PREPROCESSING - HOW TO GET WHAT YOU WANT TO GET WITH 5.0

Aleksandrs Petrovs-Gavrilovs
Technical Support Engineer
WHAT IS PREPROCESSING?
ZABBIX PREPROCESSING
WHAT IS PREPROCESSING?

- In computer science, a preprocessor is a program that processes its input data to produce output that is used as input to another program. Both require tuning (e.g., buffer and transaction log) and is important.

- Data preprocessing, used in learning and data collecting to make input data easier to work with.
WHAT IS PREPROCESSING?

- Preprocessing allows to define transformation rules for the received item values.
- One or several transformations are possible before saving values to the database.
- Transformations are executed in the order in which they are defined.
- Preprocessing is done either by Zabbix server or by Zabbix proxy (for items monitored by proxy).
WHY DO WE NEED PREPROCESSING?
WHY DO WE NEED **PREPROCESSING**?
WHY DO WE NEED **PREPROCESSING**?

- Data cleansing
- Data editing
- Data reduction
- Data wrangling
DATA **CLEANSING?**

- Define a range by specifying minimum/maximum values (inclusive).
- Specify a regular expression that will look for an error or a value that must match or not match.
- Check for an application-level error message located at JSONpath.
- Check for an application-level error message located at xpath.
DEFINING A **RANGE**

In the image, there are two screenshots showing the pre-processing steps for defining a range in Zabbix. The screenshots display the following:

1. **Preprocessing steps**:
   - **Name**: `in range`
   - **Parameters**: `1` and `100`
   - **Custom on fail**: Discard value, Set value to, Set error to

2. **Preprocessing steps**:
   - **Name**: `in range`
   - **Parameters**: `1` and `999`
   - **Custom on fail**: Discard value, Set value to, Set error to

Both screenshots include buttons for **Add**, **Test**, and **Cancel**.
# Use a Regular Expression

## Preprocessing

<table>
<thead>
<tr>
<th>Preprocessing steps</th>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Matches regular expression</td>
<td>(?error)</td>
<td></td>
<td>Test, Remove</td>
</tr>
<tr>
<td></td>
<td>Custom on fail</td>
<td>Discard</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set value to</td>
<td>Set value to</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set error to</td>
<td>Set error to</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Add**
- Add
- Test
- Cancel

## Preprocessing

<table>
<thead>
<tr>
<th>Preprocessing steps</th>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Does not match regular expression</td>
<td>(?error)</td>
<td></td>
<td>Test, Remove</td>
</tr>
<tr>
<td></td>
<td>Custom on fail</td>
<td>Discard</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set value to</td>
<td>Set value to</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set error to</td>
<td>Set error to</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Add**
- Add
- Test
- Cancel

Test all steps
USE MORE REGULAR EXPRESSIONS

☑  ^(automatic|automatic delayed|manual|disabled)$
   Windows service startup states

☑  ([1-9]+)
   collect only numbers

☑  ((?>[a-z\-0-9]{2,}\.){1,}[a-z]{2,8})(?::\s|\/)
   domain name

☑  ([0-9A-Za-z\&\-\./\(\)\=\:\;\+])((?::\|\=)\(?::\)?(?::\)|D|P))
   smiley faces
### Use JSONPath

#### Preprocessing

**Preprocessing steps**

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for error in JSON</td>
<td>$.myjson.error</td>
<td></td>
<td>Test Remove</td>
</tr>
<tr>
<td>Custom on fail</td>
<td>Discard value Set value to Set error to</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Add**

- Add
- Test
- Cancel

---

**Preprocessing steps**

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for error in XML</td>
<td>/address/message/error</td>
<td></td>
<td>Test Remove</td>
</tr>
<tr>
<td>Custom on fail</td>
<td>Discard value Set value to Set error to There is an error</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Add**

- Add
- Test
- Cancel

---

**Preprocessing steps**

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for error in XML</td>
<td>/address/message/error</td>
<td></td>
<td>Test Remove</td>
</tr>
<tr>
<td>Custom on fail</td>
<td>Discard value Set value to Set error to There is an error</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Add**

- Add
- Test
- Cancel
DATA EDITING?

☑ Replace.
☑ Trim.
☑ Use regex.
☑ Multiply.
☑ Javascript.
### Preprocessing

<table>
<thead>
<tr>
<th>Preprocessing steps</th>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Replace</td>
<td>error</td>
<td></td>
<td>Test, Remove</td>
</tr>
<tr>
<td></td>
<td>Add</td>
<td>Test, Cancel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preprocessing steps</th>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular expression</td>
<td><em>error</em></td>
<td></td>
<td>Test, Remove</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Replace</td>
<td></td>
<td>Test, Remove</td>
</tr>
<tr>
<td></td>
<td>Add</td>
<td>Test, Cancel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TRIM

Removes provided symbols on both sides (right/left) with **Trim**
- \( t \ 36 \ C \) \quad \text{Trim: } tC \quad 36

Remove provided symbols on right side with **Right trim**
- \( t \ 36 \ F \) \quad \text{Right trim: } F \quad t \ 36

Remove provided symbols on left side with **Left trim**
- \( t \ 36 \ K \) \quad \text{Left trim: } t \quad 36 \ K
USE EVEN MORE **REGULAR EXPRESSIONS**

- ^(eth[0-9]$)|(^eth[0-9]:[1-9]$)
  
  *Look for network adapters*

- ^(ntfs|fat32|zfs)$
  
  **Short filesystem regex**

- ^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$
  
  **SNMP storage devices**
MULTIPLY

Kilobytes to bytes

Millisecsconds to seconds

Note that if the item type of information is Numeric (unsigned), incoming values with a fractional part will be trimmed (i.e. '0.9' will become '0') before the custom multiplier is applied.
Get time till the end of certificate in seconds

```javascript
var split = value.split(' '),
month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
now = Date.now();
return parseInt((Date.parse(ISOdate) - now) / 1000);
```
Convert SNMP DateandTime to Timestamp

'use strict';
var str = value;
alert("str: " + str);
var y256 = str.slice(0,2); var y = str.slice(3,5); var m = str.slice(6,8);
var d = str.slice(9,11); var h = str.slice(12,14); var min = str.slice(15,17);
var y256Base10 = +("0x" + y256);
var yBase10 = +("0x" + y);
var Year = 256*y256Base10 + yBase10;
var mBase10 = +("0x" + m);
var dBase10 = +("0x" + d);
var hBase10 = +("0x" + h);
var minBase10 = +("0x" + min);
var YR = String(Year); var MM = String(mBase10); var DD = String(dBase10);
var HH = String(hBase10);
var MIN = String(minBase10);
if (mBase10 < 10) MM = "0" + MM; if (dBase10 < 10) DD = "0" + DD;
if (hBase10 < 10) HH = "0" + HH; if (minBase10 < 10) MIN = "0" + MIN;
var Date = YR + "-" + MM + "-" + DD + " " + HH + ":" + MIN;
return Date;
USE MORE JAVASCRIPT

Convert SNMP DateandTime to Timestamp
DATA REDUCTION?

- ✔ XML XPath.
- ✔ JSON Path.
- ✔ Use regex.
- ✔ Throttling.
- ✔ Replace
XML XPATH

Prepare an XML

```xml
<rhs_check>
  <network>
    <ping host="cdn.redhat.com">passed</ping>
    <reverse_dns>passed</reverse_dns>
    <port address="80">passed</port>
    <port address="8080">passed</port>
    <port address="5671">passed</port>
    <port address="443">passed</port>
    <port address="8140">passed</port>
    <port address="9090">passed</port>
  </network>
  <services>
    <service name="mongod">active</service>
    <service name="qpidd">active</service>
    <service name="qdrouterd">active</service>
    <service name="tomcat">active</service>
    <service name="pulp_resource_manager">active</service>
    <service name="pulp_workers">active</service>
    <service name="httpd">active</service>
    <service name="puppet">inactive</service>
    <service name="elasticsearch">unknown</service>
  </services>
</rhs_check>
```
**XML XPATH**

Find a service you are looking for:

/rhs_check/services/service[1]

Add some more steps:
THE PATH OF THE JSON

{ "obj2testobj2Connection": "OK", "objam.testHttpConnection": "OK", "obj2.testobj2ServerComponents": "OK", "tomcat.freememory.threshold": "2132131000", "tomcat.freememory": "OK", "tomcat.freememory.actual": "13123131121", "objam.testNFSStore": "OK" }
BEFORE THROTTLING
BEFORE THROTTLING
### Preprocessing Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Name</th>
<th>Parameters</th>
<th>Custom on fail</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Replace</td>
<td>up</td>
<td></td>
<td>Test, Remove</td>
</tr>
<tr>
<td>2</td>
<td>Replace</td>
<td>down</td>
<td></td>
<td>Test, Remove</td>
</tr>
</tbody>
</table>

**Add**

- Update
- Clone
- Test
- Delete
- Cancel
### Test Item

<table>
<thead>
<tr>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>now</td>
</tr>
</tbody>
</table>

**Previous value**

<table>
<thead>
<tr>
<th>Previous value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prev time</td>
</tr>
</tbody>
</table>

**End of line sequence**

- LF
- CRLF

**Preprocessing steps**

1. Replace
2. Replace

**Result**

- Result converted to Numeric (unsigned)
- Result with value map applied

### Test Item

<table>
<thead>
<tr>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>now</td>
</tr>
</tbody>
</table>

**Previous value**

<table>
<thead>
<tr>
<th>Previous value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prev time</td>
</tr>
</tbody>
</table>

**End of line sequence**

- LF
- CRLF

**Preprocessing steps**

1. Replace
2. Replace

**Result**

- Result converted to Numeric (unsigned)
- Result with value map applied

### Test Item

<table>
<thead>
<tr>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>now</td>
</tr>
</tbody>
</table>

**Previous value**

<table>
<thead>
<tr>
<th>Previous value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prev time</td>
</tr>
</tbody>
</table>

**End of line sequence**

- LF
- CRLF

**Preprocessing steps**

1. Replace
2. Replace

**Result**

- Result converted to Numeric (unsigned)
- Result with value map applied

### Test Item

<table>
<thead>
<tr>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>now</td>
</tr>
</tbody>
</table>

**Previous value**

<table>
<thead>
<tr>
<th>Previous value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prev time</td>
</tr>
</tbody>
</table>

**End of line sequence**

- LF
- CRLF

**Preprocessing steps**

1. Replace
2. Replace

**Result**

- Result converted to Numeric (unsigned)
- Result with value map applied

### Test Item

<table>
<thead>
<tr>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>now</td>
</tr>
</tbody>
</table>

**Previous value**

<table>
<thead>
<tr>
<th>Previous value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prev time</td>
</tr>
</tbody>
</table>

**End of line sequence**

- LF
- CRLF

**Preprocessing steps**

1. Replace
2. Replace

**Result**

- Result converted to Numeric (unsigned)
- Result with value map applied
DATA WRANGLING?

- Boolean/Octal/Hexadecimal to decimal.
- CSV to JSON.
- Prometheus pattern
- Simple change
- Change per second
**BOOLEAN TO DECIMAL**

- **TRUE** - true, t, yes, y, on, up, running, enabled, available
- **FALSE** - false, f, no, n, off, down, unused, disabled, unavailable
OCTAL/HEXADECIMAL TO DECIMAL
CSV TO JSON

[{"Identifier":"9012","First name":"Rachel","Last name":"Booker"},
{"Identifier":"2070","First name":"Laura","Last name":"Grey"},
{"Identifier":"4081","First name":"Craig","Last name":"Johnson"},
{"Identifier":"9346","First name":"Mary","Last name":"Jenkins"},
{"Identifier":"5079","First name":"Jamie","Last name":"Smith"}]}
CSV TO JSON

Test item

- Value: Identifier, First name, Last name...
- Time: now
- Previous value:
- End of line sequence: LF, CRLF
- Preprocessing steps:
  1: CSV to JSON
  2: JSONPath

Result: Result converted to Log
PROMETHEUS PATTERN

# HELP wmi_logical_disk_free_bytes Free space in bytes
(LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08

wmi_service_state{name="dhcp",state="continue pending"} 0
wmi_service_state{name="dhcp",state="pause pending"} 0
wmi_service_state{name="dhcp",state="paused"} 0
wmi_service_state{name="dhcp",state="running"} 1
wmi_service_state{name="dhcp",state="start pending"} 0
wmi_service_state{name="dhcp",state="stop pending"} 0
wmi_service_state{name="dhcp",state="stopped"} 0
wmi_service_state{name="dhcp",state="unknown"} 0
PROMETHEUS PATTERN
SIMPLE CHANGE

- Simple change - Calculate difference between the current and previous value. Evaluated as `value-prev_value`, where value - current value; prev_value - previously received value. This setting can be useful to measure a constantly growing value.
CHANGE PER SECOND

- Change per second - Calculate the value change (difference between the current and previous value) speed per second. Evaluated as \( \frac{\text{value} - \text{prev_value}}{\text{time} - \text{prev_time}} \), where value - current value; prev_value - previously received value; time - current timestamp; prev_time - timestamp of previous value.
03

HOW DOES IT WORK?
Item can be placed at the end or at the beginning of the preprocessing queue. Zabbix internal items are always placed at the beginning of preprocessing queue, while other item types are enqueued at the end.
5.0 IMPROVEMENTS

- Maximum count of dependent items for one master item is now 29999 (was 999)
- [ZBX-17694] Memory usage became significantly lower
  15K dependent items with master worth of 209 KB needed around 3 GB RAM with total busy time in manager of 2.115940 seconds, but after changes 4.6 MB is the peak and busy time is 0.688623 seconds.
- [ZBX-17720] Exclude disabled dependant items from preprocessing configuration
- [ZBX-17548] Don't store history in Proxy DB If History storage period is 0
Thank you!