

1 Supported trigger functions

All functions supported in [trigger expressions](#) are listed here.

FUNCTION		
Description	Parameters	Comments
abschange		
The amount of absolute difference between last and previous values.		Supported value types: float, int, str, text, log For example: (previous value;last value=abschange) 1;5=4 3;1=2 0;-2.5=2.5 For strings returns: 0 - values are equal 1 - values differ
avg (sec #num,<time_shift>)		
Average value of an item within the defined evaluation period.	sec or #num - evaluation period in seconds or in latest collected values (preceded by a hash mark) time_shift (optional) - evaluation point is moved the number of seconds back in time	Supported value types: float, int Examples: ⇒ avg(#5) → average value for the five latest values ⇒ avg(1h) → average value for an hour ⇒ avg(1h,1d) → average value for an hour one day ago. The <code>time_shift</code> parameter is supported since Zabbix 1.8.2. It is useful when there is a need to compare the current average value with the average value <code>time_shift</code> seconds back.
band (<sec #num>,mask,<time_shift>)		
Value of “bitwise AND” of an item value and mask.	sec (ignored, equals #1) or #num (optional) - the Nth most recent value mask (mandatory) - 64-bit unsigned integer (0 - 18446744073709551615) time_shift (optional) - see avg()	Supported value types: int Take note that <code>#num</code> works differently here than with many other functions (see <code>last()</code>). Although the comparison is done in a bitwise manner, all the values must be supplied and are returned in decimal. For example, checking for the 3rd bit is done by comparing to 4, not 100. Examples: ⇒ band(,12)=8 or band(,12)=4 → 3rd or 4th bit set, but not both at the same time ⇒ band(,20)=16 → 3rd bit not set and 5th bit set. This function is supported since Zabbix 2.2.0.
change		

FUNCTION		
Description	Parameters	Comments
abschange		
The amount of difference between last and previous values.		Supported value types: float, int, str, text, log For example: (previous value;last value=change) 1;5=+4 3;1=-2 0;-2.5=-2.5 See also: abschange for comparison For strings returns: 0 - values are equal 1 - values differ
count (sec #num,<pattern>,<operator>,<time_shift>)		

FUNCTION		
Description	Parameters	Comments
abschange		
<p>Number of values within the defined evaluation period.</p>	<p>sec or #num - evaluation period in seconds or in latest collected values (preceded by a hash mark)</p> <p>pattern (optional) - required pattern</p> <p>operator (optional)</p> <p>Supported operators: <i>eq</i> - equal <i>ne</i> - not equal <i>gt</i> - greater <i>ge</i> - greater or equal <i>lt</i> - less <i>le</i> - less or equal <i>like</i> - matches if contains pattern (case-sensitive) <i>band</i> - bitwise AND <i>regexp</i> - case sensitive match of regular expression given in pattern <i>iregexp</i> - case insensitive match of regular expression given in pattern</p> <p>Note that: <i>eq</i> (default), <i>ne</i>, <i>gt</i>, <i>ge</i>, <i>lt</i>, <i>le</i>, <i>band</i>, <i>regexp</i>, <i>iregexp</i> are supported for integer items <i>eq</i> (default), <i>ne</i>, <i>gt</i>, <i>ge</i>, <i>lt</i>, <i>le</i>, <i>regexp</i>, <i>iregexp</i> are supported for float items <i>like</i> (default), <i>eq</i>, <i>ne</i>, <i>regexp</i>, <i>iregexp</i> are supported for string, text and log items</p> <p>time_shift (optional) - see avg()</p>	<p>Supported value types: float, integer, string, text, log Float items match with the precision of 0.000001.</p> <p>With <i>band</i> as third parameter, the second pattern parameter can be specified as two numbers, separated by '/': number_to_compare_with/mask. count() calculates "bitwise AND" from the value and the <i>mask</i> and compares the result to <i>number_to_compare_with</i>. If the result of "bitwise AND" is equal to <i>number_to_compare_with</i>, the value is counted. If <i>number_to_compare_with</i> and <i>mask</i> are equal, only the <i>mask</i> need be specified (without '/').</p> <p>With <i>regexp</i> or <i>iregexp</i> as third parameter the second pattern parameter can be an ordinary or global (starting with '@') regular expression. In case of global regular expressions case sensitivity is inherited from global regular expression settings. For the purpose of regexp matching, float values will always be represented with 4 decimal digits after '.'. Also note that for large numbers difference in decimal (stored in database) and binary (used by Zabbix server) representation may affect the 4th decimal digit.</p> <p>Examples: ⇒ count(10m) → number of values for last 10 minutes ⇒ count(10m,"error",eq) → number of values for last 10 minutes that equal 'error' ⇒ count(10m,12) → number of values for last 10 minutes that equal '12' ⇒ count(10m,12,gt) → number of values for last 10 minutes that are over '12' ⇒ count(#10,12,gt) → number of values within last 10 values that are over '12' ⇒ count(10m,12,gt,1d) → number of values for preceding 10 minutes up to 24 hours ago that were over '12' ⇒ count(10m,6/7,band) → number of values for last 10 minutes having '110' (in binary) in the 3 least significant bits. ⇒ count(10m,,,1d) → number of values for preceding 10 minutes up to 24 hours ago</p> <p>The #num parameter is supported since Zabbix 1.6.1. The time_shift parameter and string operators are supported since Zabbix 1.8.2. The band operator is supported since Zabbix 2.2.0. The regexp, iregexp operators are supported since Zabbix 3.2.0.</p>

FUNCTION		
Description	Parameters	Comments
abschange		
date		
Current date in YYYYMMDD format.		Supported value types: <i>any</i> Example of returned value: 20150731
dayofmonth		
Day of month in range of 1 to 31.		Supported value types: <i>any</i> This function is supported since Zabbix 1.8.5.
dayofweek		
Day of week in range of 1 to 7 (Mon - 1, Sun - 7).		Supported value types: <i>any</i>
delta (sec #num,<time_shift>)		
Difference between the maximum and minimum values within the defined evaluation period ('max()' minus 'min()').	sec or #num - evaluation period in seconds or in latest collected values specified (preceded by a hash mark) time_shift (optional) - see avg()	Supported value types: float, int The <code>time_shift</code> parameter is supported since Zabbix 1.8.2.
diff		
Checking if last and previous values differ.		Supported value types: float, int, str, text, log Returns: 1 - last and previous values differ 0 - otherwise
forecast (sec #num,<time_shift>,time,<fit>,<mode>)		

FUNCTION		
Description	Parameters	Comments
abschange		
<p>Future value, max, min, delta or avg of the item.</p>	<p>sec or #num - evaluation period in seconds or in latest collected values specified (preceded by a hash mark) time_shift (optional) - see avg() time - forecasting horizon in seconds fit (optional) - function used to fit historical data</p> <p>Supported fits: <i>linear</i> - linear function <i>polynomialN</i> - polynomial of degree N (1 <= N <= 6) <i>exponential</i> - exponential function <i>logarithmic</i> - logarithmic function <i>power</i> - power function</p> <p>Note that: <i>linear</i> is default, <i>polynomial1</i> is equivalent to <i>linear</i></p> <p>mode (optional) - demanded output</p> <p>Supported modes: <i>value</i> - value (default) <i>max</i> - maximum <i>min</i> - minimum <i>delta</i> - max-min <i>avg</i> - average</p> <p>Note that: <i>value</i> estimates item value at the moment now + time <i>max</i>, <i>min</i>, <i>delta</i> and <i>avg</i> investigate item value estimate on the interval between now and now + time</p>	<p>Supported value types: float, int</p> <p>If value to return is larger than 99999999999.9999 or less than -99999999999.9999, return value is cropped to 99999999999.9999 or -99999999999.9999 correspondingly.</p> <p>Becomes not supported only if misused in expression (wrong item type, invalid parameters), otherwise returns -1 in case of errors.</p> <p>Examples: ⇒ forecast(#10,,1h) → forecast of item value after one hour based on last 10 values ⇒ forecast(1h,,30m) → forecast of item value after 30 minutes based on last hour data ⇒ forecast(1h,1d,12h) → forecast of item after 12 hours based on one hour one day ago ⇒ forecast(1h,,10m,exponential) → forecast of item value after 10 minutes based on last hour data and exponential function ⇒ forecast(1h,,2h,polynomial3,max) → forecast of maximum value item can reach in next two hours based on last hour data and cubic (third degree) polynomial ⇒ forecast(#2,,-20m) → estimate the value of an item which was 20 minutes ago based on last two values (this can be more precise than using last() or prev(), especially if item is updated rarely, say, once an hour)</p> <p>This function is supported since Zabbix 3.0.0. Negative time values are supported since Zabbix 3.0.6 and 3.2.2. See also additional information on predictive trigger functions.</p>
fuzzytime (sec)		

FUNCTION		
Description	Parameters	Comments
abschange		
Checking how much an item value (as timestamp) differs from the Zabbix server time.	sec - seconds	Supported value types: float, int Returns: 1 - difference between item value (as timestamp) and Zabbix server timestamp is less than or equal to T seconds 0 - otherwise Usually used with the 'system.localtime' item to check that local time is in sync with the local time of Zabbix server. <i>Note</i> that 'system.localtime' must be configured as a passive check . Can be used also with vfs.file.time[/path/file,modify] key to check that file didn't get updates for long time. Example: ⇒ fuzzytime(60)=0 → detect a problem if time difference is over 60 seconds
iregexp (<pattern>,<sec #num>)		
This function is a non case-sensitive analogue of regexp().	see regexp()	Supported value types: str, log, text
last (<sec #num>,<time_shift>)		
The most recent value.	sec (ignored, equals #1) or #num (optional) - the Nth most recent value time_shift (optional) - see avg()	Supported value types: float, int, str, text, log Take note that #num works differently here than with many other functions. For example: last() is always equal to last(#1) last(#3) - third most recent value (<i>not</i> three latest values) Zabbix does not guarantee exact order of values if more than two values exist within one second in history. The #num parameter is supported since Zabbix 1.6.2. The time_shift parameter is supported since Zabbix 1.8.2.
logeventid (<pattern>)		
Checking if event ID of the last log entry matches a regular expression.	pattern (optional) - regular expression describing the required pattern, Perl Compatible Regular Expression (PCRE) style.	Supported value types: log Returns: 0 - does not match 1 - matches This function is supported since Zabbix 1.8.5.
logseverity		

FUNCTION		
Description	Parameters	Comments
abschange		
Log severity of the last log entry.		Supported value types: log Returns: 0 - default severity N - severity (integer, useful for Windows event logs: 1 - Information, 2 - Warning, 4 - Error, 7 - Failure Audit, 8 - Success Audit, 9 - Critical, 10 - Verbose). Zabbix takes log severity from Information field of Windows event log.
logsource (<pattern>)		
Checking if log source of the last log entry matches a regular expression.	pattern (optional) - regular expression describing the required pattern, Perl Compatible Regular Expression (PCRE) style.	Supported value types: log Returns: 0 - does not match 1 - matches Normally used for Windows event logs. For example, logsource("VMware Server").
max (sec #num,<time_shift>)		
Highest value of an item within the defined evaluation period.	sec or #num - evaluation period in seconds or in latest collected values (preceded by a hash mark) time_shift (optional) - see avg()	Supported value types: float, int The <code>time_shift</code> parameter is supported since Zabbix 1.8.2.
min (sec #num,<time_shift>)		
Lowest value of an item within the defined evaluation period.	sec or #num - evaluation period in seconds or in latest collected values (preceded by a hash mark) time_shift (optional) - see avg()	Supported value types: float, int The <code>time_shift</code> parameter is supported since Zabbix 1.8.2.
nodata (sec)		
Checking for no data received.	sec - evaluation period in seconds. The period should not be less than 30 seconds because the history syncer process calculates this function only every 30 seconds. nodata(0) is disallowed.	Supported value types: <i>any</i> Returns: 1 - if no data received during the defined period of time 0 - otherwise Note that this function will display an error if, within the period of the 1st parameter: - there's no data and Zabbix server was restarted - there's no data and maintenance was completed - there's no data and the item was added or re-enabled Errors are displayed in the <i>Info</i> column in trigger configuration .
now		
Number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).		Supported value types: <i>any</i>
percentile (sec #num,<time_shift>,percentage)		

FUNCTION		
Description	Parameters	Comments
abschange		
P-th percentile of a period, where P (percentage) is specified by the third parameter.	sec or #num - evaluation period in seconds or in latest collected values (preceded by a hash mark) time_shift (optional) - see avg() percentage - a floating-point number between 0 and 100 (inclusive) with up to 4 digits after the decimal point	Supported value types: float, int This function is supported since Zabbix 3.0.0.
prev		
Previous value.		Supported value types: float, int, str, text, log Returns the same as last(#2).
regexp (<pattern>,<sec #num>)		
Checking if the latest (most recent) value matches regular expression.	pattern (optional) - regular expression, Perl Compatible Regular Expression (PCRE) style. sec or #num (optional) - evaluation period in seconds or in latest collected values (preceded by a hash mark). In this case, more than one value may be processed.	Supported value types: str, text, log Returns: 1 - found 0 - otherwise If more than one value is processed, '1' is returned if there is at least one matching value. This function is case-sensitive.
str (<pattern>,<sec #num>)		
Finding a string in the latest (most recent) value.	pattern (optional) - required string sec or #num (optional) - evaluation period in seconds or in latest collected values (preceded by a hash mark). In this case, more than one value may be processed.	Supported value types: str, text, log Returns: 1 - found 0 - otherwise If more than one value is processed, '1' is returned if there is at least one matching value. This function is case-sensitive.
strlen (<sec #num>,<time_shift>)		
Length of the latest (most recent) value in characters (not bytes).	sec (ignored, equals #1) or #num (optional) - the Nth most recent value time_shift (optional) - see avg()	Supported value types: str, text, log Take note that #num works differently here than with many other functions. Examples: ⇒ strlen()(is equal to strlen(#1)) → length of the latest value ⇒ strlen(#3) → length of the third most recent value ⇒ strlen(,1d) → length of the most recent value one day ago. This function is supported since Zabbix 1.8.4.
sum (sec #num,<time_shift>)		

FUNCTION		
Description	Parameters	Comments
abschange		
Sum of collected values within the defined evaluation period.	sec or #num - evaluation period in seconds or in latest collected values (preceded by a hash mark) time_shift (optional) - see avg()	Supported value types: float, int The <code>time_shift</code> parameter is supported since Zabbix 1.8.2.
time		
Current time in HHMMSS format.		Supported value types: <i>any</i> Example of returned value: 123055
timeleft (sec #num,<time_shift>,threshold,<fit>)		
Time in seconds needed for an item to reach a specified threshold.	sec or #num - evaluation period in seconds or in latest collected values specified (preceded by a hash mark) time_shift (optional) - see avg() threshold - value to reach fit (optional) - see forecast()	Supported value types: float, int If value to return is larger than 9999999999.9999, return value is cropped to 9999999999.9999. Returns 9999999999.9999 if threshold cannot be reached. Becomes not supported only if misused in expression (wrong item type, invalid parameters), otherwise returns -1 in case of errors. Examples: ⇒ <code>timeleft(#10,,0)</code> → time until item value reaches zero based on last 10 values ⇒ <code>timeleft(1h,,100)</code> → time until item value reaches 100 based on last hour data ⇒ <code>timeleft(1h,1d,0)</code> → time until item value reaches 0 based on one hour one day ago ⇒ <code>timeleft(1h,,200,polynomial2)</code> → time until item reaches 200 based on last hour data and assumption that item behaves like quadratic (second degree) polynomial This function is supported since Zabbix 3.0.0. Unit symbols in <code>threshold</code> parameter are supported since Zabbix 3.0.6 and 3.2.2. See also additional information on predictive trigger functions .

- 1) All functions return numeric values only. Comparison to strings is not supported, for example.
- 2) Some of the functions cannot be used for non-numeric parameters!
- 3) String arguments should be double quoted. Otherwise, they might get misinterpreted.
- 4) For all trigger functions **sec** and **time_shift** must be an integer with an optional [time unit suffix](#) and has absolutely nothing to do with item's data type.

Functions and unsupported items

Starting from Zabbix 3.2, **nodata()**, **date()**, **dayofmonth()**, **dayofweek()**, **now()** and **time()** functions are calculated for unsupported items, too. Other functions require that the referenced item is in a supported state.

Last update: 2019/04/02 17:21 manual:appendix:triggers:functions <https://www.zabbix.com/documentation/current/manual/appendix/triggers/functions?rev=1552833325>

From: <https://www.zabbix.com/documentation/current/> - **Zabbix Documentation 4.4**

Permanent link: <https://www.zabbix.com/documentation/current/manual/appendix/triggers/functions?rev=1552833325>

Last update: **2019/04/02 17:21**

