# Neues verbessertes SNMP Monitoring in Zabbix ab Version 6.4

Stefan Matzek

Zabbix Trainer, Consultant

IntelliTrend GmbH, Germany

IntelliTrend

ZABBIX

# SNMP-Grundlagen

ZABBIX
PREMIUM PARTNER



IntelliTrend

# SNMP-Grundlagen

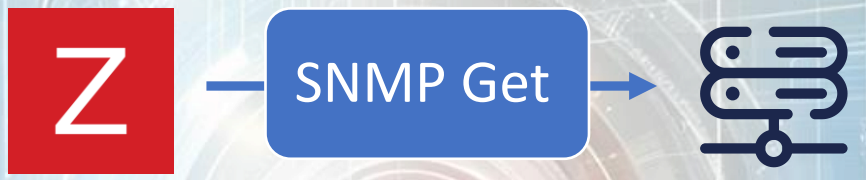
## **SNMP** steht für **S**imple **N**etwork **M**anagement **P**rotocol

SNMP-Manager -> Agent: UDP-Port 161 für Befehle.
SNMP-Agent -> Manager: UDP-Port 162 für Traps.

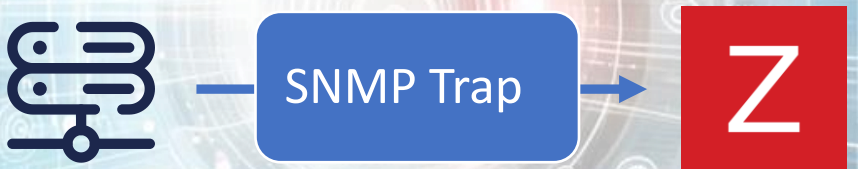
SNMPv1:
- Keine Verschlüsselung
- Community String-basierte Authentifizierung

SNMPv2c:
- Erweiterungen für Leistung und Trap-Handling
- Weiterhin nur Community String-basierte Authentifizierung
- Keine Verschlüsselung

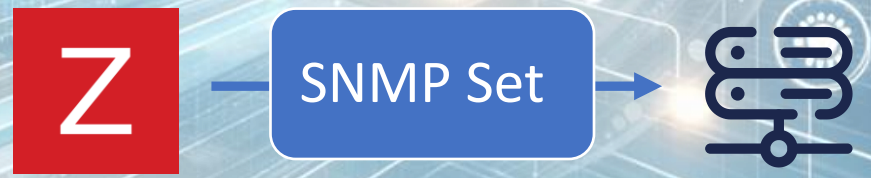
SNMPv3:
- Sichere Authentifizierung
- Verschlüsselung
- Benutzerbasierte Sicherheitsmodelle




Informationen von Geräten abrufen




Geräte schicken Alarme




Geräte können konfiguriert werden
Zabbix unterstütz diese Funktion nicht

ZABBIX PREMIUM PARTNER



IntelliTrend

# SNMP-Grundlagen

## OID

- "Object Identifier„
- Beispiel: 1.3.6.1.2.1.1.5.0
- Keine OS-Abhängigkeit in Zabbix
- Templates in Zabbix verwenden oft OIDs

## MIB

- "Managed Information Base„
- Beispiel: SNMPv2-MIB::sysName.0
- Menschenlesbare Beschreibungen zu OIDs
- Müssen im OS hinterlegt sein, damit Zabbix MIB-Namen statt OIDs verwenden kann.

```
1.3.6.1.2.1.1.5 - sysName
1.3.6.1.2.1.1 - SNMP MIB-2 System
1.3.6.1.2.1 - SNMP MIB-2
1.3.6.1.2 - IETF Management
1.3.6.1 - OID assignments from 1.3.6.1 - Internet
1.3.6 - US Department of Defense
1.3 - ISO Identified Organization
1 - ISO assigned OIDs
```

ZABBIX PREMIUM PARTNER



IntelliTrend

# SNMP-Monitoring vor Zabbix 6.4

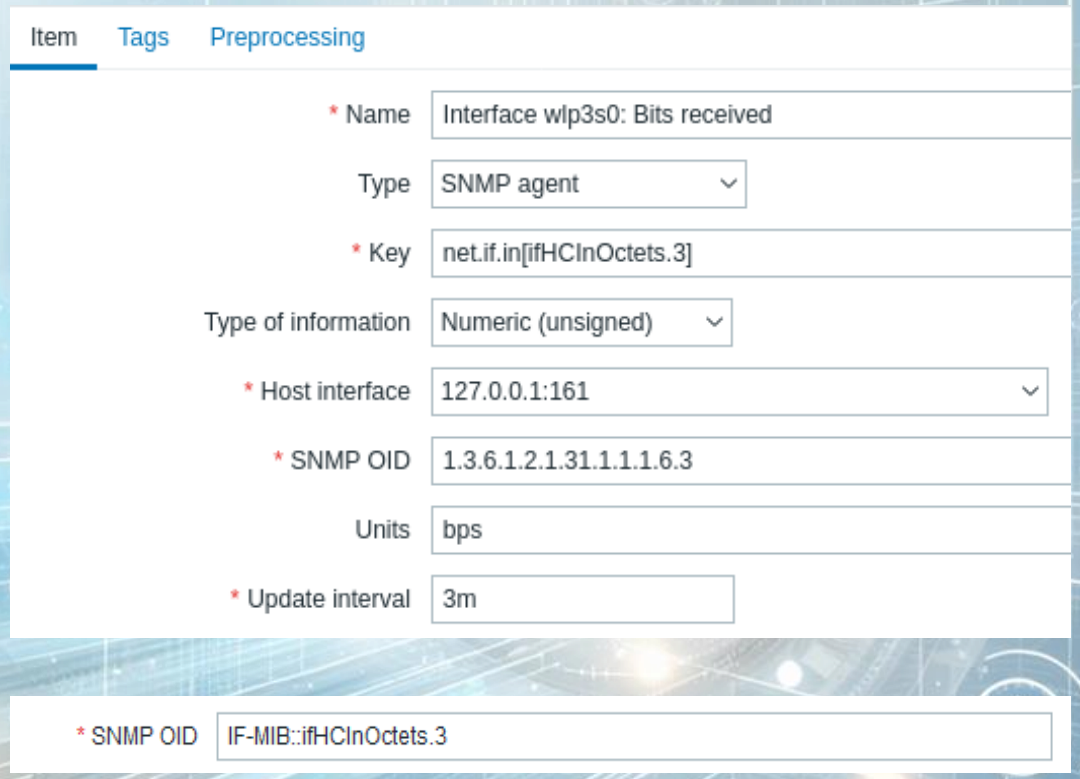
ZABBIX
PREMIUM PARTNER



IntelliTrend

# SNMP-Monitoring vor Zabbix 6.4

## **SNMP** Item




SNMP GET Request:

- „Normales“ SNMP Agent Item
- OID/MIB als Identifikator der Metrik
- Eine Metrik pro Anfrage

ZABBIX PREMIUM PARTNER



IntelliTrend

# SNMP-Monitoring vor Zabbix 6.4

## **SNMP** Item - Discovery

SNMP Discovery Syntax:

- discovery[{#LLDMACRO1},oid1, {#LLDMACRO2},oid2]

Beispiel:

- discovery[{#IFNAME},ifName]
- discovery[{#IFNAME},1.3.6.1.2.1.31.1.1.1.1]

```
[
  {
    "{#SNMPINDEX}": "1",
    "{#IFNAME}": "eth1"
  },
  {
    "{#SNMPINDEX}": "2",
    "{#IFNAME}": "eth2"
  }
]
```

- Dynamische Ressourcen-Überwachung
- Reduziert manuellen Konfigurationsaufwand

ZABBIX PREMIUM PARTNER



IntelliTrend

# SNMP-Monitoring vor Zabbix 6.4

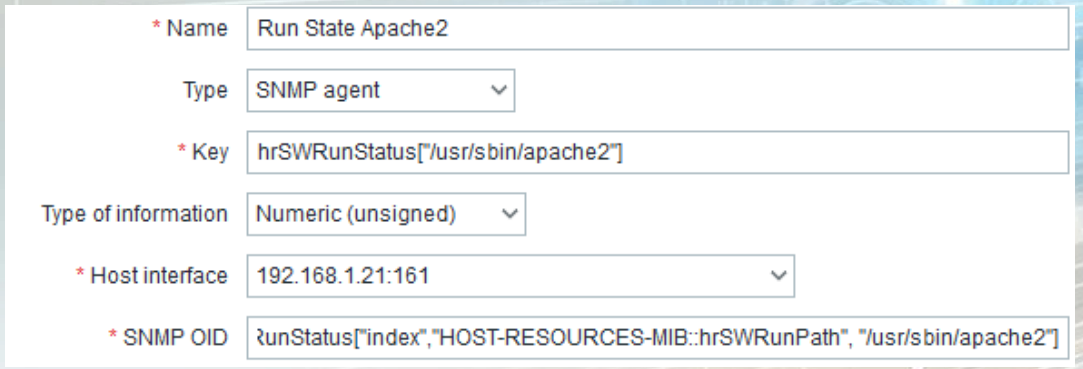
## **SNMP** Item - Dynamic Index

SNMP Dynamic Index Syntax:

- <OID of data>["index","<base OID of index>","<string to search for>"]

Beispiel:

- HOST-RESOURCES-MIB::hrSWRunStatus["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]




```
1  HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"
2  HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"
3  HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"
4  HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"
```

ZABBIX PREMIUM PARTNER

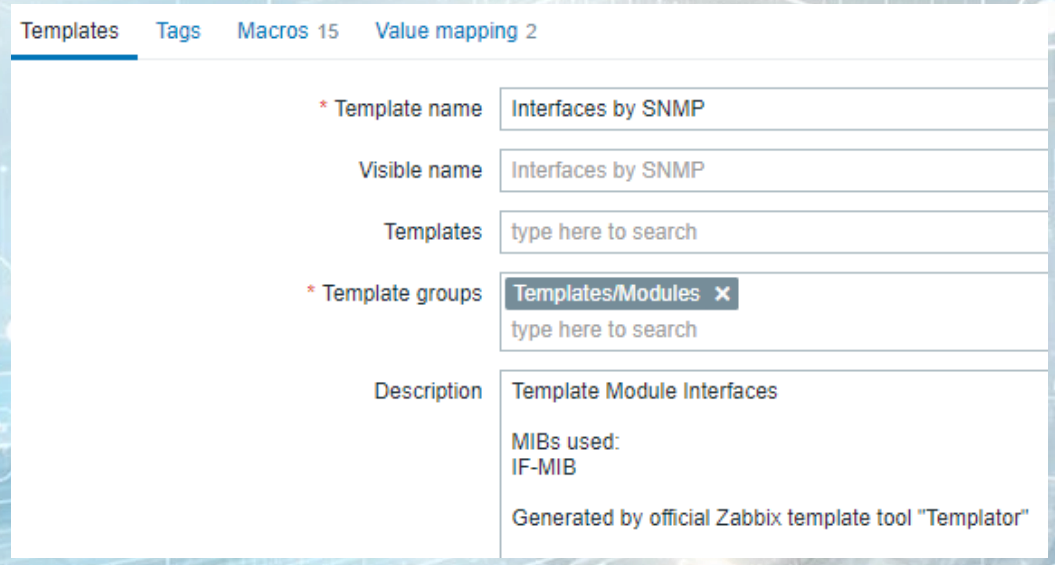


IntelliTrend

# Beispiel für eine SNMP-Discovery vor Zabbix 6.4

# Beispiel für eine SNMP-Discovery vor Zabbix 6.4

## Template: Interfaces by SNMP




- Modul-Template
- Automatisierte Schnittstellenüberwachung
- Nutzt SNMP zur Datenerfassung

ZABBIX PREMIUM PARTNER

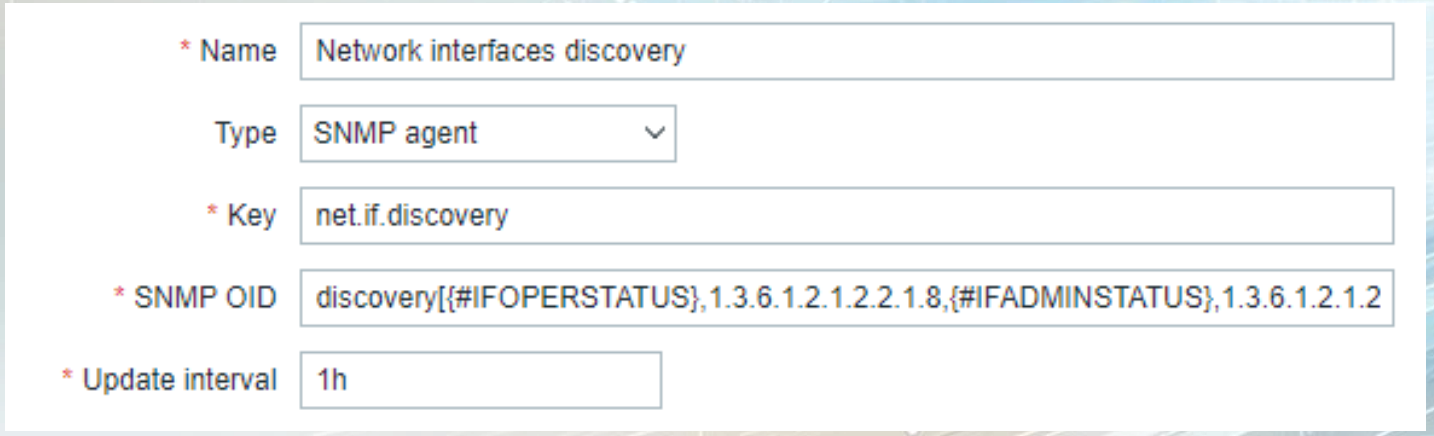


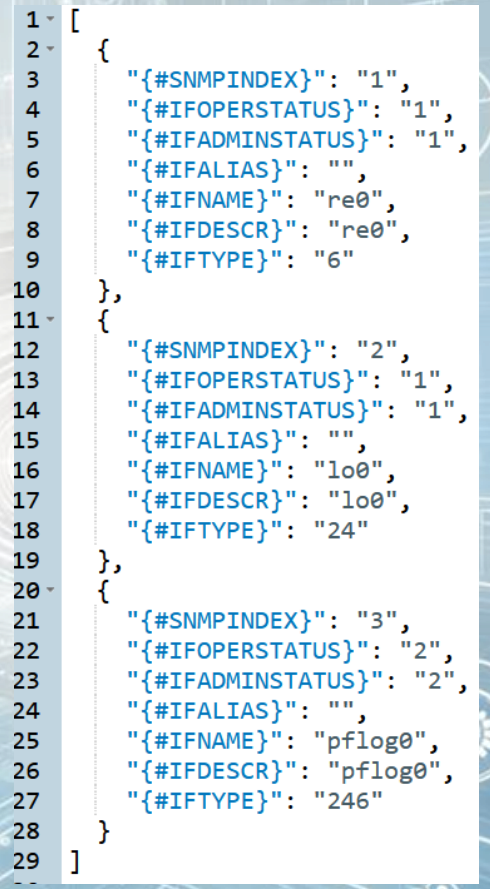
IntelliTrend

# Beispiel für eine SNMP-Discovery vor Zabbix 6.4

## Low Level Discovery rule

- Sammelt Interface-Metadaten via SNMP Discovery
- Verwendet Ergebnisse in LLD-Makros
- Periodische Ausführung (in der Regel stündlich)




ZABBIX PREMIUM PARTNER



IntelliTrend

# Beispiel für eine SNMP-Discovery vor Zabbix 6.4

## Low Level Discovery - Prototypen

- Entitäten werden auf Basis von Prototypen erstellt (Items/Trigger/Graphs)
- LLD-Makros liefern Metadaten für Entitäten

| Name ▲ | Key | Interval | History | Trends | Type |
|---|---|---|---|---|---|
| Interface {#IFNAME}({#IFALIAS}): Bits received | net.if.in[ifHCInOctets.{#SNMPINDEX}] | 3m | 7d | 365d | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Bits sent | net.if.out[ifHCOutOctets.{#SNMPINDEX}] | 3m | 7d | 365d | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded | net.if.in.discards[ifInDiscards.{#SNMPINDEX}] | 3m | 7d | 365d | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors | net.if.in.errors[ifInErrors.{#SNMPINDEX}] | 3m | 7d | 365d | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Interface type | net.if.type[ifType.{#SNMPINDEX}] | 1h | 7d | 0 | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Operational status | net.if.status[ifOperStatus.{#SNMPINDEX}] | 1m | 7d | 0 | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded | net.if.out.discards[ifOutDiscards.{#SNMPINDEX}] | 3m | 7d | 365d | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors | net.if.out.errors[ifOutErrors.{#SNMPINDEX}] | 3m | 7d | 365d | SNMP agent |
| Interface {#IFNAME}({#IFALIAS}): Speed | net.if.speed[ifHighSpeed.{#SNMPINDEX}] | 5m | 7d | 0 | SNMP agent |

ZABBIX PREMIUM PARTNER



IntelliTrend

# Beispiel für eine SNMP-Discovery vor Zabbix 6.4

## Finales Ergebnis

- Pro Interface Metrik wird ein SNMP Agent Item erstellt

| Name ▲ | Triggers | Key | Interval | History | Trends | Type |
|---|---|---|---|---|---|---|
| Network interfaces discovery: Interface re0(): Bits received | Triggers 1 | net.if.in[ifHCInOctets.1] | 3m | 7d | 365d | SNMP agent |
| Network interfaces discovery: Interface re0(): Bits sent | Triggers 1 | net.if.out[ifHCOutOctets.1] | 3m | 7d | 365d | SNMP agent |
| Network interfaces discovery: Interface re0(): Inbound packets discarded | | net.if.in.discards[ifInDiscards.1] | 3m | 7d | 365d | SNMP agent |
| Network interfaces discovery: Interface re0(): Inbound packets with errors | Triggers 1 | net.if.in.errors[ifInErrors.1] | 3m | 7d | 365d | SNMP agent |
| Network interfaces discovery: Interface re0(): Interface type | Triggers 1 | net.if.type[ifType.1] | 1h | 7d | 0 | SNMP agent |
| Network interfaces discovery: Interface re0(): Operational status | Triggers 2 | net.if.status[ifOperStatus.1] | 1m | 7d | 0 | SNMP agent |
| Network interfaces discovery: Interface re0(): Outbound packets discarded | | net.if.out.discards[ifOutDiscards.1] | 3m | 7d | 365d | SNMP agent |
| Network interfaces discovery: Interface re0(): Outbound packets with errors | Triggers 1 | net.if.out.errors[ifOutErrors.1] | 3m | 7d | 365d | SNMP agent |
| Network interfaces discovery: Interface re0(): Speed | Triggers 2 | net.if.speed[ifHighSpeed.1] | 5m | 7d | 0 | SNMP agent |

ZABBIX PREMIUM PARTNER



IntelliTrend

# Beispiel für eine SNMP-Discovery vor Zabbix 6.4

## Einschränkungen

- Potenzielle Leistungsprobleme durch viele einzelne SNMP-Abfragen
- Weniger effiziente Bündelung von SNMP-Anfragen
- Erhöhter Overhead bei SNMPv3

# Neue SNMP Discovery-Methode ab Zabbix 6.4

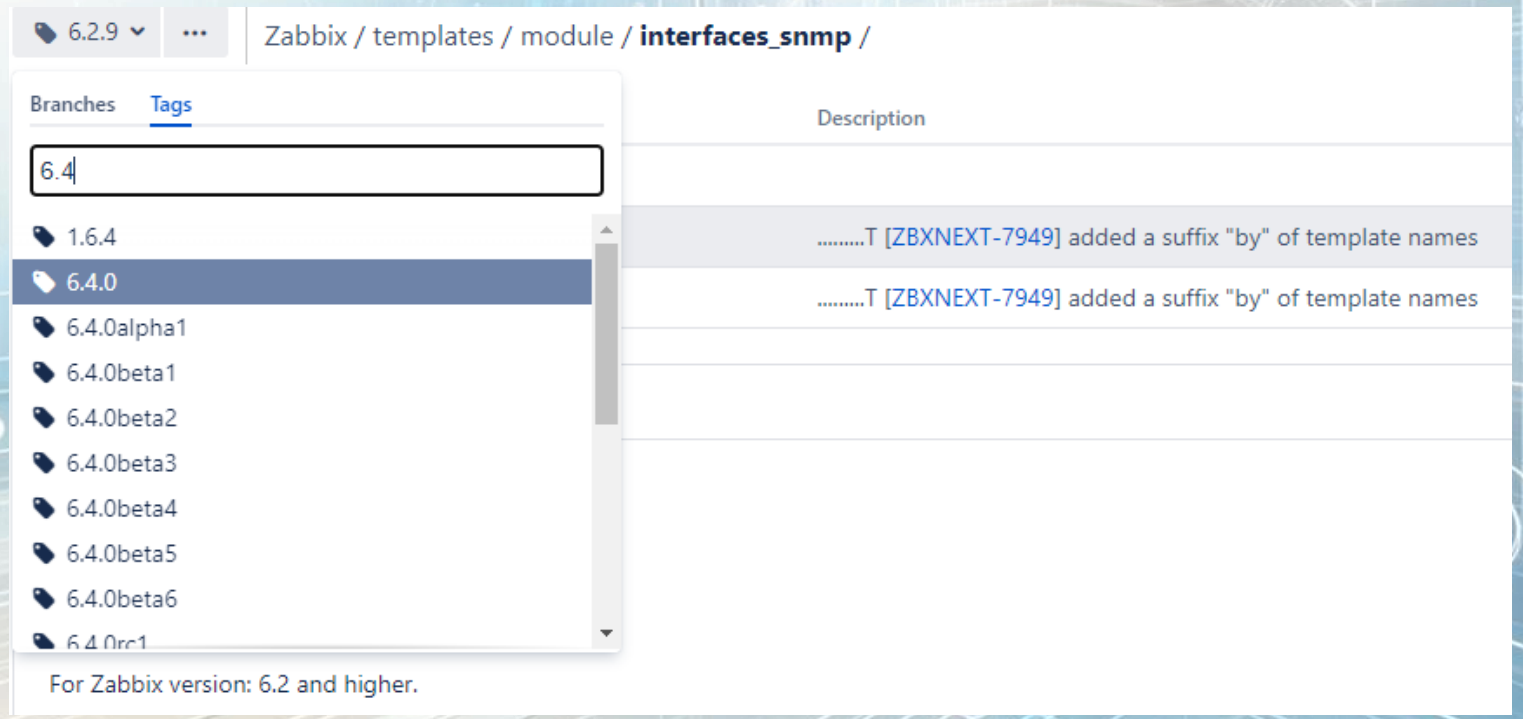
ZABBIX PREMIUM PARTNER



IntelliTrend

# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Template: Interfaces by SNMP?




ZABBIX PREMIUM PARTNER

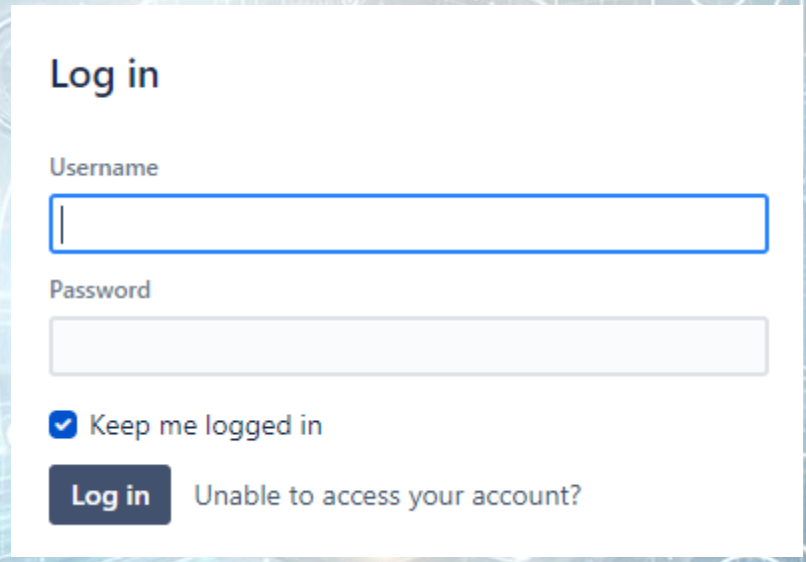


IntelliTrend

# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Template: ~~Interfaces by SNMP~~

- Modul-Templates wurden mit 6.4 entfernt!
- Alternativen?




ZABBIX PREMIUM PARTNER



IntelliTrend

# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Template: FortiGate by SNMP

- Neues Template mit Zabbix 6.4.8 hinzugefügt
- Verwendet neue Interface Discovery Techniken

https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/net/fortinet/fortigate_snmp?at=refs%2Fheads%2Frelease%2F6.4

ZABBIX PREMIUM PARTNER

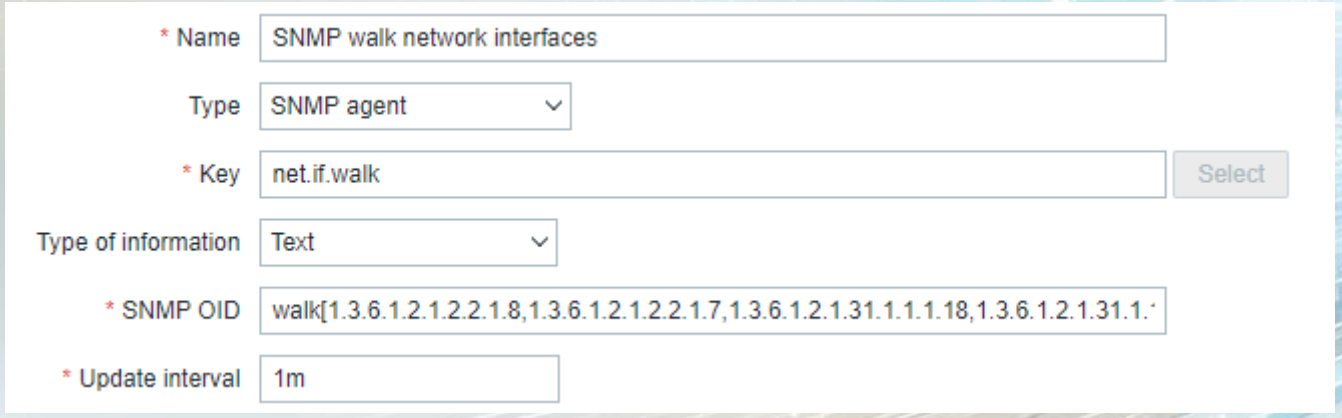


IntelliTrend

# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Master Item: SNMP walk network interfaces

- Neues SNMP walk Item als Master Item
- Ausgabe im SNMP walk Format
- Syntax: walk[oid1,oid2,oid3]




```
1  .1.3.6.1.2.1.2.2.1.13.1 = Counter32: 0
2  .1.3.6.1.2.1.2.2.1.13.2 = Counter32: 0
3  .1.3.6.1.2.1.2.2.1.13.3 = Counter32: 0
4  .1.3.6.1.2.1.2.2.1.14.1 = Counter32: 0
5  .1.3.6.1.2.1.2.2.1.14.2 = Counter32: 0
6  .1.3.6.1.2.1.2.2.1.14.3 = Counter32: 0
7  .1.3.6.1.2.1.2.2.1.19.1 = Counter32: 0
8  .1.3.6.1.2.1.2.2.1.19.2 = Counter32: 0
9  .1.3.6.1.2.1.2.2.1.19.3 = Counter32: 0
10 .1.3.6.1.2.1.2.2.1.2.1 = STRING: "re0"
11 .1.3.6.1.2.1.2.2.1.2.2 = STRING: "lo0"
12 .1.3.6.1.2.1.2.2.1.2.3 = STRING: "pflog0"
13 .1.3.6.1.2.1.2.2.1.20.1 = Counter32: 0
14 .1.3.6.1.2.1.2.2.1.20.2 = Counter32: 0
15 .1.3.6.1.2.1.2.2.1.20.3 = Counter32: 0
16 .1.3.6.1.2.1.2.2.1.3.1 = INTEGER: 6
17 .1.3.6.1.2.1.2.2.1.3.2 = INTEGER: 24
18 .1.3.6.1.2.1.2.2.1.3.3 = INTEGER: 246
19 .1.3.6.1.2.1.2.2.1.7.1 = INTEGER: 1
20 .1.3.6.1.2.1.2.2.1.7.2 = INTEGER: 1
21 .1.3.6.1.2.1.2.2.1.7.3 = INTEGER: 2
22 .1.3.6.1.2.1.2.2.1.8.1 = INTEGER: 1
23 .1.3.6.1.2.1.2.2.1.8.2 = INTEGER: 1
24 .1.3.6.1.2.1.2.2.1.8.3 = INTEGER: 2
25 .1.3.6.1.2.1.31.1.1.1.1.1 = STRING: "re0"
26 .1.3.6.1.2.1.31.1.1.1.1.2 = STRING: "lo0"
27 .1.3.6.1.2.1.31.1.1.1.1.3 = STRING: "pflog0"
28 .1.3.6.1.2.1.31.1.1.1.10.1 = Counter64: 263688948503
29 .1.3.6.1.2.1.31.1.1.1.10.2 = Counter64: 1608
30 .1.3.6.1.2.1.31.1.1.1.10.3 = Counter64: 0
31 .1.3.6.1.2.1.31.1.1.1.15.1 = Gauge32: 1000
32 .1.3.6.1.2.1.31.1.1.1.15.2 = Gauge32: 0
33 .1.3.6.1.2.1.31.1.1.1.15.3 = Gauge32: 0
```

ZABBIX PREMIUM PARTNER

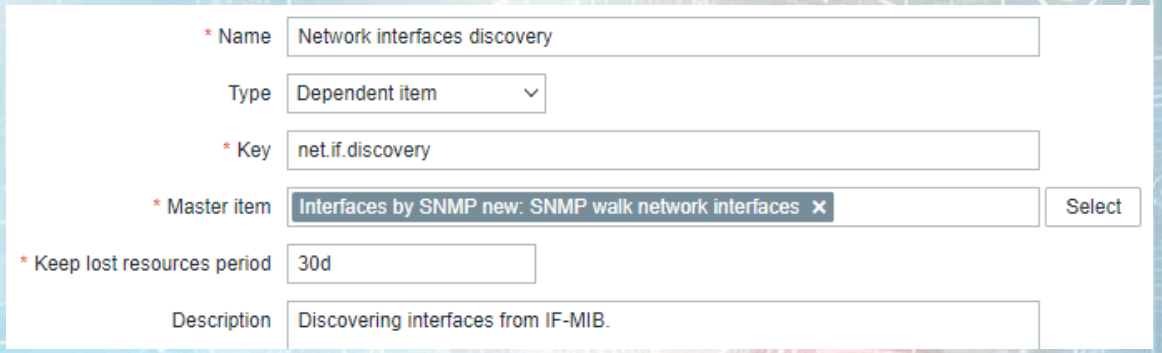


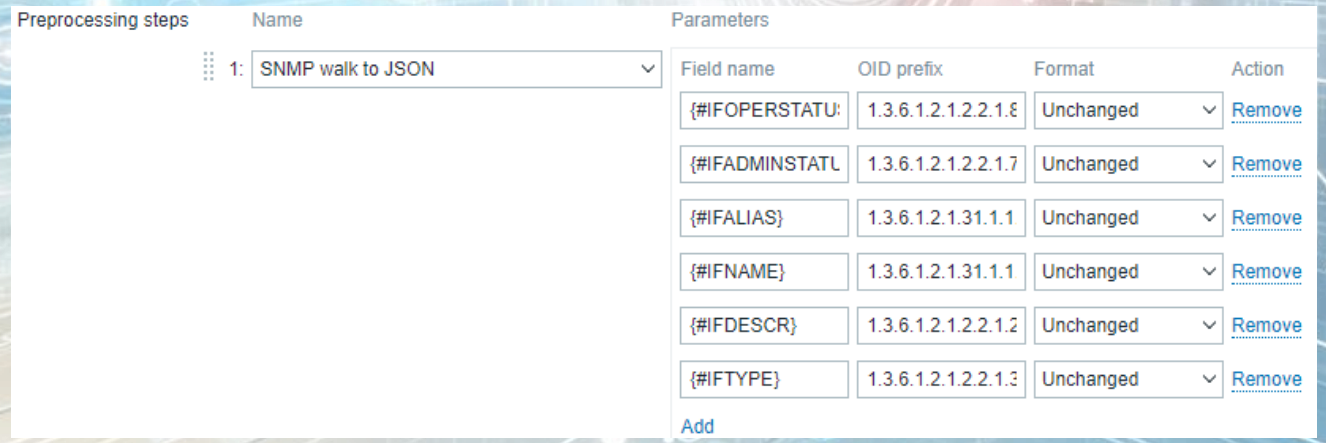
IntelliTrend

# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Dependent Discovery Rule: Network interfaces discovery

- Verwendet neuen Preprocessing Step „SNMP walk to JSON“
- Metadaten aus dem walk Format in LLD-Makros verarbeitet
- Input in die LLD nach Preprocessing Step gleich wie in der Methode vor Zabbix 6.4



# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Low Level Discovery - Item Prototypen

- Dependend Items werden aus dem SNMP walk Master Item abgeleitet
- Keine individuellen Item-Intervalle
- Preprocessing Step “Discard unchanged with Heartbeat” dringend empfohlen.

| Name ▲ | Key | Interval | History | Trends | Type |
|---|---|---|---|---|---|
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Bits received | net.if.in[ifInOctets.{#SNMPINDEX}] | | 7d | 365d | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Bits sent | net.if.out[ifOutOctets.{#SNMPINDEX}] | | 7d | 365d | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded | net.if.in.discards[ifInDiscards.{#SNMPINDEX}] | | 7d | 365d | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors | net.if.in.errors[ifInErrors.{#SNMPINDEX}] | | 7d | 365d | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Interface type | net.if.type[ifType.{#SNMPINDEX}] | | 7d | 0 | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Operational status | net.if.status[ifOperStatus.{#SNMPINDEX}] | | 7d | 0 | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded | net.if.out.discards[ifOutDiscards.{#SNMPINDEX}] | | 7d | 365d | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors | net.if.out.errors[ifOutErrors.{#SNMPINDEX}] | | 7d | 365d | Dependent item |
| SNMP walk network interfaces: Interface {#IFNAME}({#IFALIAS}): Speed | net.if.speed[ifSpeed.{#SNMPINDEX}] | | 7d | 0 | Dependent item |

ZABBIX PREMIUM PARTNER

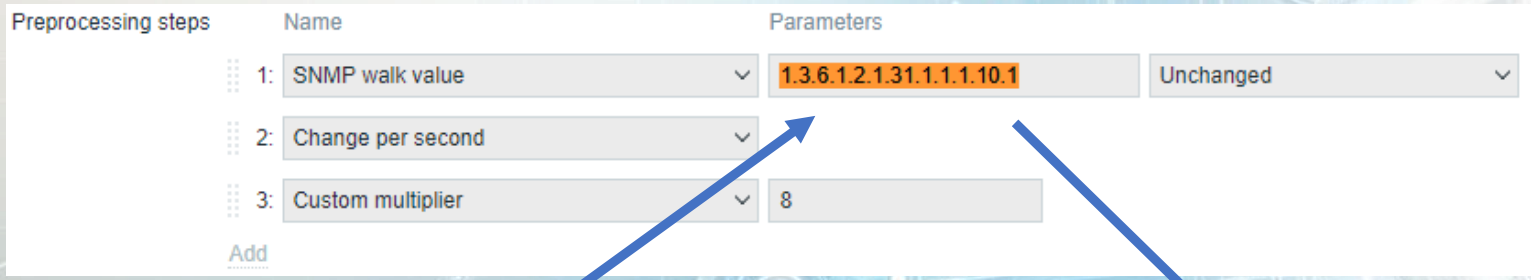


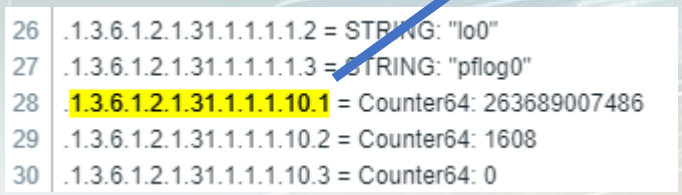
IntelliTrend

# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Low Level Discovery - Item Prototypen

- Items nutzen neuen Preprocessing Step zur Datenextraktion




# Neue SNMP Discovery-Methode ab Zabbix 6.4

## Vorteile

- Effizienz durch bessere Bündelung der Abfragen
- Schnellere Anpassung des Monitorings bei Entitätsänderungen
- Deutliche Performancevorteile mit SNMPv3

ZABBIX PREMIUM PARTNER



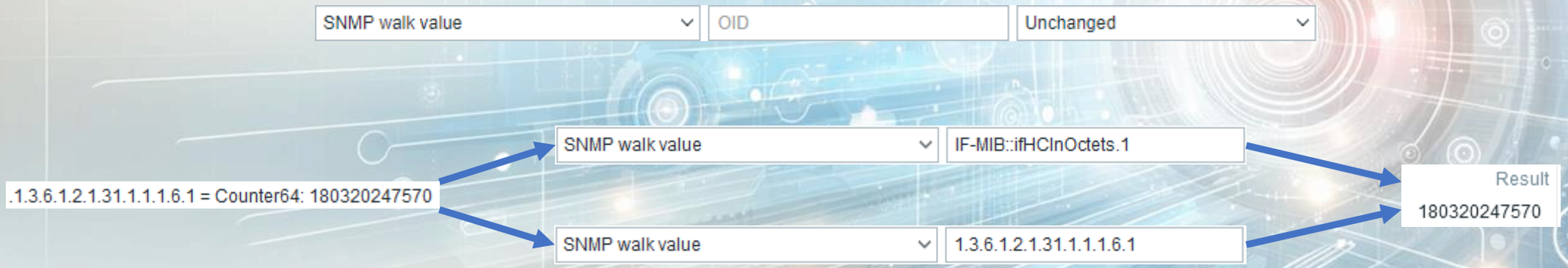
IntelliTrend

# Neue Funktionen im Detail

# Neue Preprocessing-Schritte

## SNMP walk value

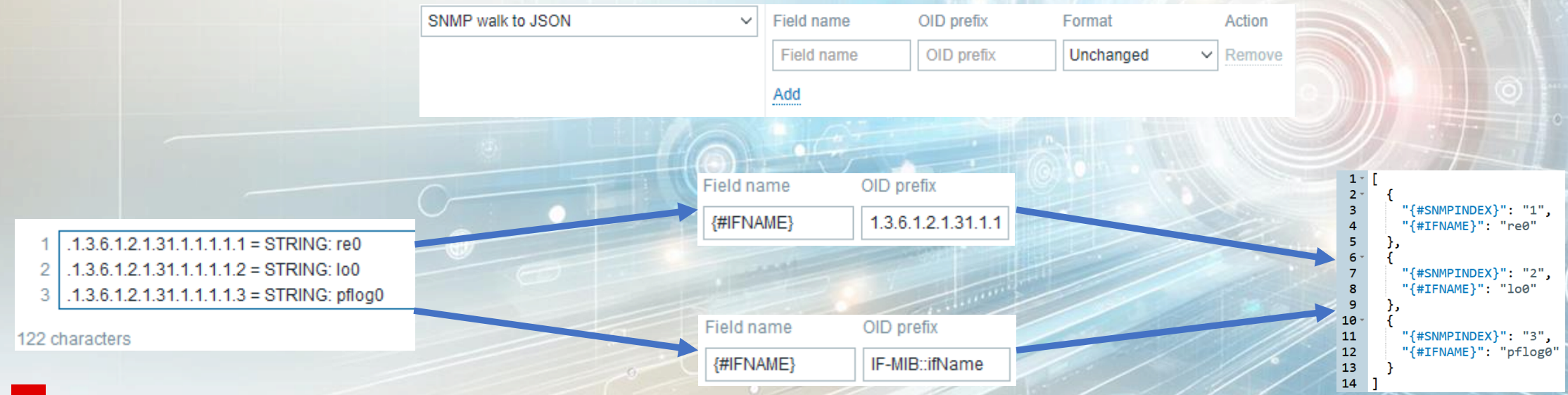
- Extraktion von Werten über spezifizierte OID/MIB-Namen.
- Anwendung von Formatierungsoptionen wie Umwandlung in UTF-8 oder MAC-Adressformatierung.



# Neue Preprocessing-Schritte

## SNMP walk to JSON

- Umwandlung von SNMP-Werten in ein JSON-Format.
- Ermöglicht das Zuordnen von Feldnamen im JSON zu bestimmten SNMP OID-Pfaden.



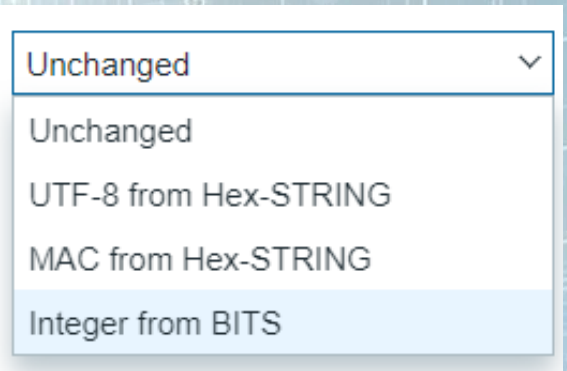

ZABBIX PREMIUM PARTNER



IntelliTrend

# Neue Preprocessing-Schritte

## Konvertierungsmethoden




- Unchanged: Gibt Hex-Strings als nicht-umgewandelte Hex-Zeichenketten zurück.
- UTF-8 from Hex-STRING: Wandelt Hex-Strings in UTF-8 Zeichenketten um.
- MAC from Hex-STRING: Überprüft Hex-Strings auf gültige MAC-Adressformate.
- Integer from BITS: Konvertiert die ersten 8 Bytes eines Bitstrings in eine 64-Bit Ganzzahl.

ZABBIX PREMIUM PARTNER

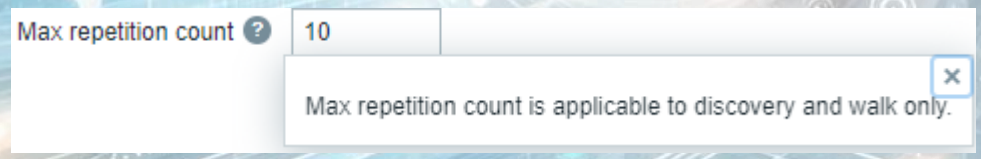


IntelliTrend

# SNMP Interface Änderungen

## Max repetition count

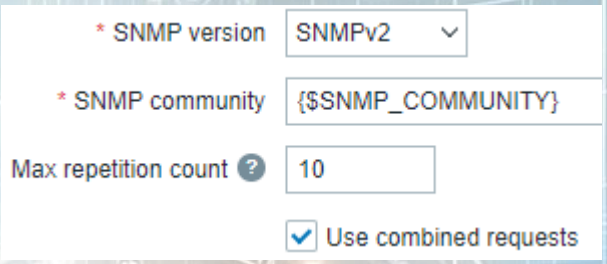
- "Max Repetition Count" definiert Anzahl der Objekte in GetBulk-Request
- Nur für discovery und walk
- Ein höherer Wert kann Netzwerk und Gerät stärker belasten, beschleunigt aber die Abfrage großer Datenmengen
- Wert 10 bedeutet: bis zu 10 Objekte in einem GetBulk-Request abfragen



# SNMP Interface Änderungen

## Use combined requests

- Vor Zabbix 6.4 als "Bulk Requests" bekannt
- In Zabbix 6.4 zu "Combined Requests" umbenannt
- Änderung nur im Namen, Funktionalität gleich geblieben
- Kombiniert SNMP-Requests für Effizienz
- Zabbix-eigene Methode zur Effizienzsteigerung
- Alternative zu nativen SNMP GetBulk-Requests




ZABBIX PREMIUM PARTNER



IntelliTrend

# Vielen Dank für Ihre Aufmerksamkeit!

Stefan Matzek

Zabbix Trainer, Consultant

IntelliTrend GmbH, Germany

IntelliTrend

ZABBIX