

Zabbix Acceptance

- LLD rules must support preprocessing functionality like it is currently supported for items and item prototypes
 - It must include preprocessing actions
- LLD will accept normal JSON containing an array, i.e. without 'data' object
- LLD will accept user defined list of LLD macros
 - LLD rule configuration form will be extended to support a new tab to configure LLD macros
 - A set of {#MACRO}=<json path> can be defined, for example: {#NAME}=\$.name, {\$FS}=\$.fs.name, {#MOUNT}=\$.fs.mountpoint
 - 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']
 - For backward compatibility Zabbix will accept older JSON notation with "data" element, but the syntax will be documented as deprecated. In this case, Zabbix will use LLD macros from the JSON
 - For simplicity, user defined LLD macros will overwrite deprecated macros (in reality it should not happen)
 - Format and accepted characters for LLD macros as well as syntax for JSON path must be documented
- Prototype form will be extended to offer autocompletion for LLD macros
 - When a user start typing '{#' Zabbix will automatically offer a list of macros defined for LLD rule
 - The list will have an ability to quickly add a new macro
 - The new UI control must be flexible enough to be potentially used for other purposes in the future (autocompletion for host name, item key, user macros, etc)
- New types of preprocessing must be supported and available for items, item prototypes and LLD rules:
 - CSV to JSON conversion
 - Zabbix will assume that column names are stored in the first row
 - CSV will be converted into JSON array of JSON objects, where each element is a combination of column name (key) and row data (value)
 - JSON array will preserve same order as in CSV
 - CSV, no header to JSON conversion
 - Zabbix will assume that row names are stored in the first column
 - CSV will be converted into JSON array of JSON objects, where each element is a combination of column number (key) and row data (value)
 - JSON array will preserve same order as in CSV
 - CSV conversions will support parameter to specify single character separator, default value is ','
 - It must not be allowed to be set to some special characters like EOL
 - CSV conversions must be placed under 'Structured data' menu item
 - CSV should conform to [RFC4180](#) with support of configurable separator
- LLD preprocessing will also support same set of actions like for existing item level preprocessing
- 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:

Code Block

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

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

CSV:

1997,Ford,E350
2000,Mercury,Cougar

JSON:

Code Block

```
[
  {
    "0": 1997,
    "1": "Ford",
    "2": "E350"
  },
  {
    "0": 2000,
    "1": "Mercury",
    "2": "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

Open questions

1. Let's consider to change [JMX discovery](#) to output using new `{#<jsonpath>}` syntax. It will help to fix this: [ZBXNEXT-4209](#) . We might introduce new key for jmx discovery (or parameter) and deprecate current discovery key for JMX
2. Let's consider to change [ODBC discovery](#) to output using new `{#<jsonpath>}` syntax. It will help to use non `[A-Z0-9_]` chars in column names. We might introduce new key for odbc discovery (or parameter) and deprecate current discovery key for ODBC
3. There are currently no item preprocessing steps available to get actual item values from master item with output similar to LLD: for CSV, JSON array, and ODBC.