2.1 Macros supported by location

Overview

The table contains a complete list of macros supported by Zabbix.

- **X** means “supported” in that location
- The numbered macro syntax of {MACRO**<1-9>**} is used to reference hosts in the order in which they appear in a trigger expression. Thus, macros like {HOST.IP1}, {HOST.IP2}, {HOST.IP3} will expand to the IP of the first, second and third host in the trigger expression, providing the expression contains those hosts. Additionally {HOST.HOST**<1-9>**} is supported within {host:key.func(param)} macro in graph names. For example, {{HOST.HOST2}:key.func()} in the graph name will refer to the host of the second item in the graph.

▼MACRO▼	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	▼DESCRIPTION▼
{ACTION.ID}	X	X	X	X	X	X												*Numeric ID of the triggered action.* Supported since 2.2.0.
{ACTION.NAME}	X	X	X	X	X	X												*Name of the triggered action.* Supported since 2.2.0.
{DATE}	X	X	X	X	X	X												*Current date in yyyy.mm.dd. format.*
{DISCOVERY.DEVICE.IPADDRESS}		X																*IP address of the discovered device.* Available always, does not depend on host being added.
{DISCOVERY.DEVICE.DNS}		X																*DNS name of the discovered device.* Available always, does not depend on host being added.
{DISCOVERY.DEVICE.STATUS}		X																*Status of the discovered device*: can be either UP or DOWN.
{DISCOVERY.DEVICE.UPTIME}		X																*Time since the last change of discovery status for a particular device.* For example: 1h 29m. For devices with status DOWN, this is the period of their downtime.
{DISCOVERY.RULE.NAME}		X																*Name of the discovery rule that discovered the presence or absence of the device or service.*
{DISCOVERY.SERVICE.NAME}		X																*Name of the service that was discovered.* For example: HTTP.
{DISCOVERY.SERVICE.PORT}		X																*Port of the service that was discovered.* For example: 80.
{DISCOVERY.SERVICE.STATUS}		X																*Status of the discovered service:* can be either UP or DOWN.

{DISCOVERY.SERVICE.UPTIME}		X																*Time since the last change of discovery status for a particular service.* For example: 1h 29m. For services with status DOWN, this is the period of their downtime.
{ESC.HISTORY}	X			X	X	X												*Escalation history. Log of previously sent messages.* Shows previously sent notifications, on which escalation step they were sent and their status (*sent*, *in progress* or *failed*).
{EVENT.ACK.HISTORY}	X																	*Log of acknowledgements on the problem.*
{EVENT.ACK.STATUS}	X																	*Acknowledgement status of the event (Yes/No).*
{EVENT.AGE}	X	X	X	X	X	X												*Age of the event that triggered an action.* Useful in escalated messages.
{EVENT.DATE}	X	X	X	X	X	X												*Date of the event that triggered an action.*
{EVENT.ID}	X	X	X	X	X	X												*Numeric ID of the event that triggered an action.*
{EVENT.RECOVERY.DATE}	X			X	X	X												*Date of the recovery event.* Can be used in recovery messages only. Supported since 2.2.0.
{EVENT.RECOVERY.ID}	X			X	X	X												*Numeric ID of the recovery event.* Can be used in recovery messages only. Supported since 2.2.0.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	
{EVENT.RECOVERY.STATUS}	X			X	X	X												*Verbal value of the recovery event.* Can be used in recovery messages only. Supported since 2.2.0.
{EVENT.RECOVERY.TIME}	X			X	X	X												*Time of the recovery event.* Can be used in recovery messages only. Supported since 2.2.0.
{EVENT.RECOVERY.VALUE}	X			X	X	X												*Numeric value of the recovery event.* Can be used in recovery messages only. Supported since 2.2.0.
{EVENT.STATUS}	X	X	X	X	X	X												*Verbal value of the event that triggered an action.* Supported since 2.2.0.
{EVENT.TIME}	X	X	X	X	X	X												*Time of the event that triggered an action.*
{EVENT.VALUE}	X	X	X	X	X	X												*Numeric value of the event that triggered an action.* Supported since 2.2.0.
{HOST.CONN<1-9>}	X			X	X	X	X	X[2]	X			X		X	X[5]	X		*IP or host DNS name, depending on host settings* [3]. Supported in trigger names since 2.0.0.
{HOST.DNS<1-9>}	X			X	X	X	X	X[2]	X			X		X	X[5]	X		*Host DNS name* [3]. Supported in trigger names since 2.0.0.
{HOST.HOST<1-9>}	X		X	X	X	X	X	X	X			X		X	X[5]	X		*Host name.* {HOSTNAME<1-9>} is deprecated.
{HOST.ID}										X								Host ID.
{HOST.IP<1-9>}	X		X	X	X	X	X	X[2]	X			X		X	X[5]	X		*Host IP address* [3]. Supported since 2.0.0. {IPADDRESS<1-9>} is deprecated.
{HOST.METADATA}			X															*Host metadata.* Used only for active agent auto-registration. Supported since 2.2.0.
{HOST.NAME<1-9>}	X		X	X	X	X	X	X	X			X		X	X[5]	X		*Visible host name.* Supported since 2.0.0.
{HOST.PORT<1-9>}	X		X	X	X	X						X						*Host (agent) port* [3]. Supported in auto-registration since 2.0.0. Supported in trigger names, trigger descriptions, internal and trigger-based notifications since 2.2.2.

{HOSTGROUP.ID}										X									*Host group ID.*
{INVENTORY.ALIAS<1-9>}	X			X	X	X													*Alias field in host inventory.*
{INVENTORY.ASSET.TAG<1-9>}	X			X	X	X													*Asset tag field in host inventory.*
{INVENTORY.CHASSIS<1-9>}	X			X	X	X													*Chassis field in host inventory.*
{INVENTORY.CONTACT<1-9>}	X			X	X	X													*Contact field in host inventory.* `{PROFILE.CONTACT<1-9>}` is deprecated.
{INVENTORY.CONTRACT.NUMBER<1-9>}	X			X	X	X													*Contract number field in host inventory.*
{INVENTORY.DEPLOYMENT.STATUS<1-9>}	X			X	X	X													*Deployment status field in host inventory.*
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17		
{INVENTORY.HARDWARE<1-9>}	X			X	X	X													*Hardware field in host inventory.* `{PROFILE.HARDWARE<1-9>}` is deprecated.
{INVENTORY.HARDWARE.FULL<1-9>}	X			X	X	X													*Hardware (Full details) field in host inventory.*
{INVENTORY.HOST.NETMASK<1-9>}	X			X	X	X													*Host subnet mask field in host inventory.*
{INVENTORY.HOST.NETWORKS<1-9>}	X			X	X	X													*Host networks field in host inventory.*
{INVENTORY.HOST.ROUTER<1-9>}	X			X	X	X													*Host router field in host inventory.*
{INVENTORY.HW.ARCH<1-9>}	X			X	X	X													*Hardware architecture field in host inventory.*
{INVENTORY.HW.DATE.DECOMM<1-9>}	X			X	X	X													*Date hardware decommissioned field in host inventory.*
{INVENTORY.HW.DATE.EXPIRY<1-9>}	X			X	X	X													*Date hardware maintenance expires field in host inventory.*
{INVENTORY.HW.DATE.INSTALL<1-9>}	X			X	X	X													*Date hardware installed field in host inventory.*
{INVENTORY.HW.DATE.PURCHASE<1-9>}	X			X	X	X													*Date hardware purchased field in host inventory.*
{INVENTORY.INSTALLER.NAME<1-9>}	X			X	X	X													*Installer name field in host inventory.*
{INVENTORY.LOCATION<1-9>}	X			X	X	X													*Location field in host inventory.* `{PROFILE.LOCATION<1-9>}` is deprecated.
{INVENTORY.LOCATION.LAT<1-9>}	X			X	X	X													*Location latitude field in host inventory.*
{INVENTORY.LOCATION.LON<1-9>}	X			X	X	X													*Location longitude field in host inventory.*
{INVENTORY.MACADDRESS.A<1-9>}	X			X	X	X													*MAC address A field in host inventory.* `{PROFILE.MACADDRESS<1-9>}` is deprecated.
{INVENTORY.MACADDRESS.B<1-9>}	X			X	X	X													*MAC address B field in host inventory.*
{INVENTORY.MODEL<1-9>}	X			X	X	X													*Model field in host inventory.*
{INVENTORY.NAME<1-9>}	X			X	X	X													*Name field in host inventory.* `{PROFILE.NAME<1-9>}` is deprecated.
{INVENTORY.NOTES<1-9>}	X			X	X	X													*Notes field in host inventory.* `{PROFILE.NOTES<1-9>}` is deprecated.
{INVENTORY.OOB.IP<1-9>}	X			X	X	X													*OOB IP address field in host inventory.*
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17		
{INVENTORY.OOB.NETMASK<1-9>}	X			X	X	X													*OOB subnet mask field in host inventory.*
{INVENTORY.OOB.ROUTER<1-9>}	X			X	X	X													*OOB router field in host inventory.*
{INVENTORY.OS<1-9>}	X			X	X	X													*OS field in host inventory.* `{PROFILE.OS<1-9>}` is deprecated.
{INVENTORY.OS.FULL<1-9>}	X			X	X	X													*OS (Full details) field in host inventory.*
{INVENTORY.OS.SHORT<1-9>}	X			X	X	X													*OS (Short) field in host inventory.*
{INVENTORY.POC.PRIMARY.CELL<1-9>}	X			X	X	X													*Primary POC cell field in host inventory.*
{INVENTORY.POC.PRIMARY.EMAIL<1-9>}	X			X	X	X													*Primary POC email field in host inventory.*
{INVENTORY.POC.PRIMARY.NAME<1-9>}	X			X	X	X													*Primary POC name field in host inventory.*
{INVENTORY.POC.PRIMARY.NOTES<1-9>}	X			X	X	X													*Primary POC notes field in host inventory.*
{INVENTORY.POC.PRIMARY.PHONE.A<1-9>}	X			X	X	X													*Primary POC phone A field in host inventory.*
{INVENTORY.POC.PRIMARY.PHONE.B<1-9>}	X			X	X	X													*Primary POC phone B field in host inventory.*

{INVENTORY.POC.PRIMARY.SCREEN<1-9>}	X			X	X	X												*Primary POC screen name field in host inventory.*
{INVENTORY.POC.SECONDARY.CELL<1-9>}	X			X	X	X												*Secondary POC cell field in host inventory.*
{INVENTORY.POC.SECONDARY.EMAIL<1-9>}	X			X	X	X												*Secondary POC email field in host inventory.*
{INVENTORY.POC.SECONDARY.NAME<1-9>}	X			X	X	X												*Secondary POC name field in host inventory.*
{INVENTORY.POC.SECONDARY.NOTES<1-9>}	X			X	X	X												*Secondary POC notes field in host inventory.*
{INVENTORY.POC.SECONDARY.PHONE.A<1-9>}	X			X	X	X												*Secondary POC phone A field in host inventory.*
{INVENTORY.POC.SECONDARY.PHONE.B<1-9>}	X			X	X	X												*Secondary POC phone B field in host inventory.*
{INVENTORY.POC.SECONDARY.SCREEN<1-9>}	X			X	X	X												*Secondary POC screen name field in host inventory.*
{INVENTORY.SERIALNO.A<1-9>}	X			X	X	X												*Serial number A field in host inventory.* `{PROFILE.SERIALNO<1-9>}` is deprecated.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	
{INVENTORY.SERIALNO.B<1-9>}	X			X	X	X												*Serial number B field in host inventory.*
{INVENTORY.SITE.ADDRESS.A<1-9>}	X			X	X	X												*Site address A field in host inventory.*
{INVENTORY.SITE.ADDRESS.B<1-9>}	X			X	X	X												*Site address B field in host inventory.*
{INVENTORY.SITE.ADDRESS.C<1-9>}	X			X	X	X												*Site address C field in host inventory.*
{INVENTORY.SITE.CITY<1-9>}	X			X	X	X												*Site city field in host inventory.*
{INVENTORY.SITE.COUNTRY<1-9>}	X			X	X	X												*Site country field in host inventory.*
{INVENTORY.SITE.NOTES<1-9>}	X			X	X	X												*Site notes field in host inventory.*
{INVENTORY.SITE.RACK<1-9>}	X			X	X	X												*Site rack location field in host inventory.*
{INVENTORY.SITE.STATE<1-9>}	X			X	X	X												*Site state/province field in host inventory.*
{INVENTORY.SITE.ZIP<1-9>}	X			X	X	X												*Site ZIP/postal field in host inventory.*
{INVENTORY.SOFTWARE<1-9>}	X			X	X	X												*Software field in host inventory.* `{PROFILE.SOFTWARE<1-9>}` is deprecated.
{INVENTORY.SOFTWARE.APP.A<1-9>}	X			X	X	X												*Software application A field in host inventory.*
{INVENTORY.SOFTWARE.APP.B<1-9>}	X			X	X	X												*Software application B field in host inventory.*
{INVENTORY.SOFTWARE.APP.C<1-9>}	X			X	X	X												*Software application C field in host inventory.*
{INVENTORY.SOFTWARE.APP.D<1-9>}	X			X	X	X												*Software application D field in host inventory.*
{INVENTORY.SOFTWARE.APP.E<1-9>}	X			X	X	X												*Software application E field in host inventory.*
{INVENTORY.SOFTWARE.FULL<1-9>}	X			X	X	X												*Software (Full details) field in host inventory.*
{INVENTORY.TAG<1-9>}	X			X	X	X												*Tag field in host inventory.* `{PROFILE.TAG<1-9>}` is deprecated.
{INVENTORY.TYPE<1-9>}	X			X	X	X												*Type field in host inventory.* `{PROFILE.DEVICETYPE<1-9>}` is deprecated.
{INVENTORY.TYPE.FULL<1-9>}	X			X	X	X												*Type (Full details) field in host inventory.*
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	
{INVENTORY.URL.A<1-9>}	X			X	X	X												*URL A field in host inventory.*
{INVENTORY.URL.B<1-9>}	X			X	X	X												*URL B field in host inventory.*
{INVENTORY.URL.C<1-9>}	X			X	X	X												*URL C field in host inventory.*
{INVENTORY.VENDOR<1-9>}	X			X	X	X												*Vendor field in host inventory.*
{ITEM.DESCRIPTION<1-9>}	X			X	X	X												*Description of the Nth item in the trigger expression that caused a notification.* Supported since 2.0.0.
{ITEM.ID<1-9>}	X			X	X	X												*Numeric ID of the Nth item in the trigger expression that caused a notification.* Supported since 1.8.12.

{ITEM.KEY<1-9>}	X			X	X	X												*Key of the Nth item in the trigger expression that caused a notification.* Supported since 2.0.0. `{TRIGGER.KEY}` is deprecated.
{ITEM.KEY.ORIG<1-9>}	X			X	X	X												*Original key (with macros not expanded) of the Nth item in the trigger expression that caused a notification.* Supported since 2.0.6.
{ITEM.LASTVALUE<1-9>}	X										X							*The latest value of the Nth item in the trigger expression that caused a notification.* It will resolve to *UNKNOWN* in the frontend if the latest history value has been collected more than the *ZBX_HISTORY_PERIOD* time ago (defined in defines.inc.php). Supported since 1.4.3. It is alias to `{{HOST.HOST}:{ITEM.KEY}.last()}`
{ITEM.LOG.AGE<1-9>}	X																	*Age of the log item event.*
{ITEM.LOG.DATE<1-9>}	X																	*Date of the log item event.*
{ITEM.LOG.EVENTID<1-9>}	X																	*ID of the event in the event log.* For Windows event log monitoring only.
{ITEM.LOG.NSEVERITY<1-9>}	X																	*Numeric severity of the event in the event log.* For Windows event log monitoring only.
{ITEM.LOG.SEVERITY<1-9>}	X																	*Verbal severity of the event in the event log.* For Windows event log monitoring only.
{ITEM.LOG.SOURCE<1-9>}	X																	*Source of the event in the event log.* For Windows event log monitoring only.
{ITEM.LOG.TIME<1-9>}	X																	*Time of the log item event.*
{ITEM.NAME<1-9>}	X			X	X	X												*Name of the Nth item in the trigger expression that caused a notification.*
{ITEM.NAME.ORIG<1-9>}	X			X	X	X												*Original name (with macros not expanded) of the Nth item in the trigger expression that caused a notification.* Supported since 2.0.6.
{ITEM.STATE<1-9>}					X													*The latest state of the Nth item in the trigger expression that caused a notification.* Possible values: **Not supported** and **Normal**. Supported since 2.2.0.
{ITEM.VALUE<1-9>}	X										X							Resolved to either: 1) the historical (at-the-time-of-event) value of the Nth item in the trigger expression, if used in the context of trigger status change, for example, when displaying events or sending notifications. 2) the latest value of the Nth item in the trigger expression, if used without the context of trigger status change, for example, when displaying a list of triggers in a pop-up selection window. In this case works the same as {ITEM.LASTVALUE} In the first case it will resolve to *UNKNOWN* if the history value has already been deleted or has never been stored. In the second case, and in the frontend only, it will resolve to *UNKNOWN* if the latest history value has been collected more than the *ZBX_HISTORY_PERIOD* time ago (defined in defines.inc.php). Supported since 1.4.3.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	
{LLDRULE.DESCRIPTION}						X												*Description of the low-level discovery rule which caused a notification.* Supported since 2.2.0.

{LLDRULE.ID}						X											*Numeric ID of the low-level discovery rule which caused a notification.* Supported since 2.2.0.
{LLDRULE.KEY}						X											*Key of the low-level discovery rule which caused a notification.* Supported since 2.2.0.
{LLDRULE.KEY.ORIG}						X											*Original key (with macros not expanded) of the low-level discovery rule which caused a notification.* Supported since 2.2.0.
{LLDRULE.NAME}						X											*Name of the low-level discovery rule which caused a notification.* Supported since 2.2.0.
{LLDRULE.NAME.ORIG}						X											*Original name (with macros not expanded) of the low-level discovery rule which caused a notification.* Supported since 2.2.0.
{LLDRULE.STATE}						X											*The latest state of the low-level discovery rule.* Possible values: **Not supported** and **Normal**. Supported since 2.2.0.
{MAP.ID}									X								*Network map ID.*
{NODE.ID<1-9>}	X	X	X	X	X	X											
{NODE.NAME<1-9>}	X	X	X	X	X	X											
{PROXY.NAME<1-9>}	X	X	X	X	X	X											*Name of the proxy*. Resolves to either: 1) proxy of the Nth item in the trigger expression (in trigger-based notifications). You may use indexed macros here, like {PROXY.NAME1}, {PROXY.NAME2}, etc. 2) proxy, which executed discovery (in discovery notifications). Use {PROXY.NAME} here, without indexing. 3) proxy to which an active agent registered (in auto-registration notifications). Use {PROXY.NAME} here, without indexing. Supported since 1.8.4.
{TIME}	X	X	X	X	X	X											*Current time in hh:mm:ss.*
{TRIGGER.DESCRIPTION}	X			X													*Trigger description.* Supported since 2.0.4. Starting with 2.2.0, all macros supported in a trigger description will be expanded if `{TRIGGER.DESCRIPTION}` is used in notification text. `{TRIGGER.COMMENT}` is deprecated.
{TRIGGER.EVENTS.ACK}	X								X								*Number of acknowledged events for a map element in maps, or for the trigger which generated current event in notifications.* Supported since 1.8.3.
{TRIGGER.EVENTS.PROBLEM.ACK}	X								X								*Number of acknowledged PROBLEM events for all triggers disregarding their state.* Supported since 1.8.3.
{TRIGGER.EVENTS.PROBLEM.UNACK}	X								X								*Number of unacknowledged PROBLEM events for all triggers disregarding their state.* Supported since 1.8.3.
{TRIGGER.EVENTS.UNACK}	X								X								*Number of unacknowledged events for a map element in maps, or for the trigger which generated current event in notifications.* Supported in map element labels since 1.8.3.
{TRIGGER.HOSTGROUP.NAME}	X			X													*A sorted (by SQL query), comma-space separated list of host groups in which the trigger is defined.* Supported since 2.0.6.
{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}									X								*Number of acknowledged PROBLEM events for triggers in PROBLEM state.* Supported since 1.8.3.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}								X										*Number of unacknowledged PROBLEM events for triggers in PROBLEM state.* Supported since 1.8.3.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	
{TRIGGER.EXPRESSION}	X			X														*Trigger expression.* Supported since 1.8.12.
{TRIGGER.ID}	X			X						X								*Numeric trigger ID which triggered this action.* Supported in trigger URLs since Zabbix 1.8.8.
{TRIGGER.NAME}	X			X														*Name of the trigger.*
{TRIGGER.NAME.ORIG}	X			X														*Original name (with macros not expanded) of the trigger.* Supported since 2.0.6.
{TRIGGER.NSEVERITY}	X			X														*Numerical trigger severity.* Possible values: 0 - Not classified, 1 - Information, 2 - Warning, 3 - Average, 4 - High, 5 - Disaster. Supported starting from Zabbix 1.6.2.
{TRIGGER.SEVERITY}	X			X														*Trigger severity name.* Can be defined in *Administration → General → Trigger severities*.
{TRIGGER.STATE}				X														*The latest state of the trigger.* Possible values: **Unknown** and **Normal**. Supported since 2.2.0.
{TRIGGER.STATUS}	X																	*Current trigger value.* Can be either PROBLEM or OK. {STATUS} is deprecated.
{TRIGGER.TEMPLATE.NAME}	X			X														*A sorted (by SQL query), comma-space separated list of templates in which the trigger is defined, or *UNKNOWN* if the trigger is defined in a host.* Supported since 2.0.6.
{TRIGGER.URL}	X			X														*Trigger URL.*
{TRIGGER.VALUE}	X										X							*Current trigger numeric value*: 0 - trigger is in OK state, 1 - trigger is in PROBLEM state.
{TRIGGERS.UNACK}									X									*Number of unacknowledged triggers for a map element, disregarding trigger state.* A trigger is considered to be unacknowledged if at least one of its PROBLEM events is unacknowledged.
{TRIGGERS.PROBLEM.UNACK}									X									*Number of unacknowledged PROBLEM triggers for a map element.* A trigger is considered to be unacknowledged if at least one of its PROBLEM events is unacknowledged. Supported since 1.8.3.
{TRIGGERS.ACK}									X									*Number of acknowledged triggers for a map element, disregarding trigger state.* A trigger is considered to be acknowledged if all of it's PROBLEM events are acknowledged. Supported since 1.8.3.
{TRIGGERS.PROBLEM.ACK}									X									*Number of acknowledged PROBLEM triggers for a map element.* A trigger is considered to be acknowledged if all of it's PROBLEM events are acknowledged. Supported since 1.8.3.
{host:key.func(param)}	X								X[4]		X[9]						X[7]	*Simple macros, as used in building trigger expressions*.
{$MACRO}							X	X			X[8]	X	X	X	X	X		*User-definable macros.* Supported in item and trigger names since 1.8.4. Supported in global script commands and confirmation texts since Zabbix 2.2.0.
	1	**2**	3	**4**	5	**6**	7	**8**	9	**10**	11	**12**	13	**14**	15	**16**	17	

Footnotes

[1] Macros for map labels are supported since 1.8.

[2] The {HOST.*} macros supported in item key parameters will resolve to the interface that is selected for the item. When used in items without interfaces they will resolve to either the Zabbix agent, SNMP, JMX or IPMI interface of the host in this order of priority, since Zabbix 2.2.16. In Zabbix 2.0.3-2.2.15 they will not resolve when used in items without interfaces e.g. “Zabbix agent (active)”, “Calculated” etc.

[3] In remote commands, global scripts, interface IP/DNS fields and web scenarios the macro will resolve to the main agent interface, however, if it is not present, the main SNMP interface will be used. If SNMP is also not present, the main JMX interface will be used. If JMX is not present either, the main IPMI interface will be used.

[4] This macro is supported in icon labels and link labels in maps. Only the **avg**, **last**, **max** and **min** functions, with seconds as parameter are supported in this macro.

[5] Supported since 2.0.3.

[6] Supported since Zabbix 2.2.0. {HOST.*} macros and user-defined macros {$MACRO} are supported in web scenario *Name* and *Variables* fields and in scenario step *Name*, *URL*, *Post* and *Required string* fields. {$MACRO} is also supported in web scenario *Authentication* (user and password), *Agent* and *HTTP proxy* fields and in the scenario step *Required status codes* field.

[7] Supported since Zabbix 2.2.0. Only the **avg**, **last**, **max** and **min** functions, with seconds as parameter are supported within this macro in graph names. The {HOST.HOST<1-9>} macro can be used as host within the macro. For example:

- {Cisco switch:ifAlias[{#SNMPINDEX}].last()}
- {{HOST.HOST}:ifAlias[{#SNMPINDEX}].last()}

[8] Only in trigger expression constants and function parameters.

[9] While supported to build trigger expressions, simple macros may not be used inside each other.

Additional support for user macros

In addition to the locations listed, user-definable macros since Zabbix 2.0 are supported in numerous other locations:

- Hosts
  - Interface IP/DNS
  - Interface port
- Passive proxy
  - Interface port
- Items and item prototypes

    - SNMPv3 context name
    - SNMPv3 security name
    - SNMPv3 auth pass
    - SNMPv3 priv pass
    - SNMPv1/v2 community
    - SNMP OID
    - SNMP port
    - SSH username
    - SSH public key
    - SSH private key
    - SSH password
    - Telnet username
    - Telnet password
    - Calculated item formula
    - Trapper item “Allowed hosts” field (*since Zabbix 2.2*)

- Discovery
    - SNMPv3 context name
    - SNMPv3 security name
    - SNMPv3 auth pass
    - SNMPv3 priv pass
    - SNMPv1/v2 community
    - SNMP OID

Macros used in low-level discovery

There is a type of macro used within the low-level discovery function - **{#MACRO}**. It is a macro that is used in an LLD rule and returns real values of file system names, network interfaces and SNMP OIDs.

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.

LLD macros can be used:

- for item prototypes in
    - names
    - key parameters
    - SNMP OIDs
    - calculated item formulas
    - SSH and Telnet scripts
    - database monitoring SQL queries
    - descriptions (supported since 2.2.0)
- for trigger prototypes in
    - names
    - expressions (insofar as when referencing an item key prototype and as standalone constants)
    - descriptions (supported since 2.2.0)

- for graph prototypes in
  - names
- for host prototypes (supported since 2.2.0) in
  - names
  - visible names
  - host group prototype names
  - (see the full list)

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.