



Data Models

Feature Document

Table of Contents

| | |
|---|----|
| Table of Contents | 2 |
| Copyright Notice..... | 3 |
| Notices | 4 |
| 1 Introduction..... | 6 |
| 1.1 What is Data Model?..... | 6 |
| 1.2 Types of Data Models..... | 6 |
| 1.3 How Motadata Uses Data Models? | 6 |
| 1.4 Default Signatures..... | 7 |
| 1.5 Default Data Models..... | 8 |
| 1.6 User Defined Models..... | 8 |
| 2 Creating a Device Signature..... | 9 |
| 2.1 Adding Scalar Model..... | 11 |
| 2.2 Adding Tabular Model | 12 |
| 3 Verifying Device Signature..... | 16 |
| 3.1 Signature OID..... | 16 |
| 3.2 Data Models..... | 16 |
| 3.3 Monitor Widgets | 17 |
| 4 Create New Data Model | 19 |
| 4.1 Custom Data Model..... | 19 |
| 4.2 Composite Data Model | 20 |
| 5 Additional Information - Add Query in Monitors..... | 22 |

Copyright Notice

The information contained in this document represents the views and opinions of Mindarray Systems Pvt. Ltd. on the issue as of the date of publication. Because of the dynamic nature of the IT Industry and the technology that is behind it, Mindarray Systems Pvt. Ltd. can make no warranty as to the long-term accuracy of the assessment. These materials are confidential and proprietary to Mindarray Systems Pvt. Ltd. and no part of these materials should be reproduced, published in any form by any means, electronic or mechanical including photocopy or any information storage or retrieval system, nor should the material be disclosed to third parties without the express written authorization of Mindarray Systems Pvt. Ltd. Information in this document is subject to change without notice and does not represent a commitment on the part of Mindarray Systems Pvt. Ltd.

Notices

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE DOWNLOADING OR USING THE SOFTWARE. BY CLICKING ON THE "I ACCEPT THE TERMS OF THE LICENSE AGREEMENT" BUTTON, OPENING THE PACKAGE, DOWNLOADING THE PRODUCT, OR USING THIS PRODUCT, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE WITH ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE "I DO NOT ACCEPT THE TERMS OF THE LICENSE AGREEMENT" BUTTON AND THE INSTALLATION PROCESS WILL NOT CONTINUE. RETURN THE PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND, OR DO NOT DOWNLOAD THE PRODUCT. YOUR GENERAL TERMS OF BUSINESS DO NOT APPLY.

General

In this software license agreement:

- a) "Mindarray" means Mindarray Systems Pvt Ltd., 14/3, Magnet Corporate Park, 100 Feet Road, S.G Highway, Near Sola Bridge Opp. Grand Cambay, Thaltej, Ahmedabad, Gujarat. India.
- b) "Customer" means the individual(s), organization or business entity buying a license of the software from Mindarray or its distributors or its resellers.
- c) "Software" means computer programs (and their storage medium) supplied by Mindarray and known collectively as "Mindarray IP Address Manager" in which Mindarray has propriety rights for its any user manuals, example code, operating instructions, brochures and all other documentation relating to the said computer programs (the expression "software" to include all or any part or any combination of software).

License Grant

This license grants you the following rights:

- a) Software product: Mindarray grants to you an exclusive license to use the software for the sole purposes of designing, developing, and testing your software components or applications ("applications"). You may install the software on any computer in your organization.
- b) Electronic documents: Solely with respect to electronic documents included with the software, you may make an unlimited number of copies (either in hardcopy or electronic form), provided that such copies shall be used only for internal purposes and are not republished or distributed to any third party.
- c) License file: A file provided at the time of sale uniquely identifies each license. This license grant is contingent upon the purchase of a license file from Mindarray or one of Mindarray's resellers.
- d) Sample code: Mindarray grants you the right to use and modify the source code parts of the software that are listed in the "projects" and "scripts" subdirectories (if available).
- e) Redistribution: The software is made available for download solely for use by end users according to the license agreement. Any reproduction or redistribution of the software not in accordance with the license agreement is expressly prohibited.
- f) Trial software: If the software is installed without a serial number then, notwithstanding other sections of this license, you may use the software for up to 30 days after installation.
- g) Not for resale software: If the software is labeled as "not for resale " or "NFR" then, notwithstanding other sections of this license, you may not resell, or otherwise transfer the value of software, neither distribute any redistributables.
- h) Reservation of rights: Mindarray reserves all rights not expressly granted to you in this license agreement. The license is granted to the customer on a non-exclusive-basis which means that Mindarray will grant the license also to their individuals, organizations and business entities.
- i) This license agreement consists of no obligations for Mindarray to offer support (services), help (services) or maintenance (services) relating to the software. Obligations for Mindarray to offer maintenance (services) relating to the software can only arise from a maintenance agreement between Mindarray and customer. General terms of business of the customer do not apply.

Upgrades and Supplements

If the software is labeled as an upgrade, you must be properly licensed to use a product identified by Mindarray as being eligible for the upgrade in order to use the software. Software labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this license unless we provide other terms along with the update or supplement. If the software is an upgrade of a component or a package or software programs that you licensed as a single product, the software may be used and transferred only as part of that single product package.

Limitation on Reverse Engineering, Decompilation, And Disassembly

Customer may not reverse engineer, decompile, or disassemble the software, except and only to the extent that it is expressly permitted by applicable law notwithstanding this limitation.

Termination

Without prejudice to any other rights, Mindarray may cancel or dissolve this license agreement if the customer does not abide by the terms and conditions of this license agreement, in which case customer must destroy all copies of the software and all of its component parts.

Limited Warranty

Mindarray warrants that for a period of ninety (90) days from the date of shipment from Mindarray. The media on which the software is furnished will be free of defects in materials and workmanship under normal use.

The software substantially conforms to its published specifications. Except for the foregoing, the software is provided as is. This limited warranty extends only to the customer as the original licensee. Customer's exclusive remedy and the entire liability of Mindarray and its suppliers under this limited warranty will be, at Mindarray or its service center's option, repair, replacement, or refund of the software if reported (or, upon request, returned) to the party supplying the software to the customer. In no event does Mindarray warrants that the software is error-free or that customer will be able to operate the software without problems or interruptions. The customer will safeguard Mindarray against any claim relating to the use of the software by the customer. This warranty does not apply if the software: (a) has been altered, except by Mindarray; (b) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by Mindarray; (c) has been subjected to abnormal physical or electrical stress misuse, negligence, or accident; (d) is used in high-risk activities, including the operation of nuclear facilities, aircraft navigation, air traffic control, weapons systems, life support or medical applications for use in any circumstance in which the failure of the software could lead directly to death, personal injury or damage to properties or the environment.

Limitation of Liability and Remedies

NOTWITHSTANDING ANY DAMAGES THAT YOU MIGHT INCUR FOR ANY REASON WHATSOEVER (INCLUDING, WITHOUT LIMITATION ALL INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR MULTIPLE DAMAGES SUCH AS BUT NOT LIMITED TO, LOST BUSINESS OR PROFITS, LOSS OF GOODWILL, WORKS TOP PAGE AND DATA LOSS), THE ENTIRE LIABILITY OF MINDARRAY AND ANY OF ITS SUPPLIERS UNDER ANY PROVISION OF THIS LICENSE AGREEMENT AND YOUR EXCLUSIVE REMEDY FOR ALL OF THE FOREGOING (EXCEPT FOR ANY REMEDY OF REPAIR OR REPLACEMENT ELECTED BY MINDARRAY WITH RESPECT TO ANY BREACH OF THE LIMITED WARRANTY) SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE. MINDARRAY IS RELIEVED OF ANY OBLIGATION TO PAY DAMAGES IF THE CUSTOMER HAS NOT UPGRADED THE SOFTWARE WHEN POSSIBLE. THE FOREGOING LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS (INCLUDING SECTIONS 4, 5 AND 6 ABOVE) SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE.

Entire Agreement

This license agreement (including any addendum or amendment to this license agreement which is included with the software) is the entire agreement between you and Mindarray relating to the software and the support services (if any) and they supersede all prior or contemporaneous oral or written communications, proposals and representations with respect to the software or any other subject matter covered by this license agreement. To the extent the terms of any Mindarray policies or programs for support services conflict with the terms of this license agreement, the terms of this license agreement shall control.

The customer is not allowed to alienate or transfer any rights relating to this license agreement without the written approval of Mindarray.

THIS AGREEMENT SHALL BE CONSTRUED IN ACCORDANCE WITH THE LAWS OF THE INDIAN GOVERNMENT AND THE INDIAN COURTS SHALL HAVE SOLE JURISDICTION IN ANY DISPUTE RELATING TO THESE CONDITIONS. ALL DISPUTES HEREUNDER SHALL BE RESOLVED EXCLUSIVELY IN THE APPROPRIATE COURT IN THE CITY OF AHMEDABAD, INDIA. If any part of these conditions shall be or become invalid or unenforceable in any way and to any extent by any existing or future rule of law, order, statute or regulation applicable thereto, then the other conditions shall remain in full force and effect as all other provisions.

The conditions of this license agreement remain applicable after the termination of this license agreement if this results from the nature of the condition.

Copyright

The software is protected by copyright and other intellectual property laws and treaties. Mindarray or its suppliers own the title, copyright, and other intellectual property rights in the software. The granting of a license does not constitute a transfer of any intellectual property right. The software is licensed, not sold.

1 Introduction

In this document, we'll see the data models, (Admin > Data Models), their behavior and how we can use them. Working with these data models need some level of network knowledge and understanding of OIDs.

By default Motadata has stored the data models of majorly available devices in the market. This means that configuration of generic counters like RTT, status, packet loss etc. are already available in the system. The data model section is useful when you are using any device and Motadata doesn't have default configuration for it.

1.1 What is Data Model?

Any network or SNMP device requires a model consisting of the system OID, vendor and device type. Motadata uses this data model and the OID's of the counters associated with it to fetch information from the device.

Using the system OID, Motadata understands which monitor it is. When the monitor is identified, Motadata uses OID of counters to fetch specific information of metrics. The metric information then can be seen in monitor's detail page and in widgets.

1.2 Types of Data Models

In Motadata, we have divided OID in two broad segments:

- Scalar Metrics: Returns single metric value only. E.g. CPU (%), Memory, Disk etc.
- Tabular Metrics: Returns multiple metric values. E.g. Fan speed etc.

1.3 How Motadata Uses Data Models?

Motadata uses the system OID, vendor name and device type to create 'Device Signature'. For each device type of one vendor, there can be only one device signature profile. In this profile, add all the counter IDs you want to monitor.

Let's understand through an example. A Fortinet firewall has a set of OIDs for different counters. You can see the list of OID from [here](#). For CPU usage, the OID is: 1.3.6.1.4.1.12356.101.13.2.1.1.3. Enter this OID with SNMP walk command in Motadata backend.

`snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.13.2.1.1.3`

The output of command is:

```
root@ubuntu:~# snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.13.2.1.1.3
iso.3.6.1.4.1.12356.101.13.2.1.1.3.1 = Gauge32: 4
```

The output is an 'Integer' with value: 4. This example gives us following observations:

- Motadata can do SNMP walk in your system. That means all configurations are correct and Motadata is functioning properly.
- We have correct community and credentials to do SNMP walk.

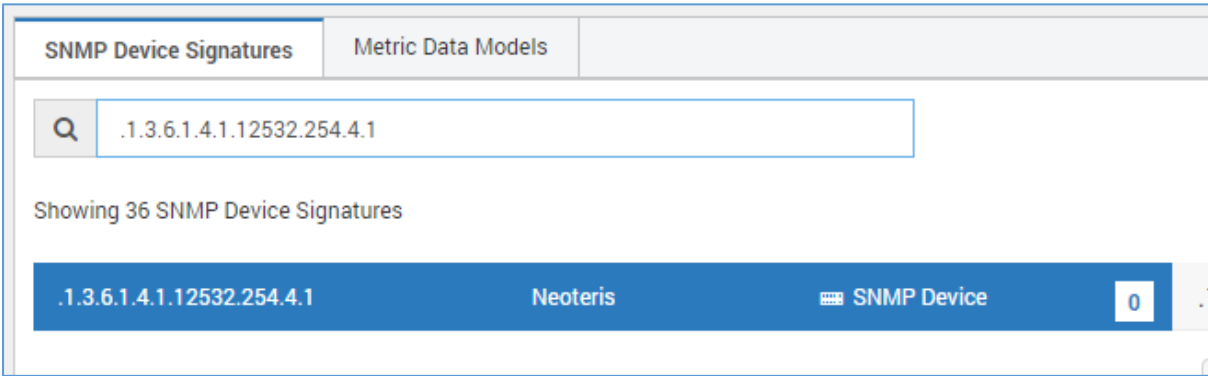
- Motadata can discover the IP 172.16.10.1 and it is network/SNMP device.
- The OID is returning the value and its type. Means we can configure device signature in Motadata GUI.
- The OID returned only 1 value. This is a scalar OID.

****Note:** We shall use the same process of SNMP walk before adding each OID for device counter.

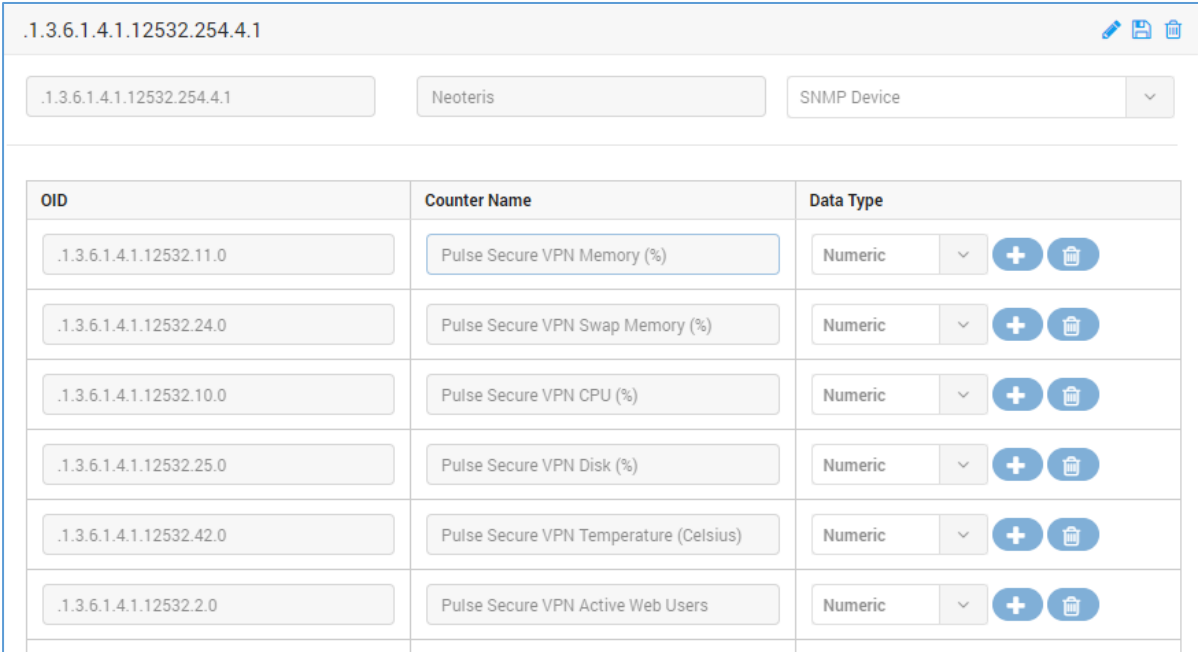
1.4 Default Signatures

Motadata comes with a lot of data models already configured in the system. Ideally, your device signature should be present in the system. To check if your device signature is present or not, you should know system OID of the device. When known, do following:

- Go to Admin > Data Models
- In SNMP Device Signatures tab, search for the system OID.



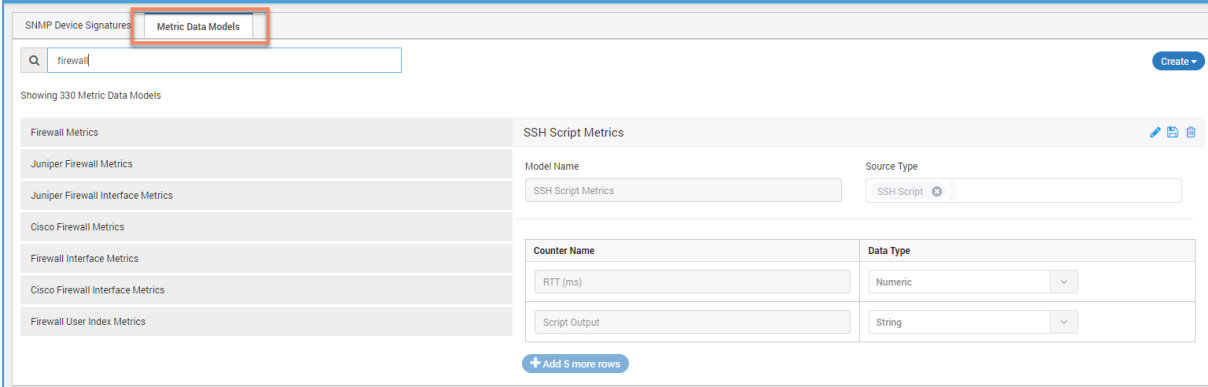
- If your device signature is available, look for all the counter OIDs on the right hand side. If the OID is not available, you can edit the signature and add them.



| OID | Counter Name | Data Type |
|-------------------------|--|-----------|
| .1.3.6.1.4.1.12532.11.0 | Pulse Secure VPN Memory (%) | Numeric |
| .1.3.6.1.4.1.12532.24.0 | Pulse Secure VPN Swap Memory (%) | Numeric |
| .1.3.6.1.4.1.12532.10.0 | Pulse Secure VPN CPU (%) | Numeric |
| .1.3.6.1.4.1.12532.25.0 | Pulse Secure VPN Disk (%) | Numeric |
| .1.3.6.1.4.1.12532.42.0 | Pulse Secure VPN Temperature (Celsius) | Numeric |
| .1.3.6.1.4.1.12532.2.0 | Pulse Secure VPN Active Web Users | Numeric |

1.5 Default Data Models

Based on the signature OID present, Motadata automatically creates the metric data models (see second tab).



Even when you create a new device signature, Motadata automatically adds the data model for it. Again, these data models serve 90% of most people's requirements. In case if they need something specific, they can create composite data models or use queries for it.

1.6 User Defined Models

In Motadata you can also create your own device signatures and data models. To create a device signature you need system OID of the device. Along with it, you should know following things:

- System OID of device
- Vendor of the device
- Device type of the device
- Scalar metric OIDs (if required)
- Tabular metric OIDs (if required)

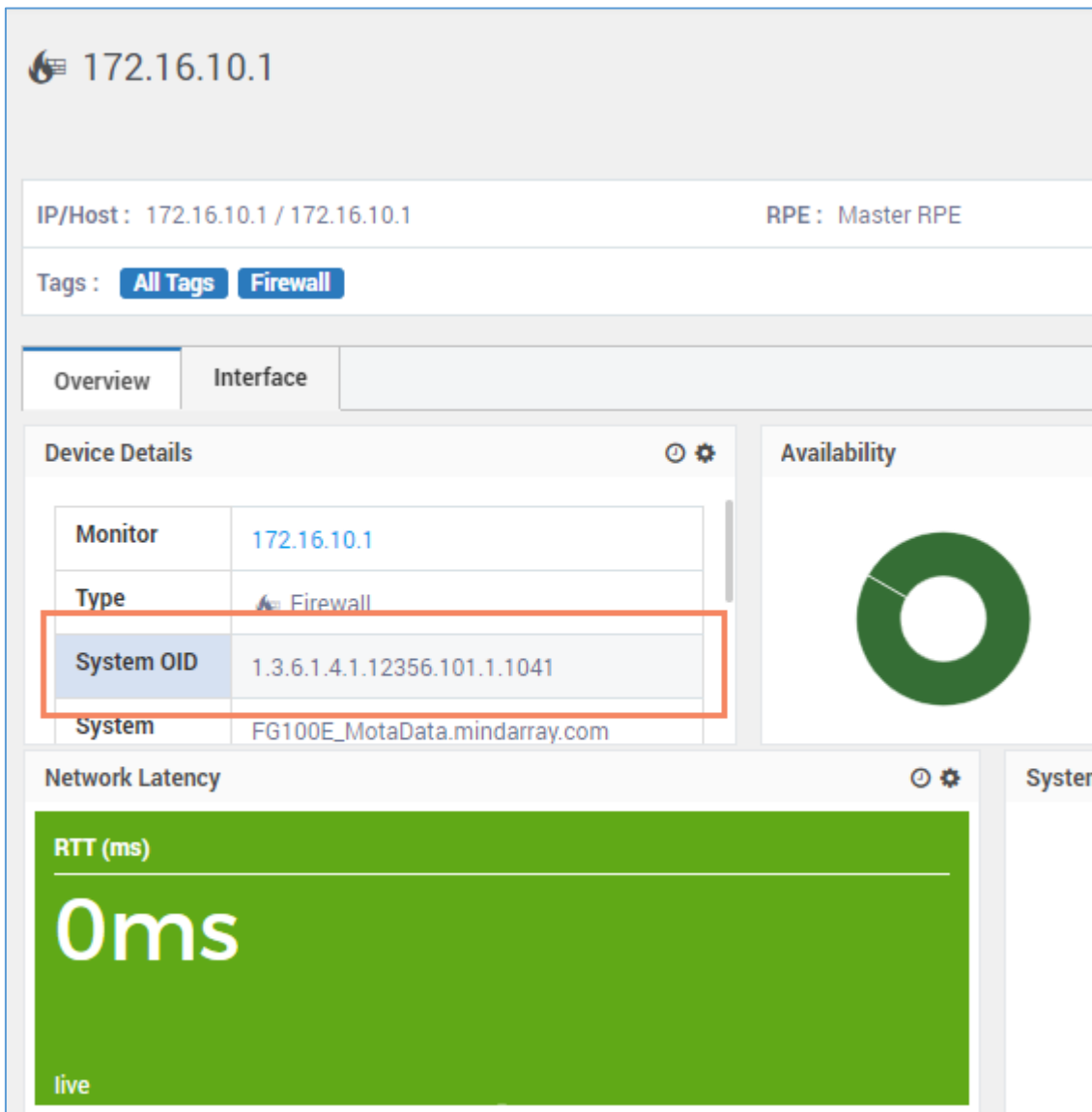
When you'll create a device signature, system will automatically create the data models for it. It will also group the data models using the signature name.

To create specific data models, you can also create composite data model or run it via writing a query.

2 Creating a Device Signature

While we are creating a custom data model, we assume that following requirements are satisfied.

- You know the system OID and metric OIDs you need to monitor. If you don't know the system OID, you can get the system OID from monitor's detail page.



The screenshot shows the Motadata interface for a device with IP 172.16.10.1. The 'Device Details' section contains a table with the following information:

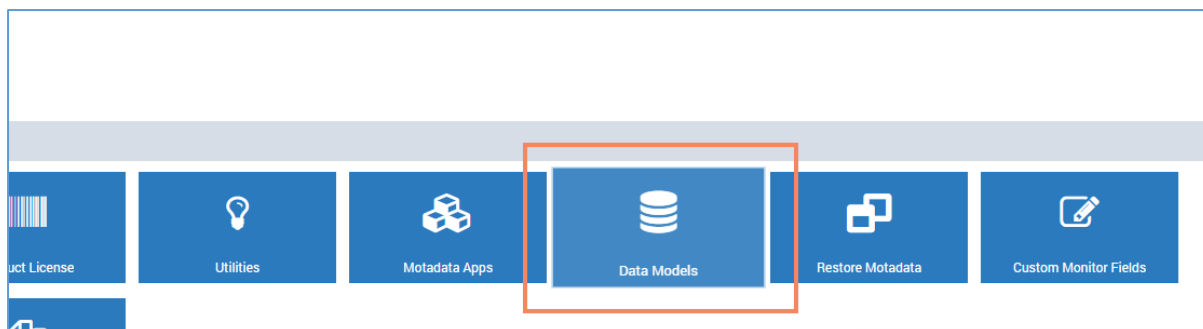
| | |
|------------|-------------------------------|
| Monitor | 172.16.10.1 |
| Type | Firewall |
| System OID | 1.3.6.1.4.1.12356.101.1.1041 |
| System | FG100E_MotaData.mindarray.com |

The 'System OID' field is highlighted with a red box. The 'Availability' section shows a green donut chart indicating 100% availability. The 'Network Latency' section shows a green bar with '0ms' and 'live' status.

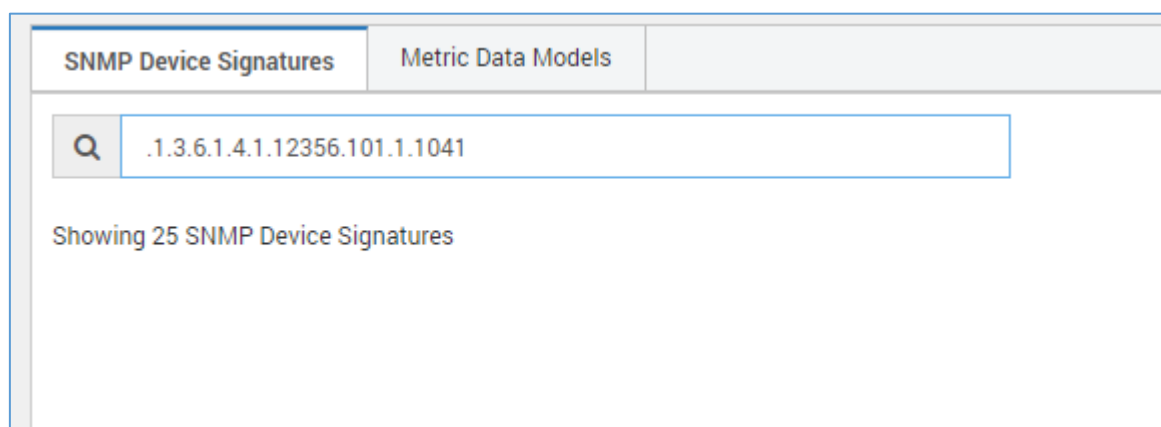
- The device is not already present in the system. In this case, you may directly jump to section 2.1 and 2.2.

Steps:

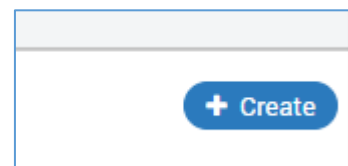
- Go to Amin > Data models.



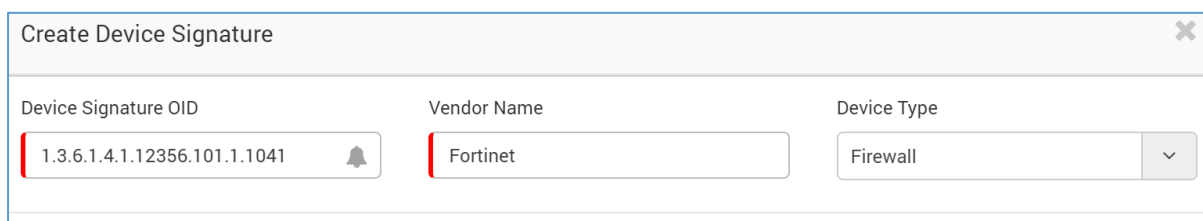
2. On data models page, to be sure, let's first search the system OID is not already present in the system. Search the OID in the search box. In case if it is there, jump to section 2.1 and 2.2.



3. When the device signature is not present, click on 'Create' button to add a new device signature.



4. On 'Create Device Signature' page, add the system OID, vendor name and device type.



5. Add at least one scalar metric to the signature. Before entering the OID, do an SNMP walk in the backend.

SNMP Walk Command: `snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.13.2.1.1.3`

```
root@ubuntu:~# snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.13.2.1.1.3
iso.3.6.1.4.1.12356.101.13.2.1.1.3.1 = Gauge32: 4
```

In our example, we'll add 2 scalar metrics.

- OID: Type the OID of the scalar metric.
- Counter Name: Type the name of the counter to recognize the OID.
- Data Type: Select the data type that OID is going to return.

****Note:** For a scalar metric, you have to add .0 at the end of the OID. Since, our scalar OID is 1.3.6.1.4.1.12356.101.13.2.1.1.3, we'll add counter as: 1.3.6.1.4.1.12356.101.13.2.1.1.3.0

6. Click 'Save'.

2.1 Adding Scalar Model

We learned that scalar metric returns only single value in SNMP walk. Also, we learned that to add scalar OID, we have to add .0 at the end of the OID. You can add scalar metric while creating a new device signature or to an existing signature.

- Edit the device signature to add scalar metrics.

- Click on 'Add 5 more rows' (see above picture).

Before adding a new OID, perform the SNMP walk in the system. From the SNMP walk, you'll understand the data type for 3rd column.

```
root@ubuntu:~# snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.13.2.1.1.3
iso.3.6.1.4.1.12356.101.13.2.1.1.3.1 = Gauge32: 4
```

data type - integer

- Type the OID. At the end of the OID, add a .0 at the end.

.1.3.6.1.4.1.12356.101.1.1041

.1.3.6.1.4.1.12356.101.1.1041 Fortinet Firewall

| OID | Counter Name | Data Type |
|--------------------------------|--------------|-----------|
| .3.6.1.4.1.12356.101.4.1.3.0 | CPU(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric |

+ Add 5 more rows

4. Delete unwanted rows and click Save.

.1.3.6.1.4.1.12356.101.1.1041

.1.3.6.1.4.1.12356.101.1.1041 Fortinet Firewall

| OID | Counter Name | Data Type |
|--------------------------------|--------------|-----------|
| .3.6.1.4.1.12356.101.4.1.3.0 | CPU(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.6.0 | Disk(%) | Numeric |

+ Add 5 more rows

2.2 Adding Tabular Model

Adding a tabular model is very similar to adding a scalar model. The only difference here is that the SNMP walk returns multiple values.

For our example, we used 'user index' counters OID and ran an SNMP walk.

```
snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.5.2.2.1.1
```

```

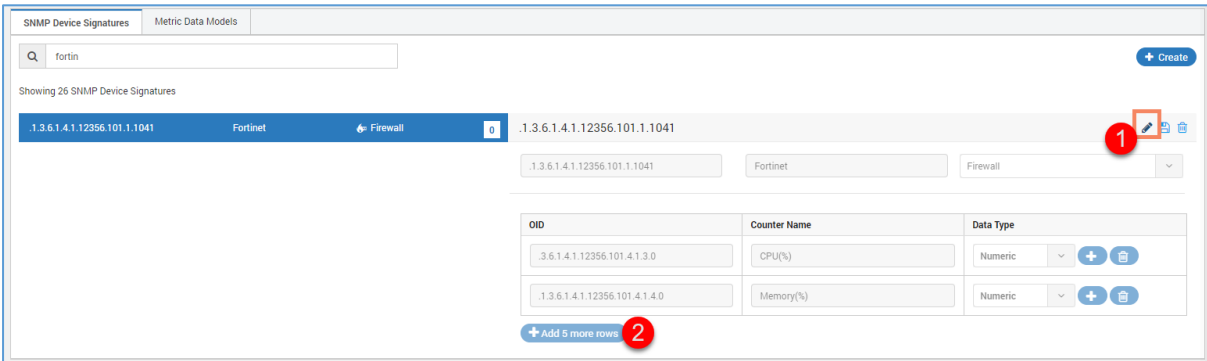
MOTADATA

Last login: Mon Nov 11 16:00:44 2019 from 172.16.8.76
root@ubuntu:~# snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.5.2.2.1.1
1
iso.3.6.1.4.1.12356.101.5.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.4.1.12356.101.5.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.4.1.12356.101.5.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.4.1.12356.101.5.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.4.1.12356.101.5.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.4.1.12356.101.5.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.4.1.12356.101.5.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.4.1.12356.101.5.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.4.1.12356.101.5.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.4.1.12356.101.5.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.4.1.12356.101.5.2.2.1.1.12 = INTEGER: 12
iso.3.6.1.4.1.12356.101.5.2.2.1.1.13 = INTEGER: 13
iso.3.6.1.4.1.12356.101.5.2.2.1.1.14 = INTEGER: 14
iso.3.6.1.4.1.12356.101.5.2.2.1.1.15 = INTEGER: 15
iso.3.6.1.4.1.12356.101.5.2.2.1.1.16 = INTEGER: 16
iso.3.6.1.4.1.12356.101.5.2.2.1.1.17 = INTEGER: 17
iso.3.6.1.4.1.12356.101.5.2.2.1.1.18 = INTEGER: 18
iso.3.6.1.4.1.12356.101.5.2.2.1.1.19 = INTEGER: 19
iso.3.6.1.4.1.12356.101.5.2.2.1.1.20 = INTEGER: 20
iso.3.6.1.4.1.12356.101.5.2.2.1.1.21 = INTEGER: 21
iso.3.6.1.4.1.12356.101.5.2.2.1.1.22 = INTEGER: 22
iso.3.6.1.4.1.12356.101.5.2.2.1.1.23 = INTEGER: 23
iso.3.6.1.4.1.12356.101.5.2.2.1.1.24 = INTEGER: 24
iso.3.6.1.4.1.12356.101.5.2.2.1.1.25 = INTEGER: 25
iso.3.6.1.4.1.12356.101.5.2.2.1.1.26 = INTEGER: 26
iso.3.6.1.4.1.12356.101.5.2.2.1.1.27 = INTEGER: 27
root@ubuntu:~#

```

You can add tabular metric while creating a new device signature or to an existing signature.

1. Edit the device signature to add scalar metrics.



| OID | Counter Name | Data Type |
|-------------------------------|--------------|-----------|
| 1.3.6.1.4.1.12356.101.1.1041 | CPU(%) | Numeric |
| 1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric |

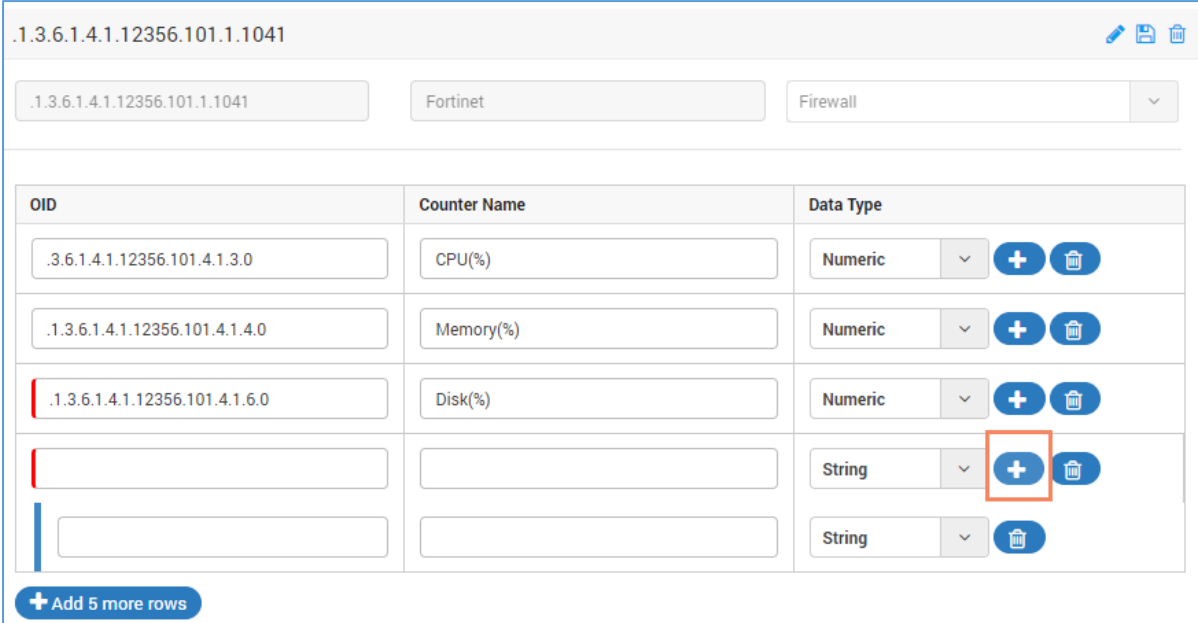
2. Click on 'Add 5 more rows' (see above picture).

Before adding a new OID, perform the SNMP walk in the system. From the SNMP walk, you'll understand the data type for 3rd column.

3. Add the OID, its counter name and the data type. Use the most unique OID from the list.

****Note:** Type OID as it is. Don't add anything at the end.

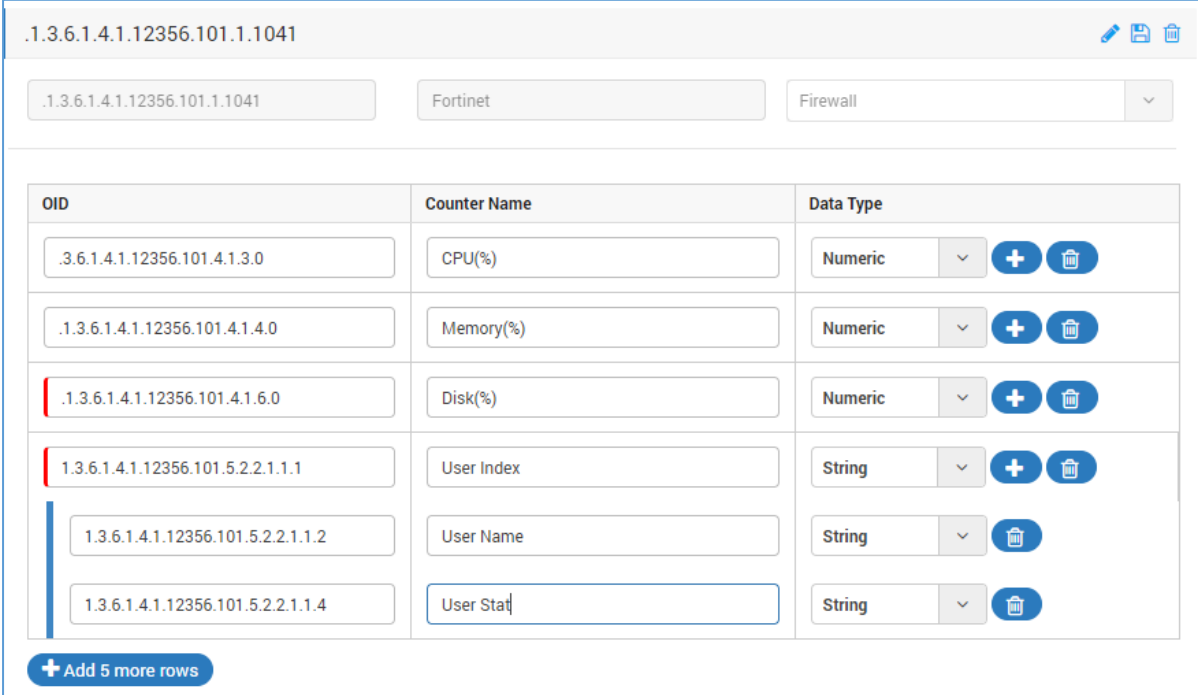
4. Click on the + sign.



| OID | Counter Name | Data Type |
|--------------------------------|--------------|-----------|
| .3.6.1.4.1.12356.101.4.1.3.0 | CPU(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.6.0 | Disk(%) | Numeric |
| | | String |
| | | String |

+ Add 5 more rows

5. Add other OIDs below the unique ID. The boxes are identified using blue color.



| OID | Counter Name | Data Type |
|-----------------------------------|--------------|-----------|
| .3.6.1.4.1.12356.101.4.1.3.0 | CPU(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric |
| .1.3.6.1.4.1.12356.101.4.1.6.0 | Disk(%) | Numeric |
| 1.3.6.1.4.1.12356.101.5.2.2.1.1.1 | User Index | String |
| 1.3.6.1.4.1.12356.101.5.2.2.1.1.2 | User Name | String |
| 1.3.6.1.4.1.12356.101.5.2.2.1.1.4 | User Stat | String |

+ Add 5 more rows

6. Delete unwanted rows and click Save.

.1.3.6.1.4.1.12356.101.1.1041

.1.3.6.1.4.1.12356.101.1.1041

Fortinet

Firewall

| OID | Counter Name | Data Type |
|--------------------------------|--------------|-----------|
| .3.6.1.4.1.12356.101.4.1.3.0 | CPU(%) | Numeric + |
| .1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric + |
| .1.3.6.1.4.1.12356.101.4.1.6.0 | Disk(%) | Numeric + |

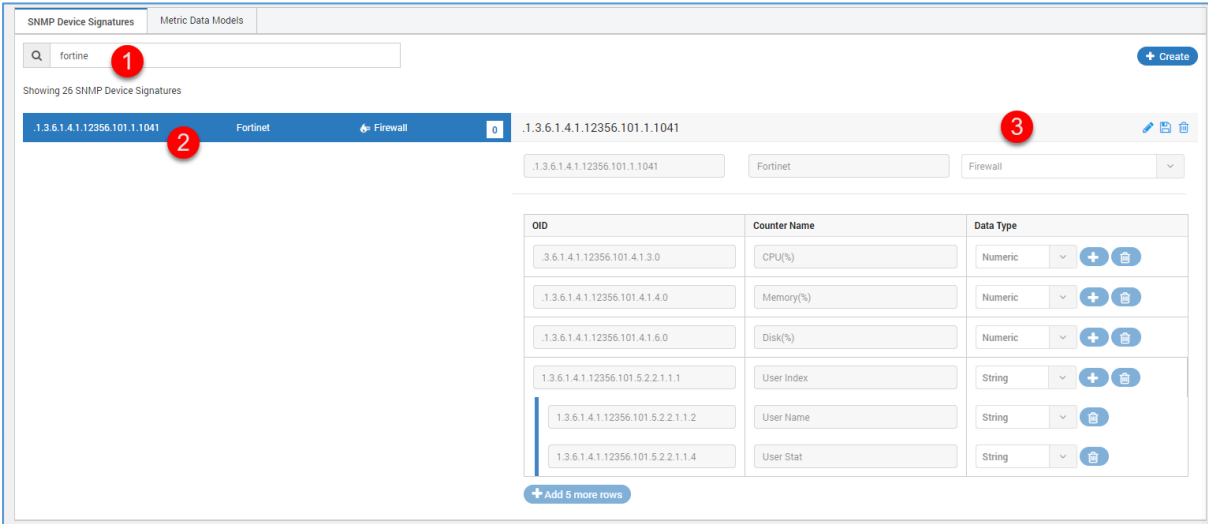
+ Add 5 more rows

3 Verifying Device Signature

3.1 Signature OID

You can look for the signature OID by searching the signature in data models section. In the search you can type the OID, vendor name or device type to find the signature.

1. Go to Admin > Data models.
2. In SNMP Device Signatures tab, search your signature.
3. Click on the signature. On right hand side you'll see the data model you created.

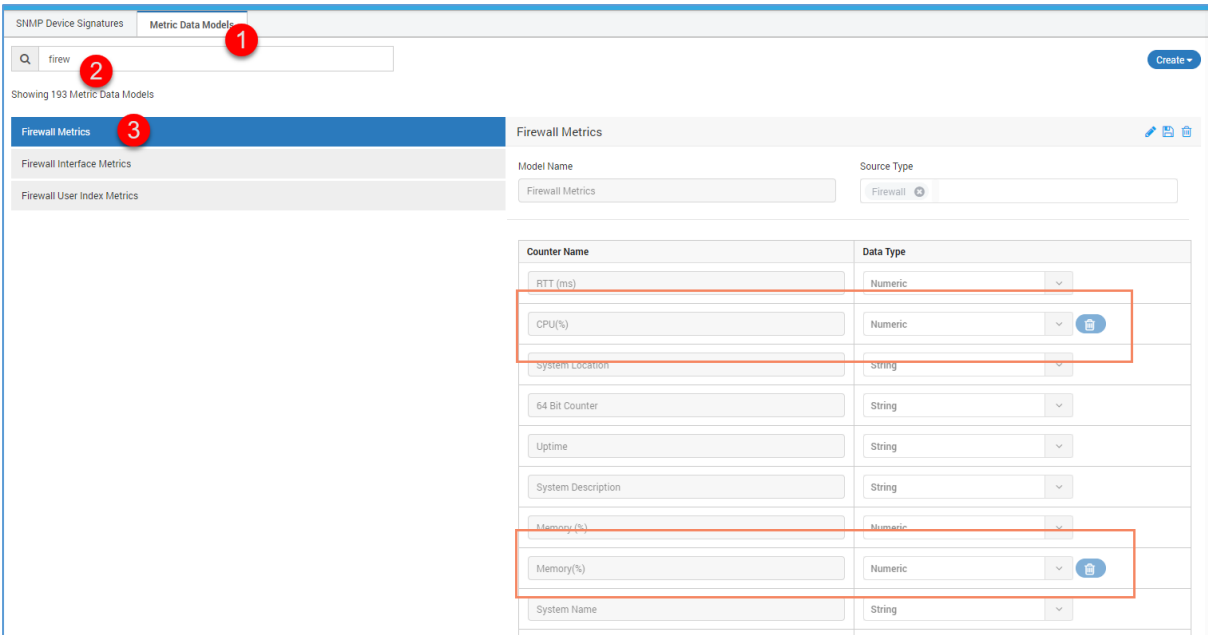


| OID | Counter Name | Data Type |
|-----------------------------------|--------------|-----------|
| 1.3.6.1.4.1.12356.101.4.1.3.0 | CPU(%) | Numeric |
| 1.3.6.1.4.1.12356.101.4.1.4.0 | Memory(%) | Numeric |
| 1.3.6.1.4.1.12356.101.4.1.6.0 | Disk(%) | Numeric |
| 1.3.6.1.4.1.12356.101.5.2.2.1.1.1 | User Index | String |
| 1.3.6.1.4.1.12356.101.5.2.2.1.1.2 | User Name | String |
| 1.3.6.1.4.1.12356.101.5.2.2.1.1.4 | User Stat | String |

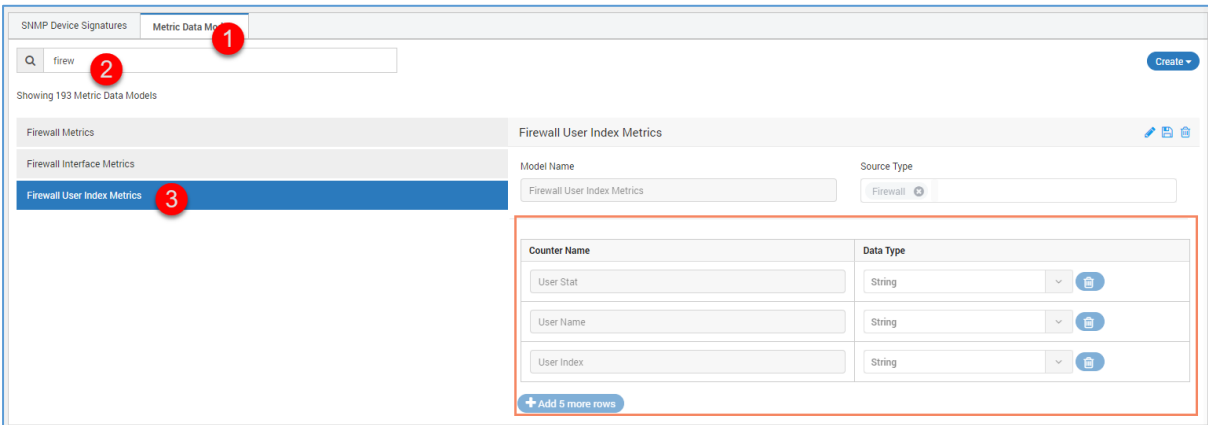
3.2 Data Models

The metric data models are automatically placed by the system under appropriate section. In our example of Fortinet firewall, the metrics were placed in two places under 'Metric Data Models' tab.

1. Firewall Metrics - This is a default category. The singular metrics are added in this category.



- For tabular metric, system created a new category. This category is named on the unique counter name we used. We used the name 'User Index'. Hence, system created the data model with the name: Firewall User Index Metric. Here, 'firewall' is the device type. 'User Metric' is the counter name and 'Metric' is the suffix.



****Note:** You can make changes in counter metrics from this page also.

3.3 Monitor Widgets

To look at the counter metrics in monitor widget, the device should be provisioned in Motadata. In the system, create a widget with counter metrics that you created.

In below figure, we are creating a widget for firewall instance counter (tabular) metric. In the source type, you'll see the data model name 'Firewall User Index Metrics'. This name is visible because we created a device signature. The metric counters are visible in Analytics section. The output shows the data similar to the SNMP walk.

Widget Name
Firewall Instance Metrics

Time Range Live

1 Filter

Source Type: Firewall User Index Metrics

Filter: Monitor

Filter Values:

Instance Filter:

2 Rule

3 Analytics

User Name: Value

User Stat: Value

4 Visualization

Group: Monitor

View Type: List

Update Preview

Creating a metric for device signature

| Monitor | Type | User Index | User Name | User Stat |
|-------------|----------|------------|------------------|-----------|
| 172.16.10.1 | Firewall | 1 | Nihal.singh | 2 |
| 172.16.10.1 | Firewall | 10 | guest | 2 |
| 172.16.10.1 | Firewall | 11 | jubin.thomas | 2 |
| 172.16.10.1 | Firewall | 12 | keyur.ranpura | 2 |
| 172.16.10.1 | Firewall | 13 | nilesh | 2 |
| 172.16.10.1 | Firewall | 14 | nisha.gohel | 2 |
| 172.16.10.1 | Firewall | 15 | paramdeep.ramola | 2 |
| 172.16.10.1 | Firewall | 16 | qa | 1 |
| 172.16.10.1 | Firewall | 17 | qa1 | 1 |

Output of the metric

```
root@ubuntu:~# snmpwalk -v2c -c mind 172.16.10.1 1.3.6.1.4.1.12356.101.5.2.2.1.1.2
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.1 = STRING: "Nihal.singh"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.2 = STRING: "Rahul.singh"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.3 = STRING: "account1"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.4 = STRING: "account2"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.5 = STRING: "ankit.sanghavi"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.6 = STRING: "chetan.vadukar"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.7 = STRING: "devendra.khatana"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.8 = STRING: "gaurav.patel"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.9 = STRING: "gk"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.10 = STRING: "guest"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.11 = STRING: "jubin.thomas"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.12 = STRING: "keyur.ranpura"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.13 = STRING: "nilesh"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.14 = STRING: "nisha.gohel"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.15 = STRING: "paramdeep .ramola"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.16 = STRING: "qa"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.17 = STRING: "qa1"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.18 = STRING: "raj.pandya"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.19 = STRING: "red"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.20 = STRING: "shrivan.hirakani"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.21 = STRING: "sunil.kumar"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.22 = STRING: "uday.hathaliya"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.23 = STRING: "vaarisha.gogdani"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.24 = STRING: "vishal.vankar"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.25 = STRING: "vpn"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.26 = STRING: "vpn.guest"
iso.3.6.1.4.1.12356.101.5.2.2.1.1.2.27 = STRING: "vpn.test"
root@ubuntu:~#
```

SNMP walk output for comparison

4 Create New Data Model

You'll need to create a new data model only in following cases:

- You have SSH, PowerShell or Database Server provisioned in Motadata and you want to see the output of a specific query.
- You want to create a composite data model to compare the different metrics of the system.

4.1 Custom Data Model

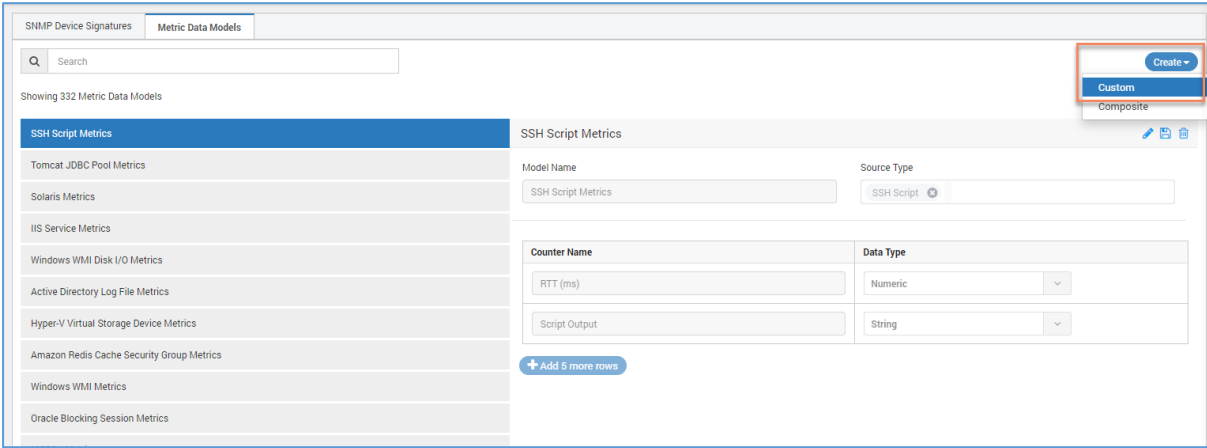
**** Requires knowledge of writing a query.**

To execute your custom query in PowerShell, Database server or SSH server; you can create a custom data model. While creating a custom data model, you need to make sure following things:

- Your query is producing desired results when we run it in backend.
- While adding the Database/PowerShell/SSH server in Motadata, you have added the script.

To create a custom data model:

1. Go to Admin > Data Models.
2. In the 'Metric Data Models' tab (second tab), click on Create >> Custom.



The screenshot shows the 'Metric Data Models' tab in the Motadata Admin interface. On the left, there is a search bar and a list of existing models including 'SSH Script Metrics', 'Tomcat JDBC Pool Metrics', 'Solaris Metrics', 'IIS Service Metrics', 'Windows WMI Disk I/O Metrics', 'Active Directory Log File Metrics', 'Hyper-V Virtual Storage Device Metrics', 'Amazon Redis Cache Security Group Metrics', 'Windows WMI Metrics', and 'Oracle Blocking Session Metrics'. On the right, the 'SSH Script Metrics' model is selected, showing its configuration. The 'Model Name' is 'SSH Script Metrics' and the 'Source Type' is 'SSH Script'. Below this, there are two rows for 'Counter Name' and 'Data Type'. The first row has 'RTT (ms)' as the counter name and 'Numeric' as the data type. The second row has 'Script Output' as the counter name and 'String' as the data type. A '+ Add 5 more rows' button is visible below the table. In the top right corner, a 'Create' button is highlighted with a red box, and a dropdown menu is open showing 'Custom' as the selected option.

3. A page 'Create Metric Data Model' will open up. In this page, provide the name of the counters that will appear as the output of the query. For safe side, first run the query in the backend and then copy the output column names over here.

| Counter Name | Data Type |
|----------------------|---------------------------------------|
| <input type="text"/> | String + 🗑️ |
| <input type="text"/> | String + 🗑️ |
| <input type="text"/> | String + 🗑️ |
| <input type="text"/> | String + 🗑️ |
| <input type="text"/> | String + 🗑️ |

+ Add 5 more rows

Fields Description

Model Name: Type the name of the model.

Source Type: Select the type of the query for which you are building the model. You can choose from 3 types of queries: Database query, PowerShell and SSH.

Counter Name: Type the name of the output counters (from the query) and their datatypes.

For this to work, you have to follow following rules:

- You **must** enter the counter name exactly as the output of your query. System compares the counter name here and in the query for mapping.
- The counters are categorized as parent and child counters. By default, you only see parent counters.
- In the parent counter, enter only those column names that contains unique value.
- Add child counters by clicking on '+' button in parent counter row. Input rest of the columns over here.

Create Metric Data Model

Model Name

Database Query Builder

Source Type

Database Query

| Counter Name | Data Type |
|---------------------------------------|---------------------------------------|
| Parent Counter (accepts unique value) | String + 🗑️ |
| Child Counters (accepts other values) | String 🗑️ |
| | String 🗑️ |
| | String 🗑️ |

+ Add 5 more rows

Save Reset

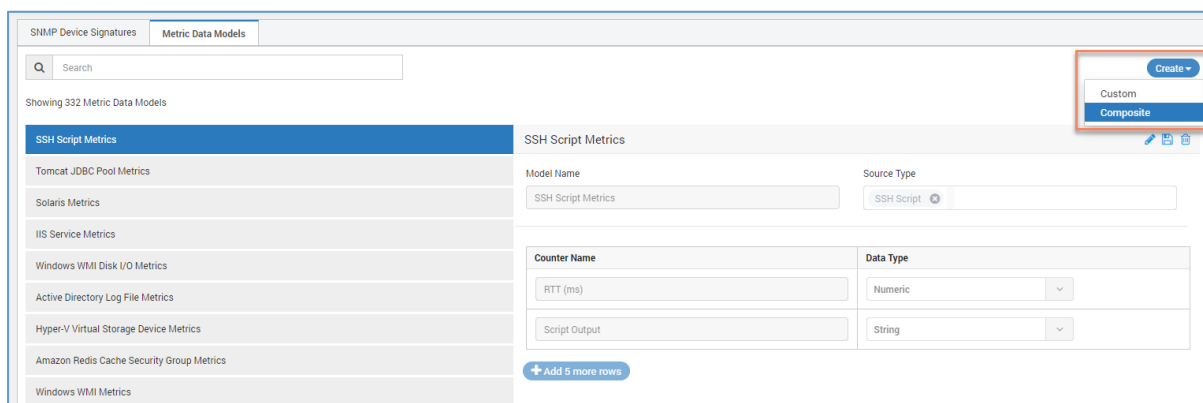
- If you have used multiple queries and there are multiple outputs, then only use second parent counter.

4.2 Composite Data Model

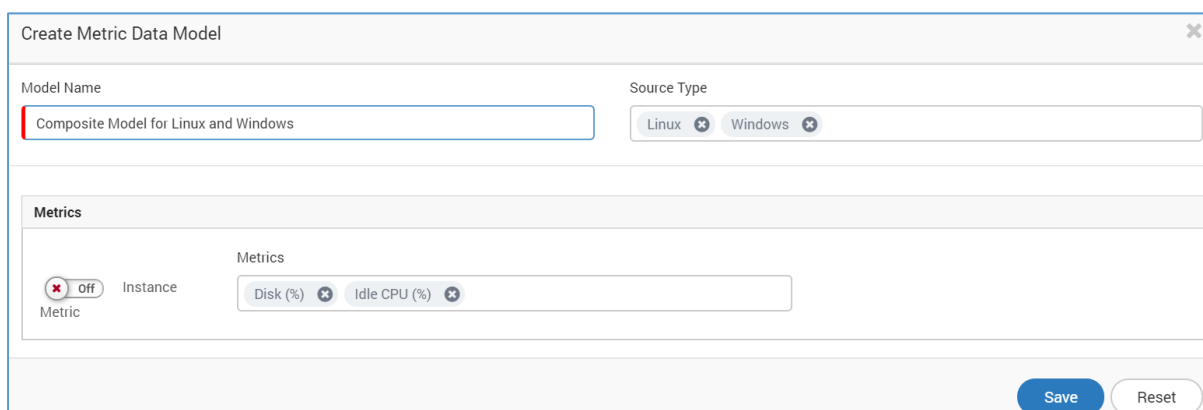
In widgets you can select only one source type. What if you want to a composite widget with different source types? For such scenario, we can create a composite data model. In composite data models, you can see information from multiple source types. It is used when you want to see say: CPU consumption of Linux and Windows servers.

To create a composite data model:

1. Go to Admin > Data Models.
2. In the 'Metric Data Models' tab (second tab), click on Create >> Composite.



3. A page 'Create Metric Data Model' will open. On this page, you can select the multiple source types and the counters that you want to see in the widget.



Fields Description

Model Name: Type the name of the model.

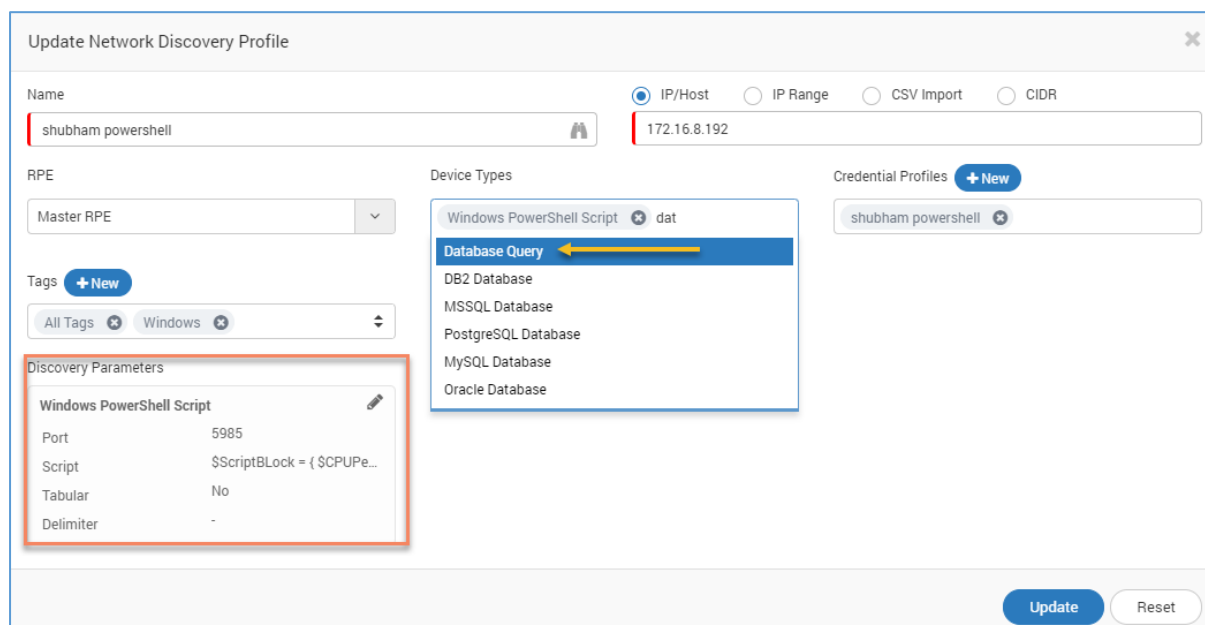
Source Type: Select the sources from the list for which you want to create a model.

Instance (On/Off): Select instance as 'Off' to create scalar metrics. Select instance as 'On' to create tabular metric.

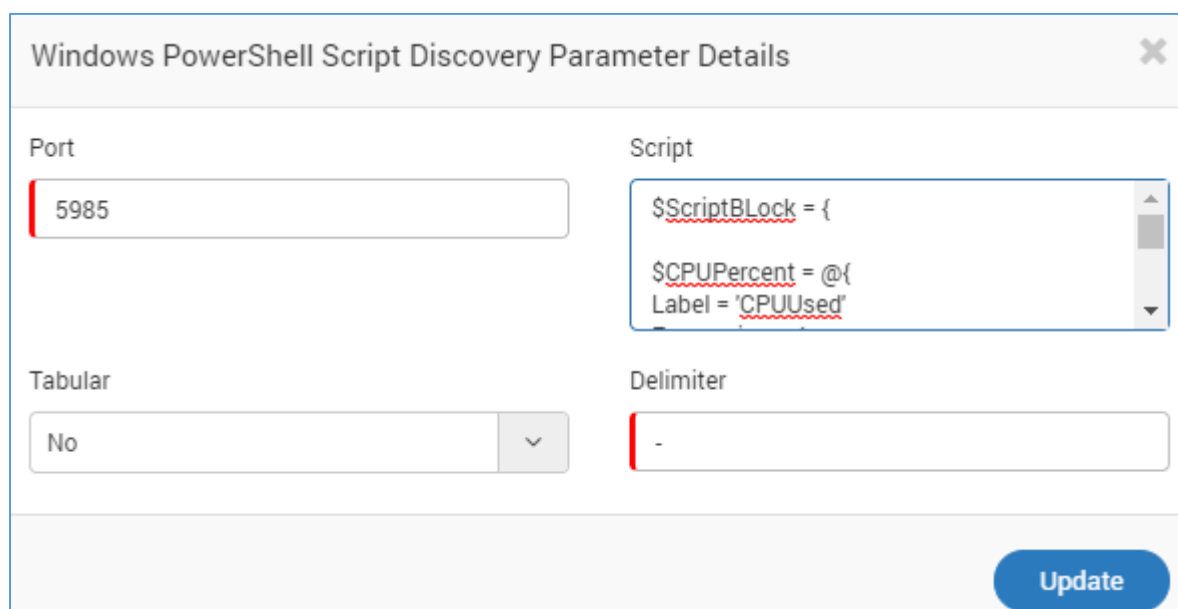
Metrics: Select the metrics from the list. The list shows all the metrics from all the source types (whether they are common or not).

5 Additional Information - Add Query in Monitors

In the network profile, suppose the device type is: PowerShell, SSH or Database Query. To add the query, edit the discovery parameters.



Edit the discovery parameters and type the script. You can add multiple scripts by separating them with comma. **(Again! Make sure that everything you paste in 'Script' produces desired result in backend).**



Keep in touch

www.motadata.com, sales@motadata.com,

India: +91 79-2680-0900, USA: +1 408-418-5229

About Motadata

Mindarray Systems Pvt. Ltd. a global IT product company, offers state of the art affordable yet powerful product suite - Motadata consisting of Network Management & Monitoring, Log & Flow Management, and IT Service Management Platforms. The platform empowers both IT administrators and CXOs to analyze, track & resolve IT operational issues by effectively monitoring various systems and devices from multiple vendors through a unified and centralized dashboard.

Motadata is industry's first IT ops solution that truly correlates the metric, flow and log events and turns them into actionable insights. Our global customers from Telecom, Government and Enterprise domain, rely on Motadata for proactively monitor their network infrastructure.

For more information, visit www.motadata.com.

© 2018 Mindarray Systems Pvt. Ltd. All rights reserved.

All trademarks, service marks, trade names, tradedress, product names and logos appearing on the document are the property of the irrespective owners. Any rights not expressly granted here in are reserved.

