3.4.0. User macros are supported, since Zabbix 3.4.0. <i>Note:</i> If set to '0', the item will not be polled. However, if a custom interval (flexible/scheduling) also exists with a non-zero value, the item will be polled during the custom interval duration.

Parameter	Description
Custom intervals	<p>You can create custom rules for checking the item:</p> <p>Flexible - create an exception to the <i>Update interval</i> (interval with different frequency)</p> <p>Scheduling - create a custom polling schedule.</p> <p>For detailed information see Custom intervals.</p> <p>Time suffixes are supported in the <i>Interval</i> field, e.g. 30s, 1m, 2h, 1d, since Zabbix 3.4.0.</p> <p>User macros are supported, since Zabbix 3.4.0.</p> <p>Scheduling is supported since Zabbix 3.0.0.</p> <p><i>Note:</i> Not available for Zabbix agent active items.</p>
History storage period	<p>Duration of keeping detailed history in the database (1 hour to 25 years). Older data will be removed by the housekeeper.</p> <p>Stored in seconds. Time suffixes are supported, e.g. 2h, 1d, since Zabbix 3.4.0.</p> <p>User macros are supported, since Zabbix 3.4.0.</p> <p>This value can be overridden globally in <i>Administration</i> → <i>General</i> → Housekeeper. If the global setting exists, a warning message is displayed:</p> <p>History storage period <input type="text" value="1w"/> Overridden by global housekeeping settings (1d)</p> <p>It is recommended to keep the recorded values for the smallest possible time to reduce the size of value history in the database. Instead of keeping a long history of values, you can keep longer data of trends.</p> <p>See also History and trends.</p>
Trend storage period	<p>Duration of keeping aggregated (hourly min, max, avg, count) history in the database (1 day to 25 years). Older data will be removed by the housekeeper.</p> <p>Stored in seconds. Time suffixes are supported, e.g. 24h, 1d, since Zabbix 3.4.0.</p> <p>User macros are supported, since Zabbix 3.4.0.</p> <p>This value can be overridden globally in <i>Administration</i> → <i>General</i> → Housekeeper. If the global setting exists, a warning message is displayed:</p> <p>Trend storage period <input type="text" value="365d"/> Overridden by global housekeeping settings (7d)</p> <p><i>Note:</i> Keeping trends is not available for non-numeric data - character, log and text.</p> <p>See also History and trends.</p>
Show value	<p>Apply value mapping to this item. Value mapping does not change received values, it is for displaying data only.</p> <p>It works with integer items only.</p> <p>For example, "Windows service states".</p>
Log time format	<p>Available for items of type Log only. Supported placeholders:</p> <ul style="list-style-type: none"> * y: Year (1970-2038) * M: Month (01-12) * d: Day (01-31) * h: Hour (00-23) * m: Minute (00-59) * s: Second (00-59) <p>If left blank the timestamp will not be parsed.</p> <p>For example, consider the following line from the Zabbix agent log file: " 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)." It begins with six character positions for PID, followed by date, time, and the rest of the line.</p> <p>Log time format for this line would be "pppppp:yyyyMMdd:hhmmss". <i>Note</i> that "p" and ":" chars are just placeholders and can be anything but "yMdhms".</p>
New application	Enter the name of a new application for the item.
Applications	Link item to one or more existing applications.

Parameter	Description
<i>Populates host inventory field</i>	You can select a host inventory field that the value of item will populate. This will work if automatic inventory population is enabled for the host.
<i>Description</i>	Enter an item description.
<i>Enabled</i>	Mark the checkbox to enable the item so it will be processed.

Item type specific fields are described on [corresponding pages](#).

When editing an existing [template](#) level item on a host level, a number of fields are read-only. You can use the link in the form header and go to the template level and edit them there, keeping in mind that the changes on a template level will change the item for all hosts that the template is linked to.

Text data limits

Text data limits depend on the database backend. Before storing text values in the database they get truncated to match the database value type limit:

Database	Type of information		
	Character	Log	Text
MySQL	255 characters	65536 bytes	65536 bytes
PostgreSQL	255 characters	65536 characters	65536 characters
Oracle	255 characters	65536 characters	65536 characters
DB2	255 bytes	2048 bytes	2048 bytes

Unit blacklist

By default, specifying a unit for an item will result in a multiplier prefix being added - for example, value 2048 with unit B would be displayed as 2KB. For a pre-defined, hardcoded list of units this is prevented:

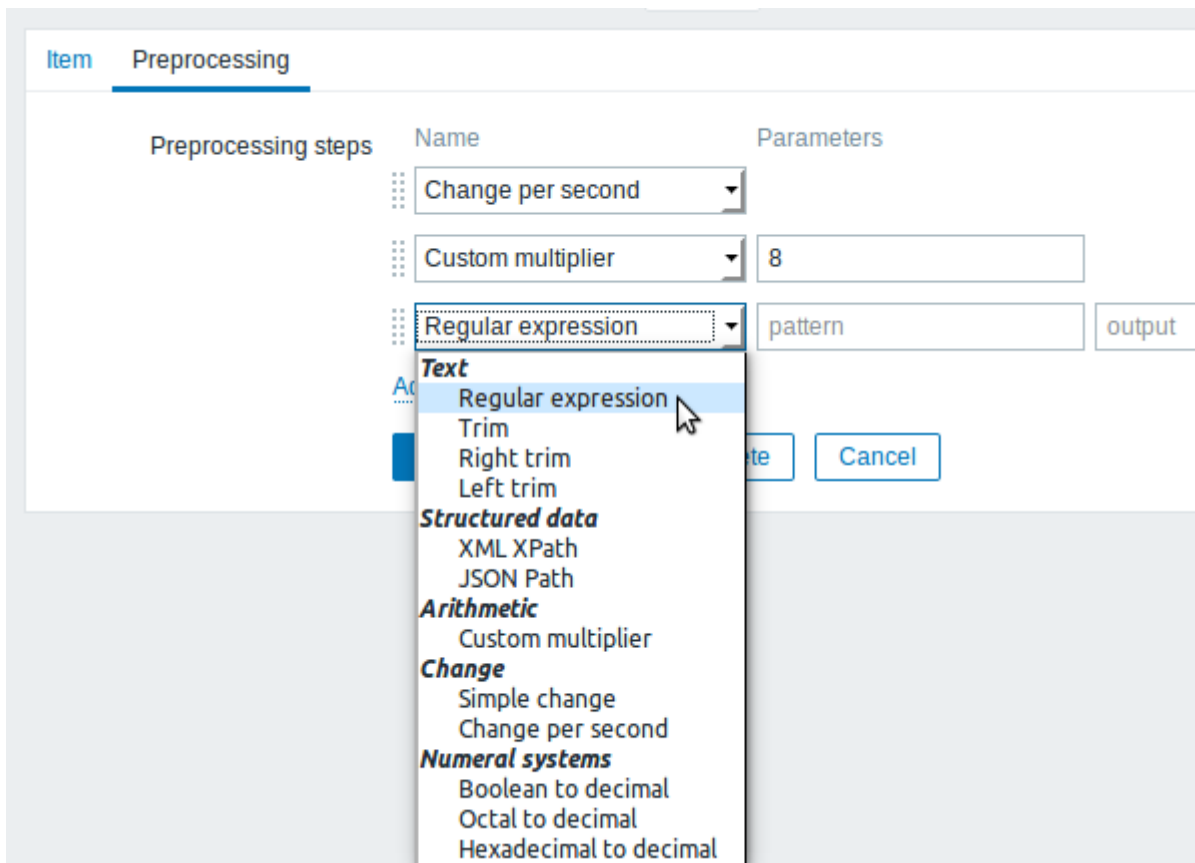
- ms
- RPM
- rpm
- %

Note that both lowercase and uppercase **rpm** (*rpm* and *RPM*) strings are blacklisted.

Item value preprocessing

The **Preprocessing** tab allows to define transformation rules for the received values. One or several transformations are possible before saving values to the database. Transformations are executed in the order in which they are defined. All preprocessing is done by Zabbix server.

See also: [Preprocessing details](#)



An item can become [unsupported](#) if any of the preprocessing steps fails.

Transformation	Description
<i>Regular expression</i>	Match the value to the <pattern> regular expression and replace value with <output>. The regular expression supports extraction of maximum 10 captured groups with the \N sequence. Failure to match the input value will make the item unsupported. Parameters: pattern - regular expression output - output formatting template. An \N (where N=1...9) escape sequence is replaced with the Nth matched group. A \0 escape sequence is replaced with the matched text. Supported since 3.4.0. Please refer to regular expressions section for some existing examples.
<i>Trim</i>	Remove specified characters from the beginning and end of the value.
<i>Right trim</i>	Remove specified characters from the end of the value.
<i>Left trim</i>	Remove specified characters from the beginning of the value.
<i>XML XPath</i>	Extract value or fragment from XML data using XPath functionality. For this option to work, Zabbix server must be compiled with libxml support. Examples: number(/document/item/value) will extract 10 from <document><item><value>10</value></item></document> number(/document/item/@attribute) will extract 10 from <document><item attribute="10"></item></document> /document/item will extract <item><value>10</value></item> from <document><item><value>10</value></item></document> Note that namespaces are not supported. Supported since 3.4.0.

Transformation	Description
<p><i>JSON Path</i></p>	<p>Extract value or fragment from JSON data using a simple subset of JSONPath functionality. JSONPath can be specified using the dot notation: <code>\$.document.item[0].value</code> or the bracket notation: <code>\$['document']['item'][0]['value']</code> The former, dot notation, can be used only if object names consist of alphanumeric + underscore characters: <code>\$.document.item_0.value</code> If object name contains other characters, e. g. blanks, dashes, you must use the bracket notation: <code>\$['document']['item 0']['value-0']</code> Both notations can be mixed: <code>\$.document['item'][0].value</code> For both notations only direct paths to single objects are supported. Extracting multiple values is not supported. More examples: <code>\$.document.item.value</code> will extract 10 from <code>{"document":{"item":{"value": 10}}}</code> <code>\$.document.item</code> will extract <code>{"value": 10}</code> from <code>{"document":{"item":{"value": 10}}}</code> <code>\$['a document'].item.value</code> will extract 10 from <code>{"a document":{"item":{"value": 10}}}</code> <code>\$.document.items[1].value</code> will extract 20 from <code>{"document":{"items":[{"value": 10}, {"value": 20}]}}</code> Supported since 3.4.0.</p>
<p><i>Custom multiplier</i></p>	<p>Multiply the value by the specified integer or floating-point value. Use this option to convert values received in KB, MBps, etc into B, Bps. Otherwise Zabbix cannot correctly set prefixes (K, M, G etc). Starting with Zabbix 2.2, using scientific notation is also supported. E.g. 1e+70.</p>
<p><i>Simple change</i></p>	<p>Calculate difference between the current and previous value. Evaluated as value-prev_value, where <i>value</i> - current value; <i>prev_value</i> - previously received value This setting can be useful to measure a constantly growing value. If the current value is smaller than the previous value, Zabbix discards that difference (stores nothing) and waits for another value. Only one change operation per item is allowed.</p>
<p><i>Change per second</i></p>	<p>Calculate the value change (difference between the current and previous value) speed per second. Evaluated as (value-prev_value)/(time-prev_time), where <i>value</i> - current value; <i>prev_value</i> - previously received value; <i>time</i> - current timestamp; <i>prev_time</i> - timestamp of previous value. This setting is extremely useful to get speed per second for a constantly growing value. If current value is smaller than the previous value, Zabbix discards that difference (stores nothing) and waits for another value. This helps to work correctly with, for instance, a wrapping (overflow) of 32-bit SNMP counters. <i>Note:</i> As this calculation may produce floating point numbers, it is recommended to set the 'Type of information' to <i>Numeric (float)</i>, even if the incoming raw values are integers. This is especially relevant for small numbers where the decimal part matters. If the floating point values are large and may exceed the 'float' field length in which case the entire value may be lost, it is actually suggested to use <i>Numeric (unsigned)</i> and thus trim only the decimal part. Only one change operation per item is allowed.</p>

Transformation	Description
<i>Boolean to decimal</i>	Convert the value from boolean format to decimal. Textual representation is translated into either 0 or 1. Thus, 'TRUE' is stored as 1 and 'FALSE' is stored as 0. All values are matched in a case-insensitive way. Currently recognized values are, for: <i>TRUE</i> - true, t, yes, y, on, up, running, enabled, available <i>FALSE</i> - false, f, no, n, off, down, unused, disabled, unavailable Additionally, any non-zero numeric value is considered to be TRUE and zero is considered to be FALSE.
<i>Octal to decimal</i>	Convert the value from octal format to decimal.
<i>Hexadecimal to decimal</i>	Convert the value from hexadecimal format to decimal. See also: known issues for 3.4.0-3.4.13 .

If you use a custom multiplier or store value as *Change per second* for items with the type of information set to *Numeric (unsigned)* and the resulting calculated value is actually a float number, the calculated value is still accepted as a correct one by trimming the decimal part and storing the value as integer.

Unsupported items

An item can become unsupported if its value cannot be retrieved for some reason. Such items are still rechecked at a fixed interval, configurable in [Administration section](#).

Unsupported items are reported as having a NOT SUPPORTED state.

From: <https://www.zabbix.com/documentation/3.4/> - **Zabbix Documentation 3.4**

Permanent link: <https://www.zabbix.com/documentation/3.4/manual/config/items/item?rev=1542200849>

Last update: **2018/11/14 13:07**

