



Wizard Components



<https://pandorafms.com/manual/!775/>

Permanent link:

https://pandorafms.com/manual/!775/en/documentation/pandorafms/technical_annexes/25_pfms_new_wizard_component

2014/03/18 21:03



Wizard Components

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

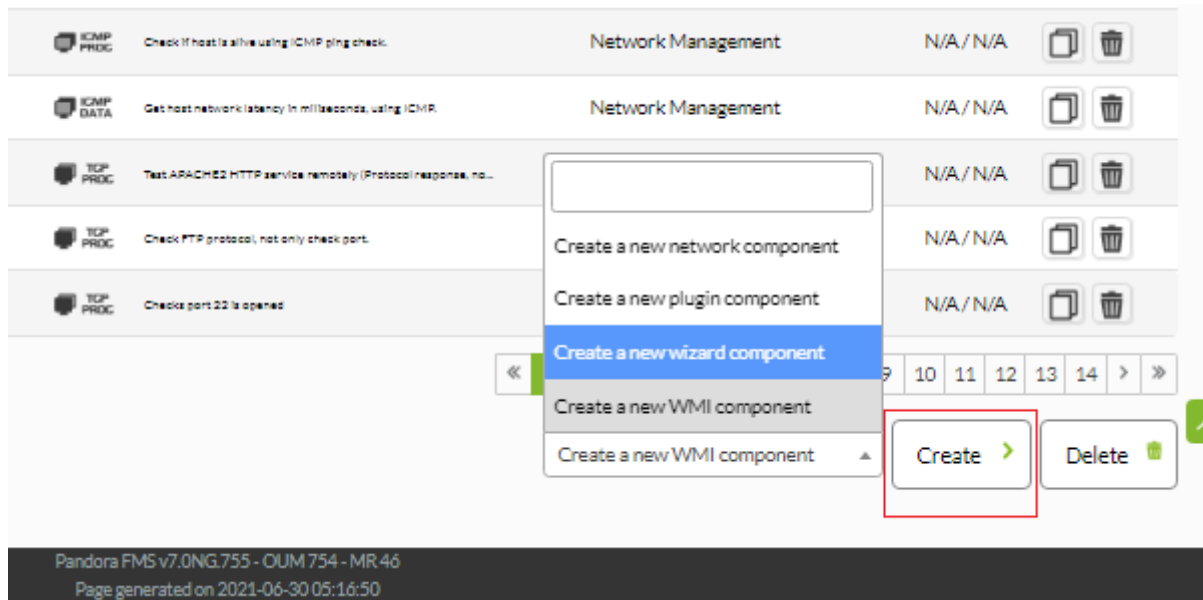
Within the capabilities of the SNMP wizard and the WMI wizard, there is a type of remote components called Wizard Components.

These components allow establishing a base configuration for the modules that will be generated in the agents when executing any of the wizards (SNMP or WMI). It also offers the possibility that with a single component several modules can be generated dynamically. For example, a component to scan the different storage units of a device or the running processes.

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

The screenshot displays the Pandora FMS Enterprise web interface. At the top left is the Pandora FMS logo. The main header area includes the text "Pandora FMS the Flexible Monitoring System" and a search bar containing "Net". Below the search bar, a dropdown menu is open, showing "Go to" followed by a wrench icon and "Network Component". The main content area is titled "MODULE MANAGEMENT » REMOTE COMPONENT MANA". Below this title, there is a "Group" dropdown menu set to "All" and a "Free Search" field with an information icon. A large configuration menu is overlaid on the right side of the screen, listing various options such as "Templates", "Inventory modules", "Manage agent autoconfiguration", "Software agents repository", "Manage policies", "Collections", "Duplicate configuration", "Bulk operations", and "Credential store". The "Remote components" option is highlighted, and a sub-menu is visible showing several component types, including "GNMP DATA MIB", "GNMP DATA CPU Load in Operating System (UNIX MIB)", "GNMP TEXT Get name of system using SNMP standard MIB", and "GNMP DATA Active users in Operating System (UNIX MIB)". At the bottom of the interface, a footer contains the text "Pandora FMS v7.0NG.755 - OUM 754 - MR 46" and "Page generated on 2021-06-25 04:00:50".

Select the Create a new wizard component option and press the Create button.



In the creation dialog there is a vitally important option: the module protocol to use. There are two protocols available, SNMP (default) and WMI, and although both share common fields they also differ in other fields.

MODULE MANAGEMENT » REMOTE COMPONENT MANAGEMENT ?

Enabled

Module name

Name OID

Module type

Module unit

Add by default

Module protocol

Manufacturer ID

Component Group

The 'Module protocol' dropdown menu is highlighted with a red box, showing options for 'SNMP' (selected) and 'WMI'.

The common fields are:

- **Enabled:** Activating this token indicates that the component will try to scan when launching the wizard.
- **Add by default:** Allows you to choose whether the modules generated by the component will be marked to be added by default when launching the wizard. That is, if the token is activated, the modules generated by the component will be marked by default in a view that we will find later and will be added to the Agent. Then in that 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. You can make use of some macros that are explained in the following sections.
- **Module protocol:** Allows you to choose between SNMP and WMI (some fields change).
- **Module type:** In this dropdown 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. Allows you 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 fully editable field, so you can add the measurement you need.
- **Warning status:** In this section you can establish a default threshold for the warning status of the modules generated by the wizard. Despite indicating a range here, there will be the possibility of customizing it for each module in the final view that includes all the modules found.
- **Critical status:** In this section you can set a default threshold for the critical status of the modules generated by the wizard. Despite indicating a range here, there will be the possibility of customizing it for each module in the final view that includes all the modules found.
- **Description:** Description that the component will have and, in turn, the modules generated by it. Some macros may be used.
- **Scan type:** Allows you to choose between two scanning modalities that wizards can perform with this component. This field determines if a component will generate a single module or several. The selected value will affect how other specific fields of each wizard should be filled.
 - **Fixed:** The component will only generate a module. For example, get the uptime (time running) of the device by SNMP.
 - **Dynamic:** The component may generate one or more modules. For example, get the percentage of disk drive usage by WMI.
- **Execution type:** This field indicates the type of execution for the modules generated by the component. It is used to determine the Pandora FMS server to which the modules will belong at the moment of their creation depending on where the wizard is executed from.
 - **Network:** The modules generated by the component will obtain their data with the mechanisms of Pandora FMS for SNMP and WMI modules. These are: the [network servers](#), [WMI server](#) and [satellite server](#).
 - **Plugin:** The modules generated by the component will obtain their data from the execution of commands, plugins or custom scripts. In this way, they will be executed by the [plugins server](#) or by the satellite server through exec type modules.

SNMP wizard

MODULE MANAGEMENT » REMOTE COMPONENT MANAGEMENT ?

Enabled Add by default

Module name Module protocol

Name OID Manufacturer ID

Module type Component Group

Module unit

Warning Min. Max. Critical Min. Max.

Inverse interval Inverse interval

Description

Scan Type Execution type

Value OID

Go back ✖ Create ✖

Pandora FMS v7.0NG.755 - OUM 754 - MR 46
Page generated on 2021-06-30 05:23:29

The specific fields for components of the SNMP wizard are:

- Name OID
- Manufacturer ID
- When Execution type is set to Network:
 - Value OID
- When Execution type is set to 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

Warning

Min. Min. Max.

Critical


Max. 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 

Allows you to indicate an OID from which a value will be obtained that can be added to the module name by means of a macro. It is especially useful when getting multiple modules generated by a dynamic component. In this way it is achieved that they have different names by default. But it is not limited to dynamic components as it can be used for fixed scan components as well.

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

If used in dynamic components, the OID indicated in this field must be an SNMP branch 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 endings that the branch can have.

If used in constructs, 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 that the macro will have in the module will be obtained directly from that OID.

Manufacturer ID

Allows you to indicate the ID of a specific manufacturer for which the SNMP wizard component will take effect. In this way, for all the 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 try to obtain the modules generated by it. For example, a component assigned to `general_snmp`, will be scanned for all devices with PEN 2021 and 8072.

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

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

SNMP Network type execution

When the execution type is Network:

Value OID:

Allows you to indicate the OID from which the data of the modules generated by the component will be obtained. If used in dynamic components, the OID indicated in this field must be an SNMP branch 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`. Also, the x node of each OID must have the same value for the X node of the Name OID field. in the case of being used.

If used in constructs, 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.

Scan Type	Fixed ▼	Execution type	Network ▼
Value OID	1.3.6.1.4.1.2021.11.9.0		

SNMP plugin type execution

When the execution type is Plugin:

OID Macros → `_oid_N_`

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

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

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

If used in dynamic components, the OIDs listed in these fields must be an SNMP branch 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 constructs, the OIDs listed 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



Allows you to indicate an arithmetic operation with which the current value of each module generated by the component in the preview of the wizard modules will be obtained. In no case does it affect:
`wiki:pfms-configuration-templates-remote_components-create_new_wizard-module_protocol-snmp.png?600x650 }`

The specific fields for components of the SNMP wizard are:

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

MODULE MANAGEMENT » REMOTE COMPONENT MANAGEMENT ?

Enabled	<input checked="" type="checkbox"/>	Add by default	<input checked="" type="checkbox"/>
Module name	<input type="text"/>	Module protocol	SNMP 
Name OID	<input type="text"/>	Manufacturer ID	All <input type="text"/>
Module type	Boolean <input type="text"/>	Component Group	General group <input type="text"/>
Module unit	<input type="text"/> 		
Warning	Min. <input type="text"/> 0 Max. <input type="text"/> 0	Critical	Min. <input type="text"/> 0 Max. <input type="text"/> 0
	Inverse interval <input type="checkbox"/>		Inverse interval <input type="checkbox"/>
Description	<input type="text"/>		
Scan Type	Fixed <input type="text"/>	Execution type	Plugin <input type="text"/>
Macros OID	<input type="text"/>		
	<input type="text"/>		
	<input type="text"/>		
Value operation	<input type="text"/>		
Satellite execution	<input type="text"/>		
Server plugin	IPMI Plugin <input type="text"/>	Plugin to get IPMI monitors from a IPMI Device.	
Target IP	<input type="text"/>		
Password	<input type="text"/>	Username	<input type="text"/>
Additional Options	<input type="text"/>	Sensor	<input type="text"/>

Go back Create 

Allows you to indicate an OID from which a value will be obtained that can be added to the module name by means of a macro. It is especially useful when getting multiple modules generated by a dynamic component. In this way it is achieved that they have different names by default. But it is not limited to dynamic components as it can be used for fixed scan components as well.

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

If used in dynamic components, the OID indicated in this field must be an SNMP branch 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 endings that the branch can have.

If used in constructs, 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 that the macro will have in the module will be obtained directly from that OID.

Manufacturer ID

Allows you to indicate the ID of a specific manufacturer for which the SNMP wizard component will take effect. In this way, for all the 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 try to obtain the modules generated by it. For example, a component assigned to `general_snmp`, will be scanned for all devices with PEN 2021 and 8072.

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

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

SNMP Network type execution

When the execution type is Network:

Value OID:

Allows you to indicate the OID from which the data of the modules generated by the component will be obtained. If used in dynamic components, the OID indicated in this field must be an SNMP branch 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`. Also, the x node of each OID must have the same value for the X node of the Name OID field. in the case of being used.

If used in constructs, 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.

Scan Type	Fixed ▼	Execution type	Network ▼
Value OID	1.3.6.1.4.1.2021.11.9.0		

SNMP plugin type execution

When the execution type is Plugin:

OID Macros → `_oid_N_`

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

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

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

If used in dynamic components, the OIDs listed in these fields must be an SNMP branch 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 constructs, the OIDs listed 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

Allows you to indicate an arithmetic operation with which the current value of each module generated by the component in the preview of the wizard modules will be obtained. In no case does it affect to final execution of the generated modules.

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

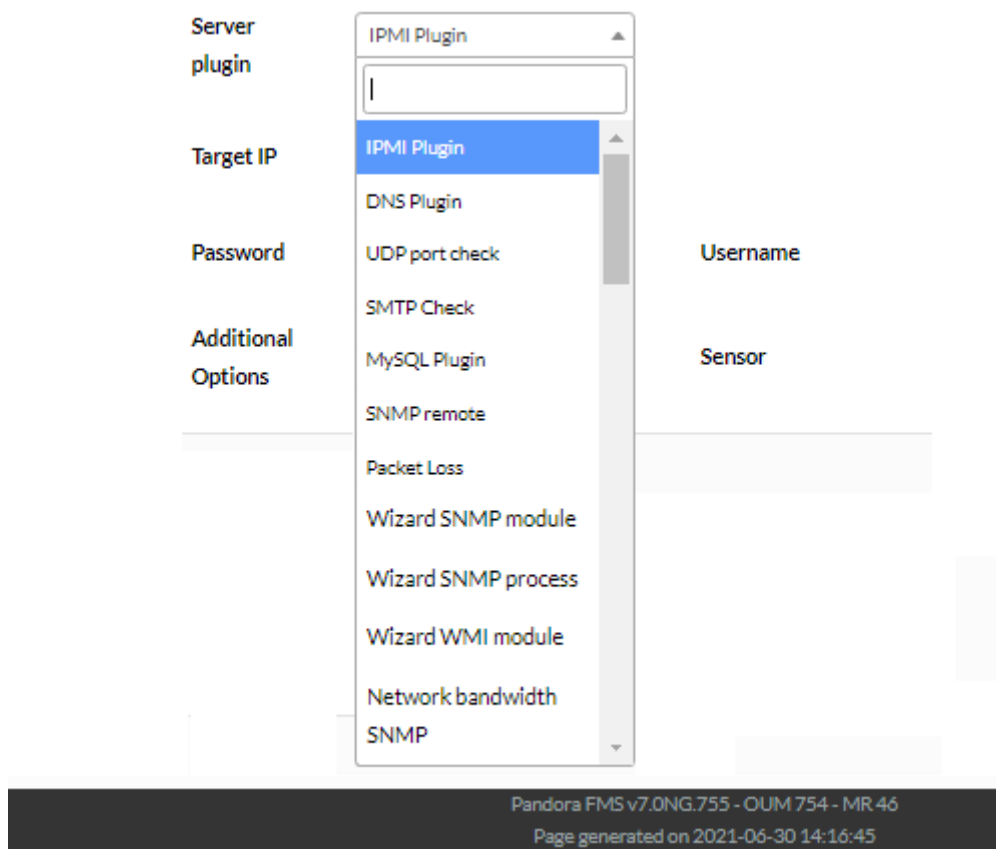
It offers the possibility of indicating the execution that a Satellite Server should carry out for the modules generated when the wizard is launched from a Satellite Server by using the exec server. This is the command, plugin, or script to be used in a satellite server module_exec.

It accepts the use of the macros for the SNMP wizard (will be detailed later) and the `_oid_N_` macros to obtain the OID 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 drop-down it will be possible to indicate a plugin previously registered in the Pandora FMS Console, which will be used by the plugin server with each module generated by the component. Choosing a plugin shows the specific fields for its use in the form.

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

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

- Wizard SNMP module.
- Wizard SNMP process.

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

Scan Type	<input type="text" value="Fixed"/>	Execution type	<input type="text" value="Plugin"/>
Macros OID			
oid 1	<input type="text" value="1.3.6.1.4.1.2021.4.6.0"/>		
oid 2	<input type="text" value="1.3.6.1.4.1.2021.4.5.0"/>		
 			
Value operation	<input type="text" value="(_oid_1_ * 100) / _oid_2_"/>		
Satellite execution	<input "_cc"="" "_port_"="" "_version_"="" -community="" -port="" -version="" _address_"="" type="text" value="/etc/pandora/satellite_plugins/wizard_snmp_module -host "/>		
Server plugin	<input type="text" value="Wizard SNMP module"/> Get the result of an arithmetic operation using several OIDs values.		
Port	<input type="text" value="_port_"/>	Host	<input type="text" value="_address_"/>
Community	<input type="text" value="_community_"/>	Version	<input type="text" value="_version_"/>
Username (v3)	<input type="text" value="_auth_user_"/>	Security level (v3)	<input type="text" value="_sec_level_"/>
Authentication password (v3)	<input type="text" value="_auth_pass_"/>	Authentication method (v3)	<input type="text" value="_auth_method_"/>
Privacy password (v3)	<input type="text" value="_priv_pass_"/>	Privacy method (v3)	<input type="text" value="_priv_method_"/>
Operation	<input type="text" value="(_o1_ * 100) / _o2_"/>	OID list	<input type="text" value="_oid_1_ _oid_2_"/>

The specific macros for the SNMP wizard components that can be used in the plugin 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 the values 1, 2c or 3.
- **_community_** : SNMP community used in the SNMP wizard.
- **_sec_level_** : SNMPv3 security level used in the SNMP wizard. Can have the 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 can have the values MD5 or SHA.
- `_auth_pass_` : SNMPv3 authentication password used in the SNMP wizard.
- `_priv_method_` : SNMPv3 privacy method used in the SNMP wizard. It can have the values DES or AES.
- `_priv_pass_` : SNMPv3 privacy password used in the SNMP wizard.

WMI wizard

MODULE MANAGEMENT » REMOTE COMPONENT MANAGEMENT ?

Enabled

Add by default

Module name

Module protocol WMI ⌵ 🇺🇸

Module type Boolean ⌵

Module unit 📄

Component Group General group ⌵

Warning Min. 0 Max. 0 Critical Min. 0 Max. 0

Inverse interval *Inverse interval*

Description

Scan Type Fixed ⌵ Execution type Network ⌵

WMI class

Query key field (`_field_wmi_0_`)

Query extra fields

`_field_wmi_1_`

Query filters ⊕ 🗑️

Scan

Execution

Field value Key string

Go back ✖
Create ✚

Pandora FMS v7.0NG.755 - OUM 754 - MR 46

Page generated on 2021-06-30 05:23:29

WMI execution type Network

When the execution type is Network:

WMI class

Allows you to specify the WMI class to query. WMI classes are predefined that are included in every Windows Management Instrumentation® (WMI) kernel namespace.

Query key field (`_field_wmi_0_`)

Query extra fields (`_field_wmi_1_`)

Query filters → Execution:

This space allows you to indicate the conditions for the WMI query launched by each module generated by the component. Accepts the use of macros with the names of the query fields (`_FIELDNAME_`) to get the value of each record in that field. For example:

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

The final execution query of a module generated by the component to get the free space on drive C: is:

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

Field value

We will indicate the number of the WMI query field from which we want to obtain the value of the module, with field 0 being the key field of the class and fields 1 and higher being the additional fields of the class.

Keystring

Allows you to convert the module value to a 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 be ignored when the wizard is launched from a Satellite Server via exec server

Scan Type	Dynamic ▾	Execution type	Network ▾
WMI class	Win32_Processor		
Query key field (_field_wmi_0_)	DeviceID		
Query extra fields			
_field_wmi_1_	LoadPercentage		
	⊕ 🗑		
Query filters			
Scan			
Execution	DeviceID = '_DeviceID_'		
Field value	1	Key string	

WMI Plugin type execution

Scan Type	Dynamic ▾	Execution type	Plugin ▾
WMI class			
Query key field (_field_wmi_0_)			
Query extra fields			
_field_wmi_1_			
	⊕ 🗑		
Query filters			
Scan			
Value operation			
Satellite execution			
Server plugin	IPMI Plugin ▾	Plugin to get IPMI monitors from a	
	IPMI Device.		
Target IP			
Password		Username	
Additional Options		Sensor	

When the execution type is Plugin:

Share common fields with Network execution up to the Scan field (Query filters).

Value operation

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

This field allows you to indicate an arithmetic operation with which the current value of each module generated by the component in the preview of the wizard modules will be obtained. In no case does it affect the final execution of the generated modules.

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

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

Satellite execution

Allows you to indicate the execution that a Satellite Server should carry out for the modules generated when the wizard is launched from a Satellite Server by using the exec server. This is the command, plugin or script to be used in a `module_exec` of the Satellite Server.

It accepts the use of the macros for the WMI wizard and the macros `_class_wmi_` 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.

The Satellite Server distributes a recommended plugin for these components:

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

Server plugin

Allows you to indicate a plugin registered in the Pandora console that will be used by the plugin server with each module generated by the component. Choosing a plugin shows the specific fields for its use in the form.

The plugin's own fields accept the use of the macros for the WMI wizard and the macros `_class_wmi_` 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 Pandora FMS Console has a plugin already registered recommended for these components: Wizard WMI module.

Scan Type	<input type="text" value="Dynamic"/>	Execution type	<input type="text" value="Plugin"/>
WMI class	<input type="text" value="Win32_LogicalDisk"/>		
Query key field (_field_wmi_0_)	<input type="text" value="DeviceID"/>		
Query extra fields			
_field_wmi_1_	<input type="text" value="Size"/>		
_field_wmi_2_	<input type="text" value="FreeSpace"/>		
	<input type="button" value="+"/> <input type="button" value="🗑"/>		
Query filters			
Scan	<input type="text" value="DriveType = 3"/>		
Value operation	<input type="text" value="((_Size_ - _FreeSpace_) * 100) / _Size_"/>		
Satellite execution	<input "_namespac"="" -namespace="" _address_"="" type="text" value="/etc/pandora/satellite_plugins/wizard_wmi_module -host "/>		
Server plugin	<input type="text" value="Wizard WMI module"/>	Get the result of an arithmetic operation using distinct fields in a WMI quer...	
Namespace (Optional)	<input type="text" value="_namespace_wmi_"/>	Host	<input type="text" value="_address_"/>
Password	<input type="text" value="_pass_wmi_"/>	User	<input type="text" value="_user_wmi_"/>
Fields list	<input type="text" value="_field_wmi_1_ _field_wmi_2_"/>	WMI Class	<input type="text" value="_class_wmi_"/>
Operation	<input type="text" value="((_f1_ - _f2_) * 100) / _f1_"/>	Query filter (Optional)	<input type="text" value="DeviceID = '_DeviceID_'"/>

The specific macros for the WMI wizard components that can be used in the fields of plugin type executions 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 specific 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 of this same form through the `_class_wmi_` macro.

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. WMI classes generally have a key field that is always returned in any query whether it is specified or not. That would be the field that should be indicated here. For example, the key field of the `Win32_Processor` class would be `DeviceID`.

The name of this field can be obtained from other fields on the form using the `_field_wmi_0_` macro, and the value of the field for each record in the WMI query can be obtained from a macro with the same field name (`_FIELDNAME_`). These `_FIELDNAME_` macros can be used, among others, in the Module name and Description fields of the component, to generate names and descriptions dynamically. For example, for the `DeviceID` field the macro with the value would be `_DeviceID_`.

Query extra fields → `_field_wmi_N_`:

In these fields we will 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 from other fields on the form using the `_field_wmi_N_` macros, and the values that the fields have for each record in the WMI query can be obtained from macros with the same field names (`_FIELDNAME_`). These `_FIELDNAME_` macros can be used, among others, in the Module name and Description fields of the component, to generate names and descriptions dynamically. For example, for the field `FreeSpace` the macro with the value would be `_FreeSpace_`.

Query filters → Scan

This space indicates the conditions for the WMI query launched in the scan, which will allow one or more records to be obtained. For example: `DriveType = 3`.

In the WMI wizard components, a separate module will be generated for each record returned by the WMI scan query. Based on the examples provided so far, the query that would be performed would obtain the free space of the disk drives of the Windows computer:

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

Scan Type	Dynamic ▾	Execution type	Plugin ▾
WMI class	Win32_LogicalDisk		
Query key field (_field_wmi_0_)	DeviceID		
Query extra fields			
_field_wmi_1_	Size		
_field_wmi_2_	FreeSpace		
	 		
Query filters			
Scan	DriveType = 3		

[Back to Pandora FMS Documentation Index](#)