# PANDORAFMS

Wizard Components

Wizard Components

Within the capabilities of the wizard SNMP and wizard WMI, there are remote component types 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 Wizard (SNMP or WMI). It also offers the possibility of dynamically generating several modules with a single component.

To create these components use the menu Management → Configuration → Templates → Remote components. The Create a new wizard component option is selected and the Create button is clicked.
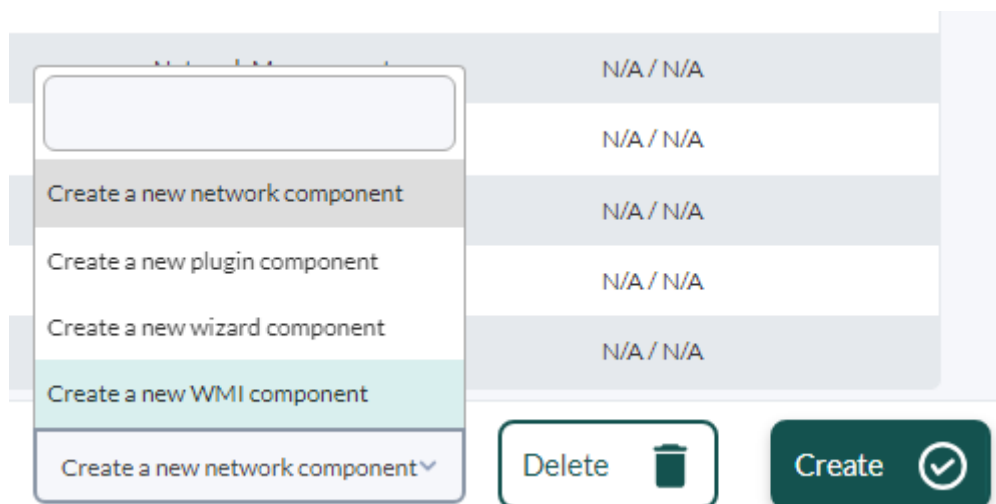
In the creation dialog box there is an important option: the Module protocol to be used. There are two protocols available, SNMP (default) and WMI, and although both share common fields they also differ in other fields.

Common fields

- Enabled: Enabling this *token* indicates that the component will attempt to scan when launching the *wizard*.
- Add by default: Allows to choose if the modules generated by the component will be marked to be added by default when launching the *wizard*.
- Module name: Name that the component will have, and default name for the modules generated by it. Some macros can be used.
- Module type: Type of data to be obtained by the modules generated by the component.
- 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.
- Warning: In this section you can set a default threshold for the warning status of the modules generated by the *wizard*.
- Critical status: In this section you can set a default threshold for the critical state of the modules generated by the *wizard*.
- Description: Description of the component and the modules generated by it. Some macros can be used.

Scan type: Allows you to choose between two exploration modes that *wizards* will be able to perform with this component:

- Fixed: The component will generate only one module.
- Dynamic: The component may generate one or more modules.

Execution type: This field indicates the execution type 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 Pandora FMS own mechanisms for SNMP and WMI modules. These are: the network servers, WMI server y satellite server.
- Plugin: The modules generated by the component will obtain their data from the execution of custom commands, *plugins* or *scripts*. In this way, they will be executed by the plug-in server or by the

satellite server through modules of type `exec`.

SNMP wizard

Management → Configuration → Templates → Remote components → Create a new wizard component → Create → Module protocol → SNMP menu.

Specific fields:

- Name OID: Allows to indicate an OID from which a value will be obtained that can be added to the module name by means of a macro. The value of this OID is stored in the macro _nameOID_, that can be used in the Module name field. If All is indicated as manufacturer, the component will be scanned for any PEN registered in Pandora FMS.
- Manufacturer ID: Allows to indicate the ID of a specific manufacturer for which the *wizard* SNMP component will take effect.
- When Execution type is established in Network: Value OID.
- When Execution type is established in Plugin: Macros OID, Value operation, Satellite execution, Server plugin.

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

SNMP Network execution

Management → Configuration → Templates → Remote components → Create a new wizard component → Create → Module protocol → SNMP (Execution type → Network) menu.

When the execution type is Network the Value OID allows 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 an end OID.
- If used on fixed components, the OID indicated in this field must be a final OID.

SNMP Plugin execution

Management → Configuration → Templates → Remote components → Create a new wizard component → Create → Module protocol → SNMP (Execution type → Plugin) menu.

The main purpose of using components of type *plugin* is to be able to perform operations with the values of one or more OIDs in the same device. That is why in these components you can indicate as many OIDs as you need to use them in other fields.

Macros OID → _oid_N_

In addition, 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, either the OID value or the OID itself will be used.

If used on dynamic components, the OIDs indicated in these fields must be an SNMP branch and not an end OID. If used on fixed components, the OIDs indicated in these fields must be an end OID.

Value operation

Allows you to specify an arithmetic operation with which the current value of each module generated by the component will be obtained in the *wizard* module preview. In any case it does not affect the final execution of the generated modules.

Accepts the characters + - * / ( ) ., numbers and macros _oid_N_ from which the values for the operation will be obtained.

Satellite execution

It offers the possibility of indicating the execution to be performed by a Satellite Server for the modules generated when the *wizard* is executed 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. Accepts the use of macros for the SNMP wizard and _oid_N_ macros to obtain the OIDs used in each module.

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

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

Server plugin

Through this list you can indicate a *plugin* previously registered in the Pandora FMS Console, which will be used by the plugins server with each module generated by the component. The choice of a *plugin* shows in turn in the form the specific fields to use it.

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

Pandora FMS console has already registered plugins recommended for these components: Wizard SNMP module and Wizard SNMP process. Depending on the use of one or the other, the rest of the respective fields should be completed to carry out the monitoring through SNMP.

WMI wizard

Management → Configuration → Templates → Remote components → Create a new wizard component → Create → Module protocol → WMI menu.

WMI Network execution

Management → Configuration → Templates → Remote components → Create a new wizard component → Create → Module protocol → WMI (Execution type → Network) menu.

When the execution type is Network:

- WMI class: Allows you to specify the WMI class to query. WMI classes are predefined classes that are included in each Windows Management Instrumentation® (WMI) kernel namespace.
- Query filters → Execution: This space allows to indicate 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 fields (_FIELDNAME_) to obtain the value of each record in that field.
- Field value: To indicate the number of the WMI query field from which the module value is to be obtained, where field 0 is the key field of the class and fields 1 and above are the additional fields of the class.
- Key string: Allows to convert the module value to *boolean* (1 or 0) depending on whether the value of the field indicated in *Field value* matches the string indicated in this field.

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

WMI Plugin execution

Management → Configuration → Templates → Remote components → Create a new wizard component → Create → Module protocol → WMI (Execution type → Plugin) menu.

When the execution type is Plugin:

- It shares common fields with the Network run up to the Query filters (Scan) field.
- Value operation: This field allows you to specify an arithmetic operation to obtain the current value of each module generated by the component in the *wizard* module preview. In any case it does not 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.
- Satellite execution: Allows to indicate the execution to be performed by a Satellite Server 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 WMI wizard and the macros _class_wmi_ to get the WMI class name and _field_wmi_N_ to get the names of the class fields 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 the Pandora FMS console that will be used by

the *plugins* server with each module generated by the component. The choice of a *plugin* shows at the same time in the form the specific fields to use it.

Wizard WMI module

Pandora FMS Console has a *plugin* already registered recommended for these components: Wizard WMI module.

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

Value operation

((_Size_ - _FreeSpace_) * 100) / _Size_

Satellite execution

/etc/pandora/satellite_plugins/wizard_wmi_module -host "_address_" -namespace "_namespace_wmi_" -user "_user_wm

Server plugin

Wizard WMI module

Get the result of an arithmetic operation using distinct fields in a WMI quer...

Host

address

Namespace (Optional)

_namespace_wmi_

User

_user_wmi_

Password

_pass_wmi_

WMI Class

_class_wmi_

Fields list

_field_wmi_1_,_field_wmi_2_

Query filter (Optional)

DeviceID = '_DeviceID_'

Operation

((_f1_ - _f2_) * 100) / _f1_

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

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 to be used in the queries of the modules generated by the component. For example, *Win32_LogicalDisk*.
- They can be used in other fields of this same form by means of 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 that they always return in any query whether it is indicated or not. That would be the field that should be indicated here. For example, the key field of the *Win32_Processor* would be *DeviceID*.

The name of this field can be obtained in other fields of the form through the macro `_field_wmi_0_`, and the value that the field has for each record of the WMI query can be obtained through a macro with the same name of the field _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 the names of the additional fields to be used in the WMI query used in the generated modules will be indicated.

The names of these fields can be obtained in other fields of the form by means of `_field_wmi_N_` macros, and the values that the fields have for each record of the WMI query can be obtained by means of macros with the same names of the _FIELDNAME_ fields. 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 FreeSpace field 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 to obtain one or more records. For example: `DriveType = 3`.

In the WMI wizard components, a separate module will be generated for each record returned by the WMI scan query.

Back to Pandora FMS Documentation Index