

4 Low-level discovery macros

Overview

There is a type of macro used within the [low-level discovery](#) (LLD) function:

```
{#MACRO}
```

It is a macro that is used in an LLD rule and returns real values of the file system name, network interface, SNMP OID, etc.

These macros can be used for creating item, trigger and graph *prototypes*. Then, when discovering real file systems, network interfaces etc., these macros are substituted with real values and are the basis for creating real items, triggers and graphs.

These macros are also used in creating host and host group *prototypes* in virtual machine [discovery](#).

Some low-level discovery macros come “pre-packaged” with the LLD function in Zabbix - `{#FSNAME}`, `{#FSTYPE}`, `{#IFNAME}`, `{#SNMPINDEX}`, `{#SNMPVALUE}`. However, adhering to these names is not compulsory when creating a [custom](#) low-level discovery rule. Then you may use any other LLD macro name and refer to that name.

Supported locations

LLD macros can be used:

- in the low-level discovery rule filter
- for item prototypes in
 - name
 - key parameters
 - unit
 - update interval¹
 - history storage period¹
 - trend storage period¹
 - item value preprocessing steps
 - SNMP OID
 - IPMI sensor field
 - calculated item formula
 - SSH script and Telnet script
 - database monitoring SQL query
 - JMX item endpoint field
 - description
 - HTTP agent URL field
 - HTTP agent HTTP query fields field
 - HTTP agent request body field
 - HTTP agent required status codes field
 - HTTP agent headers field key and value
 - HTTP agent HTTP authentication username field
 - HTTP agent HTTP authentication password field

- HTTP agent HTTP proxy field
- HTTP agent HTTP SSL certificate file field
- HTTP agent HTTP SSL key file field
- HTTP agent HTTP SSL key password field
- HTTP agent HTTP timeout¹ field
- for trigger prototypes in
 - name
 - operational data
 - expression (only in constants and function parameters)
 - URL
 - description
 - event tag name and value
- for graph prototypes in
 - name
- for host prototypes in
 - name
 - visible name
 - host group prototype name
 - host macro value
 - (see the [full list](#))

In all those places LLD macros can be used inside static user [macro context](#).

Using macro functions

Macro functions are supported with low-level discovery macros (except in low-level discovery rule filter), allowing to extract a certain part of the macro value using a regular expression.

For example, you may want to extract the customer name and interface number from the following LLD macro for the purposes of event tagging:

```
{#IFALIAS}=customername_1
```

To do so, the `regsub` macro function can be used with the macro in the event tag value field of a trigger prototype:

Tags	
Customer	<code>{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}</code> Remove
Interface	<code>{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}</code> Remove

Note, that commas are not allowed in unquoted item [key parameters](#), so the parameter containing a macro function has to be quoted. The backslash (`\`) character should be used to escape double quotes inside the parameter. Example:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}", bytes]
```

For more information on macro function syntax, see: [Macro functions](#)

Macro functions are supported in low-level discovery macros since Zabbix 4.0.

Footnotes

¹ In the fields marked with ¹ a single macro has to fill the whole field. Multiple macros in a field or macros mixed with text are not supported.

From:

<https://www.zabbix.com/documentation/current/> - **Zabbix Documentation 5.0**

Permanent link:

https://www.zabbix.com/documentation/current/manual/config/macros/lld_macros

Last update: **2020/06/10 08:40**

