

## 8 Forecasting functions

### Overview

Sometimes there are signs of the upcoming problem. These signs can be spotted so that actions may be taken in advance to prevent or at least minimize the impact of the problem. Zabbix has tools to predict the future behaviour of the monitored system based on historic data. These tools are realized as predictive trigger functions.

### 1 Functions

Two things one needs to know is how to define a problem state and how much time is needed to take action. Then there are two ways to set up a trigger signalling about a potential unwanted situation. First: trigger must fire when system after “time to act” is expected to be in problem state. Second: trigger must fire when system is going to reach the problem state in less than “time to act”. Corresponding trigger functions to use are **forecast** and **timeleft**. Note that underlying statistical analysis is basically identical for both functions. You may set up a trigger whichever way you prefer with similar results.

### 2 Parameters

Both functions use almost the same set of parameters. Use the list of [supported functions](#) for reference.

#### 2.1 Time interval

First of all you should specify the historic period Zabbix should analyse to come up with prediction. You do it in a familiar way by means of `sec` or `#num` parameter and optional `time_shift` like you do it with **avg**, **count**, **delta**, **max**, **min** and **sum** functions.

#### 2.2 Forecasting horizon

**(forecast only)**

Parameter `time` specifies how far in the future Zabbix should extrapolate dependencies it finds in historic data. No matter if you use `time_shift` or not, `time` is always counted starting from the current moment.

#### 2.3 Threshold to reach

**(timeleft only)**

Parameter `threshold` specifies a value the analysed item has to reach, no difference if from above or from below. Once we have determined  $f(t)$  (see below) we should solve equation  $f(t) = \text{threshold}$  and return the root which is closer to now and to the right from now or 999999999999.9999 if there is

no such root.

## 2.4 Fit functions

Default fit is the *linear* function. But if your monitored system is more complicated you have more options to choose from.

fit	x = f(t)
<i>linear</i>	$x = a + b*t$
<i>polynomial</i> <sup>1)</sup>	$x = a_0 + a_1*t + a_2*t^2 + \dots + a_n*t^n$
<i>exponential</i>	$x = a*\exp(b*t)$
<i>logarithmic</i>	$x = a + b*\log(t)$
<i>power</i>	$x = a*t^b$

## 2.5 Modes

(**forecast** only)

Every time a trigger function is evaluated it gets data from the specified history period and fits a specified function to the data. So, if the data is slightly different the fitted function will be slightly different. If we simply calculate the value of the fitted function at a specified time in the future you will know nothing about how the analysed item is expected to behave between now and that moment in the future. For some fit options (like *polynomial*) a simple value from the future may be misleading.

mode	forecast result
<i>value</i>	$f(\text{now} + \text{time})$
<i>max</i>	$\max_{\text{now} \leq t \leq \text{now} + \text{time}} f(t)$
<i>min</i>	$\min_{\text{now} \leq t \leq \text{now} + \text{time}} f(t)$
<i>delta</i>	$\text{max} - \text{min}$
<i>avg</i>	average of $f(t)$ ( $\text{now} \leq t \leq \text{now} + \text{time}$ ) according to <a href="#">definition</a>

## 3 Details

To avoid calculations with huge numbers we consider the timestamp of the first value in specified period plus 1 ns as a new zero-time (current epoch time is of order  $10^9$ , epoch squared is  $10^{18}$ , double precision is about  $10^{-16}$ ). 1 ns is added to provide all positive time values for *logarithmic* and *power* fits which involve calculating  $\log(t)$ . Time shift does not affect *linear*, *polynomial*, *exponential* (apart from easier and more precise calculations) but changes the shape of *logarithmic* and *power* functions.

## 4 Potential errors

Functions return -1 in such situations:

- specified evaluation period contains no data;

- result of mathematical operation is not defined<sup>2)</sup>;
- numerical complications (unfortunately, for some sets of input data range and precision of double-precision floating-point format become insufficient)<sup>3)</sup>.

Note: No warnings or errors are flagged if chosen fit poorly describes provided data or there is just too few data for accurate prediction.

## See also

1. [Forecasting trigger functions \(pdf\)](#) on zabbix.org

<sup>1)</sup>

Polynomial degree can be from 1 to 6, *polynomial1* is equivalent to *linear*. However, use higher degree polynomials with caution ([More info here](#)). If the evaluation period contains less points than needed to determine polynomial coefficients, polynomial degree will be lowered (e.g *polynomial5* is requested, but there are only 4 points, therefore *polynomial3* will be fitted).

<sup>2)</sup>

For example fitting *exponential* or *power* functions involves calculating  $\log()$  of item values. If data contains zeros or negative numbers you will get an error since  $\log()$  is defined for positive values only.

<sup>3)</sup>

For *linear*, *exponential*, *logarithmic* and *power* fits all necessary calculations can be written explicitly. For *polynomial* only *value* can be calculated without any additional steps. Calculating *avg* involves computing polynomial antiderivative (analytically). Computing *max*, *min* and *delta* involves computing polynomial derivative (analytically) and finding its roots (numerically). Solving  $f(t) = 0$  involves finding polynomial roots (numerically).

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/manual/config/triggers/prediction?rev=1443441286>

Last update: **2017/10/06 14:12**

