HPSA Extension Pack

TMN Inventory User Reference

Release v.5.1

Legal Notices

Document id:

Table of Contents

Support

Support for the HP Open View Service Activator SPI product is available on the following mailing list:

hpsa-support@hp.com

In This Guide

This guide is meant as a user reference guide for the Lock Manager's latest version. It contains all the information about this tool, its features and how to use them.

Audience

The audience for this guide is the Solutions Integrator (SI). The SI has a combination of some or all of the following capabilities:

Understands and has a solid working knowledge of:

- UNIX® commands
- Windows® system administration

Understands networking concepts and language

Is able to program in Java™ and XML

Understands security issues

Understands the customer's problem domain

Conventions

The following typographical conventions are used in this guide.

Font	What the Font Represents	Example
Italic	Book or manual titles, and man page names	Refer to the *HP Service Activator — Workflows and the Workflow Manager* and the *Javadocs* man page for more information.
	Provides emphasis	You *must* follow these steps.
	Specifies a variable that you must supply when entering a command	Run the command: `InventoryBuilder` *<sourceFiles>*
	Parameters to a method	The *assigned_criteria* parameter returns an ACSE response.
Bold	New terms	The **distinguishing attribute** of this class...
`Computer`	Text and items on the computer screen	The system replies: `Press Enter`
	Command names	Use the `InventoryBuilder` command ...
	Method names	The `get_all_replies()` method does the following...
	File and directory names	Edit the file `$ACTIVATOR_ETC/config/mwfm.xml`
	Process names	Check to see if `mwfm` is running.
	Window/dialog box names	In the `Test and Track` dialog...
	XML tag references	Use the `<DBTable>` tag to...
`Computer Bold`	Text that you must type	At the prompt, type: **`ls -l`**
Keycap	Keyboard keys	Press **Return**.
[Button]	Buttons on the user interface	Click `[Delete]`. Click the `[Apply]` button.
Menu Items	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows	Select `Locate:Objects->by Comment`.

Install Location Descriptors

The following names are used throughout this guide to define install locations.

Descriptor	What the Descriptor Represents
$ACTIVATOR_OPT	The base install location of Service Activator. The UNIX location is /opt/OV/ServiceActivator The Windows location is <drive>:\HP\OpenView\ServiceActivator\
$ACTIVATOR_ETC	The install location of specific Service Activator configuration files. The UNIX location is /etc/opt/OV/ServiceActivator The Windows location is <drive>:\HP\OpenView\ServiceActivator\etc\
$ACTIVATOR_VAR	The install location of specific Service Activator logging files. The UNIX location is /var/opt/OV/ServiceActivator The Windows location is <drive>:\HP\OpenView\ServiceActivator\var\
$ACTIVATOR_BIN	The install location of specific Service Activator binary files. The UNIX location is /opt/OV/ServiceActivator/bin The Windows location is <drive>:\HP\OpenView\ServiceActivator\bin\
$ACTIVATOR_THIRD_PARTY	The location for new Java components such as workflow nodes and modules. Third-party libraries can also be placed in this directory. The UNIX location is /opt/OV/ServiceActivator/3rd-party The Windows location is <drive>:\HP\OpenView\ServiceActivator\3rd-party\ Customized inventory files are stored in the following locations: UNIX: $ACTIVATOR_THIRD_PARTY/inventory Windows: $ACTIVATOR_THIRD_PARTY\inventory
$JBOSS_HOME	HOME The install location for JBoss. The UNIX location is /opt/HP/jboss The Windows location is <drive>:\HP\jboss
$JBOSS_DEPLOY	The install location of the Service Activator J2EE components. The UNIX location is /opt/HP/jboss/server/default/deploy The Windows location is <drive>:\HP\jboss\server\default\deploy
$ACTIVATOR_DB_USER	The database user name you define. Suggestion: ovactivator
$ACTIVATOR_SSH_USER	The Secure Shell user name you define.

	Suggestion: `ovactusr`
`$SOSA_HOME`	The base install location of SOSA. The UNIX location is `/opt/OV/Sosa` The Windows location is <drive>:\HP\OpenView\Sosa\
`$SOSA_BIN`	The install location of specific SOSA binary files. The UNIX location is `/opt/OV/Sosa/bin` The Windows location is `<drive>:\HP\OpenView\Sosa\bin\`
`$SOSA_ETC`	The install location of specific SOSA configuration files. The UNIX location is `/opt/OV/Sosa/config` The Windows location is <drive>:\HP\OpenView\Sosa\config\
`$ECP_HOME`	The base install location of Equipment Connections Pool. The UNIX location is `/opt/OV/ECP` The Windows location is <drive>:\HP\OpenView\ECP\
`$ECP_BIN`	The install location of specific Equipment Connections Pool binary files. The UNIX location is `/opt/OV/ECP/bin` The Windows location is <drive>:\HP\OpenView\ECP\bin\
`$ECP_ETC`	The install location of specific Equipment Connections Pool configuration files. The UNIX location is `/opt/OV/ECP/conf` The Windows location is <drive>:\HP\OpenView\ECP\conf\

1. Introduction

1.1. Acronyms

MWFM: Micro Work Flow Manager

HPSA: HP Service Activator

2. General description

The TMN Inventory is a library that can be used to organize and manage the complete set of networks and equipments of an organization. It is an Inventory Builder created project, that is, a set of xml entities describing the relationships and attributes of the elements involved in the network. These entities are transformed with the IB tool into DB tables, and java classes that provide tools to use these entities.

The TMN Inventory comprises many entities, which can be network based: Network, Path, TerminationPoint…; or equipment based: NetworkElement, EquipmentComponent, EquipmentFunction, Manufacturer… Each one will be described in detail in the following chapter.

3. TMN Inventory Entities

The TMN Inventory is basically a description of a network and its elements and relationships; therefore, we will describe each element in turn, and explain the relations.

3.1. Colour

This is one of the simpler entities; it provides an RGB value and its identification for further use.

3.2. ElementTypes

It provides a list of possible types of element that can appear in the network. An example element type could be a Router.

3.3. EquipmentFunction

It presents a list of possible functions that a piece of equipment may have. An example could be a WIMAX converter.

3.4. EquipmentOS

This entity contains all the possible Operating Systems that can be installed in the inventory's system. An example Operating System could be '1.0.2.0 ciscoVersionFile1.0.2.0'

3.5. EquipmentStatus

It represents the status of a piece of equipment. The table contains all possible status that a piece of equipment may have. A possible status could be 'Active'.

3.6. PathStatus

It provides the list of all possible status a Path may have. A Path status could be AVAILABLE.

3.7. Provinces

This is a list of provinces of regions to locate the networks situation in a map.

3.8. Location

These are locations that belong to a particular province. For example a city: Madrid.

3.9. Manufacturers

These are the names of the manufacturers of the equipment. For example: HP.

3.10. Network

This table represents a Network. A Network can belong to another network. It also contains X Y parameters so that it can be located in a map. An example Network could be: 'Jonquera', which belongs to its parent network: 'Telefónica'.

3.11. ElementModels

These are the different element models available. Each ElementModel is of a particular ElementType and built by a Manufacturer. For example: RS3000, a model built by Riverstone.

3.12. EquipmentFunctionModel

This entity provides a relation between EquipmentFunctions and ElementModels.

3.13. EquipmentOSModel

This entity provides a relation between EquipmentOS and ElementModels

3.14. NetworkElement

This is perhaps the most important entity in the Inventory. It tries to describe an element inside a Network. This entity has a Name, Description and IP and has relations to the following entities:

- Status, to show the Status of the Element.
- Network to show the Network the Element belongs to.
- Parent Network of this Network.
- Manufacturer, the Manufacturer who built it.
- ElementType, to show the type of this Network Element.
- X and Y axes, to give the position inside a map.
- ElementModel, the element model of this Network Element.
- ElementFunction, the function this element provides.
- Localizaciones, to show the Location where this element resides.
- Internal Function, to show the function of this element inside the network.
- OSVersion, the version of the Operating System.

An example could be a Router inside a network.

3.15. ElementComponent

This entity is given a name and status, and provides relations with the NetworkElement it belongs to and, if there are any, to the parent's ElementComponent. Examples of element components could be: 'chassis', 'rack' or network card.

3.16. Path

A Path is a representation of the path between two NetworkElements. It contains links to the origin and destination Network Elements and to the PathStatus. A path is a virtual connection.

3.17. PathComponent

This entity is a component belonging to a Path, it therefore contains a relation with the Path, an index of the order within the Path whether the jump is loose or strict, and a relation with the Colour table. A path is made up of components, such as a switch or a router.

3.18. PathNE

This is a path to a Network Element.

3.19. TerminationPointID

This is an endpoint of an ElementComponent or NetworkElement. It has a name and links to the Network and Element Component it belongs to. It also has a description of its use, and a relation with its parent TerminationPointID. An example could be a Port.

3.20. TMNConnection

This entity describes a physical connection. It provides relations between the network of origin and the destination network, of the NetworkElement of origin and also destination, and finally the TrminationPointIDs of origin and destination. It is also related to a Path, has X-Y coordinates. An example is a real physical connection.

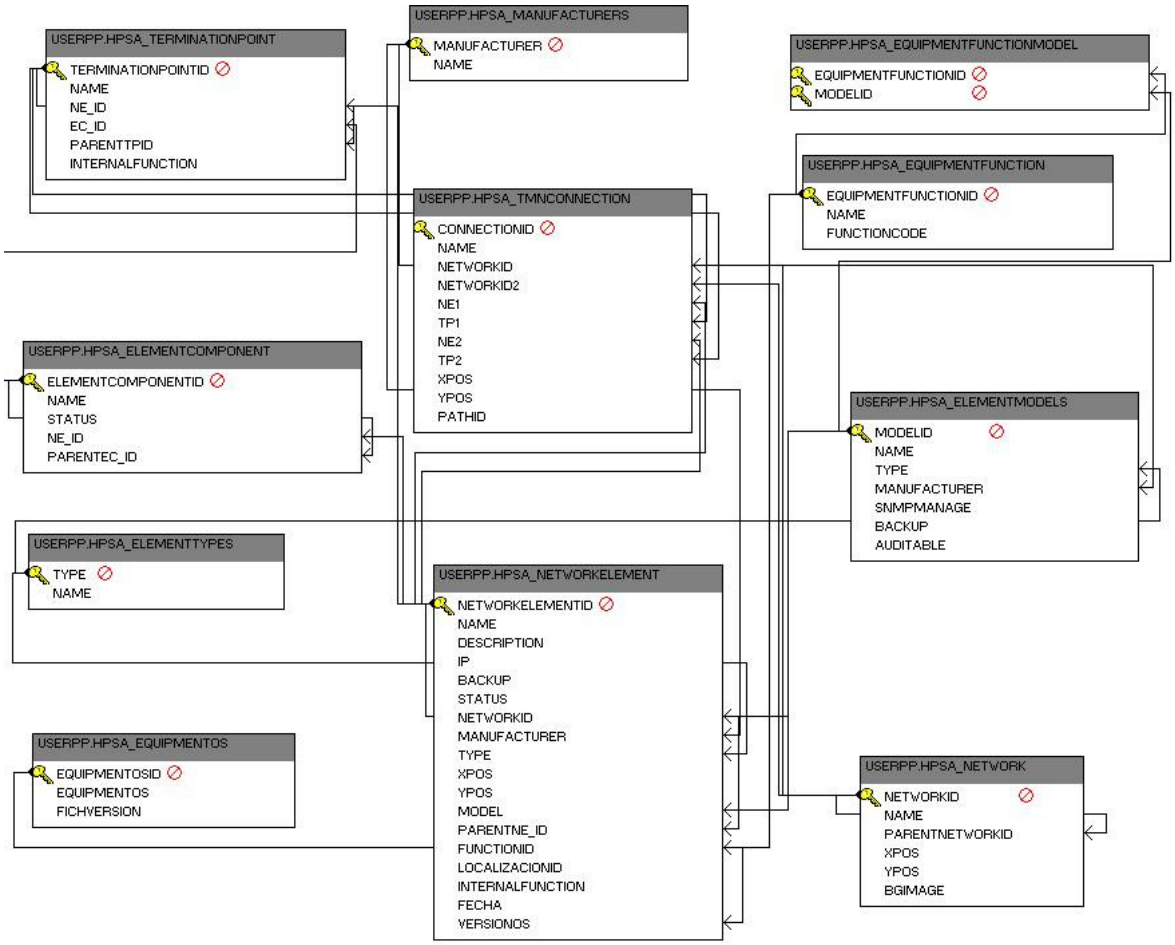
3.21. PathConnection

It provides a relation to a TMNConnection; and therefore ties a virtual connection to the physical connection that supports it.

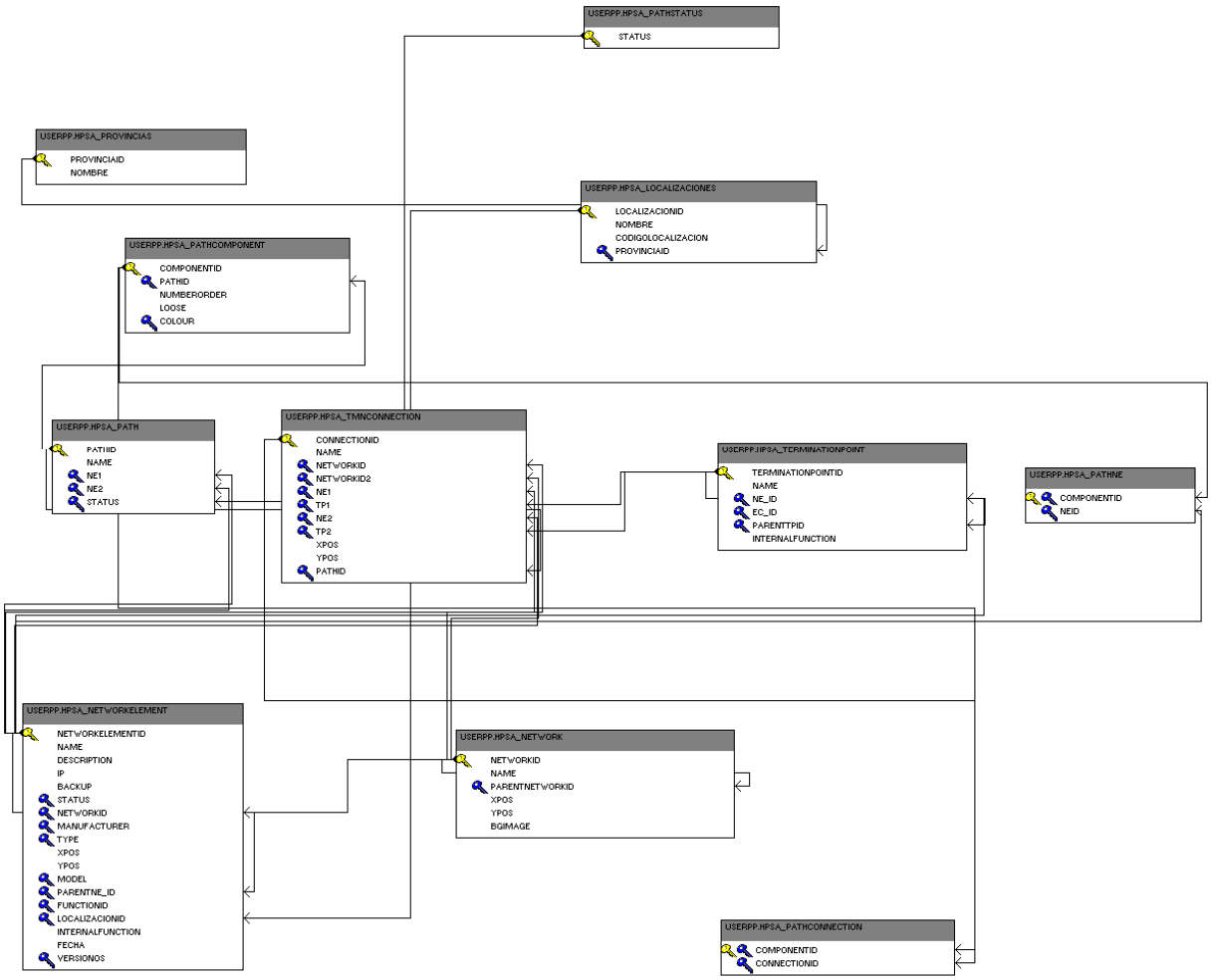
4. TMN Inventory structure

The structure will be shown with a subset of the classes, leaving the location and path entities out of the diagram, to better illustrate the Network entities.

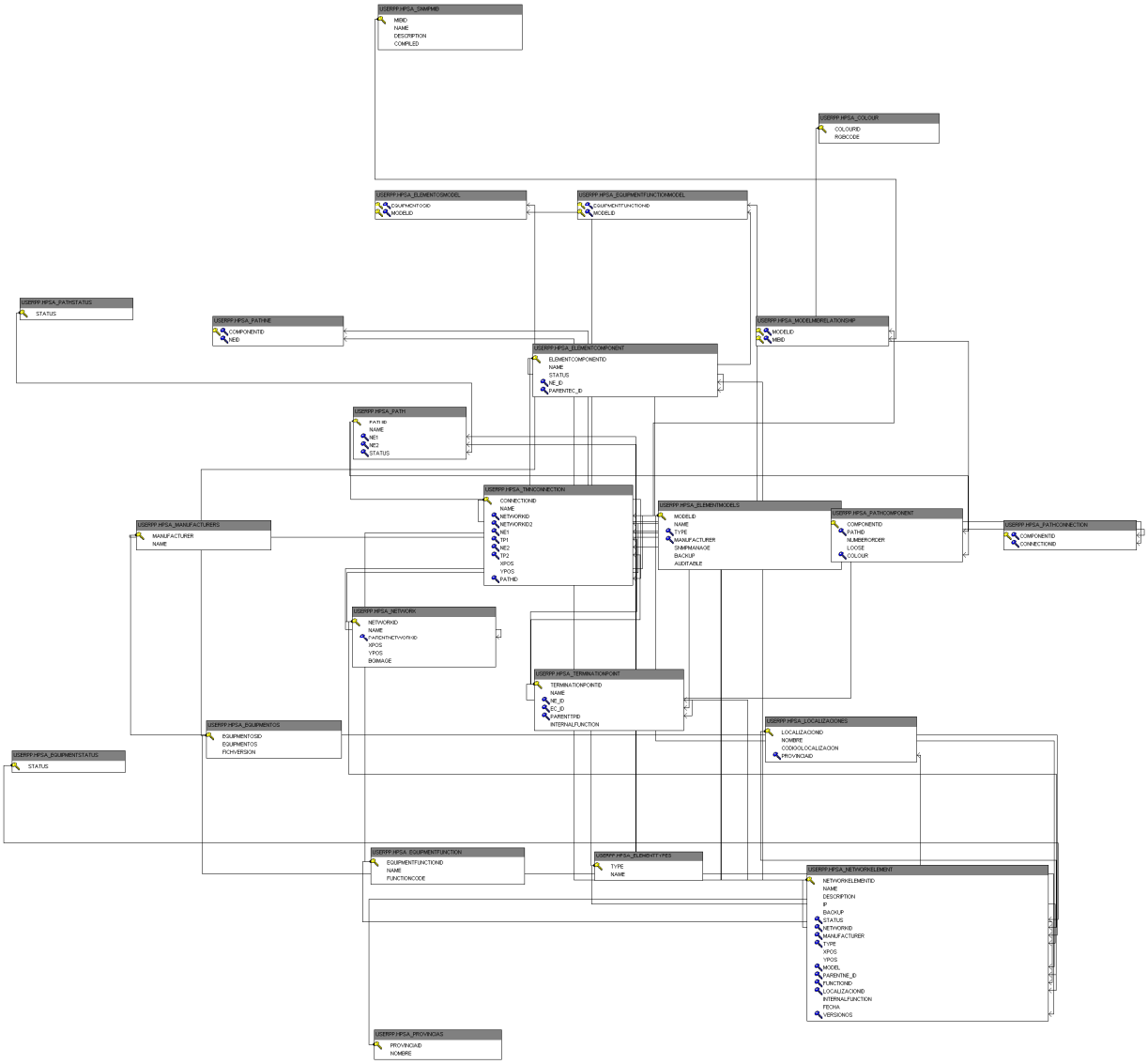
4.1. Network Entities Diagram



4.2. Path Diagram



4.3. Full Diagram



5. Inventory Views

5.1. Parameters View

This view allows the creation, deletion and maintenance of some TMN parameters such as provinces & locations, element types, equipment functions, equipment statuses, equipment OS and manufacturers & models. What follows is a brief description of the branches defined at this view.

5.1.1. TMN Parameters (Static)

This is just the root element and has no operations.

5.1.2. Provinces (Static)

This branch groups the existing provinces and allows the creation of new ones.

5.1.3. Provinces

Each existing province is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element
- Create Location… → create a new location in this province

Note that in order to delete a province it must not have locations associated to it.

5.1.4. Locations

Each existing location associated to the parent province is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element

Note that in order to delete a location it must not have equipments associated to it.

5.1.5. Element types (Static)

This branch groups the existing element types and allows the creation of new ones.

5.1.6. Types

Each existing element type is represented here by its name and accepts the following operations:

- View… → show this element's details

- Update … → modify this element's details
- Delete … → delete this element

Note that in order to delete a type it must not have neither models nor equipments associated to it.

5.1.7. Equipment functions (Static)

This branch groups the existing equipment functions and allows the creation of new ones.

5.1.8. Functions

Each existing equipment function is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element

Note that in order to delete a function it must not have neither models nor equipments associated to it.

5.1.9. Equipment status (Static)

This branch groups the existing equipment status and allows the creation of new ones.

5.1.10. Status

Each existing equipment status is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element

Note that in order to delete a status it must not have equipments associated to it.

5.1.11. Equipment OS (Static)

This branch groups the existing equipment OS versions and allows the creation of new ones.

5.1.12. OS versions

Each existing equipment OS is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element

Note that in order to delete an OS version it must not have equipments associated to it.

5.1.13. Manufacturers (Static)

This branch groups the existing manufacturers and allows the creation of new ones.

5.1.14. Manufacturers

Each existing manufacturer is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element
- Create Model… → create a new model associated to this manufacturer

Note that in order to delete a manufacturer it must not have models associated to it.

5.1.15. Models

Each existing model associated to the parent manufacturer is represented here by its name and accepts the following operations:

- View… → show this element's details
- Update … → modify this element's details
- Delete … → delete this element

Note that in order to delete a model it must not have equipments associated to it.

5.2. Equipments View

This view allows the creation of networks, equipment, and related components and ports. What follows is a brief description of the branches defined at this view.

5.2.1. Network Elements (Static)

This is just the root element and has no operations.

5.2.2. Networks (Static)

This branch groups the existing networks and allows the creation of a new one.

5.2.3. Networks

Each existing network is represented here by its name and accepts the following operations:

- View *network*… → show network details
- Update *network*… → modify network parameters
- Delete *network*… → delete this network from the system
- Create Equipment… → create a new equipment in this network

Note that in order to delete a network it must not have equipments associated to it.

5.2.4. Equipments

Each item represents an equipment of the parent network represented by its name. The following operations are supported:

- View *equipment*… → show equipment details
- Update *equipment*… → modify equipment parameters
- Delete *equipment*… → delete this equipment from the network
- Create Component… → create a new component of this equipment

Note that in order to delete an equipment it must not be connected to any other one

5.2.5. Components

Components belonging to the parent equipment are listed here by their assigned name. The following operations are supported:

- View *component*… → show component details
- Update *component* … → modify component parameters
- Delete *component* … → delete this component
- Create Port… → create a new port on this component

Note that in order to delete a component it must not have connections to other equipments on any of its ports

5.2.6. Ports

Ports created on the parent component are shown here represented by their name. The following operations are supported:

- View *port*… → show port details
- Update *port* … → modify port parameters
- Delete *port* … → delete this port

Note that in order to delete a port it must not be connected to any other equipment