

Summary

Existing functionality of Zabbix preprocessing is limited for a set of supported preprocessing rules. For ultimate flexibility Zabbix should give users a way to create their own preprocessing rules.

Support of JavaScript will allow users to extend pre-processing capabilities without modifying Zabbix code.

Zabbix Acceptance

1. A new type of pre-processing rule "Scripts -> JavaScript" should be supported
2. JavaScript will accept a non-empty JavaScript code block as an argument
 - a. To make it more user friendly, code block should be free of any must have (boilerplate) code
 - i. Therefore it should be possible to make Fahrenheit to Celsius conversion with code like "return (value - 32) * 5 / 9"
 1. For better readability 'value' is used instead of 'arguments[0]' in the above example
3. As with any other preprocessing rules, export of a template or a host will also export all script details
4. Selected JavaScript library must be compatible with GPLv2, have execution time and memory limits, lightweight, easily embeddable into Zabbix code base and have as little dependencies as possible
 - a. That is required in order to allow use of JavaScript on client side if it makes any sense in the future
5. Documentation must contain as many useful examples as possible, at least all mentioned in the **Blog post** section

Blog post

A blog post must be created on blog.zabbix.com that will explain briefly reasoning why this language(s) was chosen and also provide code examples for the following cases:

1. Mathematical formula (F to C): $T(^{\circ}\text{C}) = (T(^{\circ}\text{F}) - 32) \times 5/9$
2. String operations: number of occurrences of X in a string
3. Calculate average value based on values provided by complex multi-level JSON structure with some filtering by some attribute
4. Convert output of Apache status to JSON

Use cases

1. I want to have my own preprocessing function that will convert incoming value A to f(A)
 - a. For example: convert complex text data into JSON that can be easily used by Zabbix
2. I want preprocessing functions to be part of my template without any extra references and/or dependencies

Out of scope

1. Syntax highlighting. It may slow down implementation due to multiple color variations for all existing themes.
2. Ability to access any other information (apart of the value) from the script: item related, history, user macros, etc.