ENGLISH

ZABBIX MEETUP ONLINE '23

Optimizing your data collection with Low-level discovery and dependent items

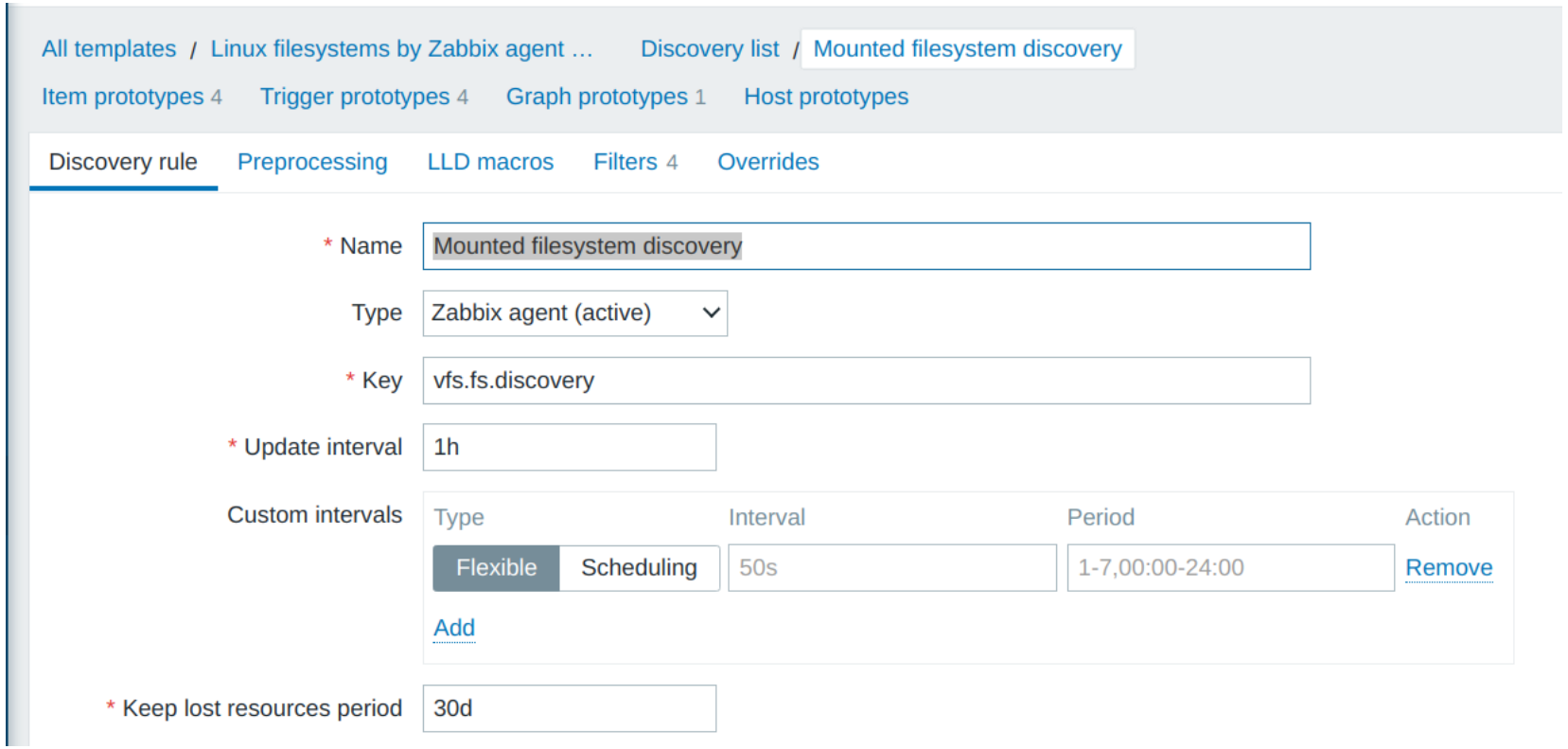
Dimitri Bellini
CEO, Quadrata Service Group, Italy and United Arab Emirates (UAE)

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

Low-level discovery

- *a way to automatically create items, triggers, graphs and hosts*
- *used in most of Out of The Box Zabbix Templates*
- *highly customizable base on your needs*



QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

ZABBIX MEETUP / 2023

Out Of The Box Discovery Rules

- mounted filesystems
- network interfaces
- CPUs and CPU cores
- SNMP OIDs
- JMX objects
- ODBC SQL queries
- Windows services
- Windows performance counter instances
- Systemd services
- host interfaces in Zabbix
- IPMI sensors
- WMI queries
- Prometheus data
- VMWare Entities

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

LLD Main components

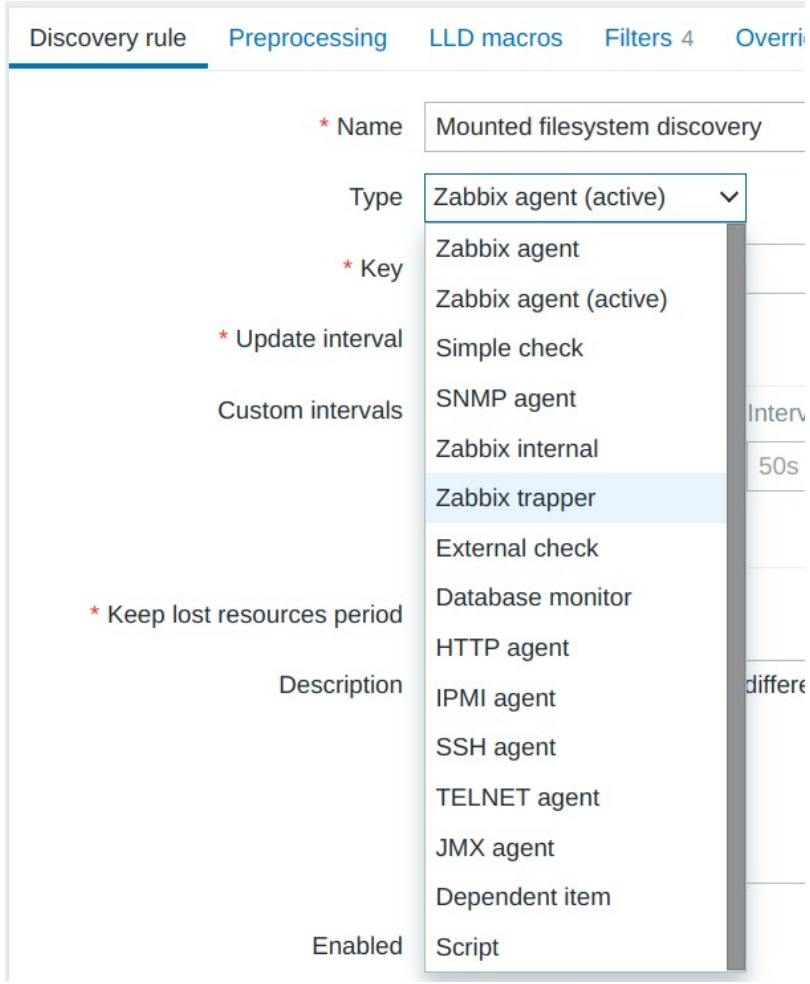
- **Discovery rule** - specifies, most importantly, the built-in item or custom script to retrieve discovery data
- **Preprocessing** - *applies some preprocessing to the discovered data*
- **LLD macros** - allows to extract some macro values to use in discovered items, triggers, etc
- **Filters** - allows to filter the discovered values
- **Overrides** - *allows to modify items, triggers, graphs or host prototypes when applying to specific discovered objects*

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

LLD how it works

Discovery Rule → Collect the required data using ALL the Zabbix Type available



Example of output of LLD

```
[
  { "{#FSNAME}":"/",              "{#FSTYPE}":"rootfs"   },
  { "{#FSNAME}":"/sys",           "{#FSTYPE}":"sysfs"    },
  { "{#FSNAME}":"/proc",          "{#FSTYPE}":"proc"     },
  { "{#FSNAME}":"/dev",           "{#FSTYPE}":"devtmpfs" },
  { "{#FSNAME}":"/dev/pts",       "{#FSTYPE}":"devpts"   },
  { "{#FSNAME}":"/lib/init/rw",   "{#FSTYPE}":"tmpfs"    },
  { "{#FSNAME}":"/dev/shm",       "{#FSTYPE}":"tmpfs"    },
  { "{#FSNAME}":"/home",          "{#FSTYPE}":"ext3"     },
  { "{#FSNAME}":"/tmp",           "{#FSTYPE}":"ext3"     },
  { "{#FSNAME}":"/usr",           "{#FSTYPE}":"ext3"     },
  { "{#FSNAME}":"/var",           "{#FSTYPE}":"ext3"     },
]
```

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

ZABBIX MEETUP / 2023

Most used Item Type

HTTP Agent → Many use cases like Kubernetes monitoring, Application Monitoring, RestAPI

SNMP Agent → For all Router/SAN Switches etc

Dependent Item → Similar use case of HTTP Agent

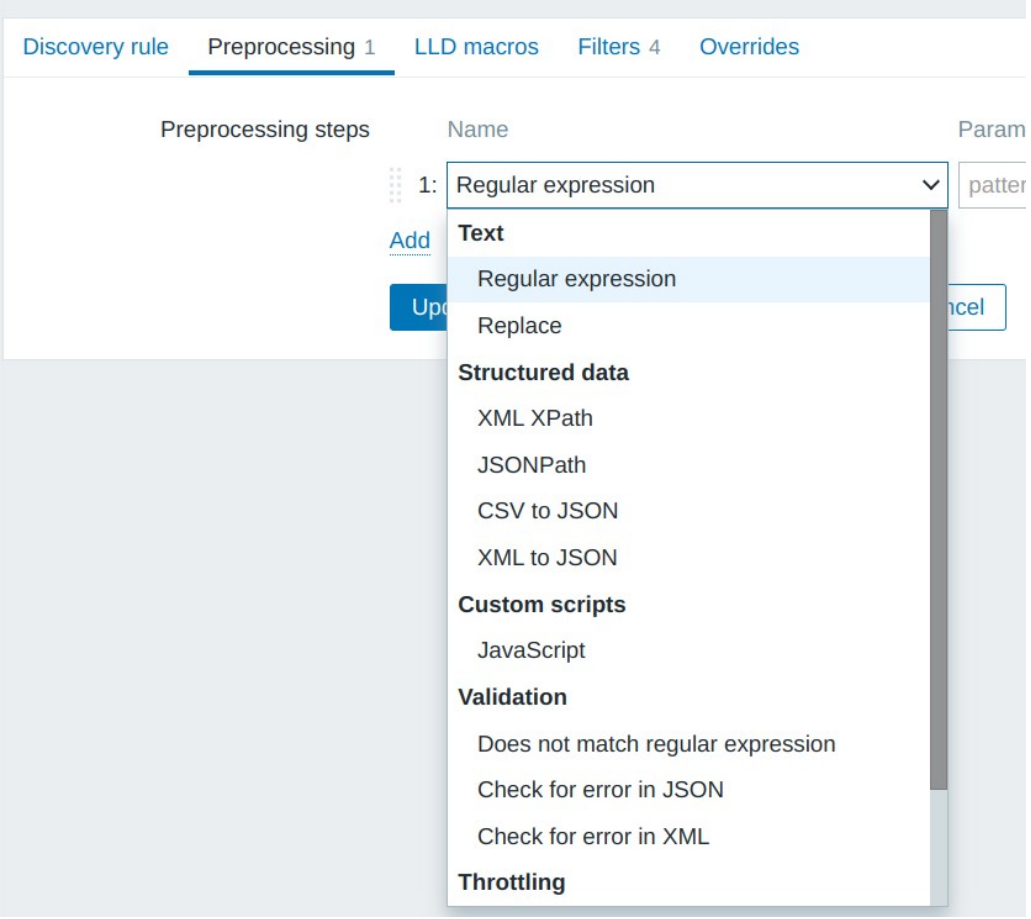
Script → More complex scenarios with multiple RestAPI calls and data conversion. Ex. RestAPI execute login and receive a Token with the Token you can request the final data...

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

Where magic comes true

Preprocessing → Multiple way to extract the information you require coming from the LLD rule



Most used Preprocessing Steps

- Regular expression
- JSONPath
- JavaScript

QUADRATA SERVICE GROUP s.r.l.

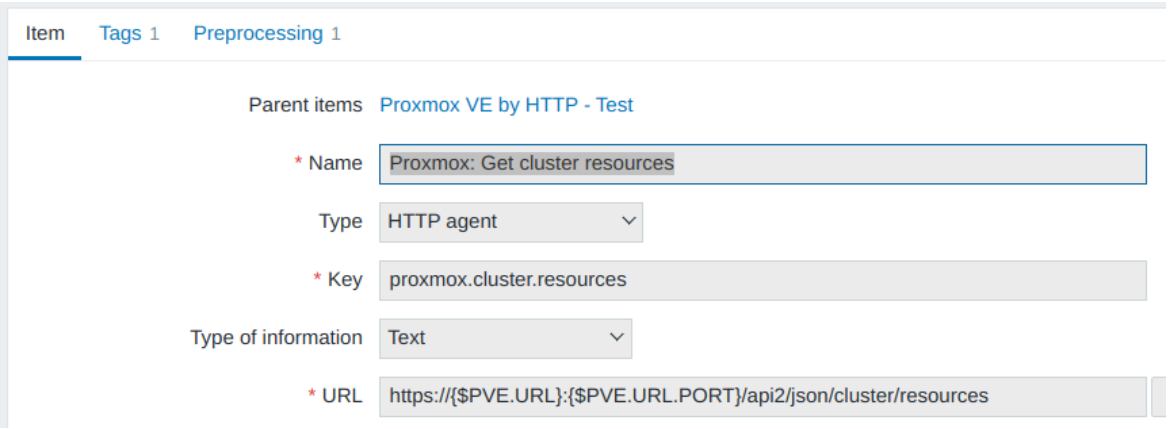
ZABBIX PREMIUM PARTNER

ZABBIX MEETUP / 2023

Next-Gen LLD discovery

- Reduce number of polling to your device using “Master/Dependant Items”
- Enhance Automation with LLD Overrides
- Enrich your LLD data using Javascript Preprocessor step
- Bring User Macros every where :-)
- Solve Tricky situations

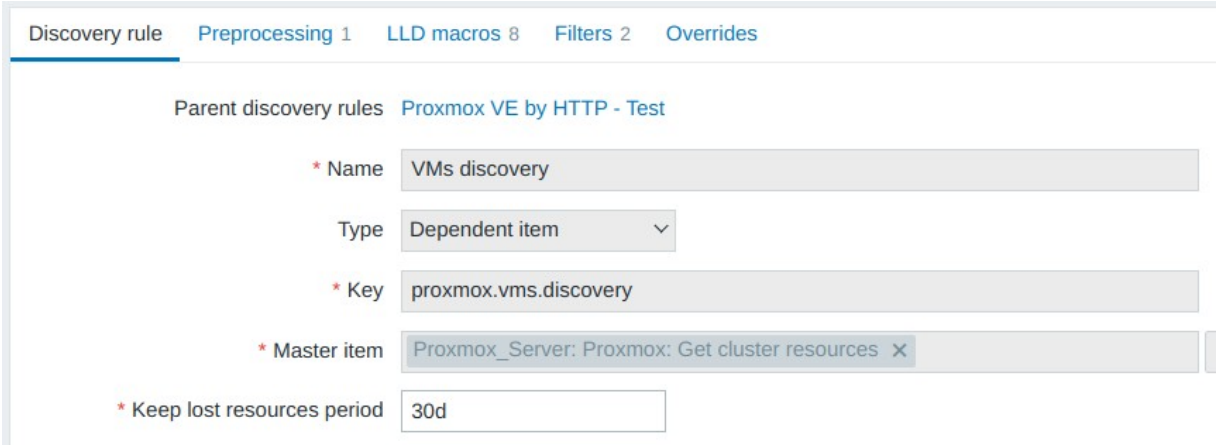
Master/Dependant on LLD

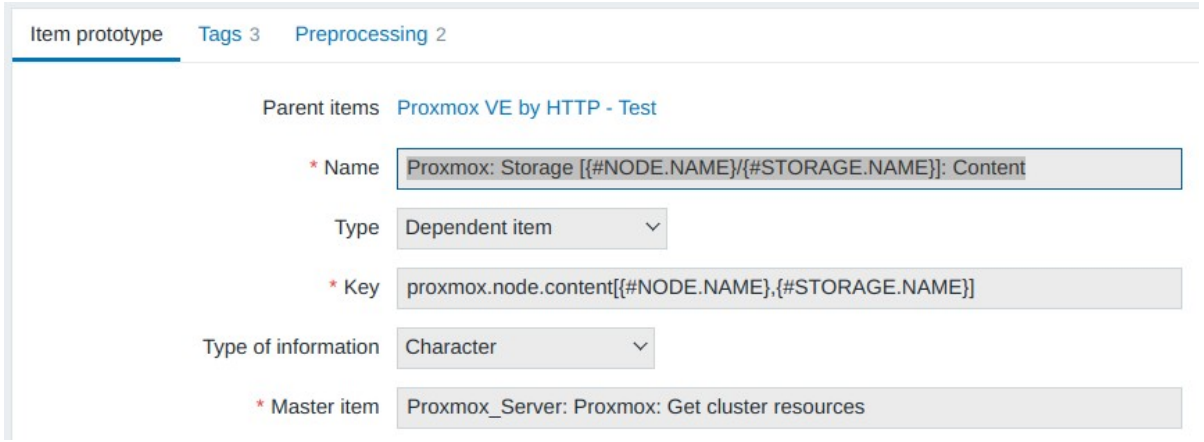


HTTP Agent → Master Item Collect the main Data

- Less polling to monitored device
- Single Fetch for all the chain

Discovery Rule → Dependant from HTTP Agent Item





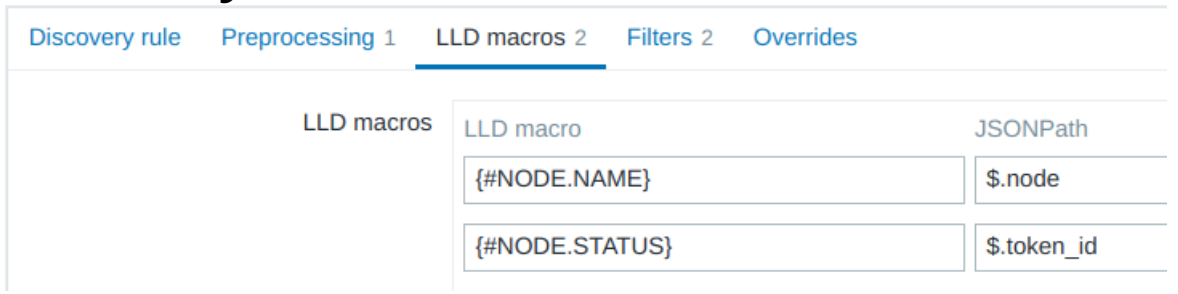
Item Prototype → Dependant from HTTP Agent Item

Master/Dependant on LLD

HTTP Agent → Raw data

```
[
  {
    "data": [
      {
        "cpu": 0.723949571936942,
        "status": "running",
        "name": "zabbix",
        "mem": 2793138336
       },
      {
        "cpu": 1.423949571936942,
        "status": "running",
        "name": "fedora",
        "mem": 2793138336
      },
      {
        "cpu": 0.2793138336,
        "status": "running",
        "name": "ubuntu",
        "mem": 2793138336
      }
     ]
  }
]
```

Discovery Rule → Hosts informations



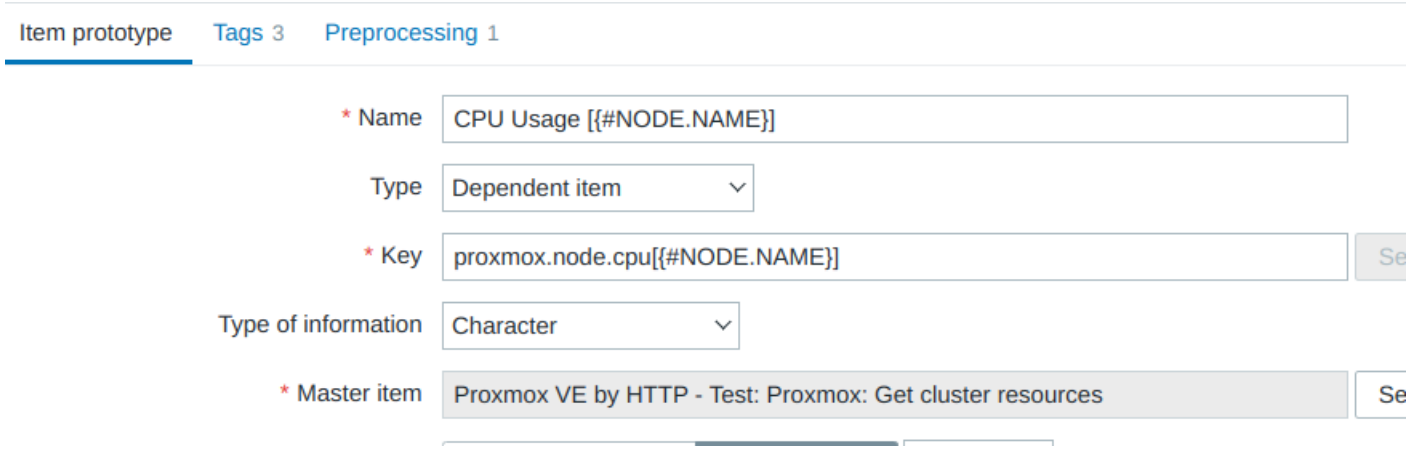

```
[
   { "{#NODE.NAME}":"zabbix", "{#NODE.STATUS}":"running" },
   { "{#NODE.NAME}":"fedora", "{#NODE.STATUS}":"running" },
   { "{#NODE.NAME}":"ubuntu", "{#NODE.STATUS}":"running" }
]
```

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

Master/Dependant on LLD

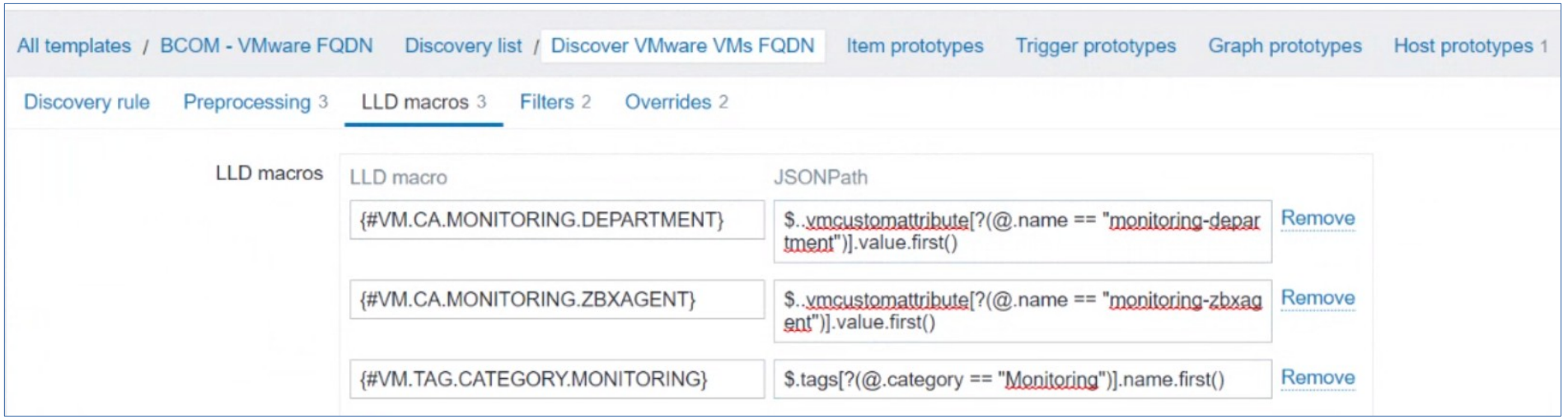
Item Prototype → For all VMs create relative Items

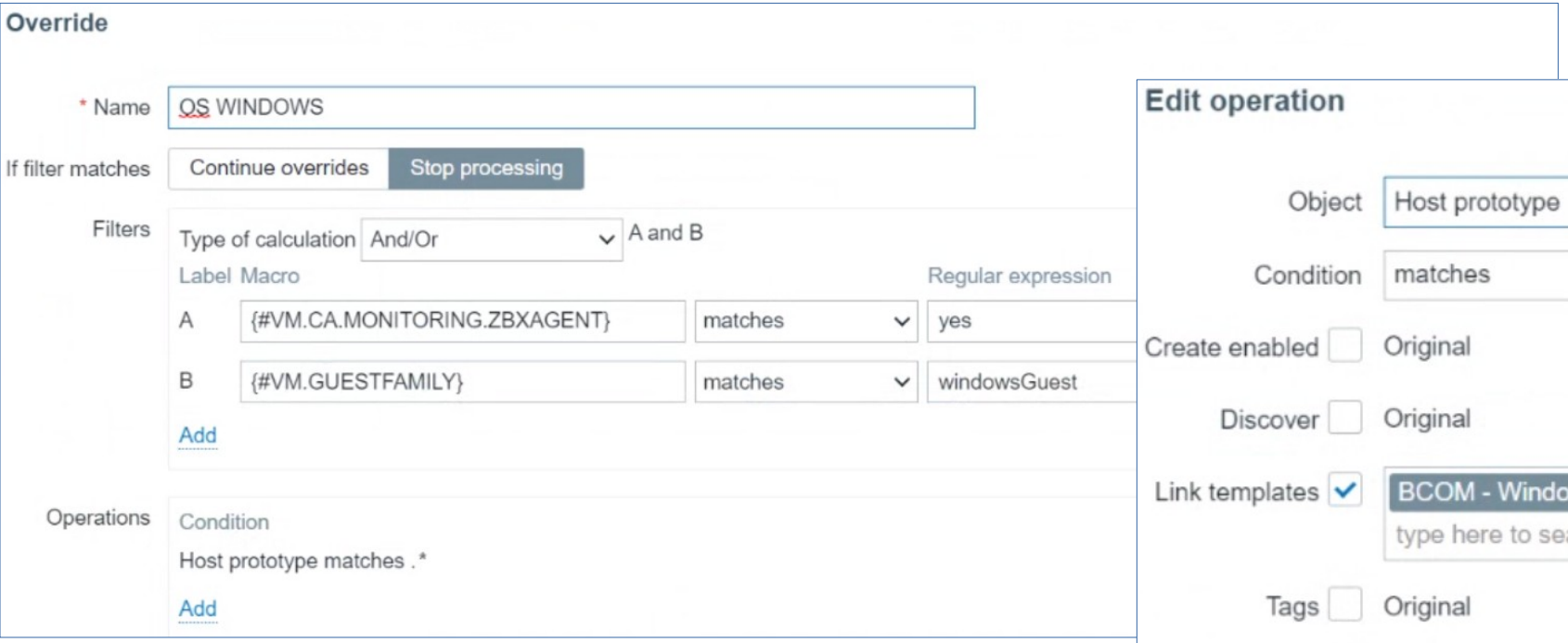


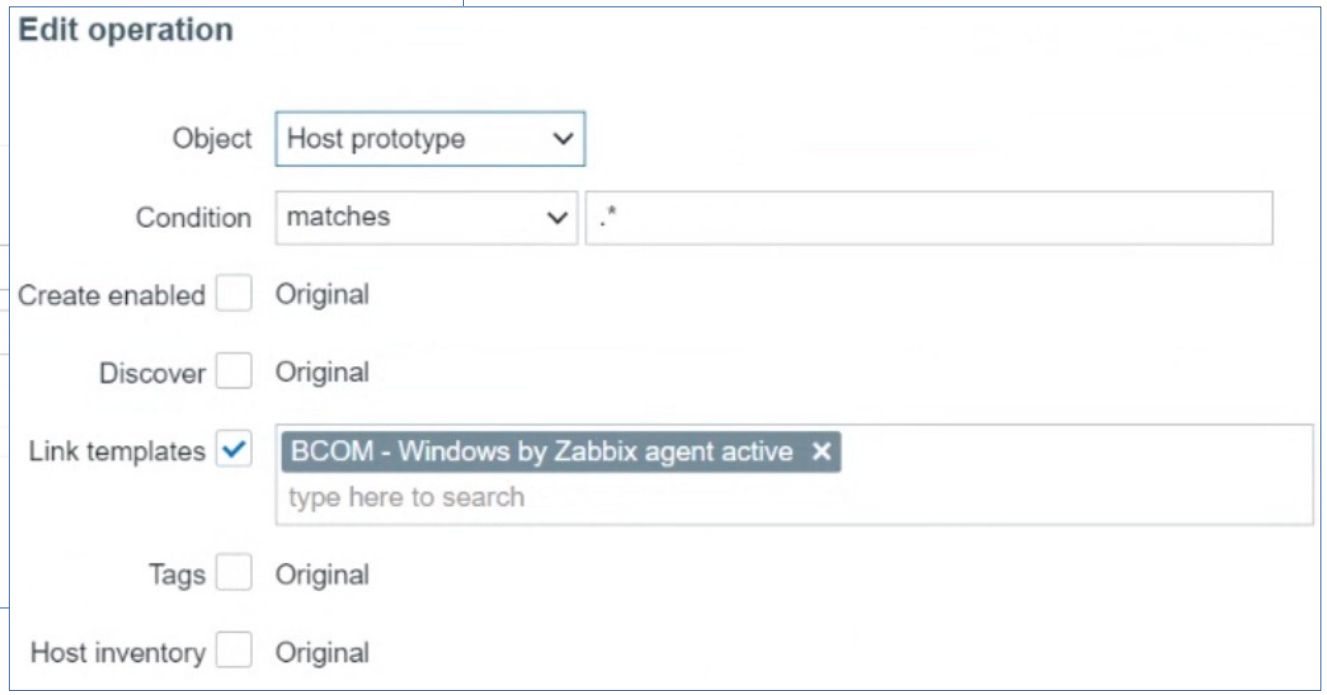
Item Preprocessing → Extract the value CPU coming from Master Item

Enhance Automation with LLD Overrides

VMWare Tag/Custom Attribute used to automate the Template association







Enrich your LLD data using Javascript Preprocessor step

All hosts / IREI

Discovery rule

JavaScript

```
var j = JSON.parse(value);
var name;
for (elem = 0; elem < Object.keys(j).length; elem++) {
        if (typeof j[elem]["{#SYSNAME}"] != 'undefined') {
                name = j[elem]["{#SYSNAME}"]
                delete j[elem]["{#SYSNAME}"];
        }
}
jclean = j.filter(function (e) {return e != null;})
if (typeof name != 'undefined') {
        var endpoint = '{$ENDPOINT}'+'/'+name;
        try {
                var fields = {};
                var req = new CurlHttpRequest();
                resp = req.Get(endpoint, JSON.stringify({"fields": fields}));
        } catch (error) {
                Zabbix.Log(3, "Brocade_Virt_Generic_v3: cannot reach external API")
                return JSON.stringify(jclean);
```

Push new Array elements to your Discovery Rule,
maybe based on result of a custom RestAPI

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

ZABBIX SUMMIT / 2022

Bring User Macros every where

All hosts / Prox

Discovery rule

JavaScript

```
function (value) {
  1  // Add User Macro to Discovery Output as an LLD Macro
  2  var vms = JSON.parse(value);
  3  var i= 0;
  4  for (i = 0; i < vms['data'].length; i++) {
  5      vms['data'][i]['token_id'] = '{$PVE.TOKEN.ID}';
  6      vms['data'][i]['token_secret'] = '{$PVE.TOKEN.SECRET}';
  7      vms['data'][i]['pve_url'] = '{$PVE.URL}';
  8      vms['data'][i]['pve_url_port'] = '{$PVE.URL.PORT}';
  9  }
 10  //Zabbix.log(3, "Proxmox Discovery VMs: "+JSON.stringify(vms));
 11  return JSON.stringify(vms);
}
```

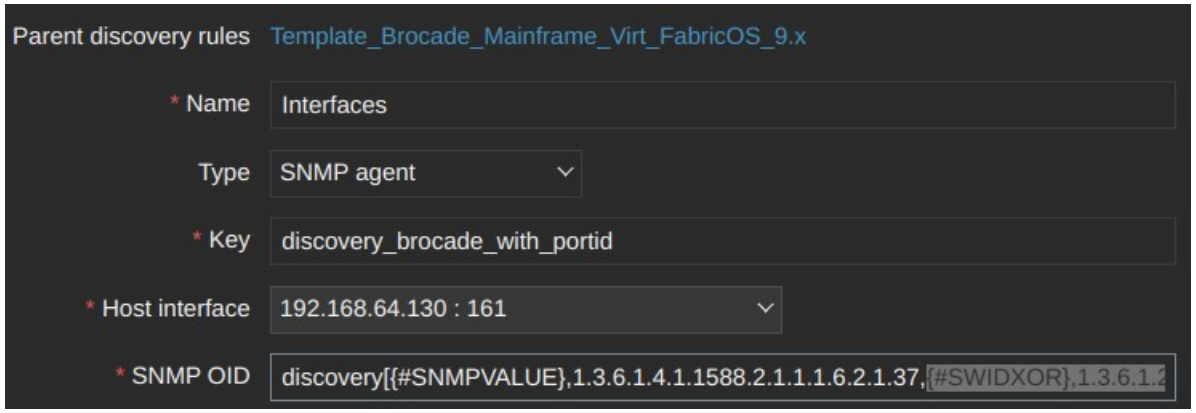
Add new Array elements to your Discovery Rule output
to bring your User MACROS everywhere

QUADRATA SERVICE GROUP s.r.l.

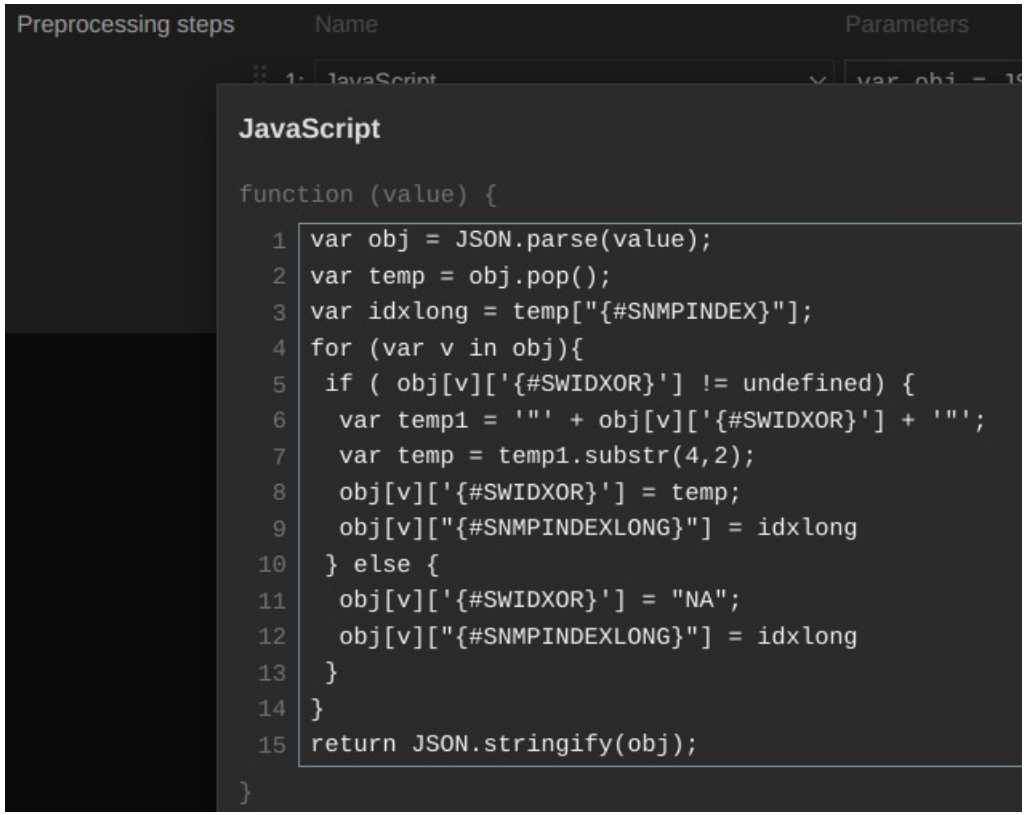
ZABBIX PREMIUM PARTNER

ZABBIX MEETUP / 2023

Solve Tricky situations



Add new Array elements to your Discovery Rule output to bring some Specific usefull OIDs or transform the value contents



QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

ZABBIX MEETUP / 2023

Solve Tricky situations

```
<Impianto Hostaddress=[redacted] Hostname=[redacted] NumeroCartelli="3" PosizioneImpianto="P" RevisioneSwCr01="1.29 ( build 5 )" SottoTipo="*" Tipologia="A__"
UltimoAggiornamento="20230213_151754">
  <Snmp EnterpriseOid="17377" SogliaOccupazioneFS1="80" SogliaOccupazioneFS2="88"/>
  <StatoFunzionamento AvvioControlloDisplay="20230212_162019" OutputLed="00010101" Protezione="1">
  ...
  </StatoFunzionamento>
  <Cartelli NumeroCartelliInclusi="3" NumeroTotaleCartelli="3">
    <Cartello Descrizione="" IdCartello="0" IdPanel="0" NomeCartello="" SottoTipo="*" Tipologia="A__">
      <StatoFunzionamento Protezione="1">
      ...
      </StatoFunzionamento>
      <Displays NumeroDisplay="2">
        <Display IdRow="0" IndirizzoDisplay="A" Installato="1" LenCodicePitto="8" NomeDisplay="PITTO1" ProtocolloCCLDisplay="3_10" RevisioneFw="DSPMV72144UBRE\D8C11
        Semaforo_acceso="0">
        ...
        </Display>
        <Display IdRow="1" IndirizzoDisplay="B" Installato="1" LenCodicePitto="8" NomeDisplay="FC1" ProtocolloCCLDisplay="3_10" RevisioneFw="DSPMV48048UBRE\D9E28C[2
        Semaforo_acceso="0">
```

First Level LLD MACRO

Second Level LLD MACRO

Single Discover with Multiple LLD MACRO Levels

```
for (j = 0; j < displayCount; j++) {
  Zabbix.log(3, "PMV Display Inizio Count: "+displayCount);
  query="/Impianto/Cartelli/Cartello[@IdCartello="+i+"]/Displays/Display[@IdRow="+j+"]";
  Zabbix.log(3, "PMV display Cartelli Query String: "+query);
  out = XML.query(value,query);
  out = XML.query(out,'/Display/@IdRow');
```

Solved with JavaScript Preproc:

- XML.query
- Array.push

Still *waiting* interesting enhancements

- **Ability to show LLD MACRO output** → ZBXNEXT-6812 (see if LLD macros are correctly populated)

QUADRATA SERVICE GROUP s.r.l.

ZABBIX PREMIUM PARTNER

Thank you!

www.quadrata-group.com

#zabbixitalia

QUADRATA
SERVICE GROUP s.r.l.