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

1. LLD rules must support preprocessing functionality like it is currently supported for items and item prototypes
    a. It must include preprocessing actions
2. LLD will accept normal JSON
    a. LLD will expect JSON array as input
3. Existing syntax of {#MACRO} will be replaced with {#<jsonpath>}
    a. A few examples: {#.name}, {#.fs.name}, {#.fs.mountpoint}
        i. JSON keys with any special characters and unicode must be supported using optional square bracket notation, like {#['unicode + special chars #1']['unicode + special chars #2']}
    b. Older syntax of macros will be deprecated, but still supported for backward compatibility
4. New types of preprocessing must be supported and available for items, item prototypes and LLD rules:
    a. CSV to JSON conversion
        i. Zabbix will assume that column names are stored in the first row
        ii. CSV will be converted into JSON array, each element of the array is a row of CSV input
        iii. JSON array will preserve same order as in CSV
    b. CSV, no header to JSON conversion
        i. Zabbix will assume that row names are stored in the first column
        ii. CSV will be converted into JSON array, each element of the array is a column of CSV input
        iii. JSON array will preserve same order as in CSV
    c. CSV conversions will support parameter to specify single character separator, default value is ','
        i. It must not be allowed to be set to some special characters like EOL
    d. CSV conversions must be placed under 'Structured data' menu item
    e. CSV should conform to RFC4180 with support of configurable separator
5. LLD preprocessing will also support same set of actions like for existing item level preprocessing
6. LLD Filters must be applied after preprocessing rules

Example of "CVS to JSON" conversion:

CSV:

```
Year,Make,Model
1997,Ford,E350
2000,Mercury,Cougar
```

JSON:

```
[
{
"Year": 1997,
"Make": "Ford",
"Model": "E350"
```

```
},
{
"Year": 2000,
"Make": "Mercury",
"Model": "Cougar"
}
]
```

Example of "CVS to JSON, no header" conversion:

CSV:

```
Year, 1997, 2000
Make, Ford, Mercury
Model, E350, Cougar
```

JSON:

```
[
{
"Year": 1997,
"Make": "Ford",
"Model": "E350"
},
{
"Year": 2000,
"Make": "Mercury",
"Model": "Cougar"
}
]
```

Use cases

1. Use normal JSON as a native data source for LLD, therefore it will handle JSON without any preprocessing
    a. Discovery from array of JSON objects with nested objects
2. Transform incoming non-JSON values into LLD ready representation. For example, convert CSV into LLD

Related tasks

1. ACC: Preprocessing rule for UTF8 encoding

Out of scope

1. Support of mass update for preprocessing rules
2. Merging of preprocessing rules and filters if it will ever be considered