

 Table of Contents

The document is ready to be reviewed by spec team, partners and community members.

Stakeholders sign off status

alex@zabbix.com -

Zabbix Acceptance

Item preprocessing rules will be internally classified as:

- transformation rules: transform input value into output value using function, fail if something goes wrong
- validation rules: validate input value, fail if input value is not valid

New validation rules must be supported (menu → submenu) for items and item prototypes:

- Validation → In range [min] [max]: check that numeric value is in range of 'min' to 'max', fail otherwise
- Validation → Matches regular expression [regex]: check that value matches 'regex', fail otherwise
- Validation → Does not match regular expression [regex]: check that value does not match 'regex', fail otherwise
- Throttling → Discard value if new value differs from last one
- Throttling → Discard value if new value differs from last one and time difference is less than N seconds [N]

Zabbix user may specify optional "On fail" action for each rule, which will be executed if this rule fails:

1. **Discard value**: preprocessing will stop immediately and Zabbix will ignore the value as it never existed
 - a. It must be set as a default action in for all Throttling validation rules in UI form
2. **Set error** + "<error message>": preprocessing will fail and error message will be set to provided non empty "<error message>"
3. If no action is specified then **Set error** is executed with internal error message set by Zabbix Server

The following preprocessing rules will never fail:

- Trim, Left trim, Right trim: always successful
- Boolean to decimal: always successful
- Throttling → Discard value if new value differs from last one
- Throttling → Discard value if new value differs from last one and time difference is less than N seconds

Other transformation and validation rules will fail if (depending on rule):

- no match or failed for whatever reason, also:
 - Custom multiplier: not numeric, out of range
 - Simple change, Change per second: not numeric, out of range
 - Octal to decimal, Hexadecimal to decimal: not octal/hex, out of range
 - In range: not numeric

Design of the preprocessing configuration must not be too cluttered, it must be as simple as possible for those who will not use this functionality.

When comparing values the new value must be casted to the type (unsigned int, float) of the last value stored in the database. Therefore, new string value "123.00" will be equal to the old numeric value of "123".

Use cases

1. Skip erroneous (incorrect, noise) values. It may happen if a counter is not utilized yet (empty or 'N/A')
2. Skip out of range values. For example, temperature sensor returns +999C if it is off, normal range is -100C up-to +100C
3. Skip values that matches some regular expression
4. Set a human readable error message in case if a preprocessing rule fails
5. Set a human readable message in case if a value is out of range or does not match some pattern. For example, I expect temperature in '[0-9]+C' format
6. Process value only if it is changed
7. Support of [subsampling](#) for dependant items. Master is updated every 10 seconds, a dependant item should be updated every minute
8. Collect a value every 10 seconds. Unconditionally process it every 60 seconds (for nice graphs, for example) or if a new value differs from last one for real-time alerting

Open questions

1. N/A