Templates and Components

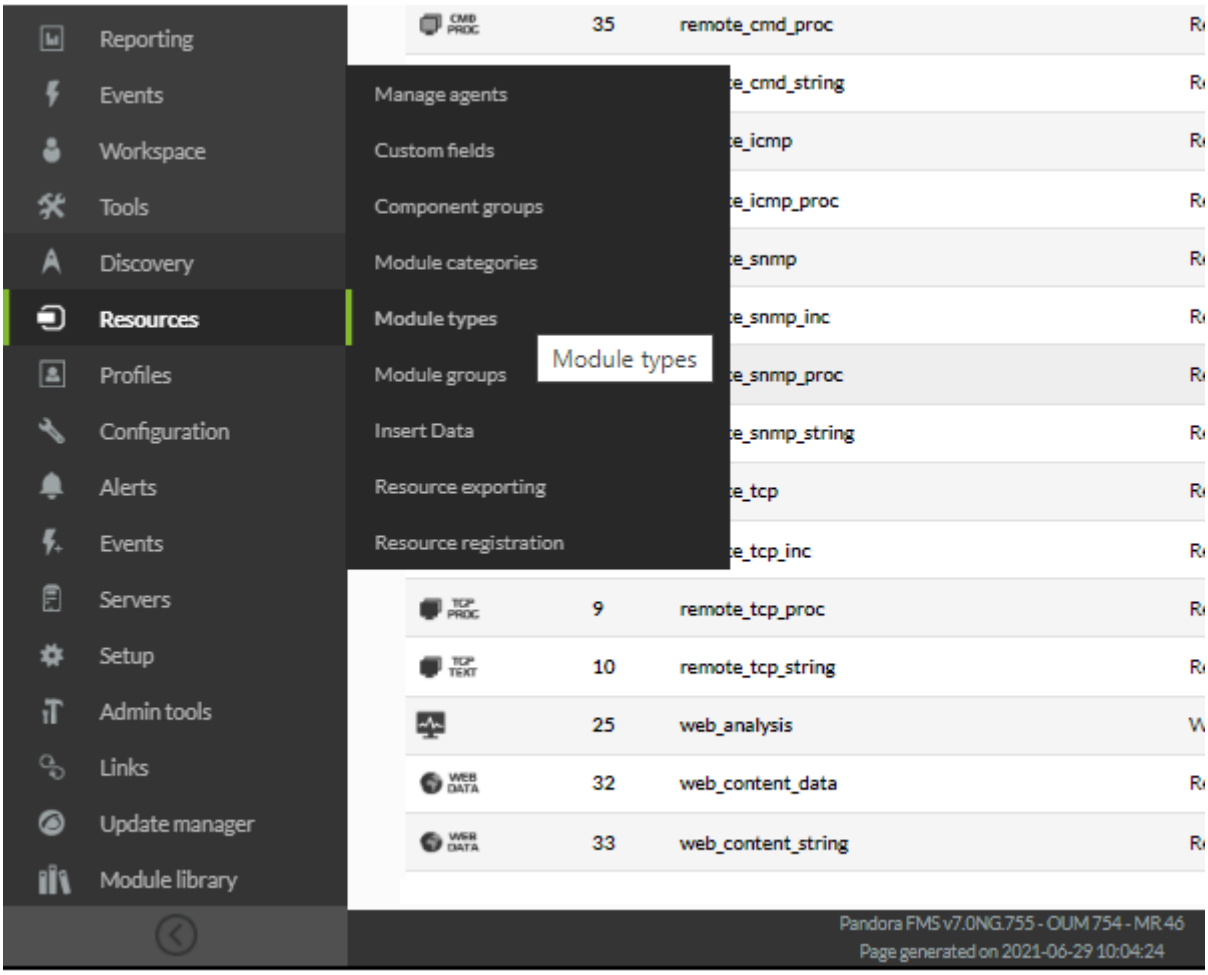
Templates and Components

We are working on the translation of the Pandora FMS documentation. Sorry for any inconvenience.

Templates and Plug Ins

Introduction

Pandora FMS performs all checks through modules allowing you to process different data types depending on the element to be monitored. The full default module list for Pandora FMS can be checked by clicking on Resources > Module Types.



There are different module types in Pandora FMS:

- async: Asynchrounous data.
- generic: Generic data.
- keep_alive: Special keepalive module, useful to control the status of the last contact with an agent.
- icmp: ICMP check (ping).

- snmp: SNMP check.
- tcp: TCP check.
- web: Network check.

These module types can stock different types of data:

- data: Numerical data.
- proc: Boolean values. ! means true and 0 means false. For example, for web modules it means that if the value exists, it returns 1 and if it does not exist, it returns 0.
- string: Text string.
- inc: Incremental data, e.g. the amount of packets sent by an interface will always grow. They show growth by time unit.
- inc_abs: Absolute incremental data, showing the value increase since the last reading.

Components

What is a component?

A component is a “generic module” which can be repeatedly applied onto an agent, as if it were a module's “master copy”, generating a module associated with an agent. That way having a database of your organization's most used components turns out to come in handy when it comes to monitoring, since you have your own components adapted to the technologies you usually use and you just have to apply these components to the new agents.

There are two types of components. Network components, which group all remote modules (wmi, tcp, snmp, icmp, plugin, web, etc.) and local components, which include the software agent module's definition as “pieces”, ready to be incorporated to the configuration file of the agent (with remote configuration enabled, Enterprise version), or they can be cut out and pasted into the agent's configuration manually (without remote configuration, Community version).

Component template

What is a component template?

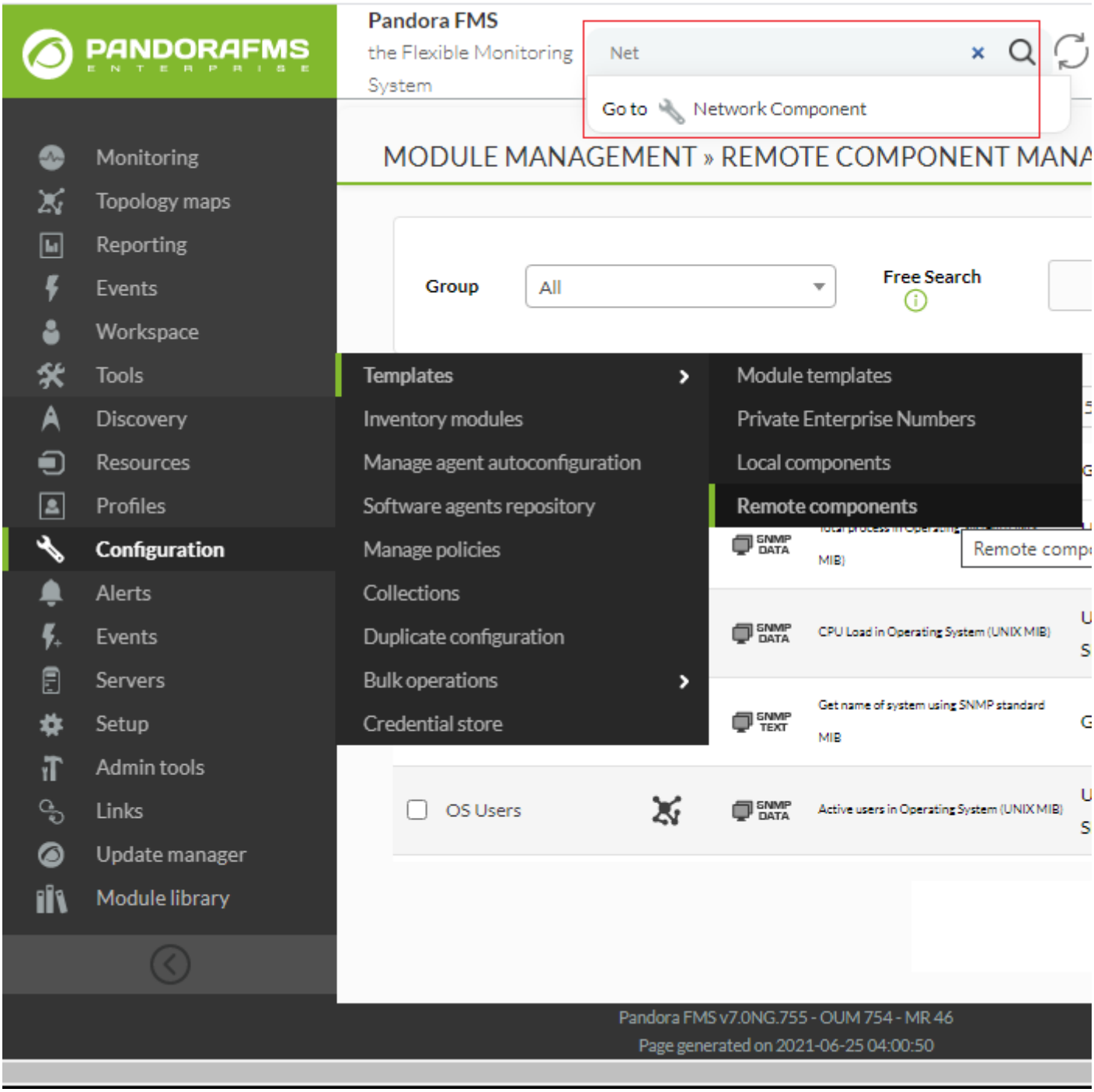
Pandora FMS offers the possibility of grouping network components in “templates” so that you may apply multiple network components directly on an agent. This makes it easy to deploy monitoring, as you create several modules simultaneously through the network components associated with a template.

The Recon server applies the network components associated to a template to detected hosts, adding the specified modules automatically and allowing a fast and automatic monitoring deployment.

Network Components

Network components are elements that enable remote network checks. Pandora FMS has about 40 preconfigured network checks, while the Enterprise Version comes with more than 400.

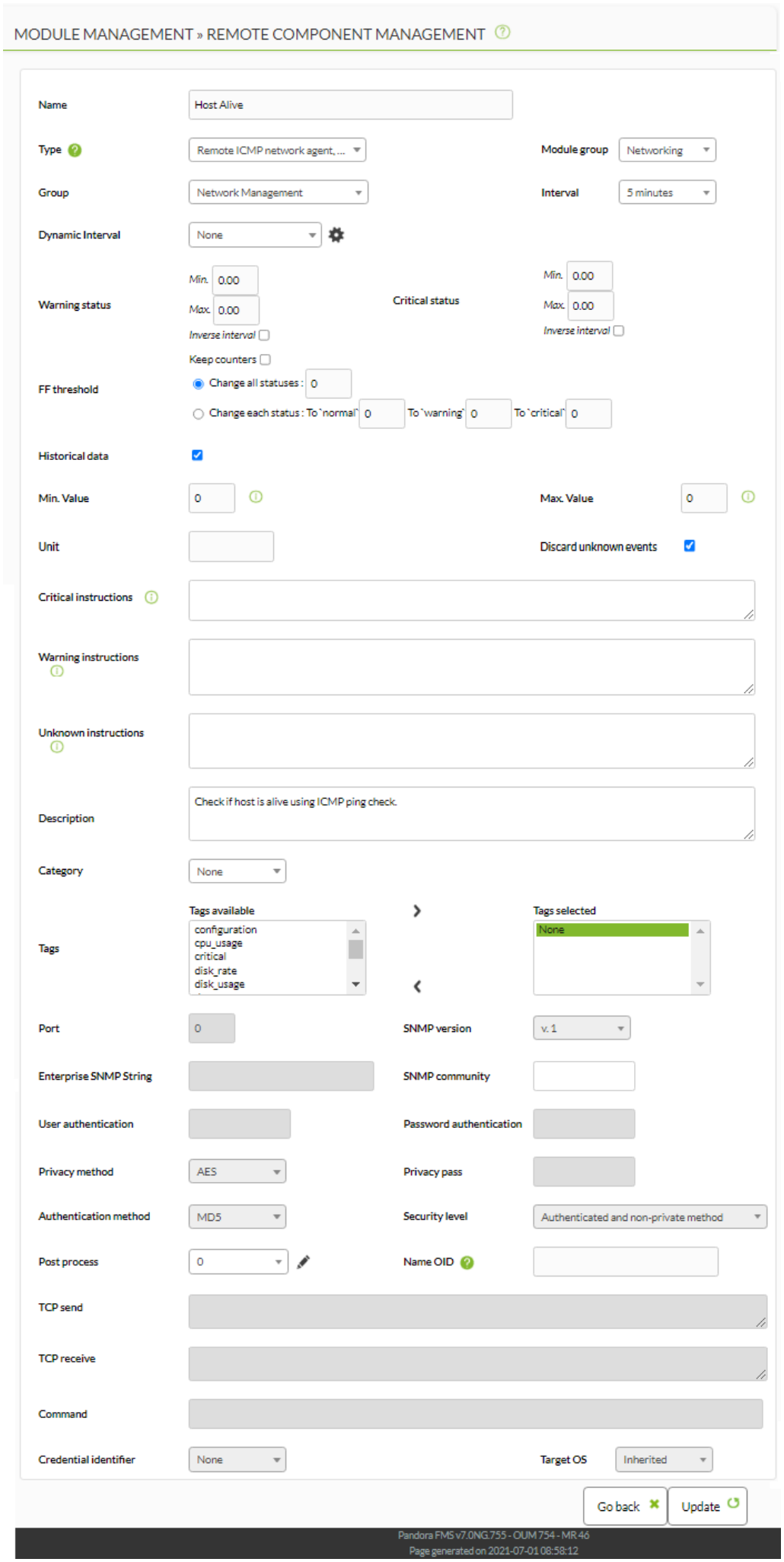
Pandora FMS network components can be created and viewed on their management page by clicking on Configuration → Templates → Remote Components.



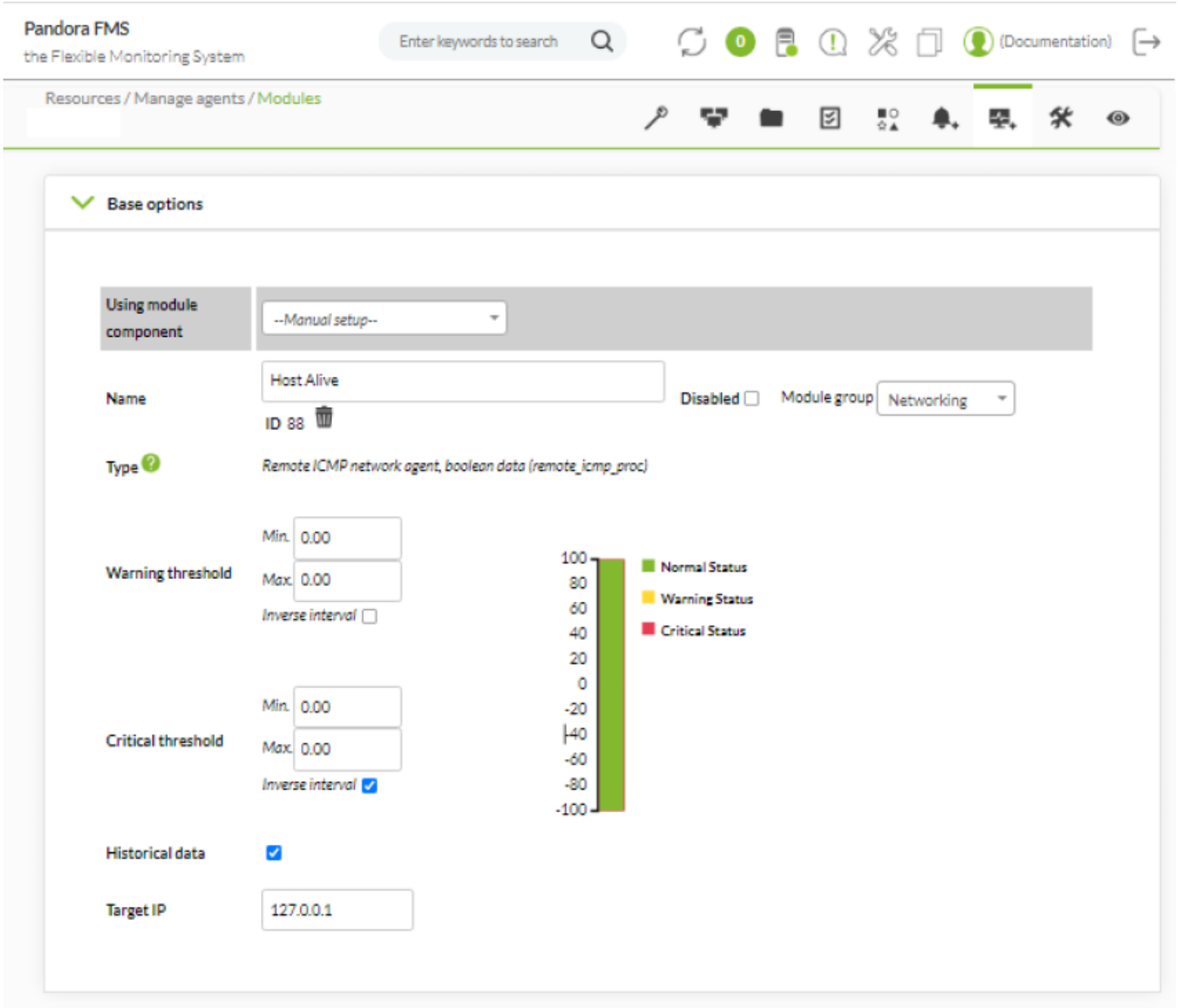
There you may look for the already existing components (by filtering 'by groups' or by 'free-text search'), see their configurations detail, modify them and create new ones.

To see the properties of any module, click on its name. It will take you to the editing page, for

instance all the Host Alive network component details:

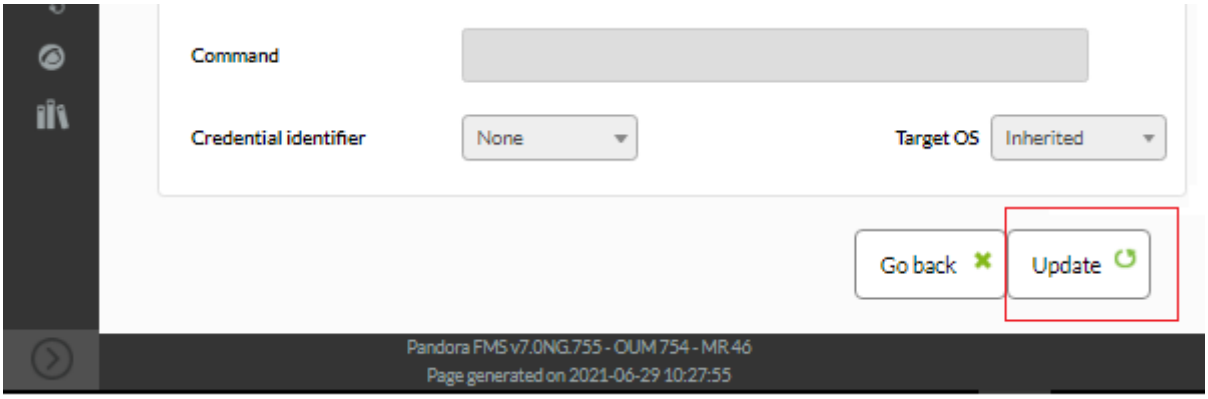

MODULE MANAGEMENT » REMOTE COMPONENT MANAGEMENT
Name
Host Alive
Type
Remote ICMP network agent, ...
Module group
Networking
Group
Network Management
Interval
5 minutes
Dynamic Interval
None
Warning status
Min. 0.00
Max. 0.00
Inverse interval
Critical status
Min. 0.00
Max. 0.00
Inverse interval
Keep counters
FF threshold
Change all statuses : 0
Change each status : To `normal` 0 To `warning` 0 To `critical` 0
Historical data
Min. Value
0
Max. Value
0
Unit
Discard unknown events
Critical instructions
Warning instructions
Unknown instructions
Description
Check if host is alive using ICMP ping check.
Category
None
Tags
Tags available
configuration
cpu_usage
critical
disk_rate
disk_usage
Tags selected
None
Port
0
SNMP version
v. 1
Enterprise SNMP String
SNMP community
User authentication
Password authentication
Privacy method
AES
Privacy pass
Authentication method
MD5
Security level
Authenticated and non-private method
Post process
0
Name OID
TCP send
TCP receive
Command
Credential identifier
None
Target OS
Inherited
Go back
Update
Pandora FMS v7.0NG.755 - OUM 754 - MR 46
Page generated on 2021-07-01 08:58:12

When applied to a module, this one will obtain the network component details, except for the IP address field, where the main agent IP address will automatically be adjusted. All parameters can be edited afterwards (for example: changing WMI module user/password).



If any template is modified, its new values will only be applied to the modules created from that moment on, not to the the ones already created.

To modify component values, click on the name of one them and modify the desired values in the editing page, e.g. the interval. Once updated, click Update at the bottom of the page to save the changes.



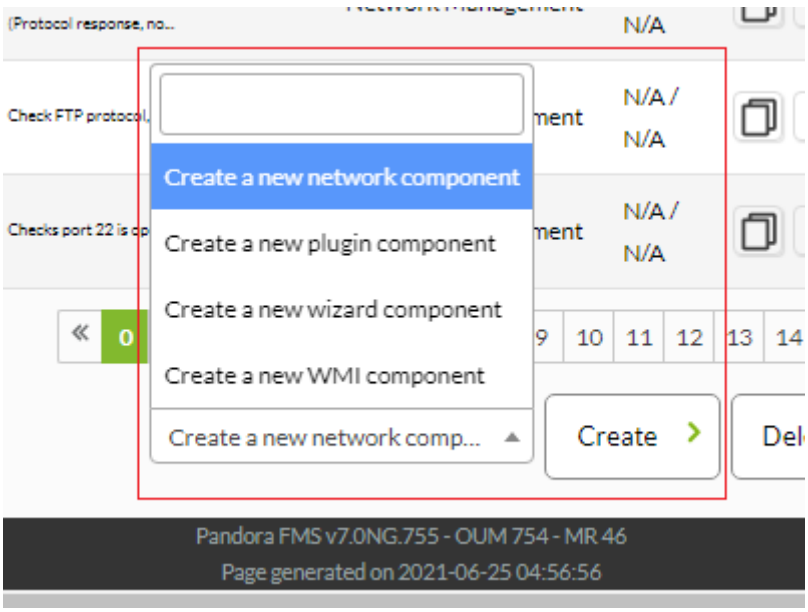
From that moment on, the new component configuration will be applied to the agents where said module is added.

Creating new Network Components

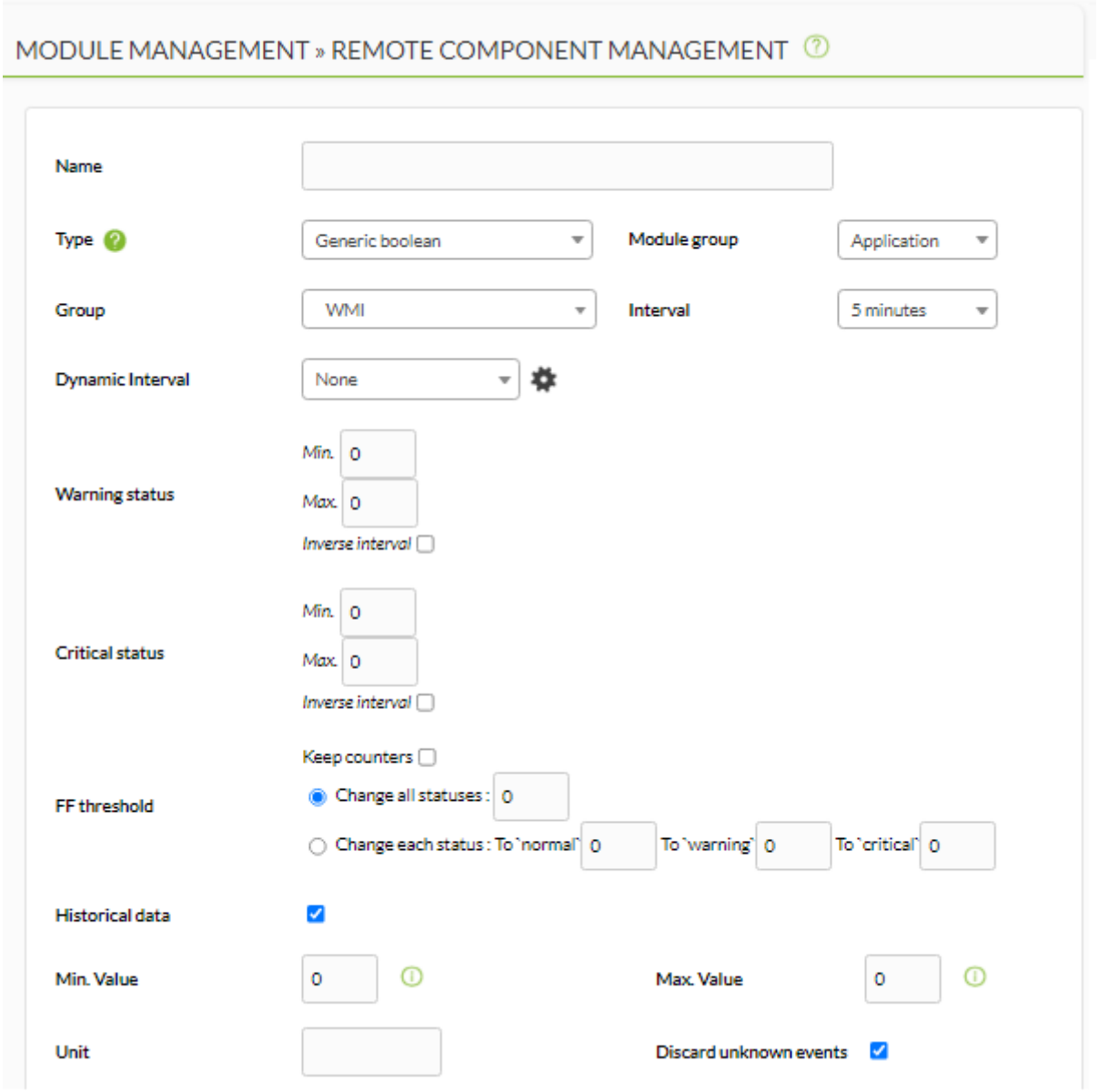
You may create four types of network components:

- Network components.
- Plug-in components (server addons).
- Wizard component.
- WMI components.

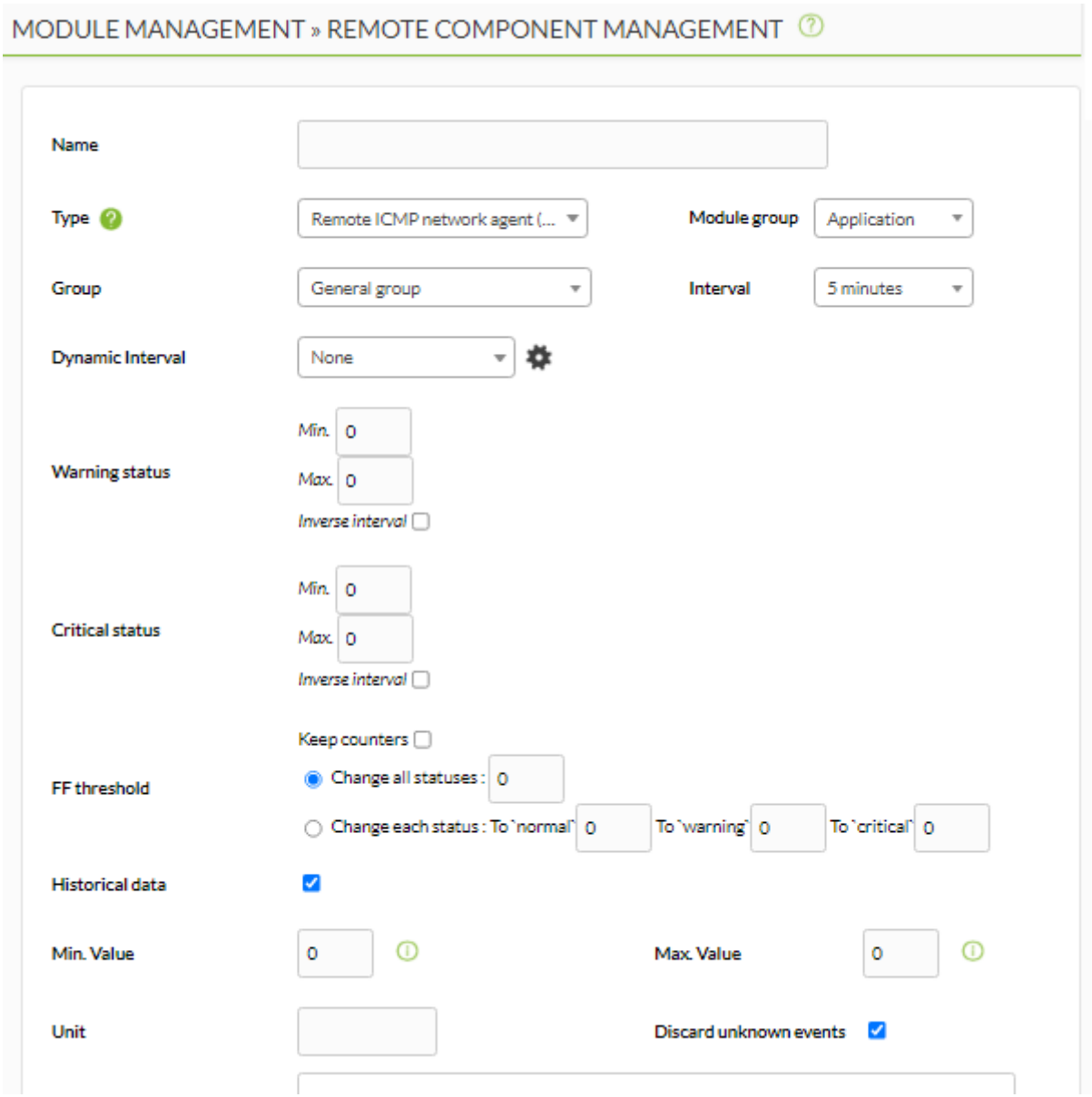
To create a new network component, go to Configuration →Templates → Remote components. Go to the bottom of the page, select a network component within the drop-down menu (Network, Plugin, Wizard or WMI) and click on Create.



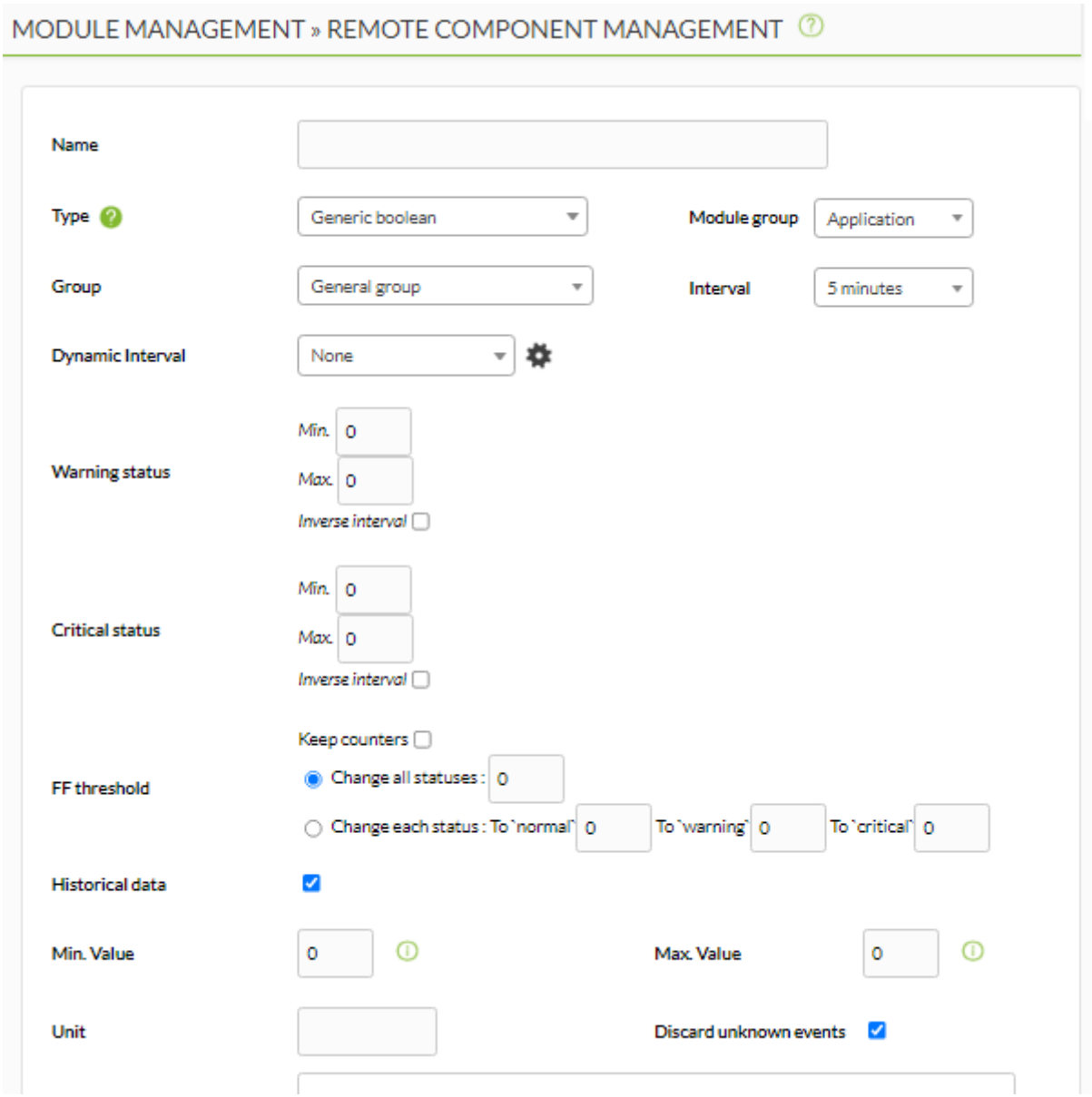
Later, configure all component fields and click on Create. This is the WMI component creation screen.



As you fill in the required fields, keep in mind that you are filling out the description of a “generic” module which will be applied to different agents. Some parameters such as snmp community, user or password may be different according to the agents that may apply the module later on, so you may leave them empty. If you have a common user policy for your systems, you may leave the modules completely configured by entering users, passwords and other data common to all agents here.



The same process applies to plugin components.



In this case, similarly to creating a plugin module, when selecting a plugin in the interface, the fields defined in the plugin macros will appear.

Local Components

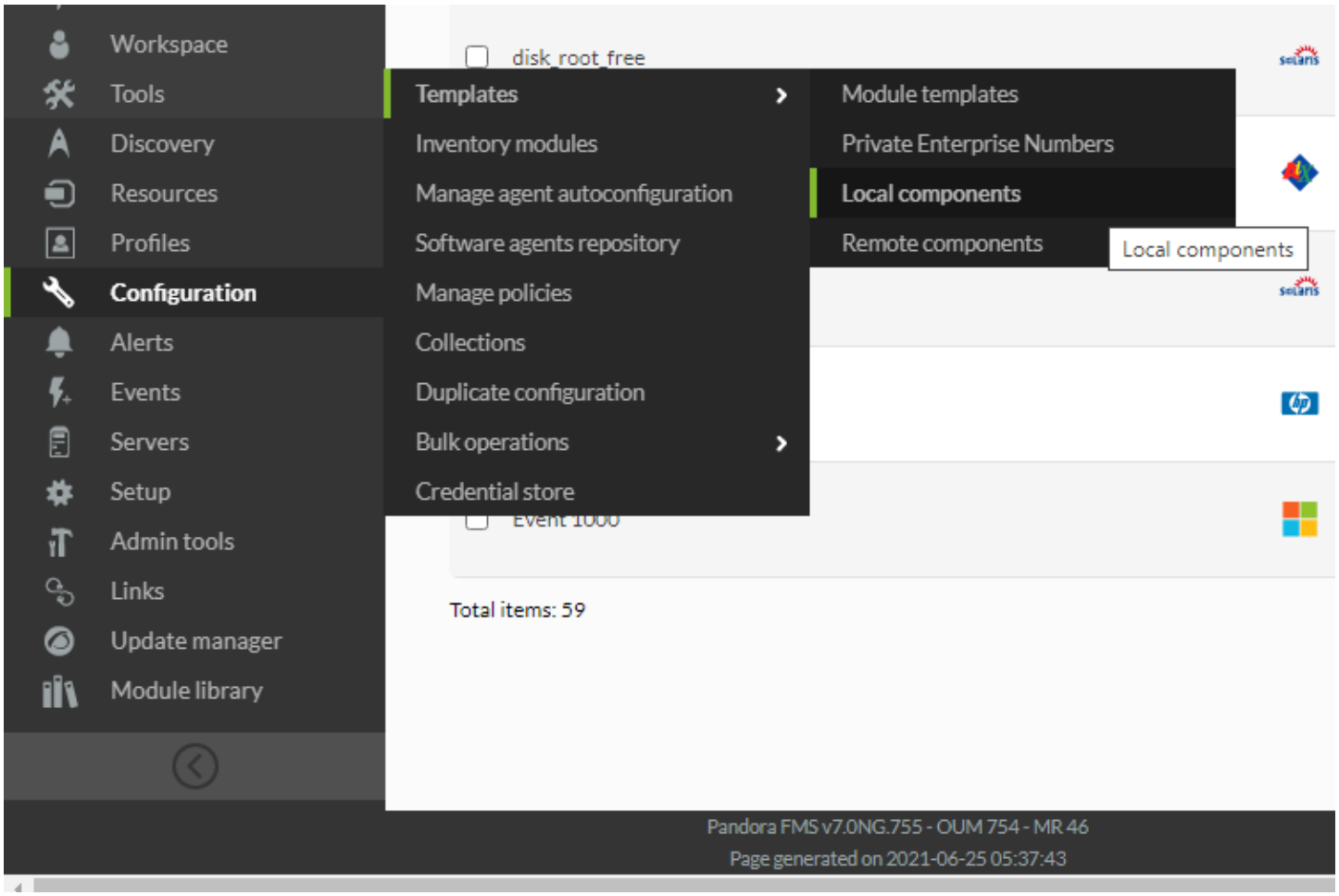
Local components are the ones that can be applied to software agents. If you have Pandora FMS Enterprise Version, these components can be applied automatically on agents through policies or manually (one by one) within the agent remote configuration editor.

Check the policy section in order to know how to remotely apply a local component to a software agent in Pandora FMS Enterprise.

Local components may also be used in Pandora FMS OpenSource version. However, they will not be applied automatically through Pandora FMS. You must access the agent directly and enter the

changes in the configuration file manually. Pandora FMS Enterprise version has dozens of local modules to apply to the policies and to the agents automatically, sorted out by categories.

Local components wor similarly to network components, once you go to their management page, Configuration > Local Components:



This screen displays the already existing local modules, which can be filtered by different parameters (group, operating system, free text query). You may also also see, modify and create new components here.

To see any module's properties, click on its name. The link will lead you to its detail page, which is shown below.

MODULE MANAGEMENT » LOCAL COMPONENT MANAGEMENT

Name: Busy XML Threads

OS: Windows

Discard unknown events:

Type: Generic numeric

Module group: Application

Group: Citrix

Interval: 5 minutes

Dynamic Interval: None

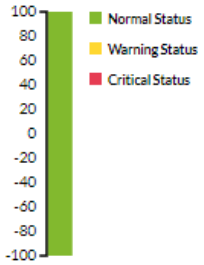
Warning status: Min. 0.00 Max. 0.00 Inverse interval

Critical status: Min. 0.00 Max. 0.00 Inverse interval

Keep counters

FF threshold: Change all statuses : 0

Change each status : To `normal` 0 To `warning` 0 To `critical` 0



Historical data:

FlipFlop timeout: Disabled

Min. Value: 0

Max. Value: 0

Unit:

Post process: 0.0000000000

Description: Total threads in the manager who manage the sessions. If there are more than XX, service fails. NOTE: Run this module only in managers

Configuration:

```
module_begin
module_name Busy XML Threads
module_description Total threads in the manager who manage the sessions. If there are more than XX, service fails. NOTE: Run this module only in managers
module_type generic_data
module_exec Typeperf '\\MAQUINA\Citrix Metaframe Presentation Server\Number of busy XML threads' -sc 1
module_end
```

Load basic

Check

Critical instructions:

Warning instructions:

Category: None

Tags

Tags available: configuration, cpu_usage, critical, disk_rate, disk_usage

Tags selected: None

Macros

Description (_field1_):

Default value (_field1_):

Help (_field1_):

Add macro

Update

As you can see, local component configuration is quite simple. The configuration's elements are described below.

- Name: Component name. This name will be visible when selecting the component to create an agent's module.
- OS: Operating system the component is intended for.
- Group: The group the module belongs to. It is quite useful to filter and assort by monitoring technologies.
- Description: Module description. A default description, which can be modified, is already in there.
- Configuration: The component's configuration like the module's configuration for software agents. For more examples or to get complementary information, check the Module definition section in Configuration.
- Warning Status: The interval in which the status changes to `warning`. If the box inverse interval is checked, the status will change to `warning` if it is not within the range of the defined interval.
- Critical Status: Interval where the status changes to critical state. If the box inverse interval is checked, the status will change to `warning` if it is not within range of the defined interval.
- Warning Instructions: Instructions to follow if the status changed to `warning`.
- Critical Instructions: Instructions to follow if the state changed to critical.
- Unknown Instructions: Instructions to follow if the state changed to unknown.
- Category: If you need to group or categorize differently, you may define categories here.
- Tags: You may assign tags here.
- Macros: You may define macros within the execution module (`module_exec`) or plugin parameters.

Creating new local components

To create a new local component, click on Configuration → Templates → Local components and click Create, which is located at the right bottom of the page.

A page containing the form for creating new local components will be displayed. Fill out the form with the information given above and click Create to save.

Local execution macros

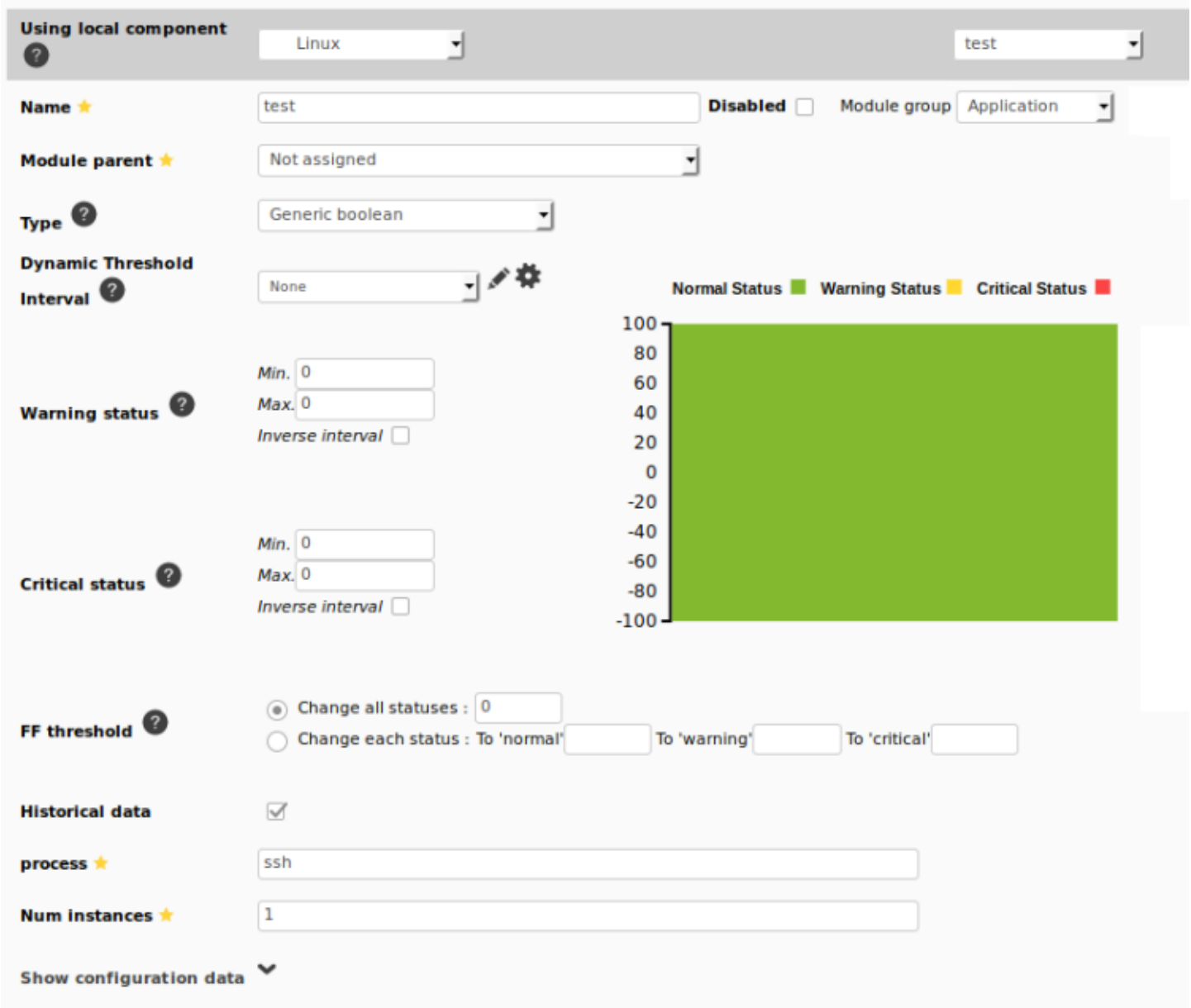
From Pandora FMS versions 5 onwards, it is possible to define macros within local components. These macros are used in the module_exec parameter. They follow the structure of `_field1_` , `_field2_` ... `_fieldN_`.

In the module edition form, macros will appear as normal fields, completely visible for the user.

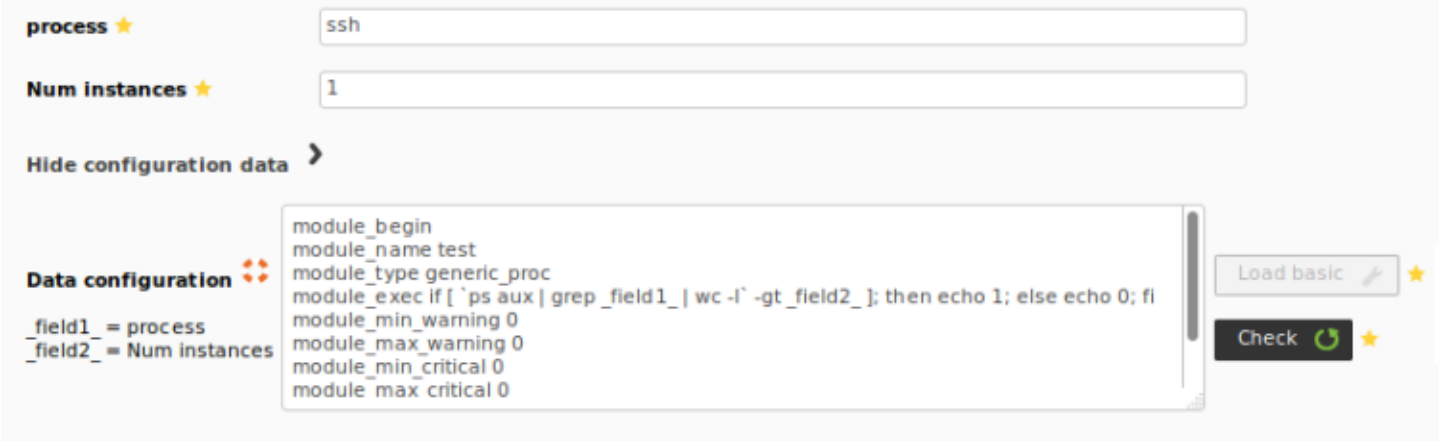
Each macro has three fields: Description, Default value and Help.

- Description: It is the label next to the field in the module form.
- Default Value: An optional value to be loaded by default in the module form field.
- Help: Optional string to add additional information to the field. If defined, a tip will appear next to the field with that string.

If a module component contains macros, the configuration data will be hidden by default to simplify the view:



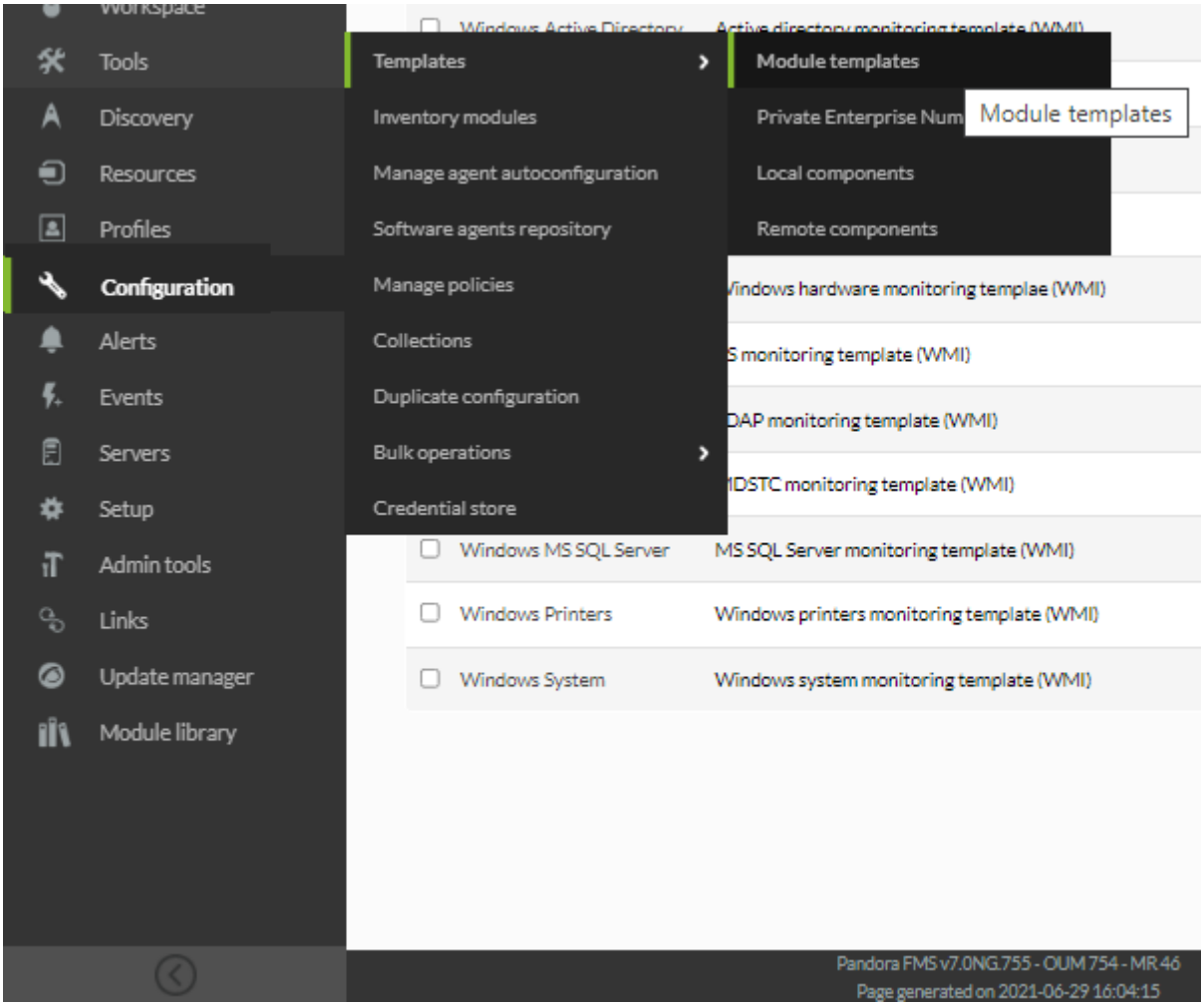
But it is possible to view and modify them.



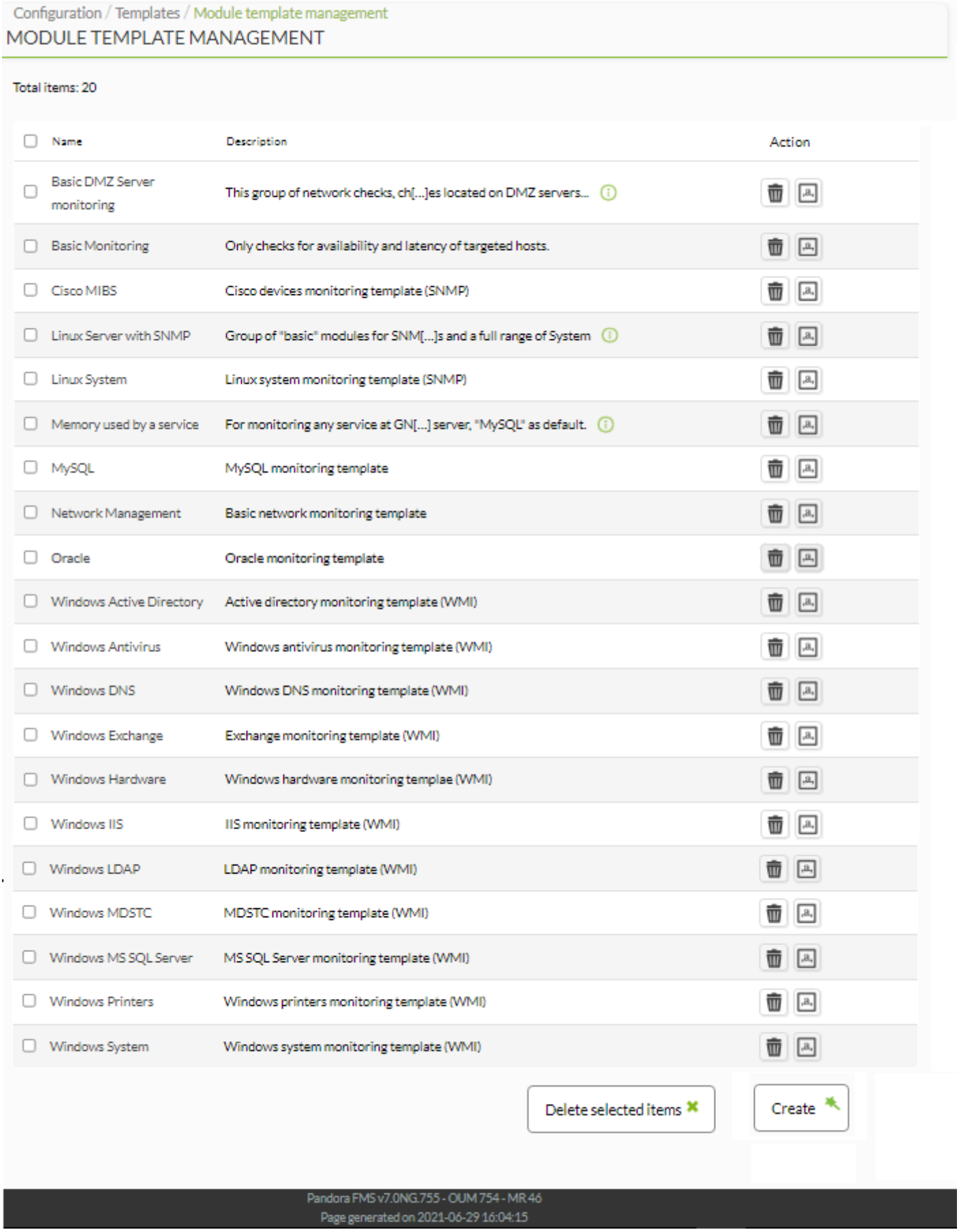
Module Templates

Module templates are templates that contain network check modules. Once created, these templates can be directly applied to agents, avoiding the need to add modules one by one, or apply the templates when carrying out a network recon task.

Click on Configuration → Templates → Module templates to manage the module templates.



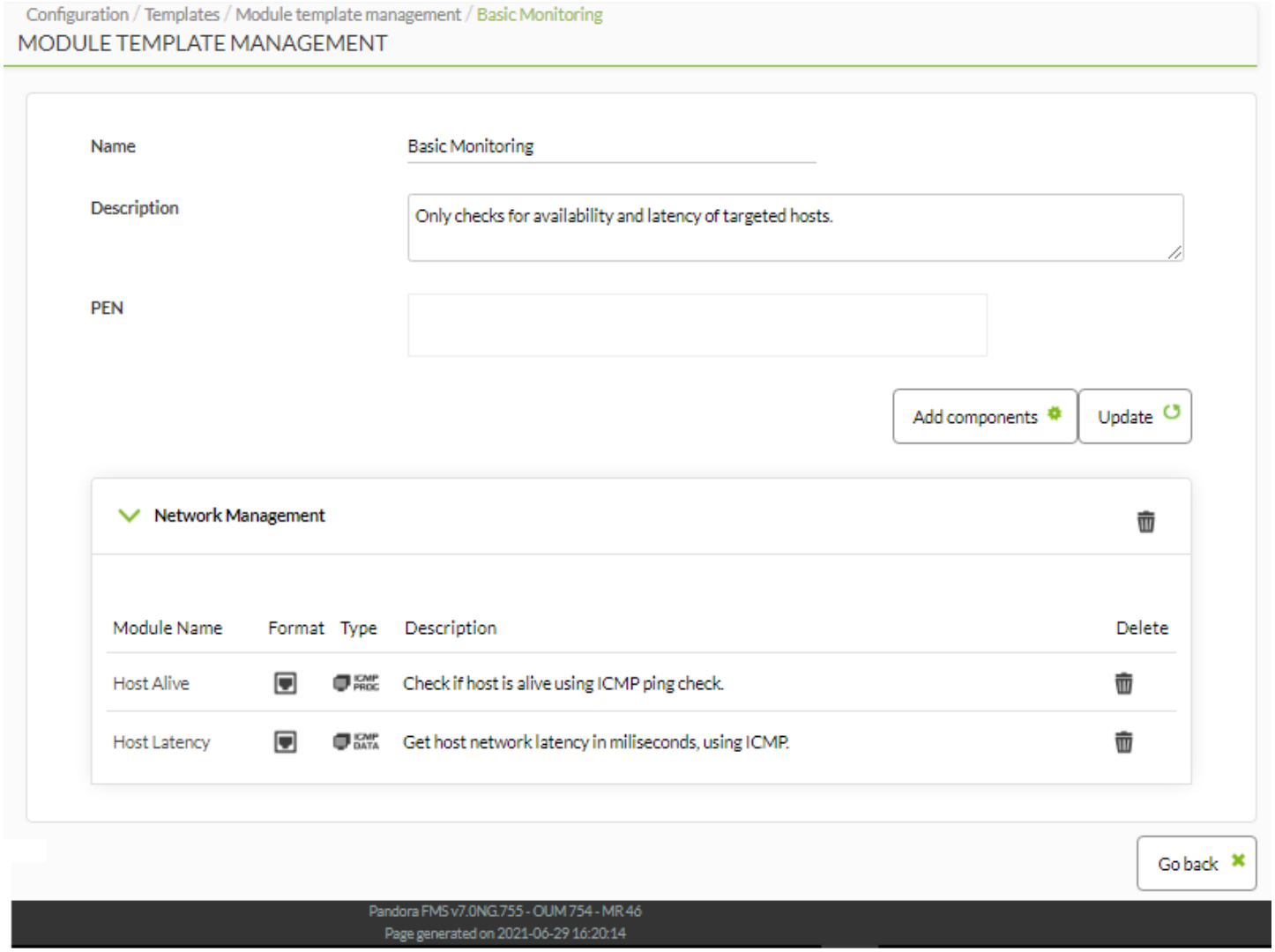
The template management window, which contains many default templates, will be displayed:



Click on any of the templates to see their details, or on the trash can icon in the right column to delete it, or on Create to create a new template.

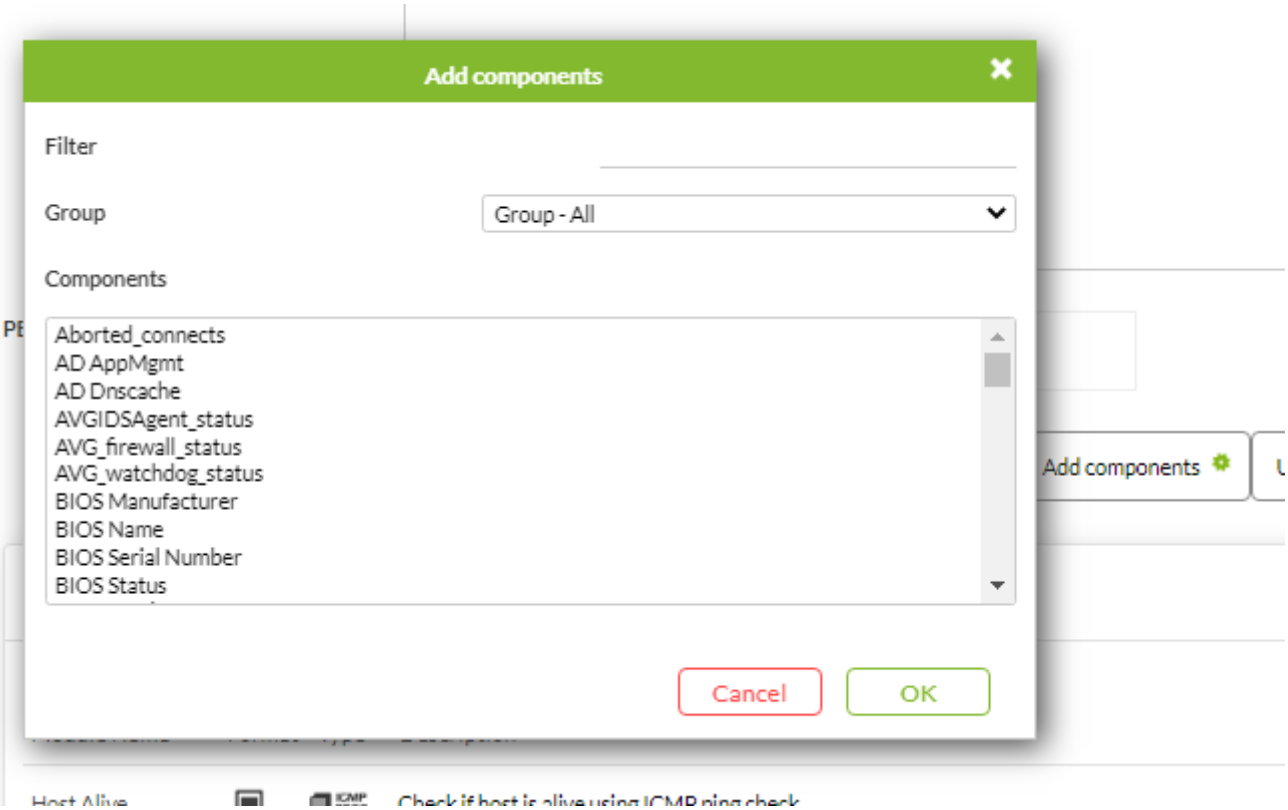
By clicking on the name of a template you will see its details, for example, the screenshot below

shows the details for the basic monitoring module template.



Below is the list of modules included in this template. In order to delete a module from that template, in the right column, click Delete and the corresponding trash icon.

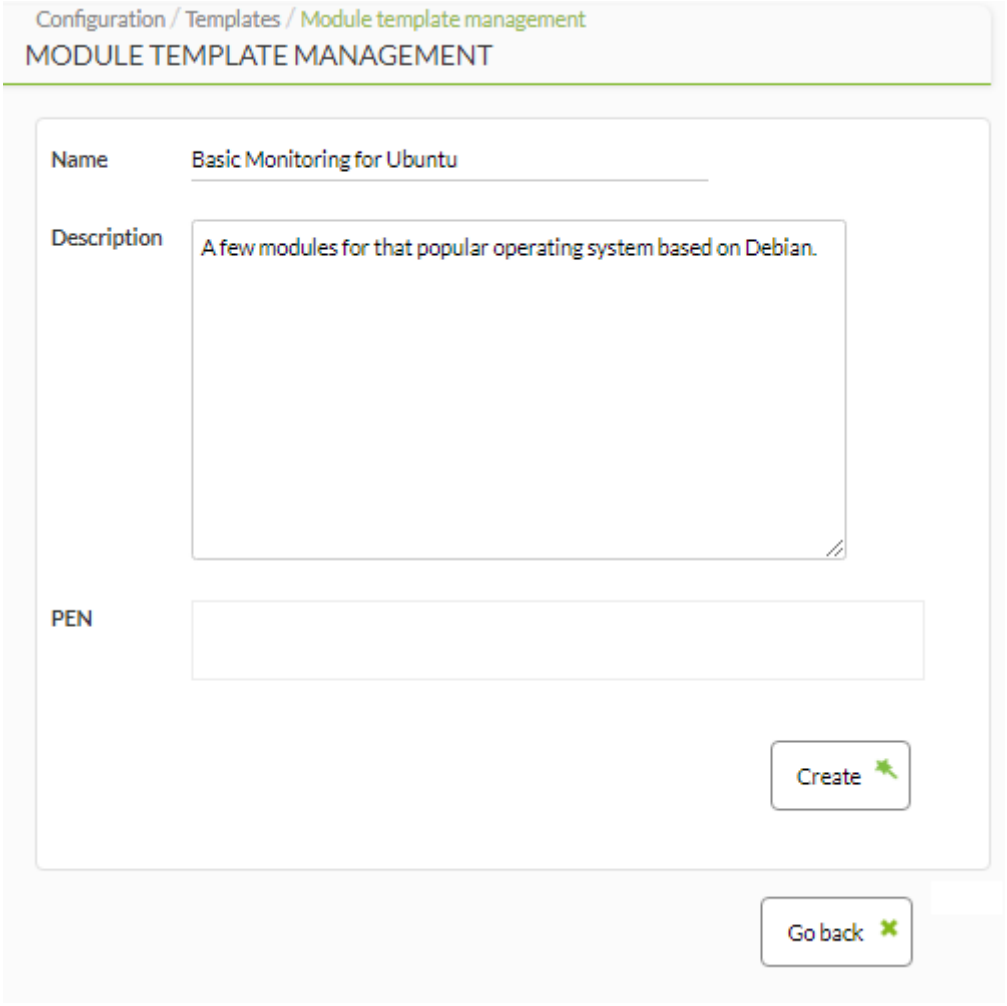
Finally, there is the form for adding modules when clicking on Add components. You may filter by module group, and then select the module and add it.



Creating new module templates

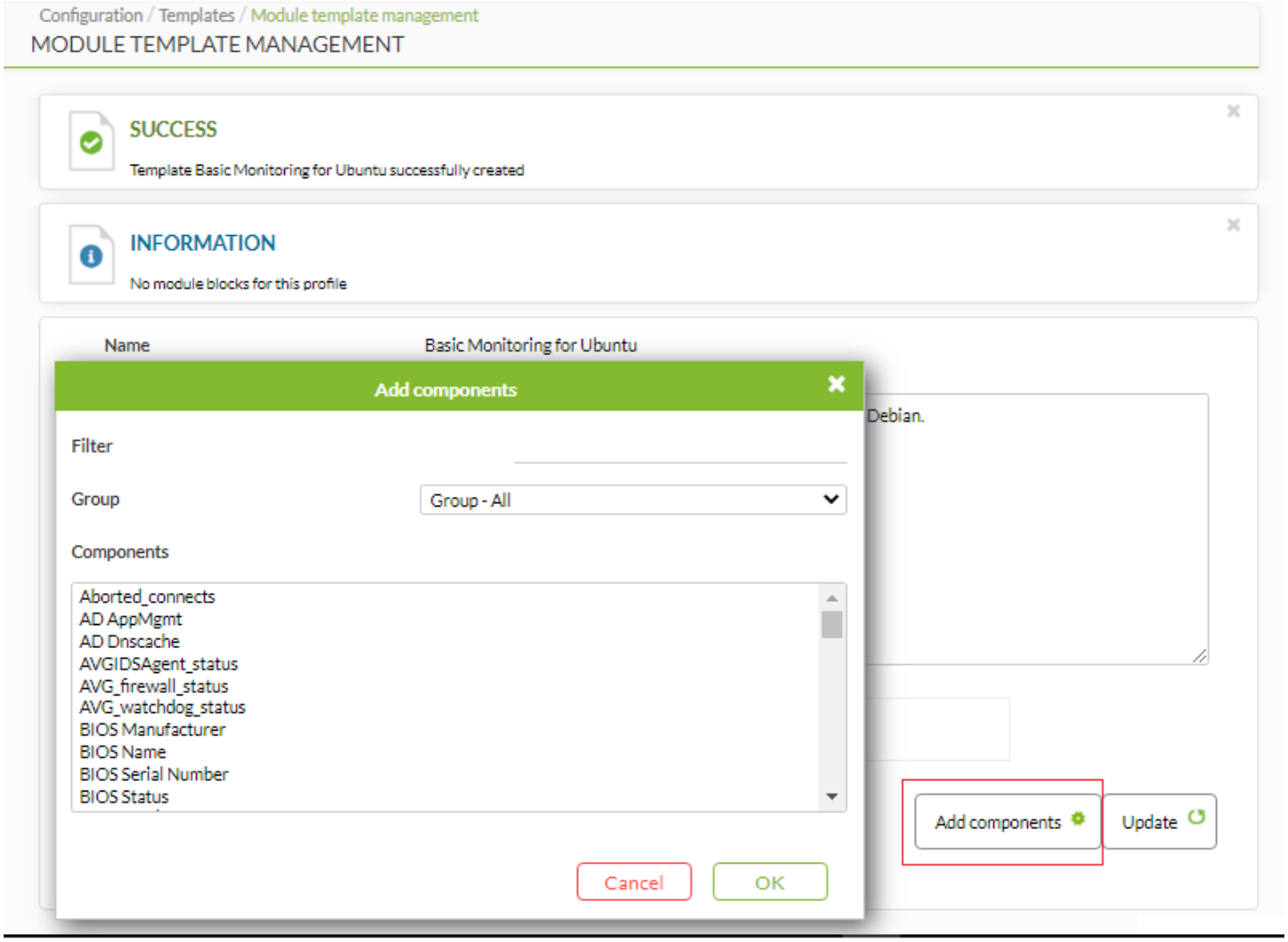
In order to create a new module template, go to the main management page, Configuration → Templates → Module templates and click on Create at the bottom-right side of the page.

A page containing the creation form for new local components will appear:



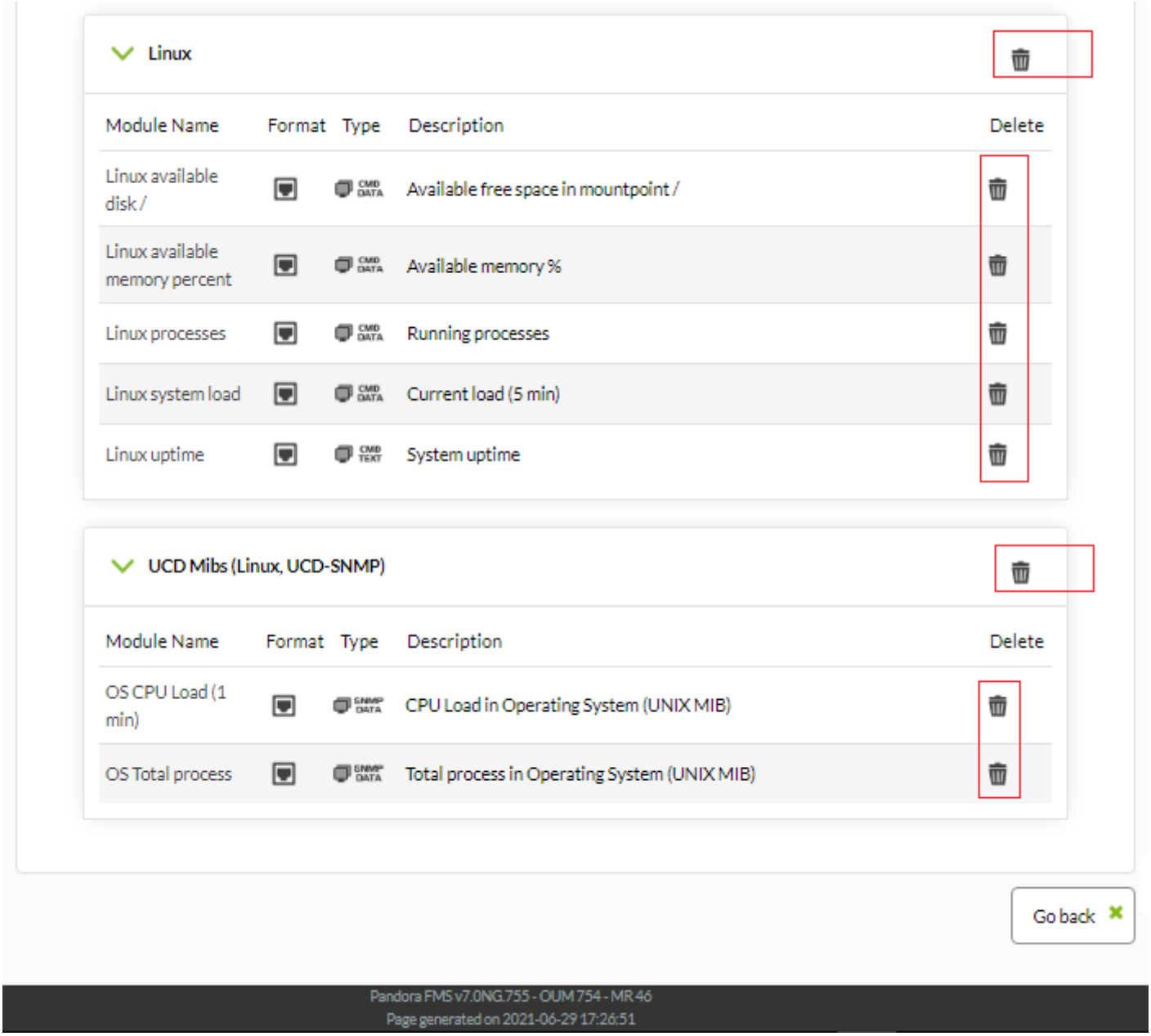
Enter the name and description for the new template and click Create.

Then you may add modules to the template by clicking on Add components.



Select the modules at the bottom, filtering them by group if necessary and click on OK.

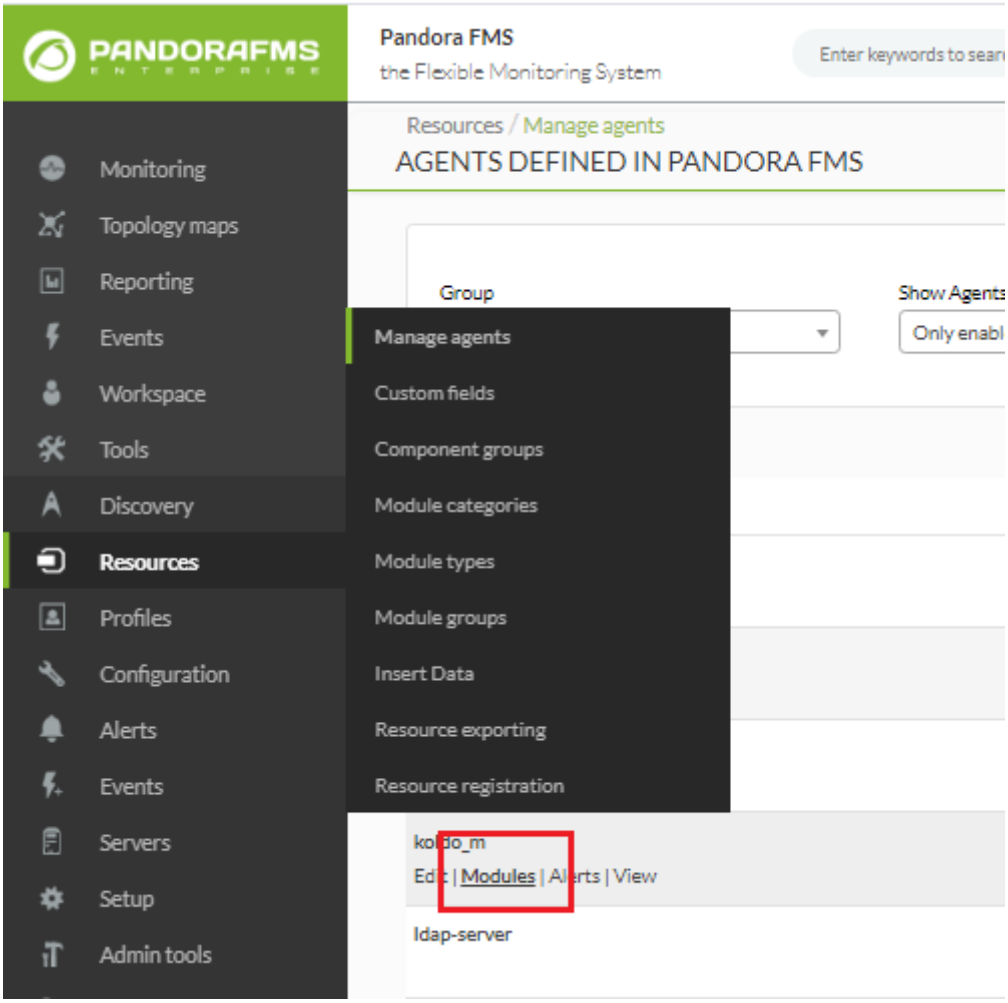
Keep in mind that you may delete the unwanted modules by selecting them and clicking Delete.



To finish off, remember to save the added modules by clicking on Update.

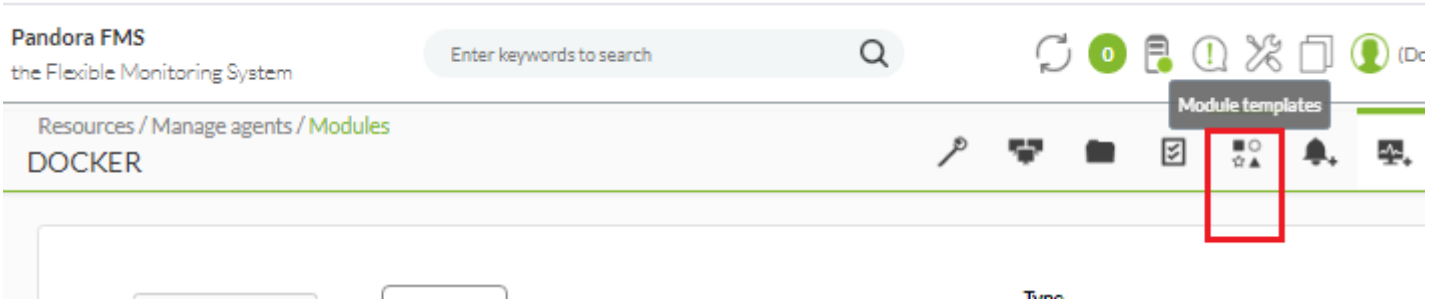
Applying a module template to an agent

In order to apply one of the existing monitoring module templates or a recently created one, go to agent configuration through the menu Resources → Manage agents.

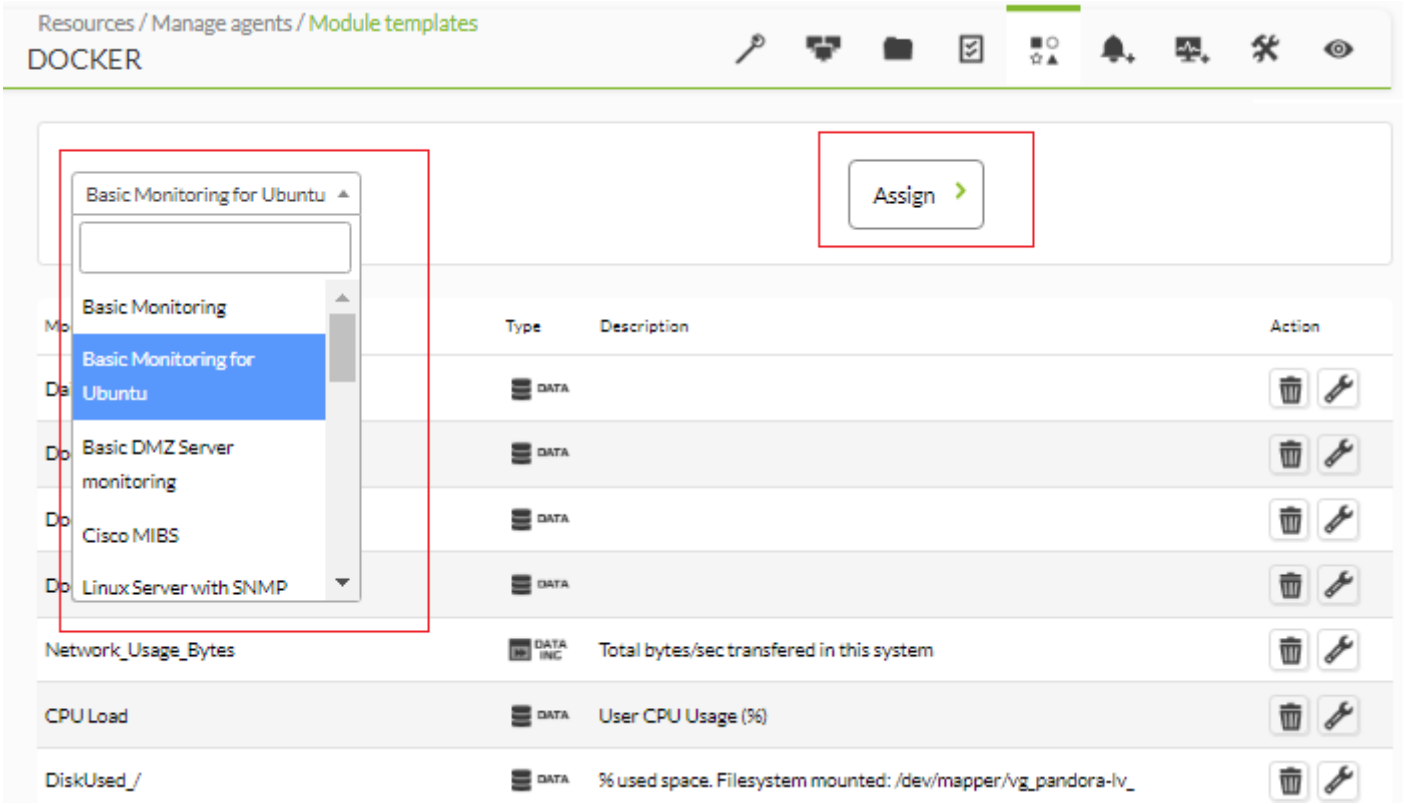


From the agent list, select one through the corresponding Modules link (see previous picture).

Once you see this window, click on Module templates at the top of the page.

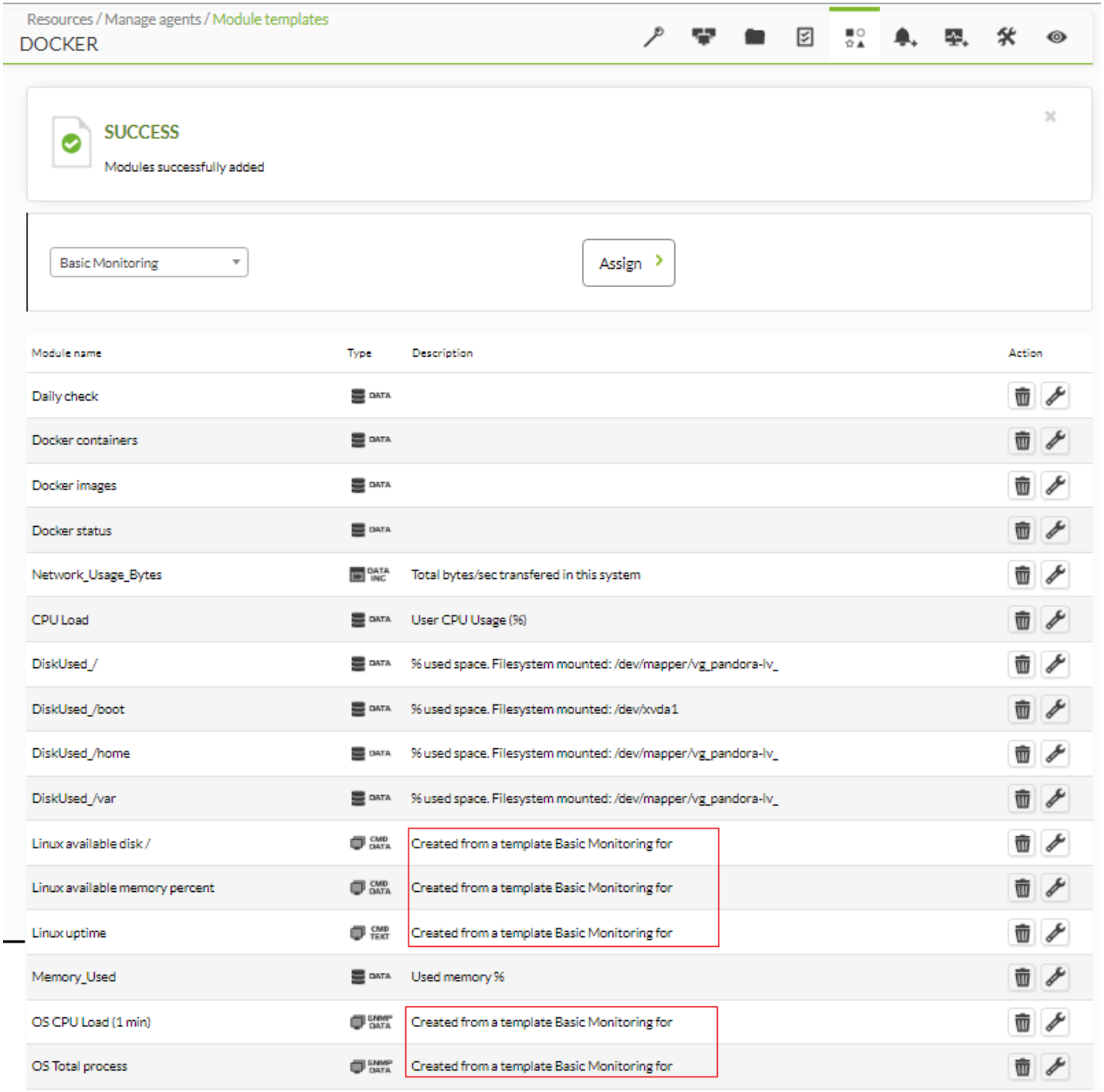


On the following picture, modules that already contain an agent and existing module templates are displayed.



Select a template and click Assign. The modules contained in this template will be added automatically. Once the template is applied, delete some of the modules by clicking on the trash can icon, or you may edit them clicking on the tool icon.

Added modules will have an automatic description based on the template's name:



The templates applied to the agent are not displayed, just the modules they contain.

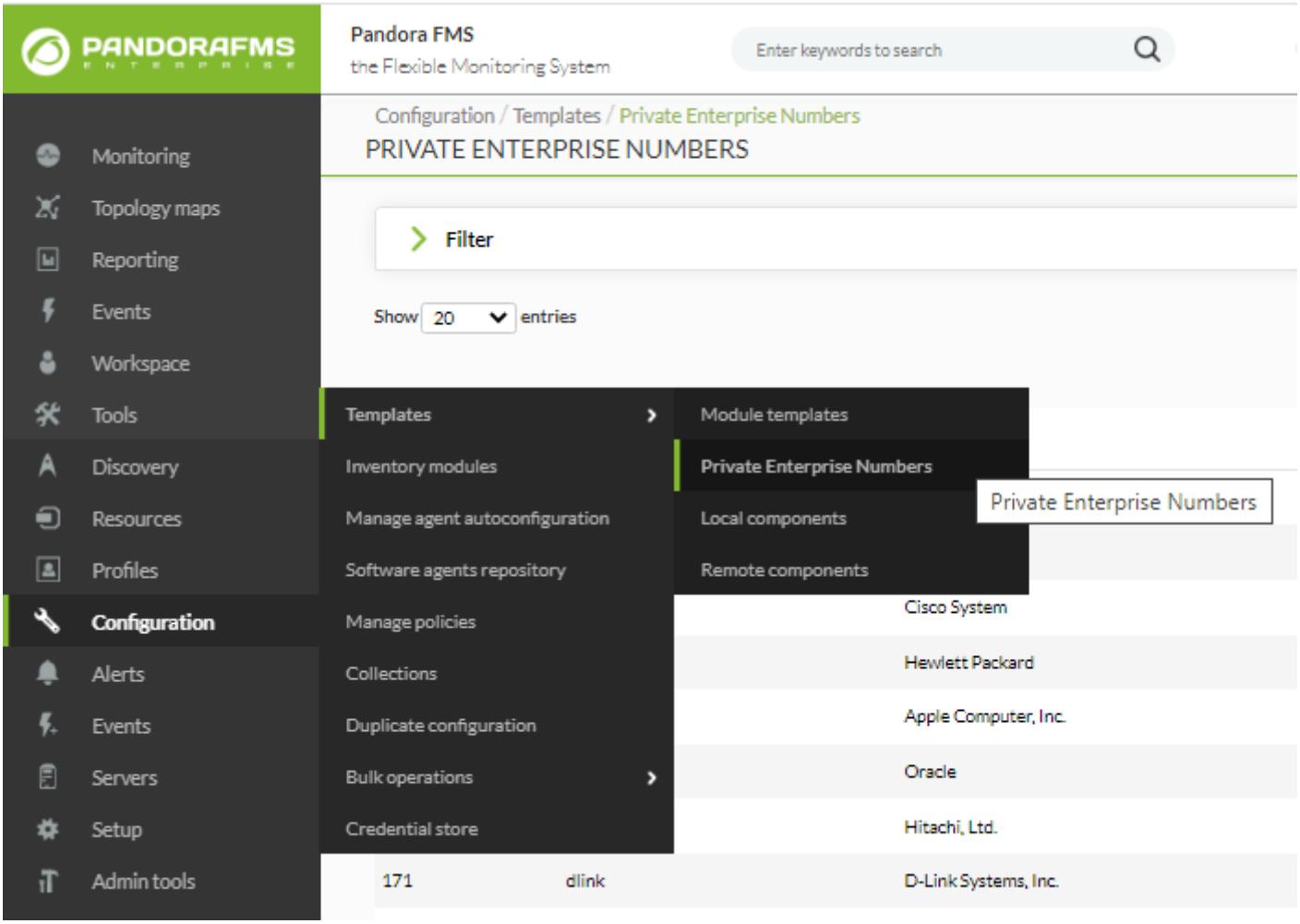
Private Enterprise Number

All SNMP devices have their own OID, which is exclusive to each device brand and model. There is a number occupying the seventh place within those strings, which is the one that gives away which manufacturer it is from.

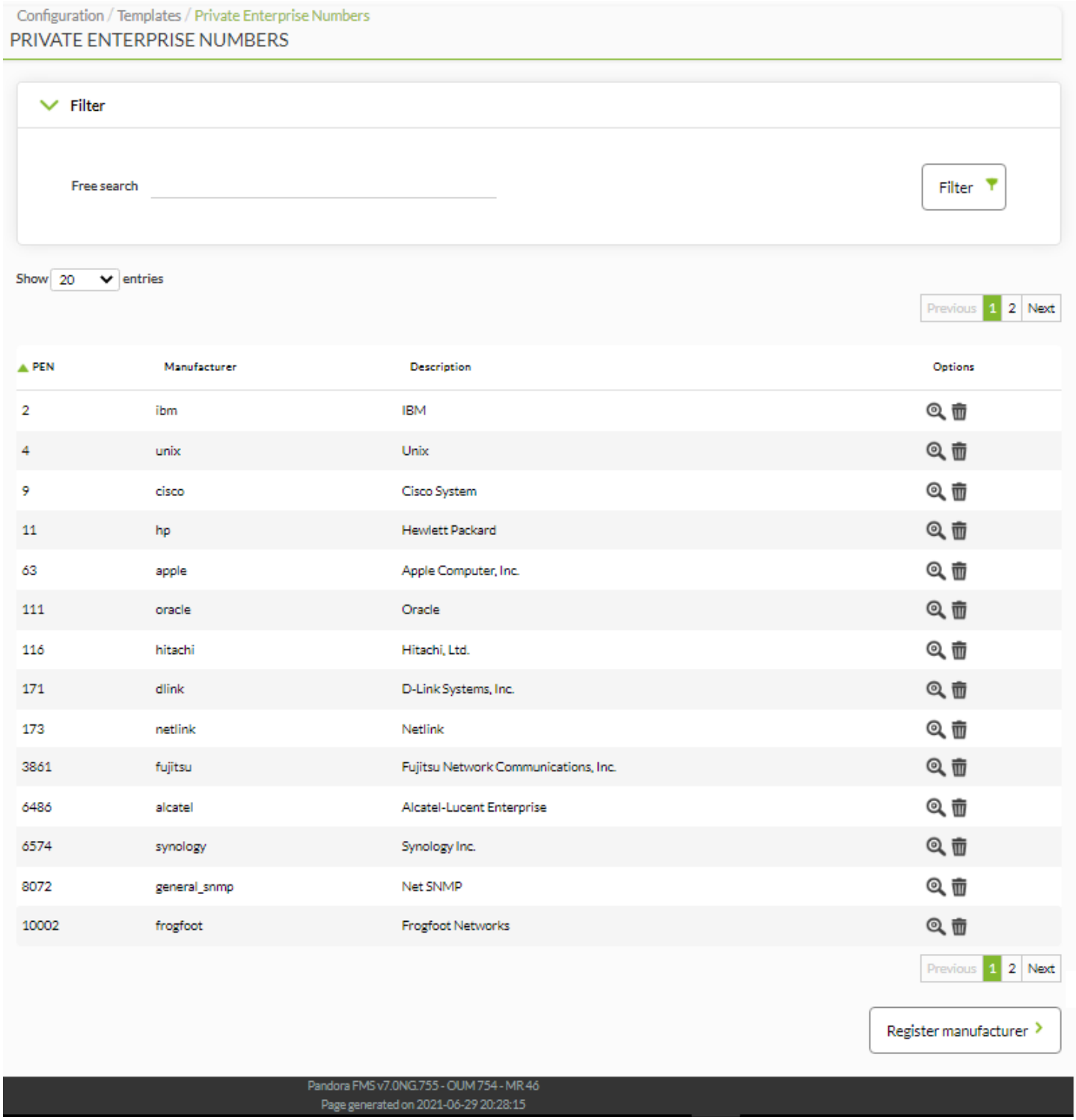
This is the manufacturer's Private Enterprise Number (PEN) and it is registered on IANA. These PEN can be configured in Pandora FMS to use them together with module templates and therefore

add dynamic monitoring.

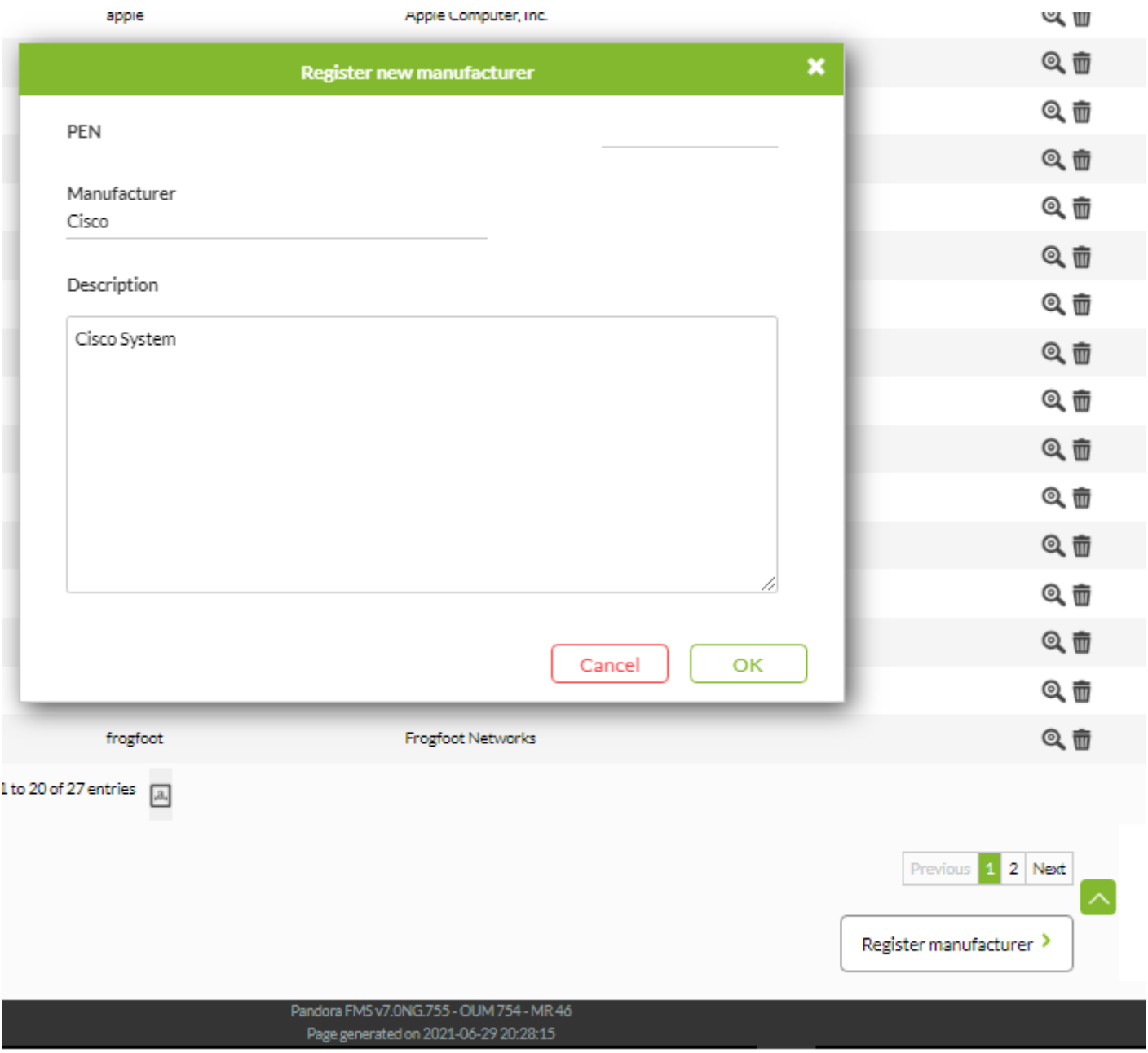
Go to Configuration → Templates → Private Enterprise Numbers to have acces to the list of registered PENs.



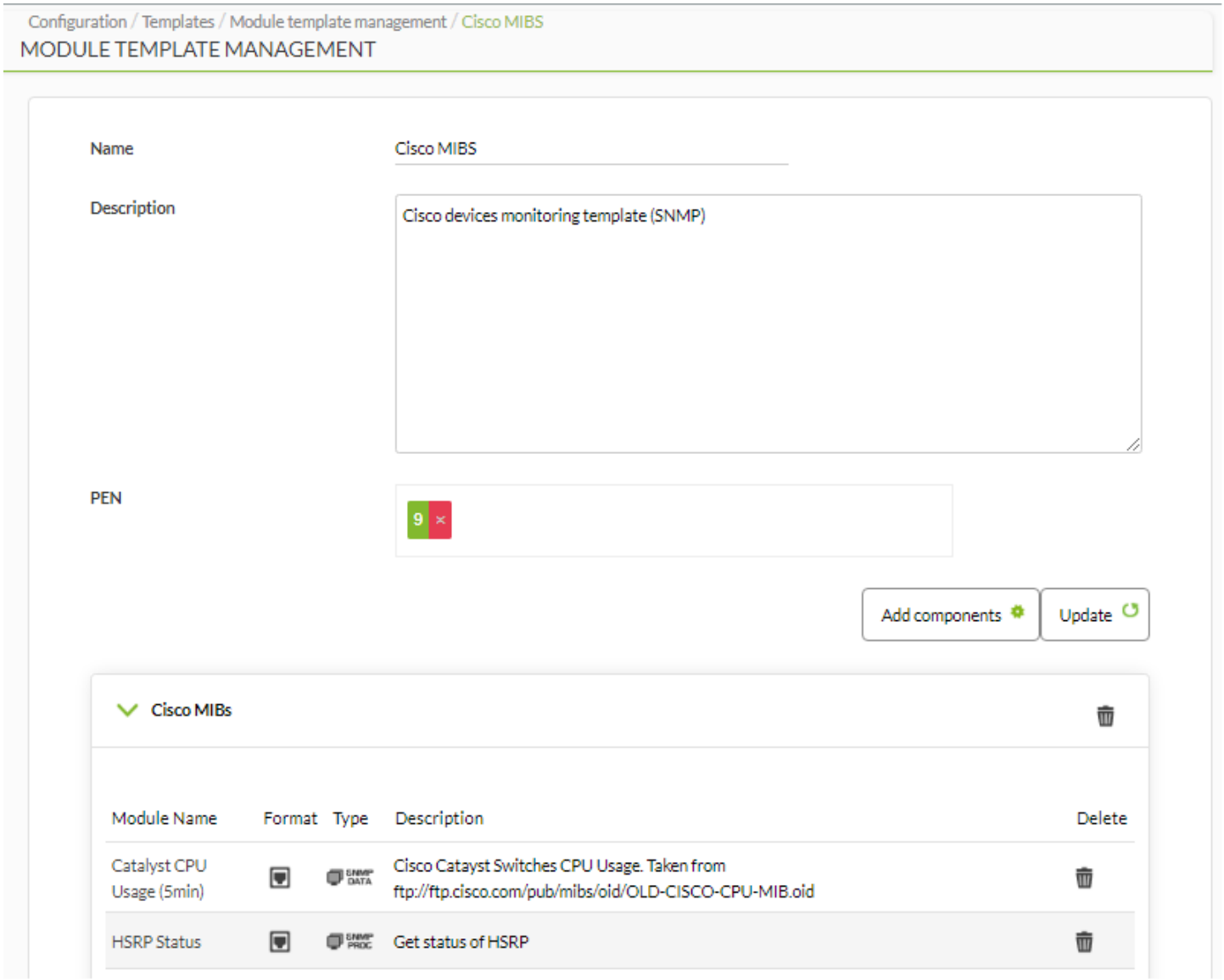
You may edit and/or delete each of the registered PENs by clicking on the corresponding icons from the Options column.



To register a new manufacturer, click on Register manufacturer. Just insert the manufacturer's corresponding PEN, indicate its name and a description. That way it will be added to the existing list.



In module templates, one or several PENs will be indicated so when there is a discovery task, Pandora FMS is able to retrieve the information about the device's manufacturer and add the appropriate monitoring information.

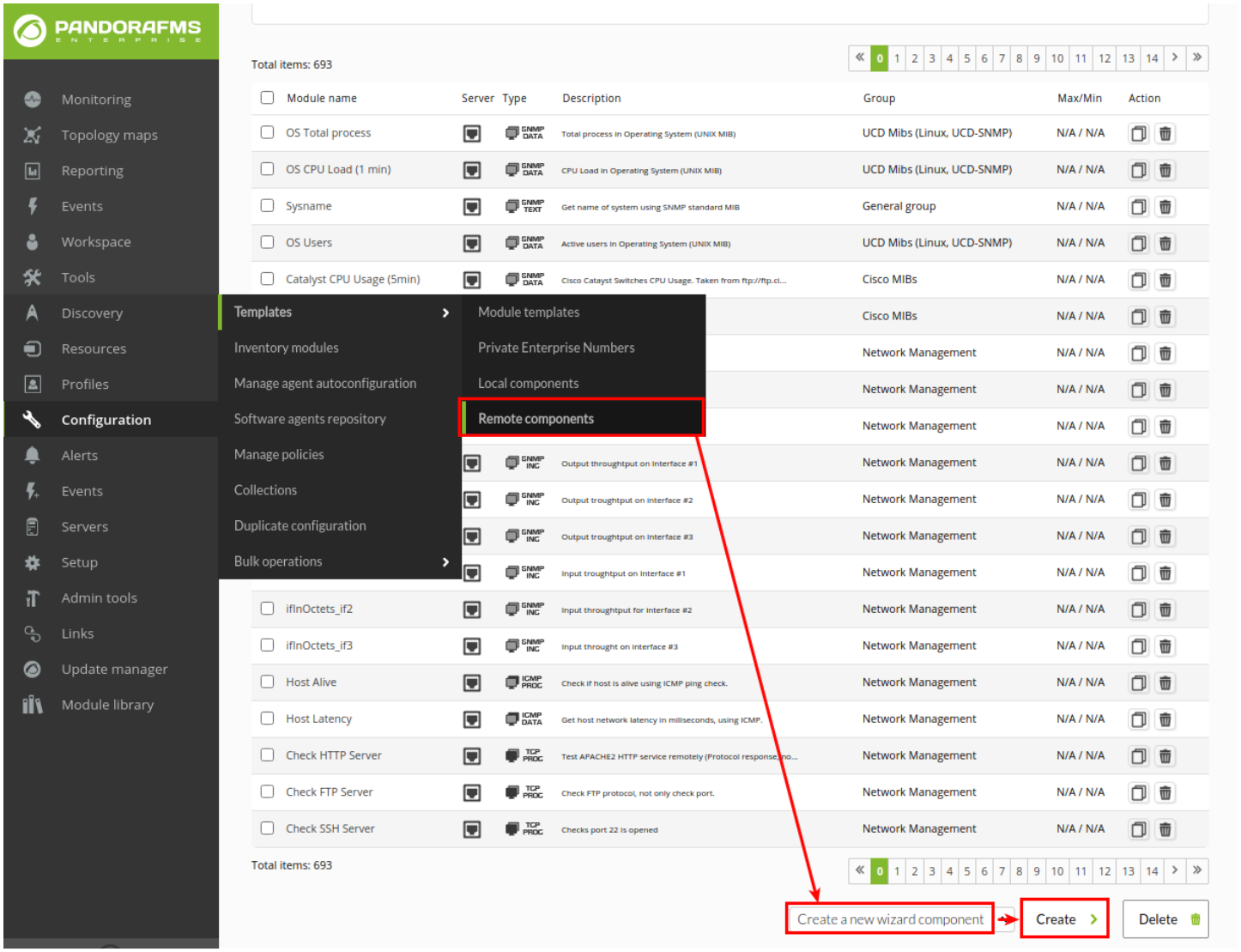


Wizard components

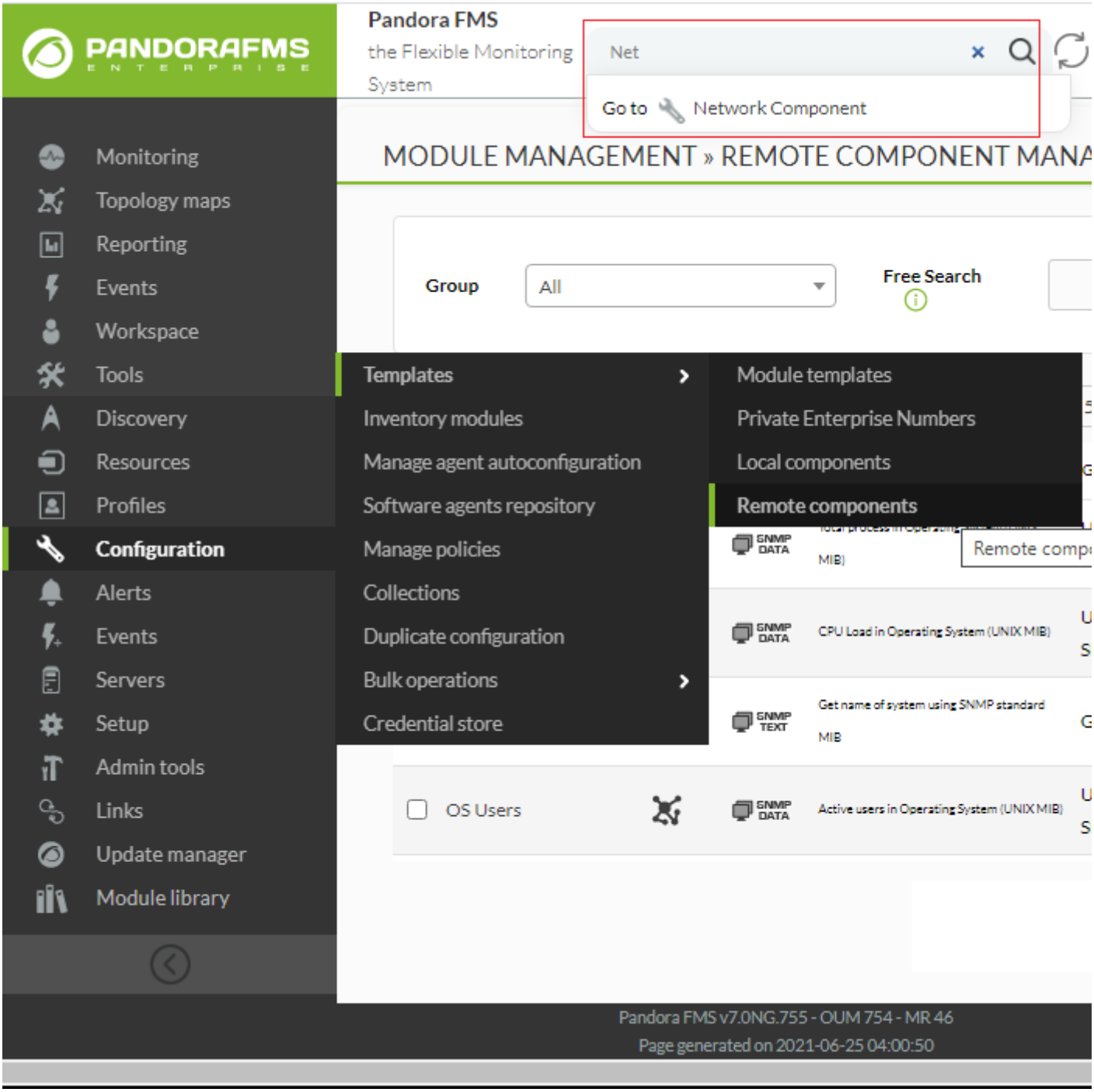
Within the capabilities of SNMP and WMI wizards, there is a type of remote components called *Wizard components*.

These components allow to set a base configuration for the modules that will be generated in the agents when executing any of the wizards (SNMP or WMI). Besides, it will offer the possibility of generating several modules dynamically through only one component. For example, a component to scan device storage units or processes in execution.

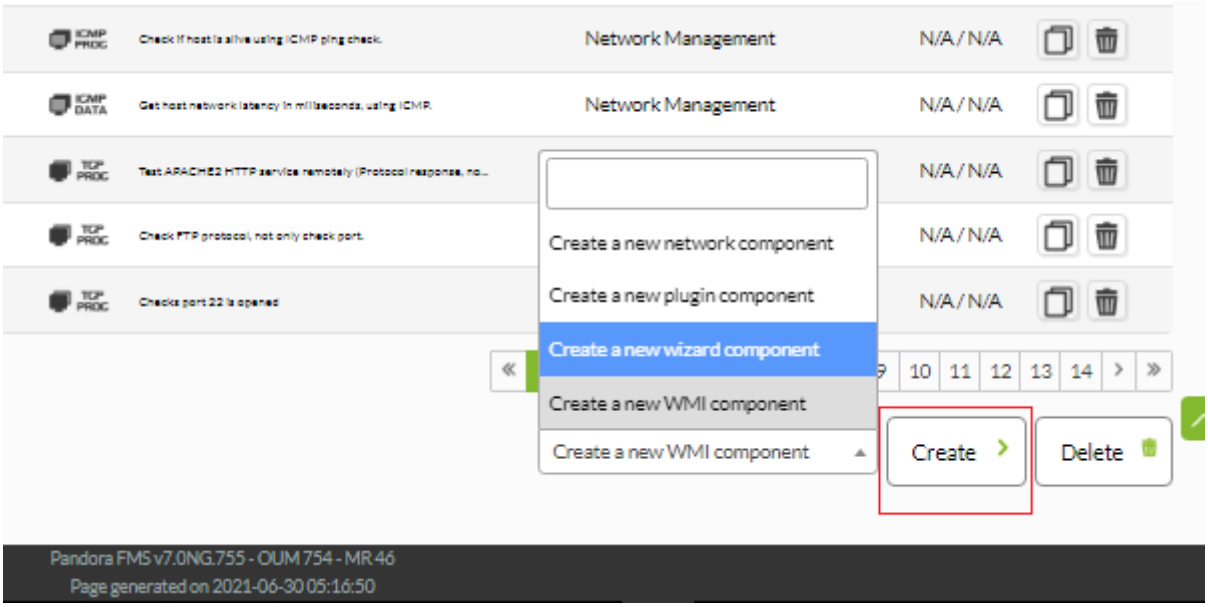
These components can be created from the menu Configuration → Templates → Remote components.



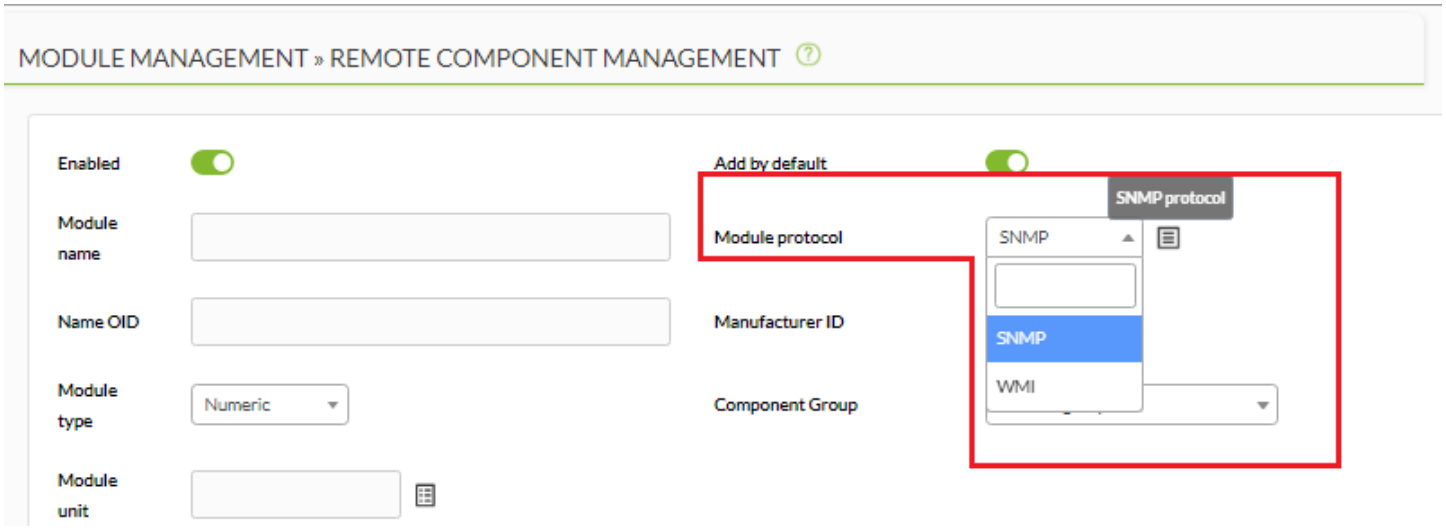
These components can be created from the menu Configuration → Templates → Remote components,



Select the option Create a new wizard component and click Create.



In the creation box, there is a key option: the protocol of the Module to be used. There are two protocols available, SNMP (by default) and WMI, and although both of them share common fields, they also have different fields.



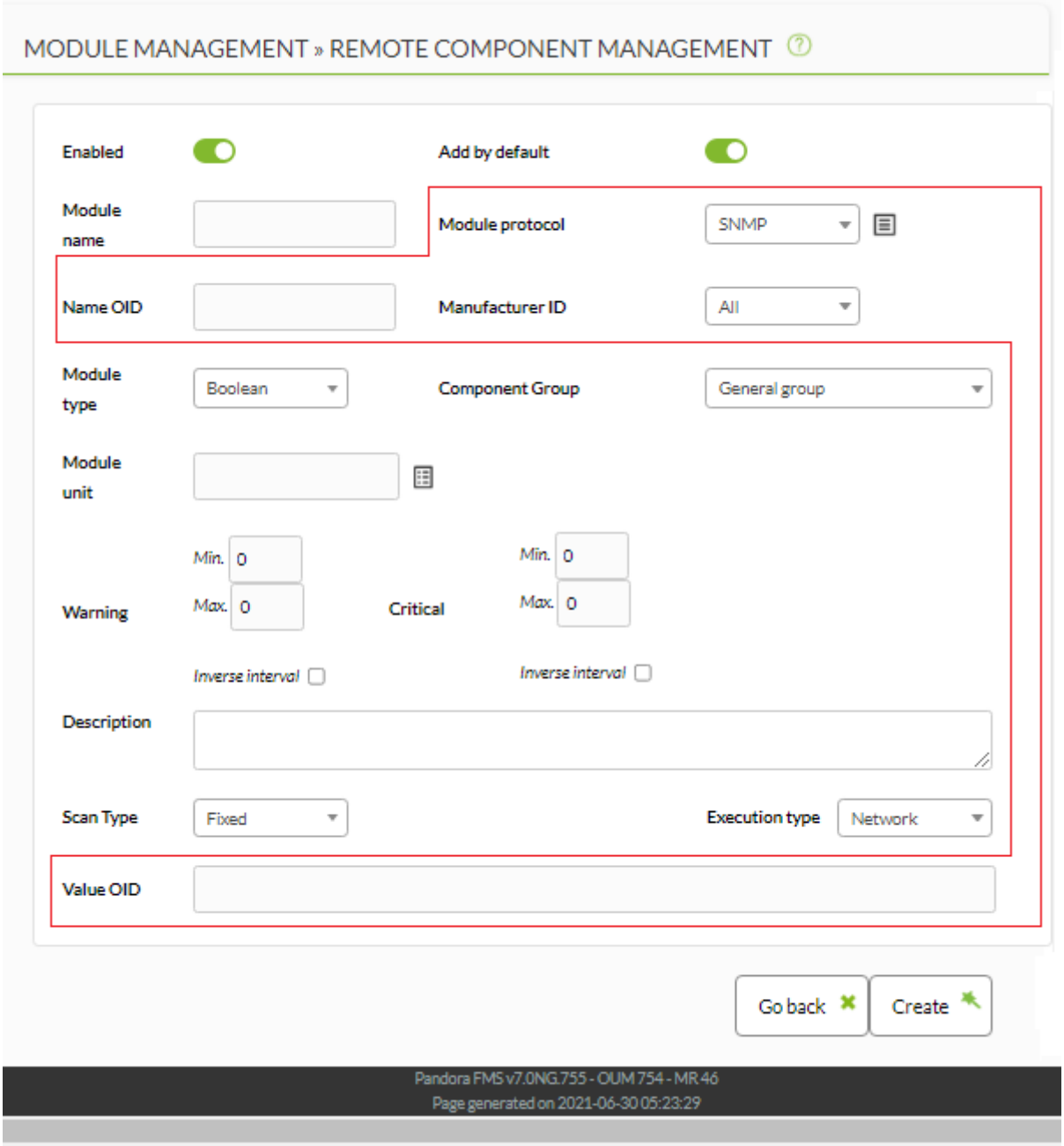
The common fields are:

- Enabled: By activating this token, you will be indicating that the component will try to scan when the wizard is launched.
- Add by default: It allows to choose whether the modules generated by the component will be mchecked to be added by default when launching the wizard. That means that if the token is activated, the modules generated by the component will be checked by default in a view that you will find later and they will be added to the agent. This action does not mean that it cannot be modified, so in this view you can make modifications and uncheck or check at will and change thresholds, descriptions, etc.
- Module name: Name that the component will have and default name for the modules generated by it. It will be possible to use some macros explained in following sections.
- Module protocol: It allows to choose between SNMP and WMI (some fields change).
- Module type: In this drop-down list, you can choose the type of data that the modules generated by the component will obtain.
- Component group: Group to which the component will belong. It allows to organize the way the

modules will be presented.

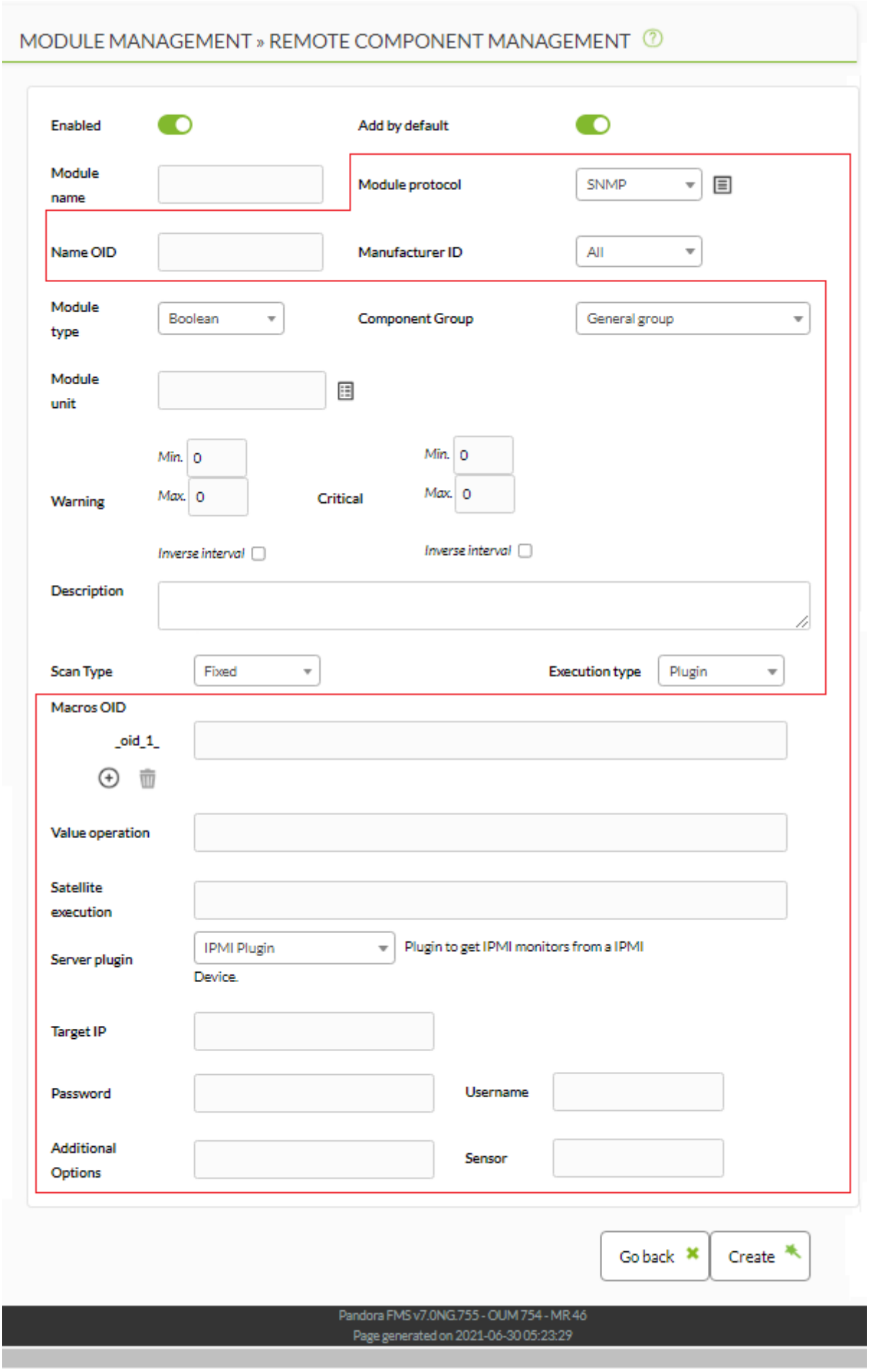
- Module unit: Unit of the data obtained by the modules generated by the component. It is a totally editable field, so you can add the measure needed.
- Warning status: In this section you can set a threshold by default for the `warning` status of the wizard-generated modules. Although here a range is indicated, there will be the possibility of customizing it for each module in the final view that collects all the found modules.
- Critical status: In this section, you may set a default threshold for the `critical` status of the wizard-generated modules. Although there is a range in here, you may customize it for each module in the final view that collects all the found modules.
- Description: This is a description that will have the component and at the same time, the modules it generated. You will be able to use some macros. (They will be shown later on).
- Scan type: It allows to choose between two scanning modes that can be performed by wizards with this component. This field determines whether a component will generate one module or several. The selected value will affect how other specific fields of each wizard must be filled out.
    - Fixed: The component will only generate one module. For example, get the uptime of the device by SNMP.
    - Dynamic: The component could generate one or more modules. For example, to obtain the percentage of disk unit usage by WMI.
- Execution type: This field indicates the execution type for component-generated modules. It is useful to determine the Pandora FMS server the modules will belong to when created depending on where the wizard is launched from.
    - Network: The modules generated by the component will get their data with Pandora FMS own system for SNMP and WMI modules. These are network server, WMI server and Satellite server.
    - Plugin: The modules generated by the component will obtain their data from the execution of commands, plugins or customized scripts. Thus, they will be executed by the plugin server or satellite server through `exec` modules.

SNMP wizard



The specific fields SNMP wizard components are:

- Name OID
- Manufacturer ID
- When Execution type is set in Network:
    - Value OID
- When Execution type is set in Plugin:
    - OID Macros
    - Value operation
    - Satellite execution
    - Server plugin


MODULE MANAGEMENT » REMOTE COMPONENT MANAGEMENT
Enabled
Add by default
Module name
Module protocol
SNMP
Name OID
Manufacturer ID
All
Module type
Boolean
Component Group
General group
Module unit
Min. 0
Max. 0
Warning
Critical
Min. 0
Max. 0
Inverse interval
Inverse interval
Description
Scan Type
Fixed
Execution type
Plugin
Macros OID
_oid_1_
Value operation
Satellite execution
Server plugin
IPMI Plugin
Plugin to get IPMI monitors from a IPMI Device.
Target IP
Password
Username
Additional Options
Sensor
Go back
Create
Pandora FMS v7.0NG.755 - OUM 754 - MR 46
Page generated on 2021-06-30 05:23:29

Name OID

It allows to indicate an OID from which a value will be obtained that could be added to the module name through a macro. It is especially useful when you get multiple modules generated by a dynamic component. That way they get different names by default. But it is not limited to dynamic components, since it can be used also for fixed scanning components.

The value of this OID is stored in the macro `_nameOID_`, that can be used in the Module name field.

If used in dynamic components, the OID indicated in this field should be a branch of SNMP and not a final OID. For example, if the OID `.1.3.6.1.4.1.2021.10.1.2` is indicated, the values that the macro will have in each module will be obtained from the OIDs `.1.3.6.1.4.1.2021.10.1.2.x`, where x represents each of the terminations that the branch may have.

If used on fixed components, the OID indicated in this field must be a final OID. For example, if the OID `.1.3.6.1.2.1.1.5.0` is indicated, the value the macro will have in the module will be retrieved directly from that OID.

Manufacturer ID

It allows to indicate the ID of a specific manufacturer for which the SNMP wizard component will take effect. That way, for all devices against which the wizard is launched, and whose Private Enterprise Number (PEN) is registered in Pandora FMS for the manufacturer ID assigned to the component, it will be tried to obtain the modules it generates. For example, a component assigned to`general_snmp` will be scanned for all devices with PEN 2021 and 8072.

If you indicate as manufacturer All, the component will be scanned for any PEN registered in Pandora FMS.

The Private Enterprise Number (PEN) must be registered in
Pandora FMS console to use Manufacturer ID

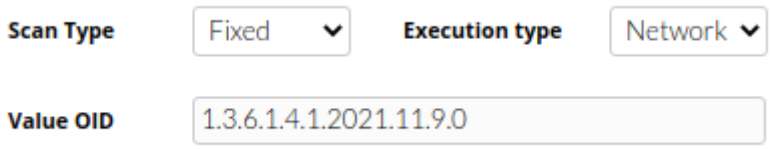
Network SNMP execution

When the type of execution is Network:

Value OID:

It allows to indicate the OID from which the component-generated module data will be obtained. If used in dynamic components, the OID indicated in this field should be a branch of SNMP and not a final OID. For example, if the OID `.1.3.6.1.4.1.2021.10.1.3` is indicated, the values that the modules will have will be obtained from the OIDs `.1.3.6.1.4.1.2021.10.1.3.x` . In addition, the *X* node of each OID must have the same value for the *X* node of the Name OID field if used.

If used in fixed components, the OID indicated in this field must be a final OID. For example, if the OID `.1.3.6.1.4.1.2021.11.9.0` is indicated, the value that the module will have will be obtained directly from that OID.



SNMP Plugin execution

When there is a plugin execution:

OID Macros → `_oid_N_`

The main purpose of using plugin components is to be able to perform operations with the values of one or more OIDs in the same device, such as obtaining the used-memory percentage from the used-memory bytes and the total available memory bytes.

That is why in these components, you can indicate as many OIDs as you need to use them in other fields.

Besides, these OIDs, or their values, can be used from the `_oid_N_` macros. Depending on which of the following fields the macro is used in, the value of the OID or the OID itself will be used.

If used in dynamic components, the OIDs indicated in these fields must be a branch of SNMP and not a final OID. For example, if the OID `.1.3.6.1.4.1.3375.2.1.7.4.2.1.3` is indicated, the values that the modules will have will be obtained from the OIDs `.1.3.6.1.4.1.3375.2.1.7.4.2.1.3.x` . In addition, the *X* node of each OID must have the same value for the *X* node of the rest of the OIDs used and the Name OID field if used.

If used in fixed components, the OIDs indicated in these fields must be a final OID. For example, if the OID `.1.3.6.1.4.1.2021.4.6.0` is indicated, the value that the module will have will be obtained directly from that OID.

Value operation

It allows you to indicate an arithmetic operation by means of which the current value of each module generated by the component will be obtained in the preview of the wizard modules. By no means does it affect the final execution of the generated modules.

It accepts the characters `+ - * / ( ) .` , numbers and the `_oid_N_` macros from which the values for the operation will be obtained. For example:

```
(_oid_1_ * 100) / _oid_2_
```

Satellite execution

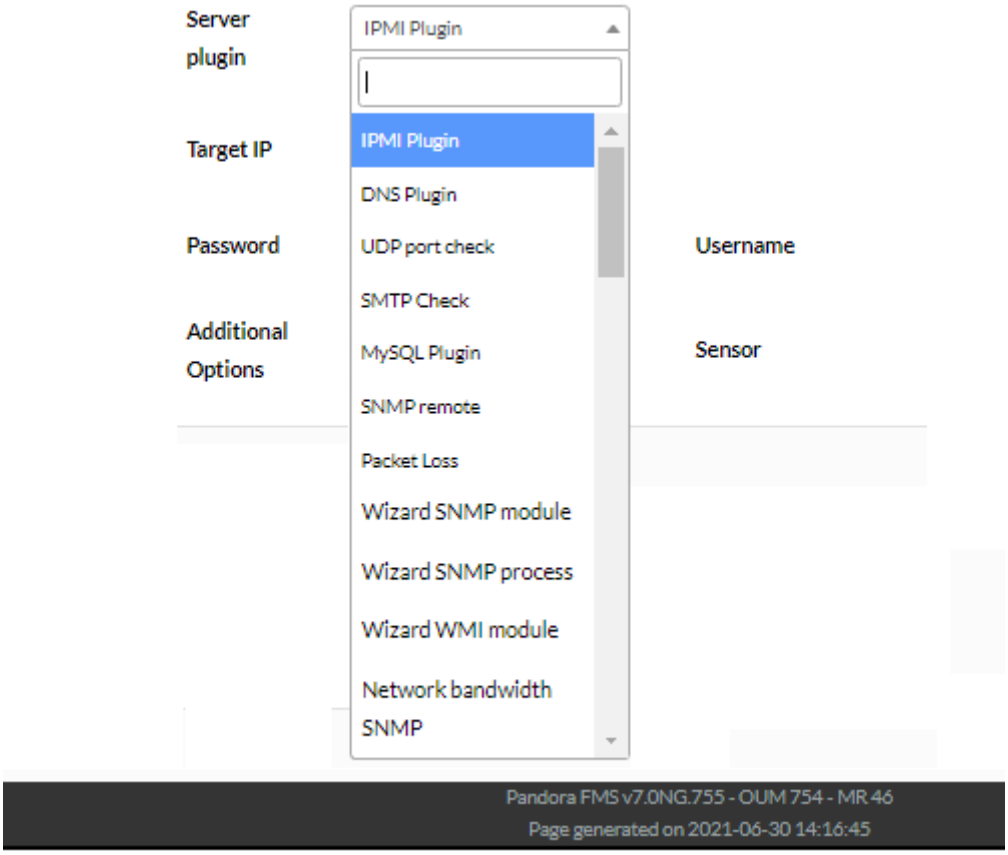
It offers the possibility to indicate the execution that a Satellite Server must do for the generated modules when the wizard is launched from a Satellite Server by using the *exec server*. This is the command, plugin or script that should be used in a module_exec of a satellite server.

It accepts the use of macros for the SNMP wizard (they will be detailed later) and of the `_oid_N_` macros to obtain the OIDs used in each module.

The Satellite Server distributes a series of recommended plugins for these components:

```
          - /etc/pandora/satellite_plugins/wizard_snmp_module
        - /etc/pandora/satellite_plugins/wizard_snmp_process
```

Server plugin



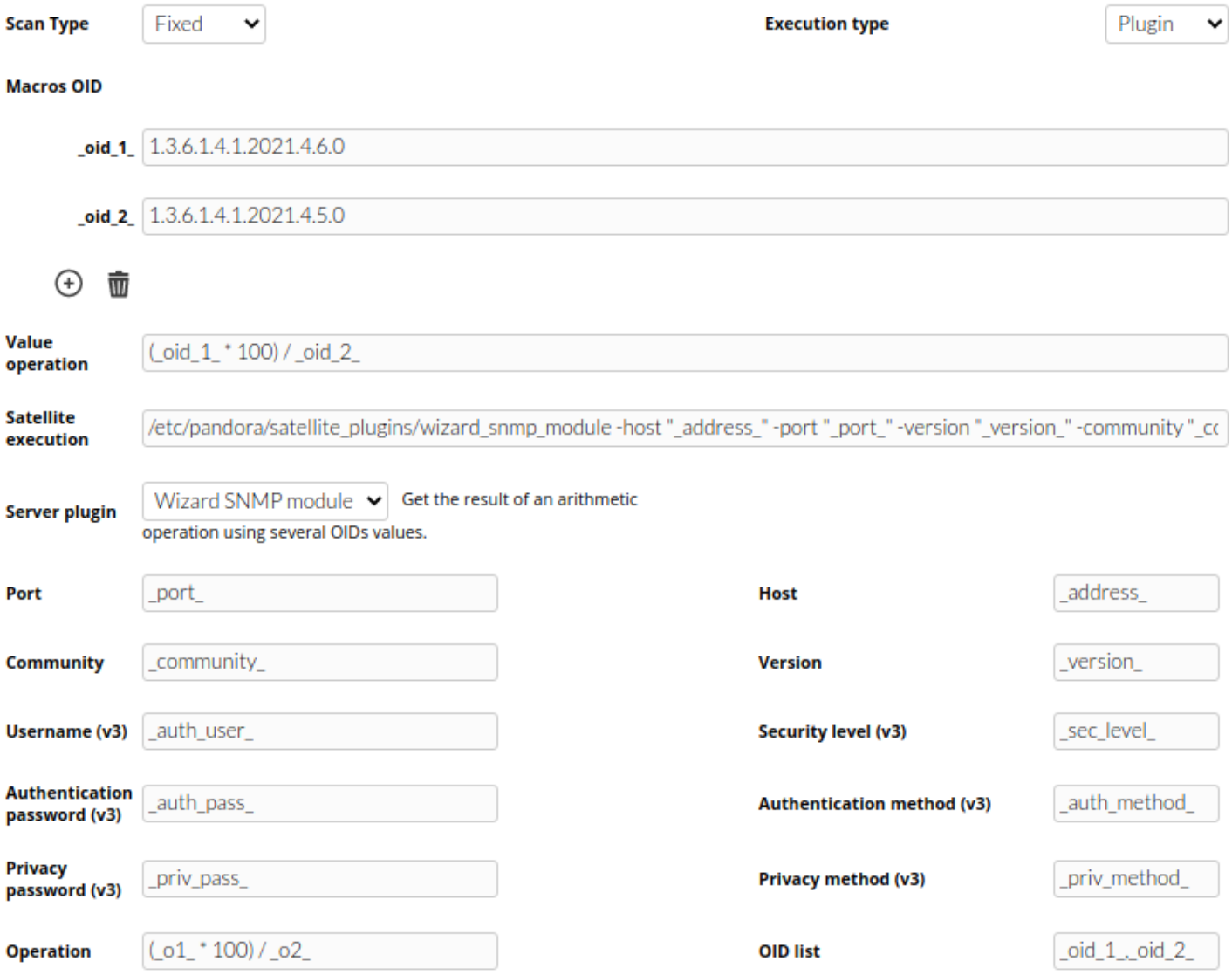
Through this dropdown you may indicate a plugin previously registered in Pandora FMS console, which will be used by the plugin server with each module generated by the component. The choice of a plugin shows at the same time the specific fields for its use in the form.

The own plugin fields accept the use of macros for the SNMP wizard and `_oid_N_` macros to get the OIDs used in each module.

Pandora FMS console has a series of already registered plugins recommended for these components:

```
        - Wizard SNMP module.
        - Wizard SNMP process.
```

For example, when selecting Wizard SNMP module you will get the following fields to fill in:

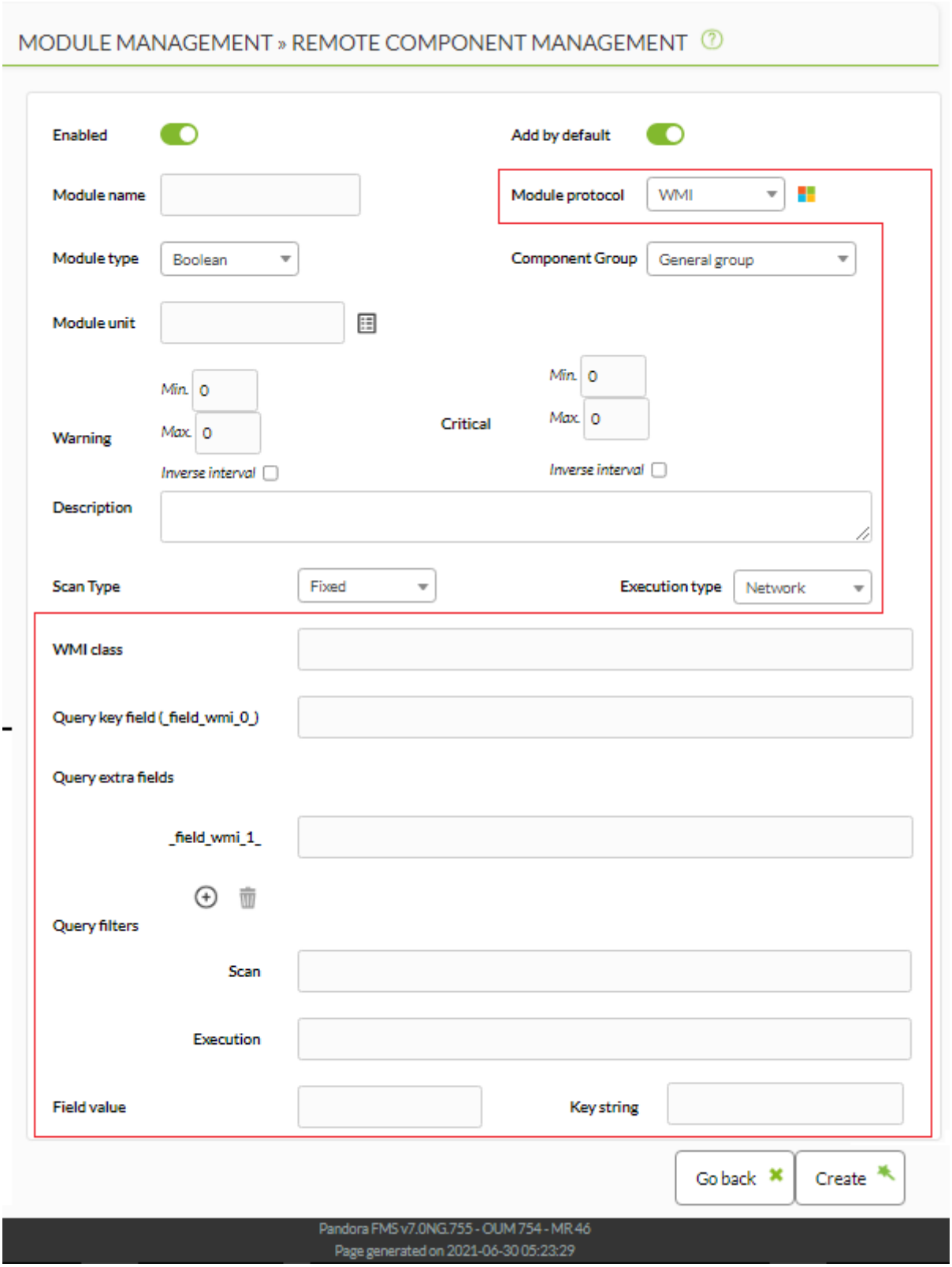


The specific macros for the SNMP wizard components that can be used in the plugin type execution fields are

- *_address_* : IP address used in the SNMP wizard. This macro will not be replaced when the wizard is launched in a policy.
- *_port_* : Port used in the SNMP wizard.
- *_version_* : SNMP version used in the SNMP wizard. It can have values 1, 2c or 3.
- *_community_* : SNMP community used in the SNMP wizard.
- *_sec_level_* : SNMPv3 security level used in the SNMP wizard. It may have values noAuthNoPriv, authNoPriv or authPriv.
- *_auth_user_* : SNMPv3 user used in the SNMP wizard.

- *_auth_method_* : SNMPv3 authentication method used in the SNMP wizard. It may have MD5 or SHA values.
- *_auth_pass_* : SNMPv3 authentication password used in the SNMP wizard.
- *_priv_method_* : SNMPv3 privacy method used in the SNMP wizard. It may have DES or AES values.
- *_priv_pass_* : SNMPv3 privacy password used in the SNMP wizard.

WMI wizard



Network WMI execution

When the type of execution is Network:

WMI class

It allows to specify the WMI class to check. WMI classes are predefined and included in each namespace from Windows Management Instrumentation® (WMI) core.

Query key field (`_field_wmi_0_`)

Query extra fields (`_field_wmi_1_`)

Query filters → Execution:

This space allows indicating the conditions for the WMI query launched by each module generated by the component. It accepts the use of macros with the names of the query felds (`_FIELDNAME_`) to obtain the value of each field log. For instance:

```
DriveType = 3 AND DeviceID = '_DeviceID_'.
```

The query is the final execution of a module generated by the component to obtain the free space of the `C:` disk unit is:

```
SELECT DeviceID, FreeSpace FROM Win32_LogicalDisk WHERE DriveType = 3 AND
DeviceID = 'C:'
```

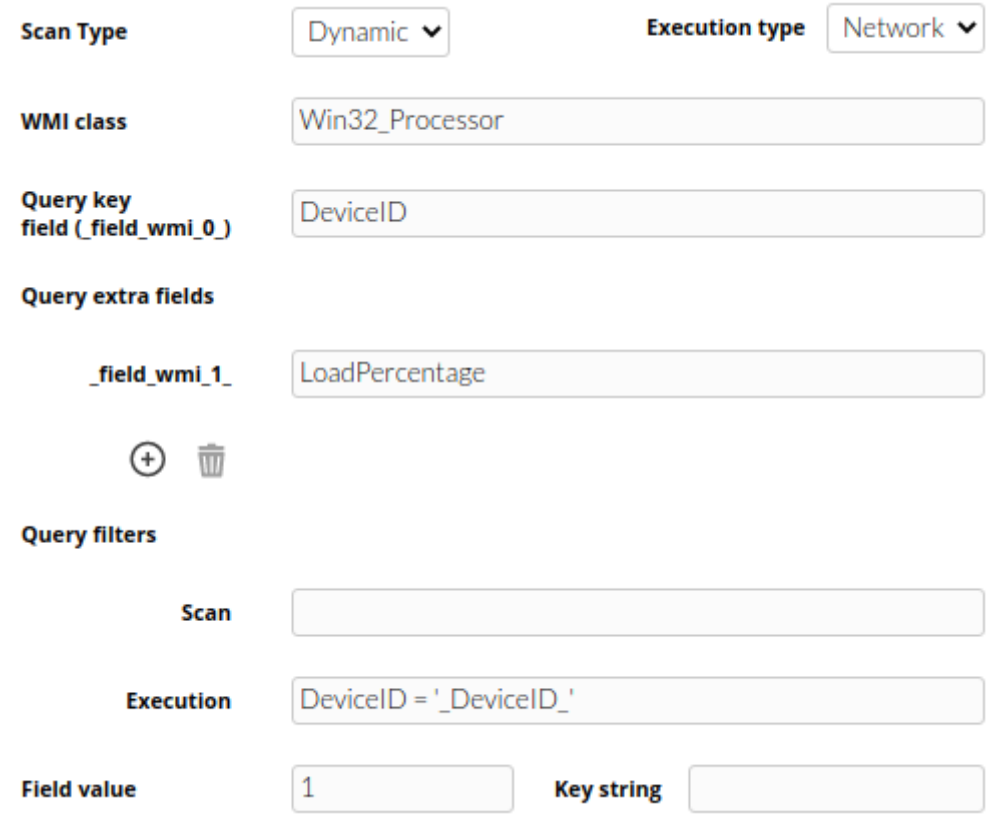
Field value

Indicate the number of the field of the WMI query from which you want to obtain the module value. Field 0 is the key field of its class and fields 1, and higher, the additional ones of their class.

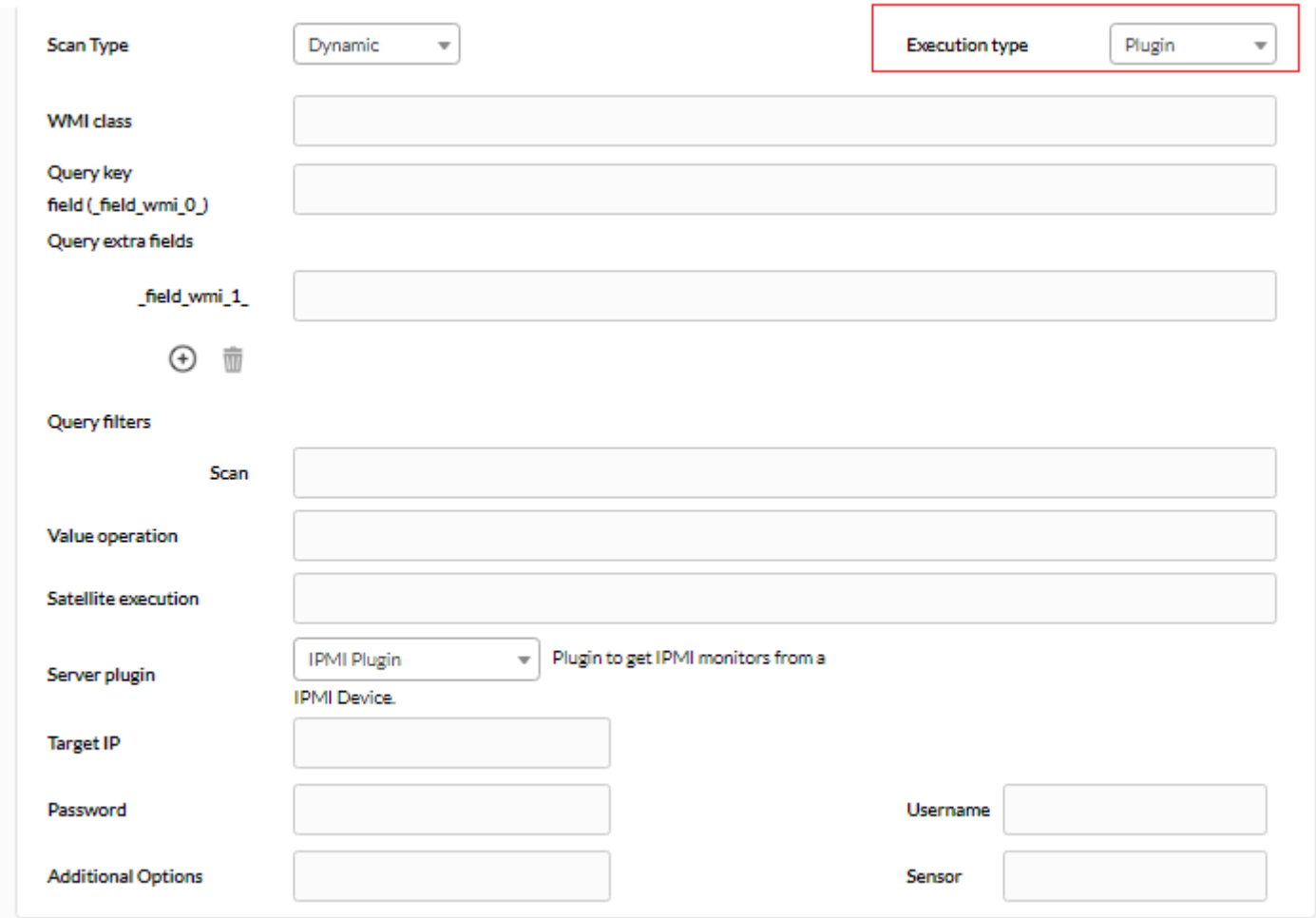
Key string

It will allow to convert the module value into boolean (1 or 0) depending on whether the value of the field indicated in *Field value* matches the text string indicated in this field.

The Key string option will not be taken into account when the wizard is launched from a Satellite Server by exec server



WMI Plugin execution



When there is a plugin execution:

It has common fields with Network execution until field Scan (Query filters).

Value operation

The main purpose of using plugin type components is to be able to perform operations with the values of different query fields, such as obtaining the used-disk percentage from the free-disk bytes and the total disk bytes available.

This field allows to indicate an arithmetic operation with which you will obtain the current value of each module generated by the component in the wizard's module preview. It does not affect in any case the final execution of the generated modules.

It accepts the characters `+ - * / ( ) .`, numbers and the macros with the names of the fields this type (`_FIELDNAME_`), from which the values for the operation will be obtained. For example:

```
((_Size_ - _FreeSpace_) * 100) / _Size_
```

Satellite execution

It allows to indicate the execution that a Satellite Server should perform for the generated modules when the wizard is launched from a Satellite Server by using the *exec server*. It is the command, plugin or script that must be used in a Satellite server `module_exec`.

It accepts the use of macros for the WMI wizard and of `_class_wmi_` macros to obtain the name of the WMI class and `_field_wmi_N_` to obtain the names of the fields of the class used in each module.

The Satellite Server distributes a recommended plugin for these components:

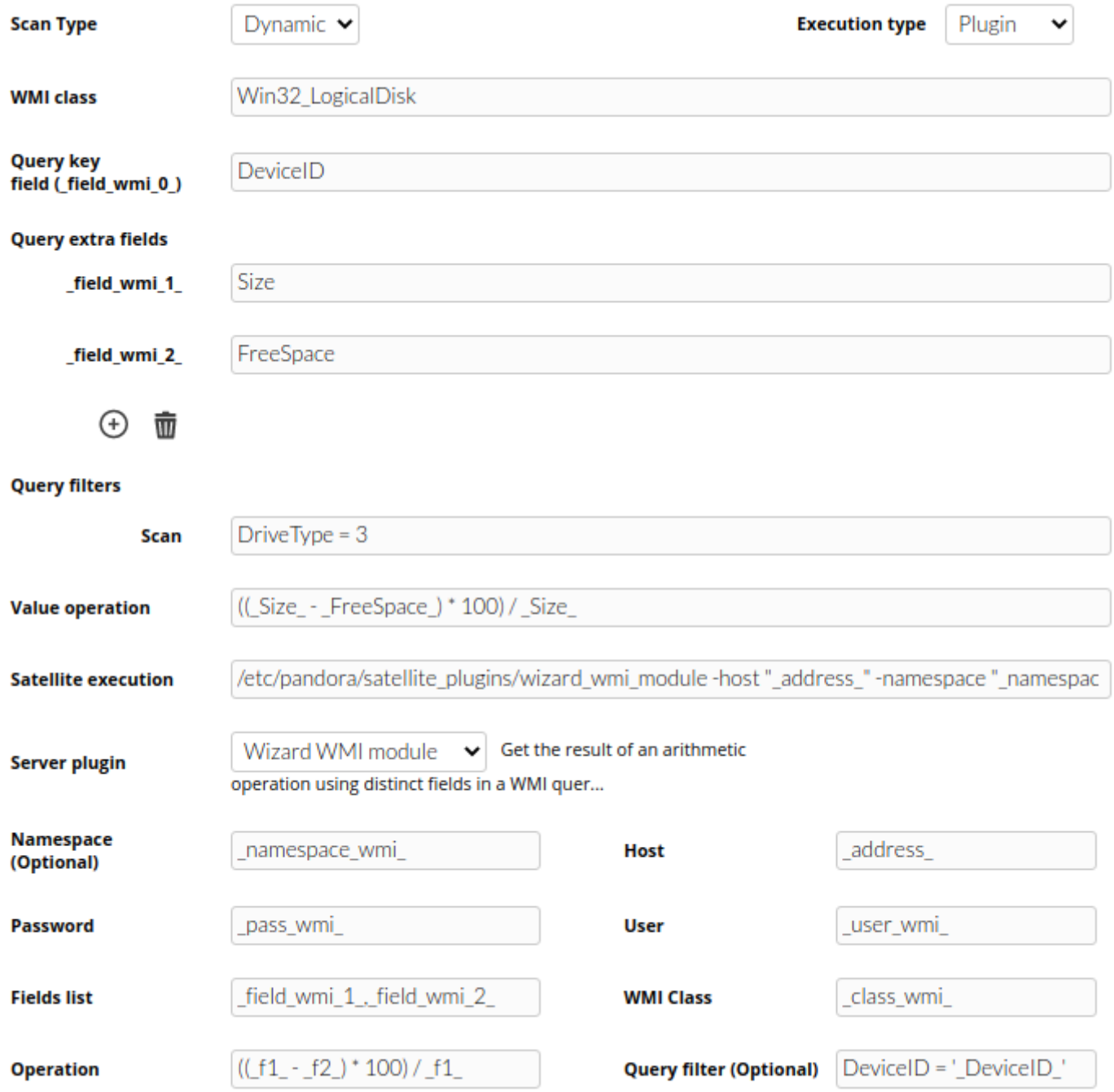
```
/etc/pandora/satellite_plugins/wizard_wmi_module.
```

Server plugin

It allows to indicate a plugin registered in Pandora FMS console that will be used by the plugin server with each module generated by the component. The choice of a plugin shows at the same time the specific fields for its use in the form.

The own plugin fields accept the use of macros for the WMI wizard and `_class_wmi_` macros to get the name of the WMI class and `_field_wmi_N_` to get the names of the fields of the class used in each module.

Pandora FMS console has an already registered and recommended plugin for these components: Wizard WMI module.



The specific macros for the WMI wizard components that can be used in the plugin execution fields are

- *_address_* : IP address used in the WMI wizard. This macro will not be replaced when the wizard is launched in a policy.
- *_namespace_wmi_* : Namespace used in the WMI wizard.
- *_user_wmi_* : User used in the WMI wizard.
- *_pass_wmi_* : Password used in the WMI wizard.

The specified fields for WMI Wizard components are:

```
* **WMI class**: It refers to the WMI class that will be used in the queries of the modules generated by the component. For example: //Win32_LogicalDisk//.
```

It can be used in other fields from the same form through the macro `_class_wmi_`.

Query key field (`_field_wmi_0_`)

It is the name of the key field that will be obtained in the WMI query used in the generated modules. Usually, WMI classes have a key field they always return in any query, whether indicated or not. That is the field that should be indicated here. For instance, the key field of class *Win32_Processor* would be *DeviceID*.

the name of this field can be obtained in other form fields through the macro `_field_wmi_0_`, and the value the field has for the WMI query log can be obtained through a macro with the same field name (`_FIELDNAME_`). These macros `_FIELDNAME_` can be used, among others, in component Module name and Description fields, to generate names and descriptions dinamically. For instance, for field *DeviceID* the macro with value would be `_DeviceID_`.

Query extra fields → `_field_wmi_N_`:

In these fields indicate the names of the additional fields that must be used in the WMI query used in the generated modules.

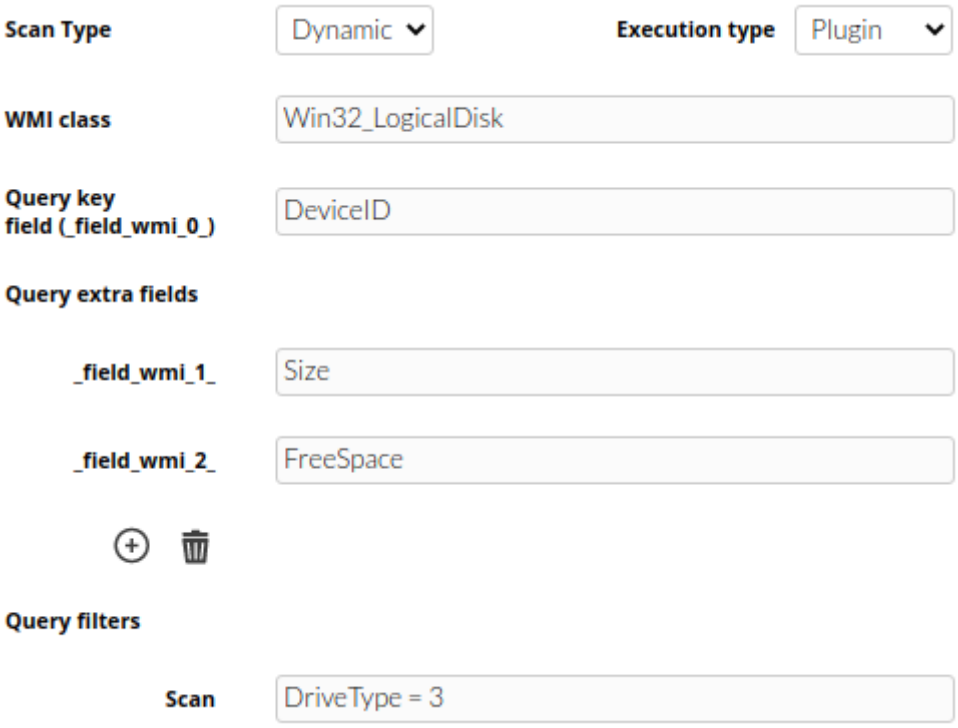
The names of these fields can be obtained in other form fields through the macros `_field_wmi_N_`, and the values that the fields for each WMI query log have can be obtained through macros with the same names as those of the fields (`_FIELDNAME_`). These macros `_FIELDNAME_` can be used, among others, in Module name and Description component fields, to generate names and descriptions dinamically. For example for the field FreeSpace the macro with the value would be `_FreeSpace_`.

Query filters → Scan

In this space the conditions for the WMI query launched in the scan are indicated, which will allow you to obtain one or more logs. E.g.: `DriveType = 3`.

In WMI wizard components, a different module is generated for each log returned by the WMI scan query. Based on the examples pruvided up to now, the scan query would obtain the free space of the disk units from the Windows computer:

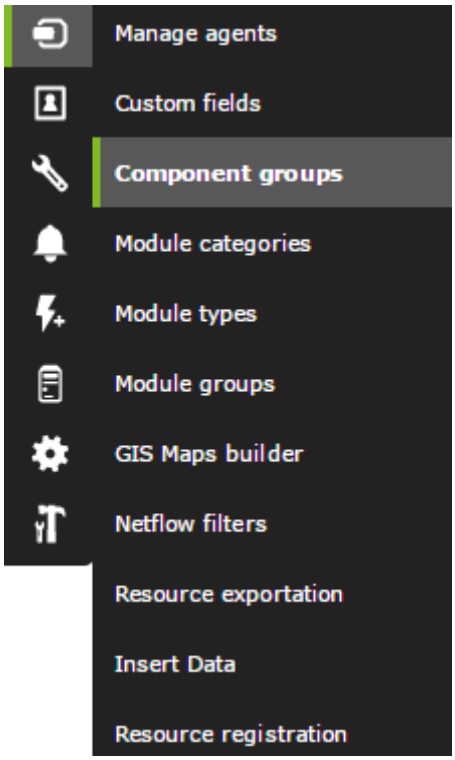
```
SELECT DeviceID, FreeSpace FROM Win32_LogicalDisk WHERE DriveType = 3
```



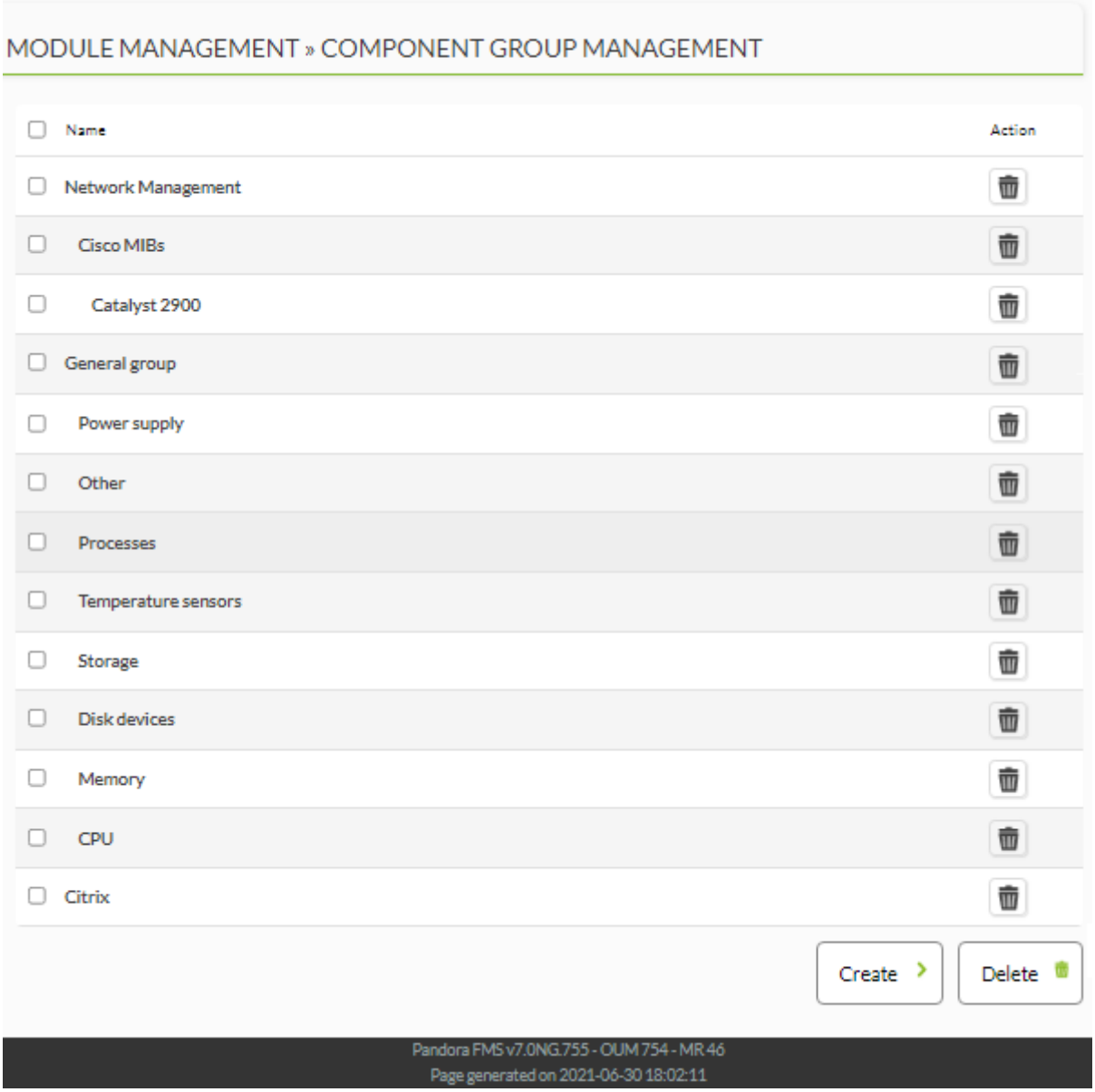
Component Groups

In order to help sorting and classifying components, component groups have been created. Components are associated to groups when created.

In order to see the existing component groups, go to Resources → Component groups:

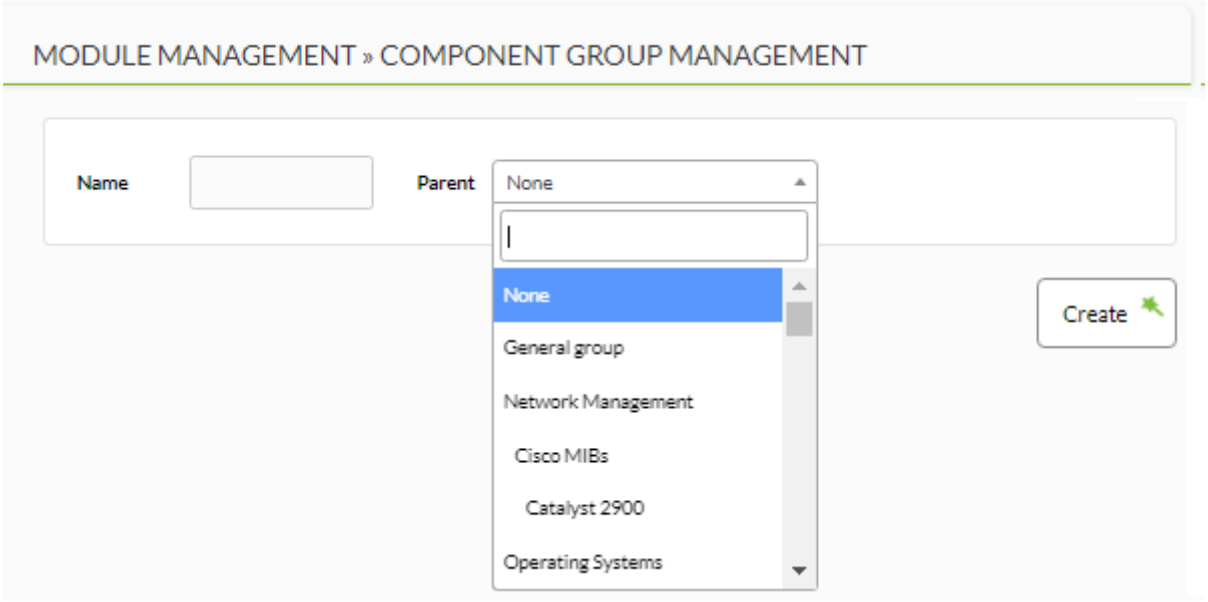


The already existing groups and their description is shown on screen:



You may view the details on the groups by clicking on their name, delete them by clicking on the trash icon at the right side, on the Action column, or create new ones by clicking Create at the bottom. you may also select several of them by their respective selection checkbox and clicking Delete.

If you intend to create a new components group, click Create and fill out the form fields.



Just provide a name for the group and determine whether it has a parent among the existing groups or not. Then click on Create once you are done.

Add as many new components to your newly created component group as you like.

Go back to Pandora FMS documentation index