



ZABBIX 5.0

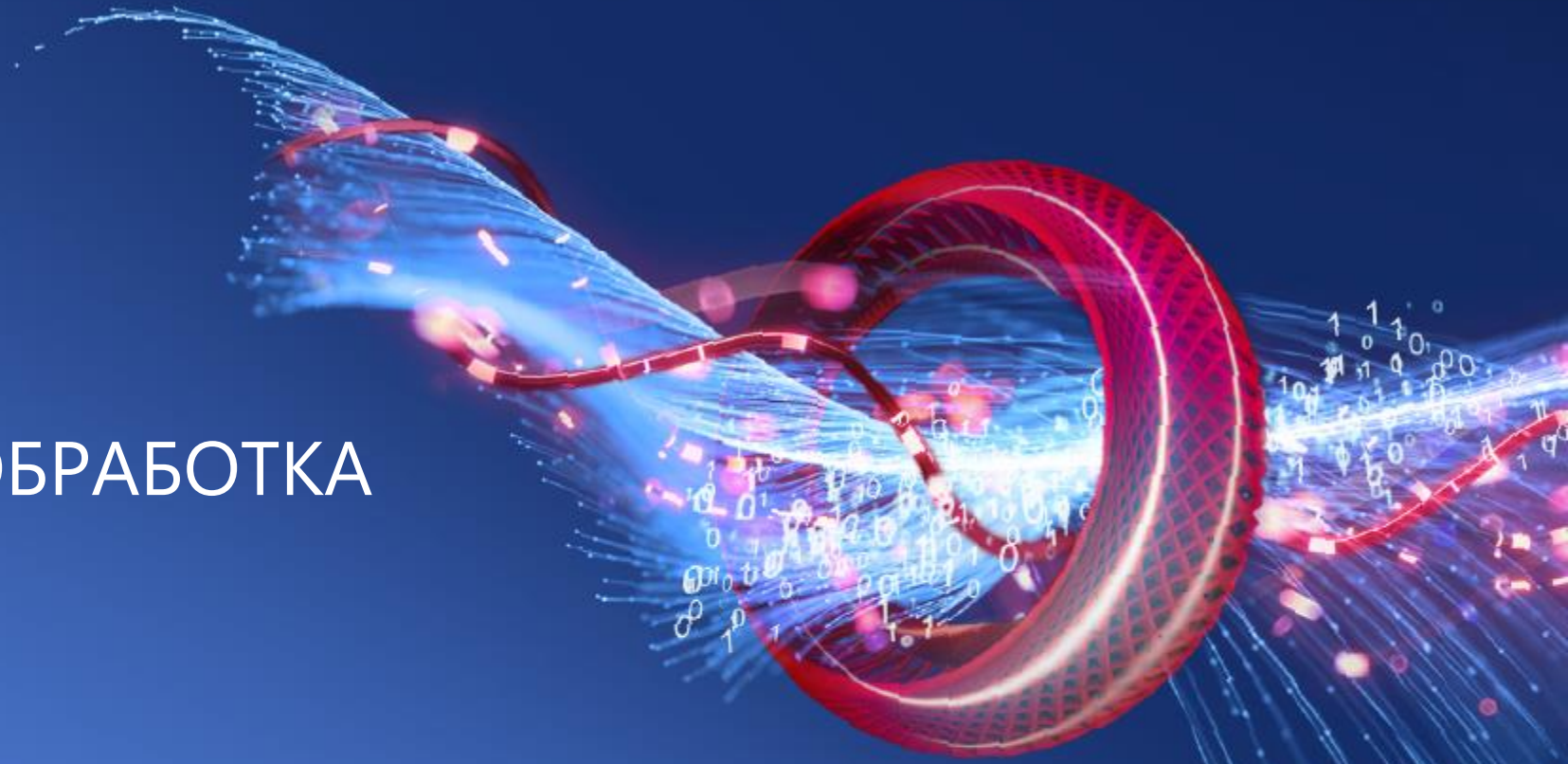
ПРЕДОБРАБОТКА ЗНАЧЕНИЙ – КАК
ЗАПОЛУЧИТЬ ТО, ЧТО ХОЧЕТСЯ В 5.0



Александр Петров-Гаврилов
Инженер технической поддержки

01

ЧТО ТАКОЕ ПРЕДОБРАБОТКА
ЗНАЧЕНИЙ?



ПРЕДОБРАБОТКА ЗНАЧЕНИЙ В ZABVIX



ЧТО ТАКОЕ ПРЕДОБРАБОТКА?

- ⊗ В компьютерной науке, предобработчик(preprocessor) это программа позволяющая обработать входные данные для использования, например, в другой программе. Оба процесса нуждаются в настройке и одинаково важны.
- ⊗ Предобработка данных используется в компьютерном обучении, для упрощения работы с входящими данными.

ЧТО ТАКОЕ ПРЕДОБРАБОТКА?

- ☑ Предобработка позволяет создать правила трансформации для получаемых метрик.
- ☑ Одна или несколько трансформаций возможны перед сохранением данных в базу данных.
- ☑ Трансформации происходят в определяемом Вами порядке.
- ☑ Предобработка выполняется или Zabbix сервером или Zabbix прокси (если сбор метрики осуществляется при помощи прокси).

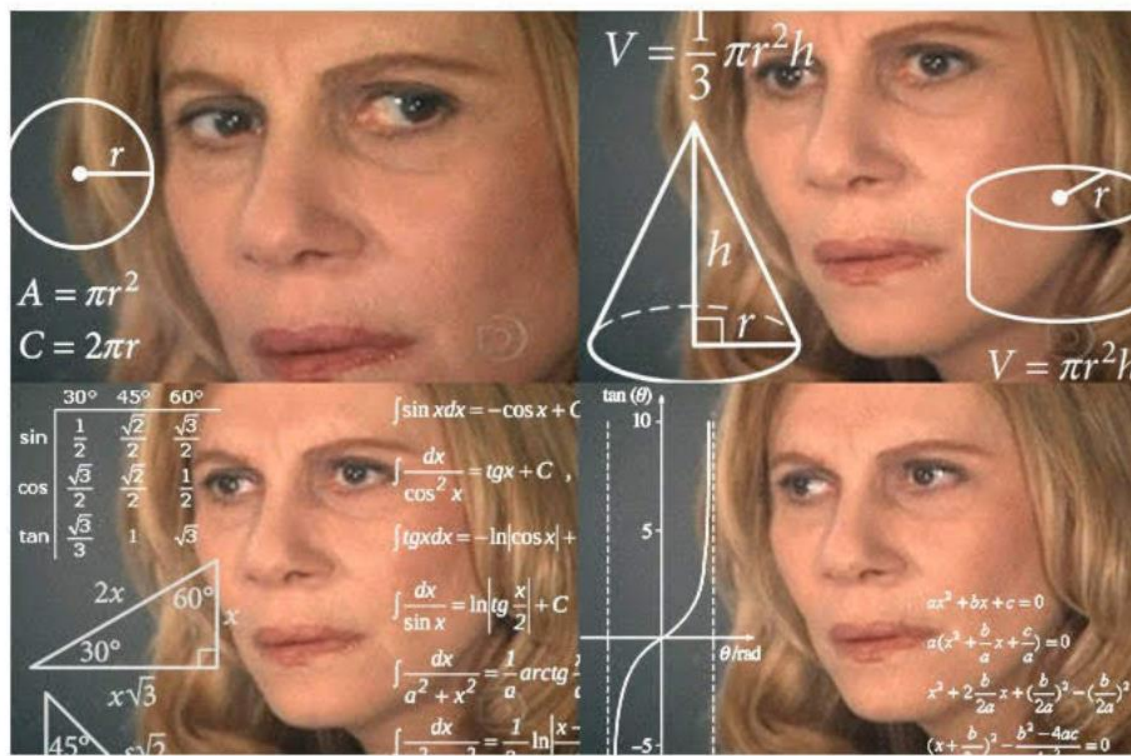
02

ДЛЯ ЧЕГО НАМ НЕОБХОДИМА
ПРЕДОБРАБОТКА ЗНАЧЕНИЙ?



ДЛЯ ЧЕГО НАМ НЕОБХОДИМА ПРЕДОБРАБОТКА?

Когда пытаешься понять данные
без предобработки



ДЛЯ ЧЕГО НАМ НЕОБХОДИМА **ПРЕДОБРАБОТКА**?

- ✓ Очистка данных
- ✓ Редактирование данных
- ✓ Уменьшение количества данных
- ✓ Трансформация данных

ОЧИСТКА **ДАНЫХХ?**

- ⊗ Определение диапазона минимальных и максимальных значений.
- ⊗ Использование регулярных выражений для поиска ошибок или нужных совпадений.
- ⊗ Поиск ошибки на уровне приложения при помощи JSONpath.
- ⊗ Поиск ошибки на уровне приложения при помощи xpath.

ОПРЕДЕЛИТЬ ДИАПАЗОН

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия	
1:	По диапазону	1	100	<input checked="" type="checkbox"/>	Тест Удалить
Другое при ошибке		Отбросить значение	Задать значение в	Задать ошибку в	
Добавить					Тест всех шагов
Добавить Тест Отмена					

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия	
1:	По диапазону	1	999	<input checked="" type="checkbox"/>	Тест Удалить
Другое при ошибке		Отбросить значение	Задать значение в	Задать ошибку в	999
Добавить					Тест всех шагов
Добавить Тест Отмена					

ИСПОЛЬЗОВАТЬ РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Совпадение регулярному выражению	(?)error	<input checked="" type="checkbox"/>	Тест Удалить

Другое при ошибке:

[Добавить](#) [Тест всех шагов](#)

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Не совпадает с регулярным выражением	(?)error	<input checked="" type="checkbox"/>	Тест Удалить

Другое при ошибке:

[Добавить](#) [Тест всех шагов](#)

БОЛЬШЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

✓ `^(automatic|automatic delayed|manual|disabled)$`

Статусы автозапуска сервисов в Windows

✓ `([1-9]+)`

Сбор только цифровых значений

✓ `((?>[a-z\-\0-9]{2,}\.){1,}[a-z]{2,8})(?:\s|/)`

Доменные имена

✓ `([0-9A-Za-z'\&\-\.\V\(\)=:;]+)|((?::|;|=)(?:-)?(?:\)|D|P))`

Эмодзи

ИСПОЛЬЗОВАНИЕ JSONPATH

Элемент данных Предобработка

Шаги предобработки

Имя	Параметры	Другое при ошибке	Действия
1: Проверьте на ошибку в JSON	<input type="text" value="\$..myjson.error"/>	<input checked="" type="checkbox"/>	Тест Удалить

Другое при ошибке

[Добавить](#) [Тест всех шагов](#)

Элемент данных Предобработка

Шаги предобработки

Имя	Параметры	Другое при ошибке	Действия
1: Проверьте на ошибку в XML	<input type="text" value="/address/message/error"/>	<input checked="" type="checkbox"/>	Тест Удалить

Другое при ошибке

[Добавить](#) [Тест всех шагов](#)

РЕДАКТИРОВАНИЕ **ДАННЫХ**?

- ☑ Заменить(Replace).
- ☑ Обрезать(Trim).
- ☑ Регулярные выражения(Regex).
- ☑ Умножение(Multiply).
- ☑ Javascript.

ЗАМЕНИТЬ

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Замена	error	0	<input type="checkbox"/>

[Добавить](#)

[Тест](#) [Удалить](#)

[Тест всех шагов](#)

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Регулярное выражение	^(error)	1	<input type="checkbox"/>
2:	Замена	error	0	<input type="checkbox"/>

[Добавить](#)

[Тест](#) [Удалить](#)

[Тест всех шагов](#)

ОБРЕЗАТЬ

Убирает символы с обеих сторон(справа и слева) с **Обрезать**

t 36 C

Обрезать: tC

36

Убирает символы с правой стороны с **Обрезать справа**

t 36 F

Обрезать справа: F

t 36

Убирает символы с правой стороны с **Обрезать слева**

t 36 K

Обрезать слева: t

36 K

ДОБАВИТЬ ЕЩЁ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

✓ `^(eth[0-9]$)|(^eth[0-9]:[1-9]$)`

Поиск сетевых адаптеров

✓ `^(ntfs|fat32|zfs)$`

Короткое выражение для поиска файловых систем

✓ `^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`

Память на SNMP устройствах

УМНОЖЕНИЕ

Килобайты в байты

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Пользовательский множитель	1024	<input type="checkbox"/>	Тест Удалить

[Добавить](#)

[Тест всех шагов](#)

Миллисекунды в секунды

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Пользовательский множитель	0.001	<input type="checkbox"/>	Тест Удалить

[Добавить](#)

[Тест всех шагов](#)

Обратите внимание, что если тип элемента информации *Числовой (целое положительное)*, входящие значения с дробной частью будут обрезаны (т. е. «0,9» станет «0») до применения пользовательского множителя.

ИСПОЛЬЗУЙТЕ JAVASCRIPT

Отслеживайте время до окончания сертификата в секундах, когда Date.parse() не подходит

```
var split = value.split(' '),
    MONTHS_LIST = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
    ISOdate = split[3] + '-' + month_index + '-' + split[1] + 'T' + split[2],
    now = Date.now();
return parseInt((Date.parse(ISOdate) - now) / 1000);
```

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги преобработки	Имя	Результат
	1: JavaScript	43297327

И ЕЩЁ JAVASCRIPT

Конвертируйте значение SNMP "DateandTime"


```
'use strict';
var str = value;
var y256 = str.slice(0,2); var y = str.slice(3,5); var m = str.slice(6,8);
var d = str.slice(9,11); var h = str.slice(12,14); var min =
str.slice(15,17);
var y256Base10 = +("0x" + y256);
var yBase10 = +("0x" + y);
var Year = 256*y256Base10 + yBase10;
var mBase10 = +("0x" + m);
var dBase10 = +("0x" + d);
var hBase10 = +("0x" + h);
var minBase10 = +("0x" + min);
var YR = String(Year); var MM = String(mBase10); var DD = String(dBase10);
var HH = String(hBase10);
var MIN = String(minBase10);
if (mBase10 < 10) MM = "0" + MM; if (dBase10 < 10) DD = "0" + DD;
if (hBase10 < 10) HH = "0" + HH; if (minBase10 < 10) MIN = "0" + MIN;
var Date = YR + "-" + MM + "-" + DD + " " + HH + ":" + MIN;
return Date;
```


И ЕЩЁ JAVASCRIPT

Конвертируйте значение SNMP "DateandTime"

Тест элемента данных ✕

Получить значение с узла сети

Значение  Время

Предыдущее значение  Пред. время

Конец строки

Шаги предобработки	Имя	Результат
	1: JavaScript	2021-09-17 13:22

УМЕНЬШЕНИЕ **КОЛИЧЕСТВА ДАННЫХ?**

- ✓ XML XPath.
- ✓ JSONPath.
- ✓ Регулярные выражения.
- ✓ Троттлинг.
- ✓ Замена.

XML XPath

Подготовьте Ваш XML

```
<rhs_check>
  <network>
    <ping host="cdn.redhat.com">passed</ping>
    <reverse_dns>passed</reverse_dns>
    <port address="80">passed</port>
    <port address="8080">passed</port>
    <port address="5671">passed</port>
    <port address="443">passed</port>
    <port address="8140">passed</port>
    <port address="9090">passed</port>
  </network>
  <services>
    <service name="mongod">active</service>
    <service name="qpidd">active</service>
    <service name="qdrouterd">active</service>
    <service name="tomcat">active</service>
    <service name="pulp_resource_manager">active</service>
    <service name="pulp_workers">active</service>
    <service name="httpd">active</service>
    <service name="puppet">inactive</service>
    <service name="elasticsearch">unknown</service>
  </services>
</rhs_check>
```

XML XPATH

Найдите нужный сервис:

/rhs_check/services/service[1]

Добавьте пару шагов:

Элемент данных | Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	XML XPath	/rhs_check/services/service[1]	<input type="checkbox"/>	Тест Удалить
2:	Регулярное выражение	(?i)(active)	<input type="checkbox"/>	Тест Удалить
3:	Замена	active	<input type="checkbox"/>	Тест Удалить

[Добавить](#) [Тест](#) [Отмена](#) [Добавить](#) [Тест всех шагов](#)

Тест элемента данных

Получить значение с узла сети

Значение: <rhs_check>... | Время: now

Предыдущее значение: | Пред. время:

Конец строки: LF CRLF

Шаги предобработки	Имя	Результат
1:	XML XPath	<service name="mongod">active</service>
2:	Регулярное выражение	active
3:	Замена	1

Результат: Результат сконвертирован в Числовой (целое положительное) | 1

“ПУТЬ JSON” (JSONPATH)

```
{ "obj2testobj2Connection": "OK", "objam.testHttpConnection": "OK",
"obj2.testobj2ServerComponents": "OK", "tomcat.freememory.threshold": "2132131000",
"tomcat.freememory": "OK", "tomcat.freememory.actual": "13123131121",
"objam.testNFSSStore": "OK" }
```

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	JSONPath	\$.["tomcat.freememory.actual"]	<input type="checkbox"/>	Тест Удалить

[Добавить](#)

[Добавить](#) [Тест](#) [Отмена](#)

[Тест всех шагов](#)

Тест элемента данных

Получить значение с узла сети

Значение Время

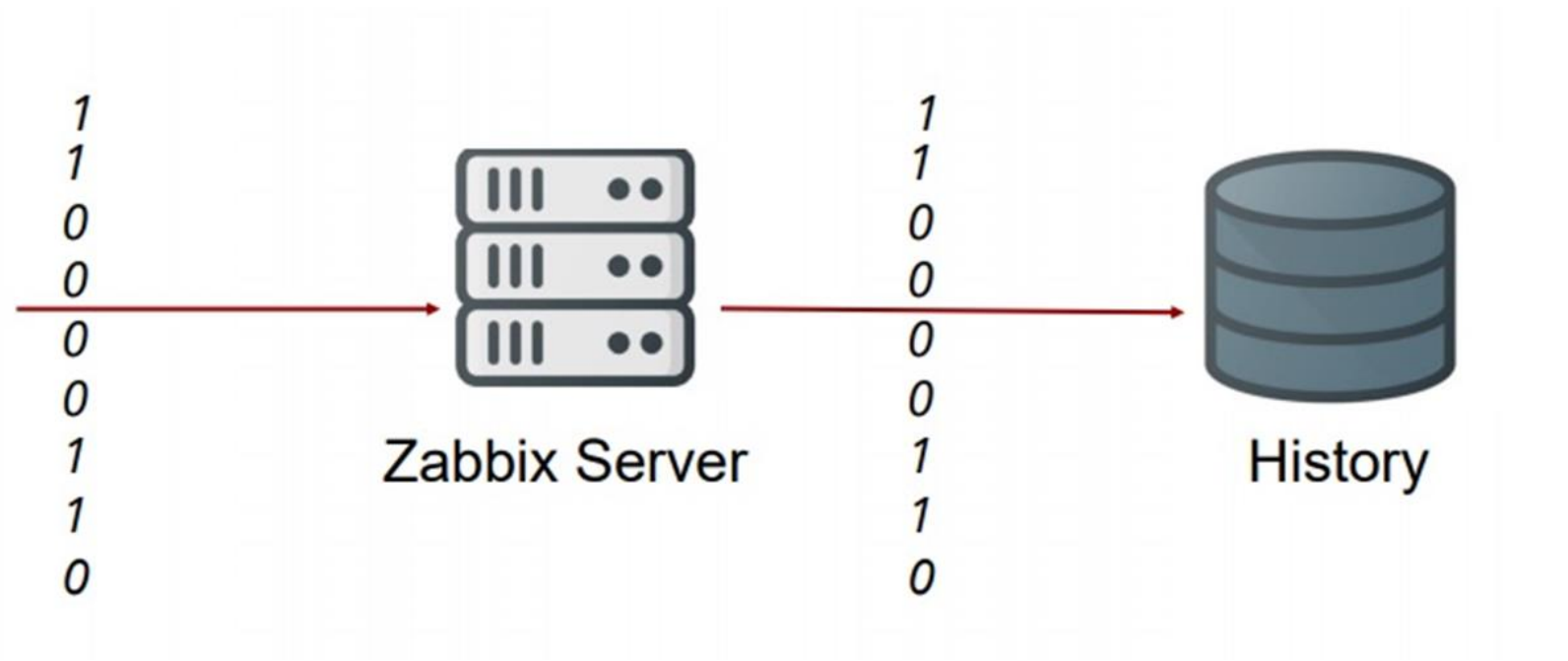
Предыдущее значение Пред. время

Конец строки

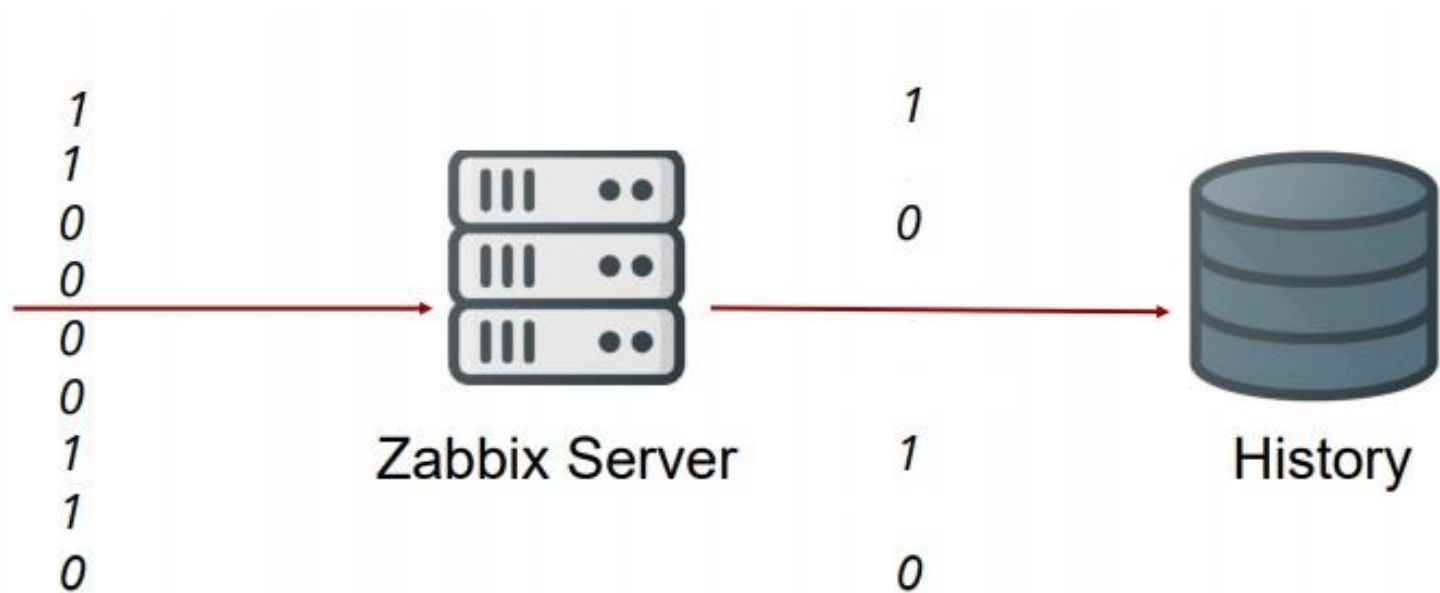
Шаги предобработки	Имя	Результат
1:	JSONPath	13123131121

[Тест](#) [Отмена](#)

БЕЗ ТРОТТЛИНГА



С ТРОТТЛИНГОМ



ЗАМЕНА

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Замена	Up	<input type="checkbox"/>	Тест Удалить
2:	Замена	Down	<input type="checkbox"/>	Тест Удалить

[Добавить](#)

[Добавить](#) [Тест](#) [Отмена](#)

[Тест всех шагов](#)

ЗАМЕНА

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги предобработки	Имя	Результат
1: Замена		1
2: Замена		1

Результат 1

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги предобработки	Имя	Результат
1: Замена		Down
2: Замена		0

Результат 0

ТРАНСФОРМАЦИЯ **ДАНЫХХ?**

- ⊗ Логические в десятичные.
- ⊗ Восьмеричные в десятичные.
- ⊗ Шестнадцатеричные в десятичные.
- ⊗ CSV в JSON.
- ⊗ Шаблон Prometheus.
- ⊗ Отслеживать изменения(Simple change).
- ⊗ Изменения в секунду(Change per second).

ЛОГИЧЕСКИЙ В ДЕСЯТИЧНЫЙ.

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
	1: Логический в десятичный		<input type="checkbox"/>	Тест Удалить

[Добавить](#)

[Тест всех шагов](#)

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги предобработки	Имя	Результат
	1: Логический в десятичный	1

TRUE - true, t, yes, y, on, up, running, enabled, available
FALSE - false, f, no, n, off, down, unused, disabled, unavailable

ВОСЬМЕРИЧНЫЕ/ШЕСТНАДЦАТЕРИЧНЫЕ В ДЕСЯТИЧНЫЕ.

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги предобработки

Имя	Результат
1: Восьмеричное в десятичное	1024

Тест элемента данных

Получить значение с узла сети

Значение Время

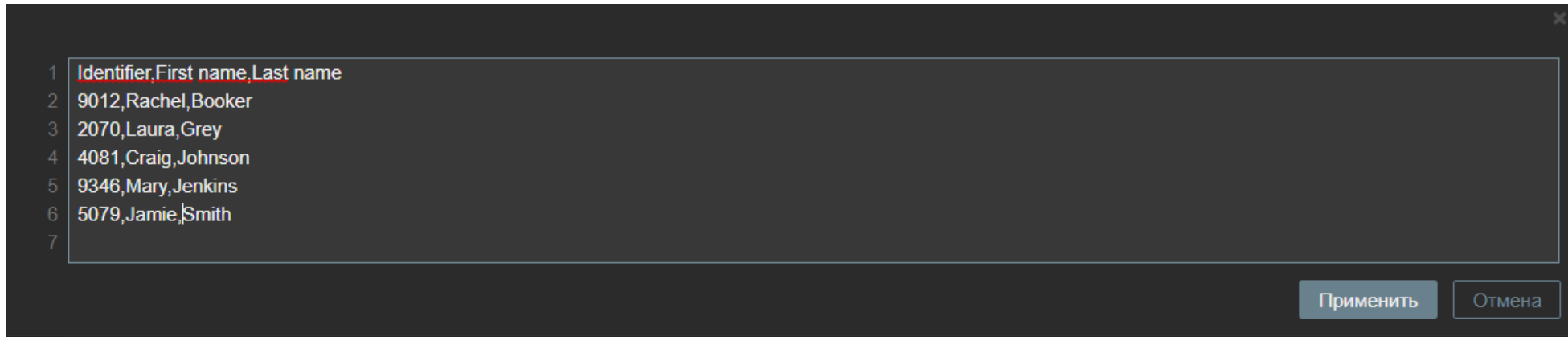
Предыдущее значение Пред. время

Конец строки

Шаги предобработки

Имя	Результат
1: Шестнадцатеричное в десятичное	2020

CSV B JSON



```
1 Identifier,First name,Last name
2 9012,Rachel,Booker
3 2070,Laura,Grey
4 4081,Craig,Johnson
5 9346,Mary,Jenkins
6 5079,Jamie,Smith
7
```

Применить Отмена

```
[{"Identifier":"9012","First name":"Rachel","Last name":"Booker"},
{"Identifier":"2070","First name":"Laura","Last name":"Grey"},
{"Identifier":"4081","First name":"Craig","Last name":"Johnson"},
{"Identifier":"9346","First name":"Mary","Last name":"Jenkins"},
{"Identifier":"5079","First name":"Jamie","Last name":"Smith"}]
```

CSV В JSON

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	CSV в JSON	, "	<input checked="" type="checkbox"/> Со строкой заголовка	Тест Удалить
2:	JSONPath	\$.[0].Identifier	<input type="checkbox"/>	Тест Удалить

[Добавить](#)

[Добавить](#) [Тест](#) [Отмена](#)

[Тест всех шагов](#)

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги предобработки	Имя	Результат
1:	CSV в JSON	{{"Identifier":"9012","First name":"Rachel","Last name":"Booker"},"Identifier":"2070"...
2:	JSONPath	9012

Результат

[Тест](#) [Отмена](#)

ШАБЛОН PROMETHEUS

```
# HELP wmi_logical_disk_free_bytes Free space in bytes
(LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"}
4.59276288e+08

wmi_service_state{name="dhcp",state="continue pending"} 0
wmi_service_state{name="dhcp",state="pause pending"} 0
wmi_service_state{name="dhcp",state="paused"} 0
wmi_service_state{name="dhcp",state="running"} 1
wmi_service_state{name="dhcp",state="start pending"} 0
wmi_service_state{name="dhcp",state="stop pending"} 0
wmi_service_state{name="dhcp",state="stopped"} 0
wmi_service_state{name="dhcp",state="unknown"} 0
```

ШАБЛОН PROMETHEUS

Элемент данных **Предобработка**

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	Шаблон Prometheus	wmi_service_state{name="dhcp"}	<имя метки>	Тест Удалить

[Добавить](#) [Тест](#) [Отмена](#)

[Тест всех шагов](#)

Тест элемента данных

Получить значение с узла сети

Значение [✎](#) Время

Предыдущее значение [✎](#) Пред. время

Конец строки

Шаги предобработки	Имя	Результат
1:	Шаблон Prometheus	1

[Тест](#) [Отмена](#)

ОТСЛЕЖИВАТЬ ИЗМЕНЕНИЯ

- ✔ Простое изменение - Вычисляется значение-пред_значение, где значение - текущее значение; пред_значение - ранее полученное значение. Эта опция может быть полезной для измерения постоянно растущего значения. Если текущее значение меньше.

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

Шаги предобработки	Имя	Результат
1: Простое изменение		768

ИЗМЕНЕНИЯ В СЕКУНДУ

- ⊙ Вычисление изменения значения (разница между текущим и предыдущим значениями) скорость в секунду. Вычисляется $(\text{значение} - \text{пред_значение}) / (\text{время} - \text{пред_время})$, где значение - текущее значение; пред_значение - ранее полученное значение; Время - текущий штамп времени; пред_значение - штамп времени предыдущего значения.

Тест элемента данных

Получить значение с узла сети

Значение Время

Предыдущее значение Пред. время

Конец строки

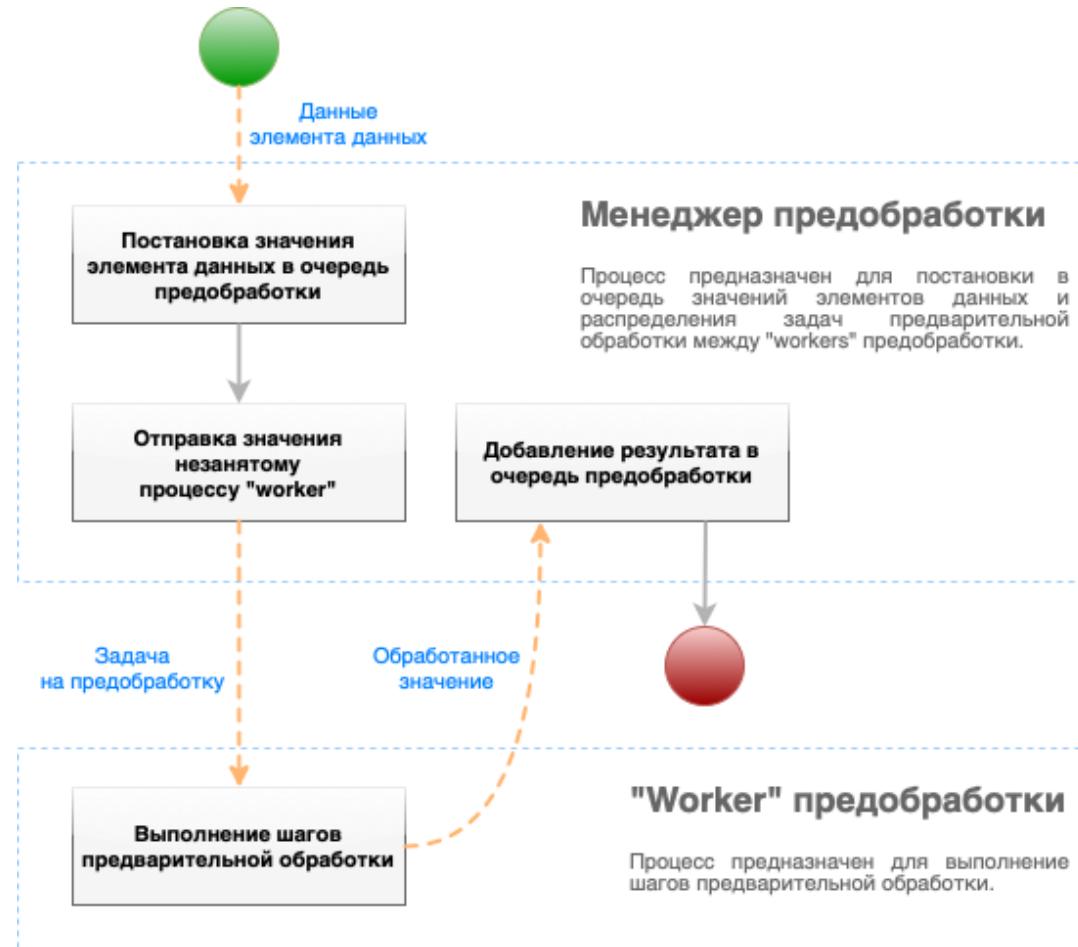
Шаги предобработки	Имя	Результат
	1: Изменение в секунду	153

03

КАК ЭТО РАБОТАЕТ?



КАК ЭТО РАБОТАЕТ?



Элемент данных может быть отправлен в начало или конец очереди. Внутренние элементы данных Zabbix (Zabbix internal items) всегда отправляются в начало очереди предобработки, в то время как остальные элементы данных – в конец очереди.

УЛУЧШЕНИЯ В 5.0

- ☑ Максимальное количество зависимых элементов для одного основного элемента данных теперь 29999 (было 999)
- ☑ [\[ZBX-17694\]](#) Количество использованной памяти значительно уменьшилось
Если раньше, 15K зависимых элементов с основной метрикой в размере 209 KB требовали около 3 GB RAM и 2.115940 секунды на обработку, с новыми изменениями это же количество потребует всего 4.6 MB RAM и займет 0.688623 seconds.
- ☑ [\[ZBX-17720\]](#) Исключение отключенных зависимых элементов из препроцессинга
- ☑ [\[ZBX-17548\]](#) Не сохраняется история в базе прокси если период хранения равен 0

ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ

- ✓ [Regex](https://regex101.com/) - <https://regex101.com/>
- ✓ [JSONpath](https://jsonpath.com/) - <https://jsonpath.com/>
- ✓ [Xpath](http://xpather.com/) - <http://xpather.com/>
- ✓ [Документация](#)

THE PREMIER ZABBIX EVENT ABOUT MONITORING!

ZABBIX



October 30-31, Riga, Latvia

- Meet [Zabbix](#) community and developers
- Learn real use cases and get all your questions answered
- Join workshops run by [Zabbix Dev](#) & Support teams

zabbix.com/events/zabbix_summit_2020

Спасибо!

