Oracle Communications IP Service Activator™
5.2.4

Network Discovery and Basic Setup

Third Edition
December 2008

ORACLE

Contents

Preface

About this document

This guide provides detailed step-by-step information on setting up the Service Activator system as well as discovering and representing the network topology. It is intended for network configuration engineers responsible for the initial setup of Service Activator after installation.

It consists of the following chapters:

- Chapter 1: *Service Activator Setup: Overview* outlines the setup tasks involved in installing and running Service Activator.
- Chapter 2: *The User Interface* provides an introduction to Service Activator's graphical user interface.
- Chapter 3: *Transactions* describes Service Activator's transaction-based model for making configuration changes and explains how to create and use transactions.
- Chapter 4: *Setting Up Users* describes how to set up user groups and users and the permissions that may be assigned to groups.
- Chapter 5: *Setting Up Domain Information* explains the tasks you need to do immediately after installation in order to set up Service Activator and create domains.
- Chapter 6: *Defining and Applying Roles* introduces roles and their function in applying policy to the network and describes how to create role assignment rules.
- Chapter 7: *Discovering and Setting Up the Network* explains how to discover the network topology for a policy domain, set up device and interface related information and manage devices.
- Chapter 8: *Representing and Mapping Objects* describes how objects appear in the user interface and how to set up and work with network topology maps.
- Chapter 9: *Checking Device Status and Capabilities* describes Service Activator's device status categories and describes how to check device capabilities.

Before contacting Oracle Global Customer Support (GCS)

If you have an issue or question, Oracle recommends reviewing the product documentation and articles on MetaLink in the Top Technical Documents section to see if you can find a solution. MetaLink is located at http://metalink.oracle.com.

In addition to MetaLink, product documentation can also be found on the product CDs and in the product set on Oracle E-Delivery.

Within the product documentation, the following publications may contain problem resolutions, work-arounds and troubleshooting information:

- Release Notes
- Oracle Installation and User's Guide
- README files

Contacting Oracle Global Customer Support (GCS)

You can submit, update, and review service requests (SRs) of all severities on MetaLink, which is available 24 hours a day, 7 days a week. For technical issues of an urgent nature, you may call Oracle Global Customer Support (GCS) directly.

Oracle prefers that you use MetaLink to log your SR electronically, but if you need to contact GCS by telephone regarding a new SR, a support engineer will take down the information about your technical issue and then assign the SR to a technical engineer. A technical support representative for the Oracle and/or former MetaSolv products will then contact you.

Note that logging a new SR in a language other than English is only supported during your local country business hours. Outside of your local country business hours, technical issues are supported in English only. All SRs not logged in English outside of your local country business hours will be received the next business day. For broader access to skilled technical support, Oracle recommends logging new SRs in English.

Oracle GCS can be reached locally in each country. Refer to the Oracle website for the support contact information in your country. The Oracle support website is located at http://www.oracle.com/support/contact.html.

Downloading products and documentation

To download the Oracle and/or former MetaSolv products and documentation, go to the Oracle E-Delivery site, located at http://edelivery.oracle.com.

You can purchase a hard copy of Oracle product documentation on the Oracle store site, located at http://oraclestore.oracle.com.

For a complete selection of Oracle documentation, go to the Oracle documentation site, located at http://www.oracle.com/technology/documentation.

Downloading a media pack

To download a media pack from Oracle E-Delivery

1. Go to http://edelivery.oracle.com.
2. Select the appropriate language and click **Continue**.
3. Enter the appropriate **Export Validation** information, accept the license agreements and click **Continue**.
4. For **Product Pack**, select **Oracle Communications Applications**.
5. For **Platform**, select the appropriate platform for your installation.
6. Click **Go**.
7. Select the appropriate media pack and click **Continue**.
8. Click **Download** for the items you wish to download.
9. Follow the installation documentation for each component you wish to install.

Service Activator publications

The Service Activator documentation suite includes a full range of publications. Refer to the Service Activator *Release Notes* for more information.

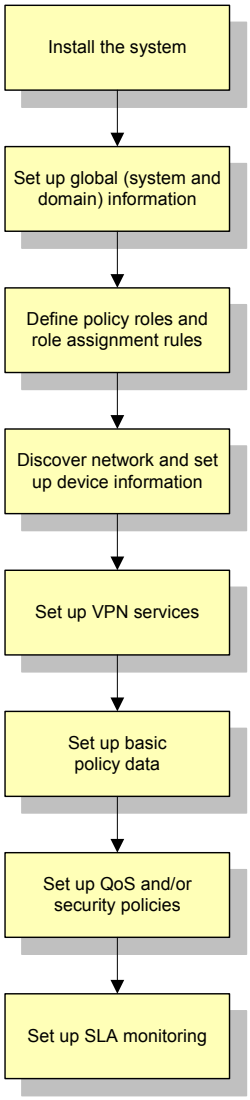
Chapter 1

Service Activator Setup: Overview

This chapter outlines the steps you need to perform to install and run Service Activator.

Steps needed to set up Service Activator

The following diagram illustrates the steps you need to perform in order to set up Service Activator and apply services and policies to the network. These are explained in more detail in the sections below:



Installing Service Activator

In a normal distributed operation, Service Activator components are installed on multiple hosts. For tests and evaluations, we recommend that you install an evaluation version of the software on a single host system. This will allow you to become familiar with Service Activator before installing a fully-operational distributed system. The Microsoft Windows version incorporates an evaluation installation that includes a local Microsoft Access database.

For full details of the installation process see the *Setup Guide*.

Setting up system and domain information

There are a number of tasks that you should do immediately after installation. These include the following:

- Setting up system users and groups – one initial system user is created the first time you log in. You need to set up security options for that user and create additional user groups and users.
- Setting up domains – you can set up multiple domains, one per Autonomous System (AS) region.
- Loading basic configuration data – you can install set-up files for each domain that automatically create the basic data that you need when setting up services and policies.

These tasks are described in detail in *Setting Up Users* on page 77 and *Setting Up Domain Information* on page 103.

Setting up roles and role assignment rules

In Service Activator, roles provide the meeting point between policy elements and policy targets, such as devices, interfaces and sub-interfaces. The roles you need to set up will depend on the service or policy you plan to implement:

- Service Activator provides a set of system-defined roles. In order to configure MPLS VPNs you must apply these roles to the devices and interfaces that are to be managed.
- User-defined roles can be created and assigned to policy targets.

Whether you are using system or user-defined roles, the recommended method for assigning roles to policy targets is role assignment rules. A role assignment rule applies a role to a device or interface if it matches the defined criteria. Role assignment rules are applied to the network during discovery and can also be applied as a standalone task.

For full details of defining roles and role assignment rules, see *Defining and Applying Roles* on page 111.

Discovering the network

The next step is to set up details of the network for each domain that you are managing. You do this by running a device discovery process that uses SNMP to find out information about devices and connected segments.

The discovery process performs the following:

- Finds out details of the devices, segments and hosts in the network
- Automatically assigns roles to devices and interfaces based on a set of role assignment rules
- Assigns devices to the proxy agents that will manage them
- Sets up the security parameters that Service Activator needs to configure devices
- Ascertains the capabilities of devices and interfaces, indicating the QoS, access control, measurement and VPN features that are available

Before running the discovery process you must set up certain device-specific information, and after discovery is complete you must set up Service Activator to manage the devices.

For full details of discovering devices and the tasks you must perform before and after device discovery, see *Discovering and Setting Up the Network* on page 137.

Mapping the network

The topology map provides a visual representation of the discovered network.

If Service Activator is set to map the network automatically, nodes are added to the topology map as they are discovered. By default, discovered nodes must be mapped manually when the discovery process is complete.

For information on mapping the network, see *Representing and Mapping Objects* on page 185.

Setting up VPN services

Once you have discovered and mapped the network, you can create the VPN services that you want to implement in each domain:

- Multi-Protocol Label Switching VPNs (MPLS VPNs) – you set up a VPN by defining customers and sites and specifying how sites are linked together.
- Transparent LAN Service (TLS) – you set up a TLS by defining customers and layer 2 sites that specify the criteria on which access to the TLS is based. Layer 2 sites are linked together in a TLS object.
- Circuit Cross Connects (CCCs) – you set up a CCC by defining customers and linking VC endpoints. CCCs are specific to Juniper M-series devices.
- Layer 2 Martini VPNs – you set up a Layer 2 Martini VPN by defining customers, creating the Layer 2 Martini VPN and linking Martini endpoints. Layer 2 Martini VPNs are specific to Cisco and Juniper M-series devices.

For more information, see the *Configuring VPN Services* guide.

Setting up basic policy data

If you plan to implement a QoS or security policy, you need to set up the basic data used in configuring these elements.

- Class of service information – defining how differentiated services are being used. You need to set this up if you are setting up a QoS policy.
- Rule components – traffic types, classifications, IP protocols, packet markings, and date and time templates, used when setting up QoS and policy rules.

For details of setting up basic data, see the *Configuring Policy Services* guide.

Setting up policies

If you have set up basic policy data, you can create the policies that you want to implement in each domain.

- QoS policies – classification rules, policing rules and PHB groups can be implemented to manage and prioritize categories of traffic at any point in the network.
- Access control policies – access rules can be set up to deny or explicitly permit defined categories of traffic.

For more information about setting up QoS policies, see the *Configuring Policy Services* guide.

Setting up SLA monitoring

Service Activator integrates with third-party reporting tools to provide SLA monitoring. Various measurement types are available to generate the source data for reporting:

- Service Assurance Agent (SAA) enables you to monitor point-to-point connections in an MPLS VPN or a measurement-only VPN.
- NetFlow and MIB-based measurements enable you to monitor activity at a specific point in the network.

More information on setting up SLA monitoring is available in the *Network and SLA Monitoring Guide* which provides details of setting up SLA measurement within Service Activator.

Transactions in Service Activator

Service Activator uses a transaction-based model for updating the system and implementing configuration changes. This means that you can make changes through the user interface and implement them immediately or save them in a pending state and implement them at some point in the future.

The tasks associated with initial system setup are probably performed by a single user as a one-off task. You may wish to implement these changes immediately. For implementing VPN services and setting up policies, you may wish to break down configuration changes into a number of transactions and perform a phased implementation.

For information on transactions, see *Transactions* on page 47.

Chapter 2confi

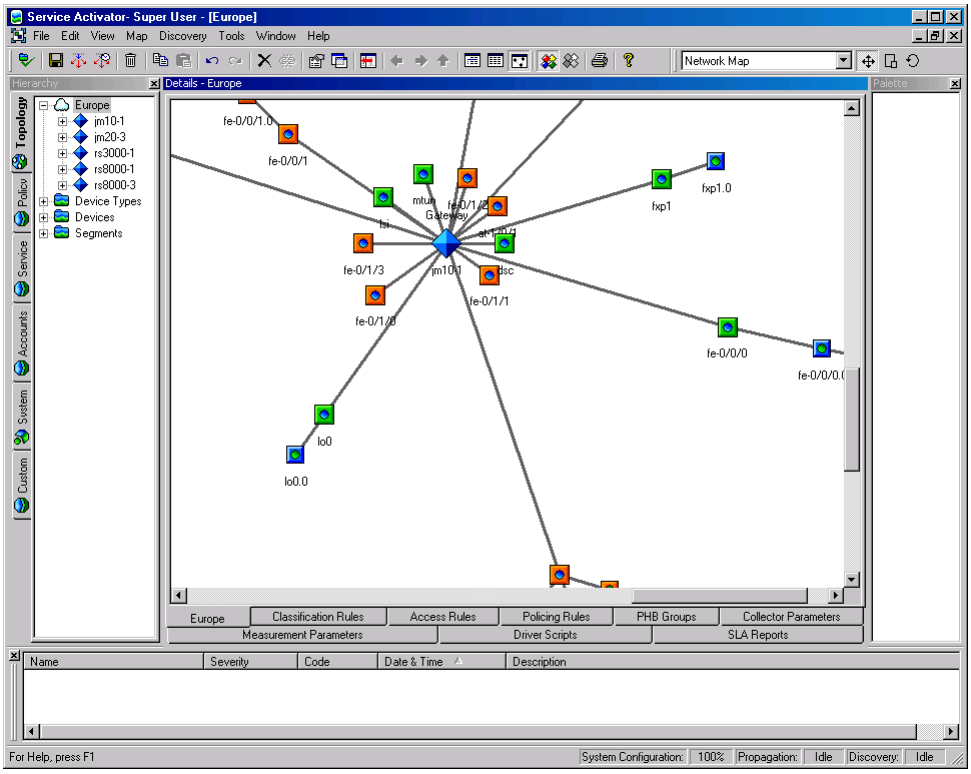
The User Interface

This chapter introduces elements of Service Activator's user interface and describes how to navigate through the system and customize the interface to suit your needs. It covers the following areas:

- Appearance of the user interface when you run the system for the first time
- Options available for issuing commands — menus, toolbars and pop-up menus
- Window types used when configuring the system
- How the system models the network applied policies in Service Activator's object model
- Available navigation tools
- How to change the appearance of the user interface
- How to search Service Activator, including searching for concrete configuration
- How to print

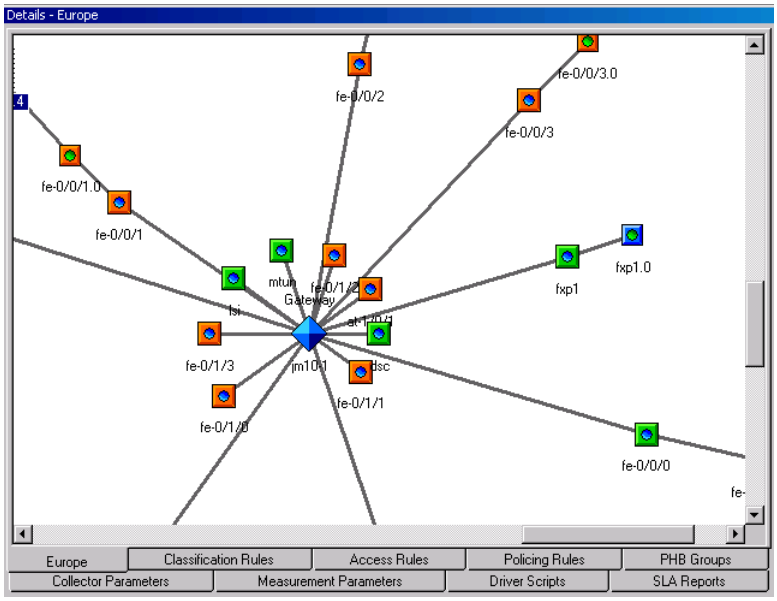
Introduction

The user interface provides a view of the system as it is currently configured and is the tool you use to enter new configuration details. Essentially, it acts as a front end to the policy server, Service Activator's core component.

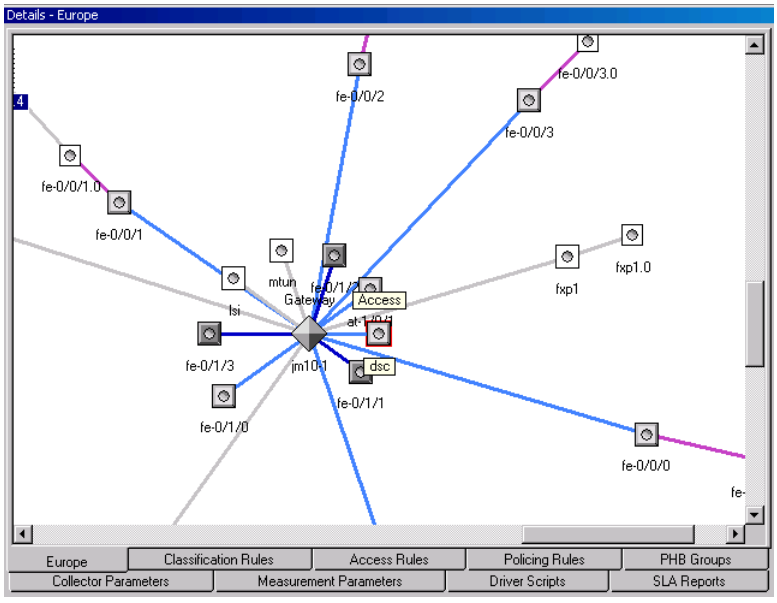


Working within the user interface, you can discover the devices to be managed, map the topology of the network and construct the policy to be applied. The information you enter in the user interface is saved to a database via the policy server.

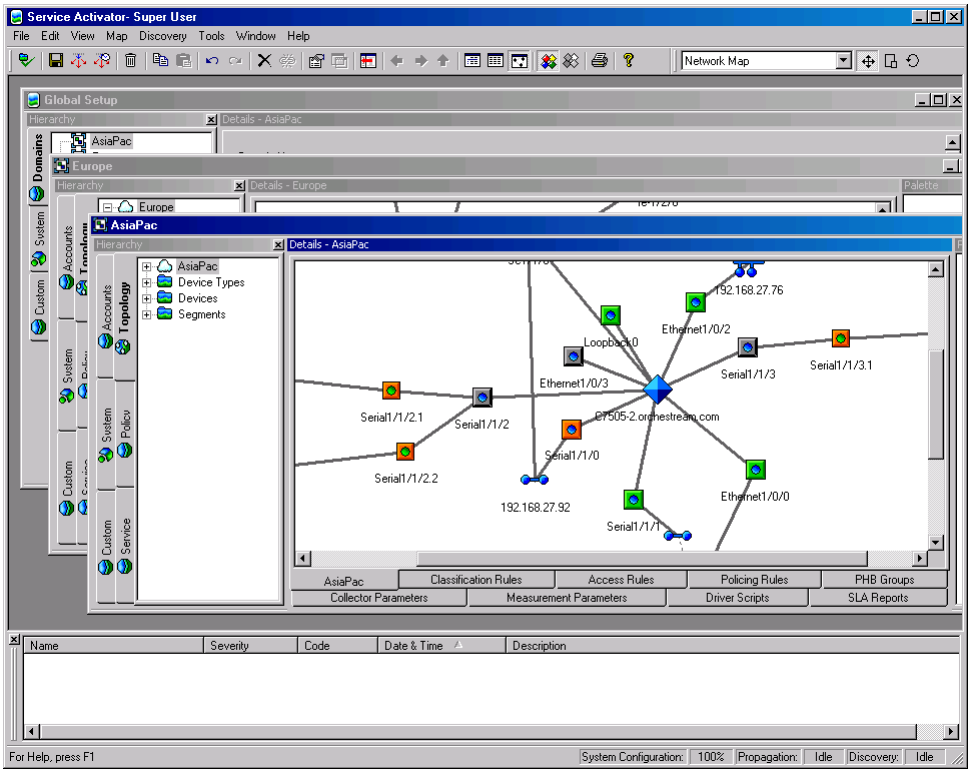
Service Activator models the network and the policy you apply to it using an object model, where every element is an object with its own properties and capabilities. Every object type has an icon associated with it. Where the state of an object may change, color is used to represent the state of the object. For example, a managed device that is reachable is represented by a green icon, while an unreachable device is represented by an orange icon.



For objects that may have policy applied to them, the system also provides a policy view, where object icons are displayed in shades of gray that reflect their policy role.



You can have several windows onto the system open simultaneously, with each one offering the same or a different view of the system. These windows are known as domain management windows.



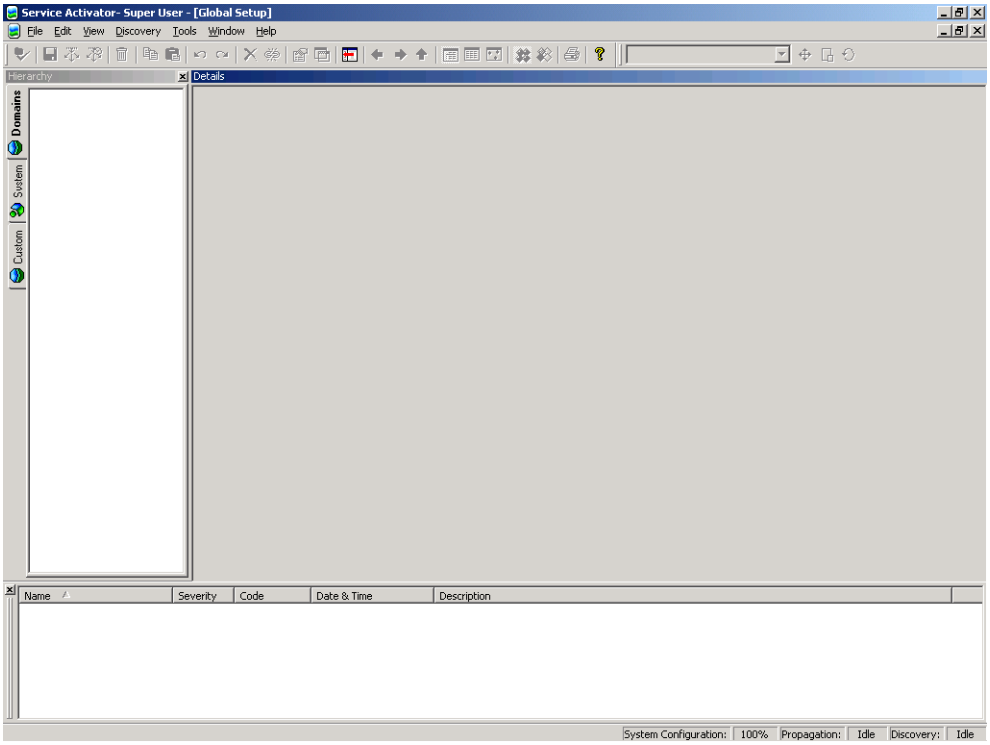
Each window is divided into several panes. Each pane displays a different type of information. You can choose whether to hide or show selected panes.

There may be several user interface components running at any one time, with each user entering new configuration details. As a user makes changes in the user interface, the changes are queued locally, and are not reflected in remote user interfaces. When a user commits a transaction, the policy server co-ordinates the information between user interfaces, ensuring that all users' views of the system are consistent.

Running Service Activator for the first time

When you run Service Activator for the first time after installation there are no domains defined and only default system information is displayed on the **System** tab.

The main Service Activator window appears:

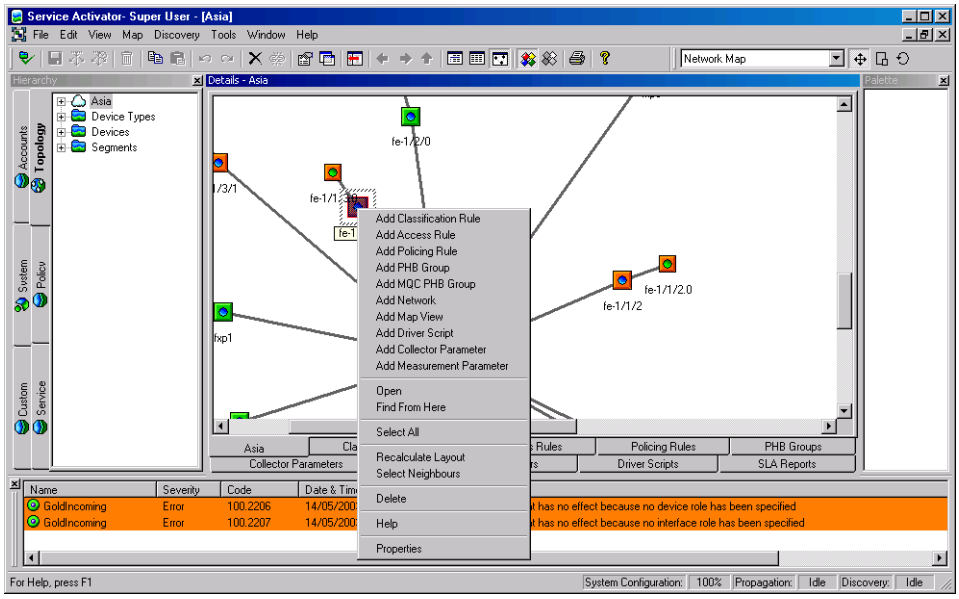


This window is referred to as the Global Setup window and is always displayed when you run Service Activator. For more information see *The global setup window* on page 21.

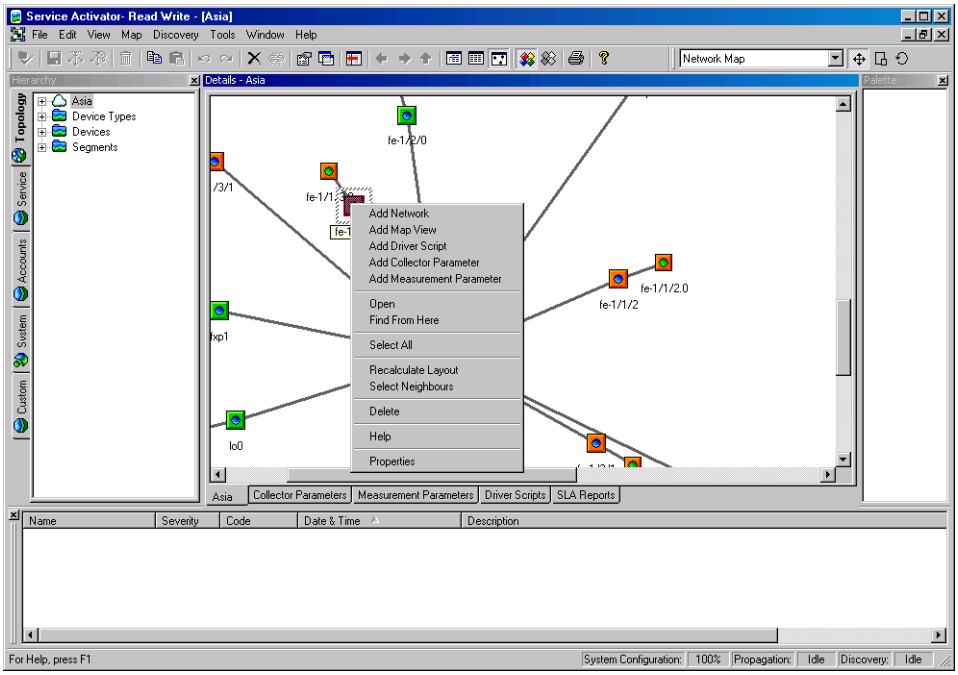
User access levels and the user interface

The appearance of the user interface is affected by your security level. You may be able to see all or only see parts of the system. Similarly you may be able to see some objects, but unable to modify their values.

For example, the following illustration shows the information displayed for a user who has permissions to view all information, and perform unlimited actions.



The following illustration shows information displayed for a user who has been denied access to Service Activator's policy area.



Note that the **Policy** tab is not displayed to the user and options for creating rules and PHB groups are missing from the device's pop-up menu. The user is therefore unable to view or create policy.

Note also that you cannot see faults associated with object types to which you do not have access. In this example, the warnings that relate to a PHB group with no roles associated with it cannot be seen in the second illustration.

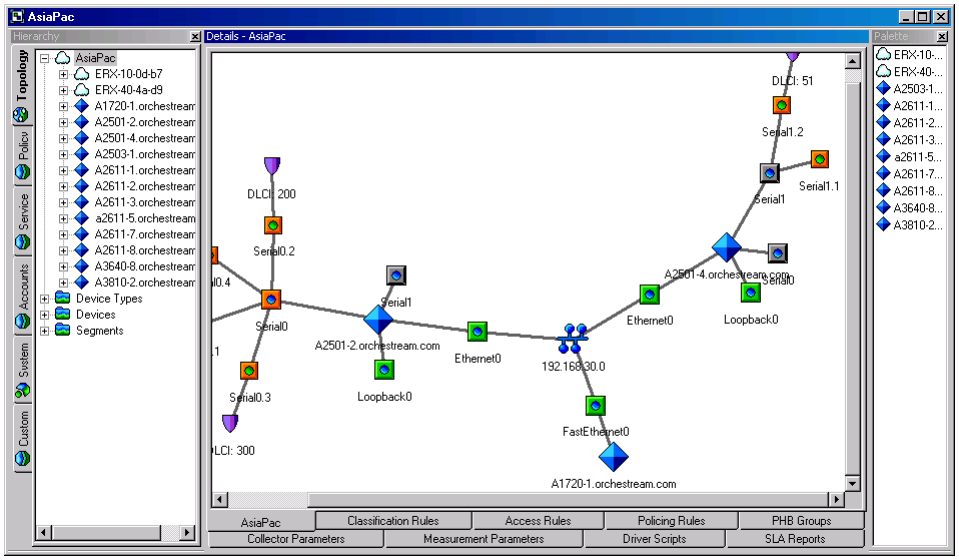
This chapter and the rest of the guide assumes you can view and modify all parts of the system. This access level is available to users with Super User access. If you have a lower access level you may be unable to access some parts of Service Activator described in this guide.

Service Activator's windows

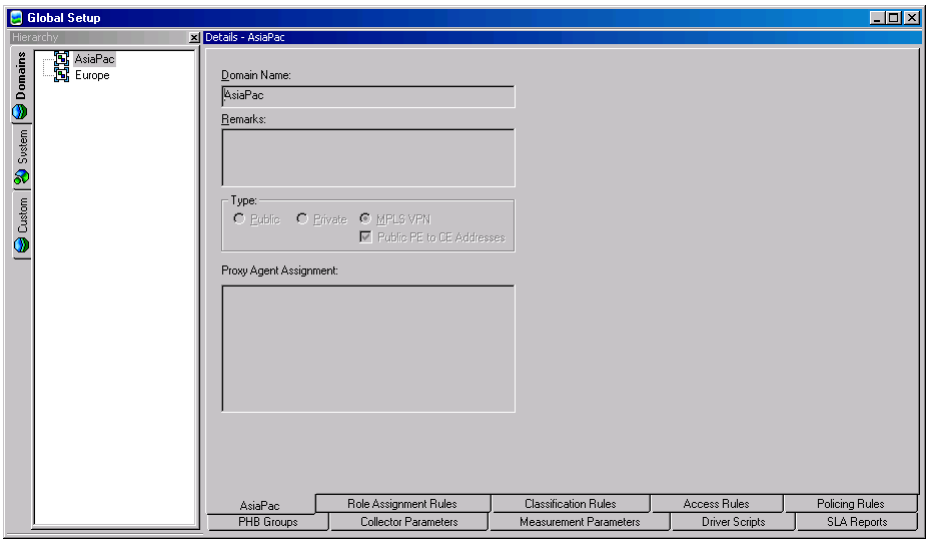
Service Activator has a Multiple Document Interface (MDI). This means you can open multiple windows within the main window and arrange them to support the current task. For example, you can display windows side-by-side to compare information between two domains, or drag and drop objects between windows on the same or different domains.

Information is displayed in the following window types:

- Domain management windows display information about a particular policy domain. This window type is your workspace area for managing the domain.



- The global setup window displays global information and allows you to create and manage multiple domains.



The information that is displayed in each window type and the actions you can perform depend on your access level. For example, in the global setup window you

may be able to view existing domains but unable to create new domains. For more information, see *User access levels and the user interface* on page 12.

To open a new domain management window

In a domain management window, either:

- From the **Window** menu, select **New Window**.

  A new domain management window is opened. The window's hierarchy pane shows the complete object model for the domain.

or:

- With an object is selected in the hierarchy pane, from the **View** menu, select **Open**.

  A new domain management window is opened. The window's hierarchy pane shows only those objects that exist below the object you selected in the hierarchy pane.

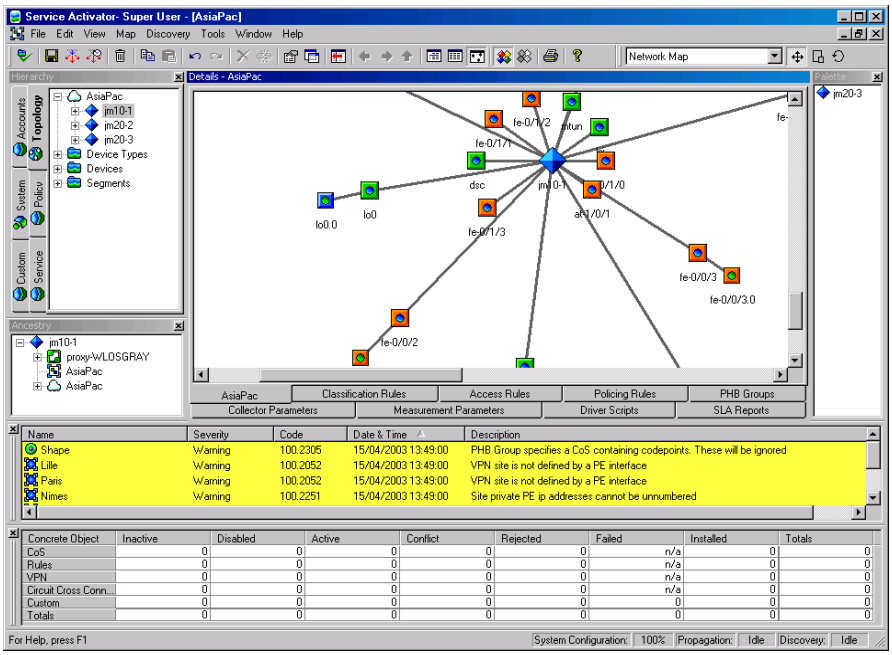
You can also create a new window by double-clicking on a domain on the **Domains** tab in a global setup window.

The domain management windows

Domain management windows provide a workspace for configuring a policy domain. You work in a domain management window to:

- Discover the network you intend to manage
- Create and view topology maps of the network
- Set up the policies and services you want to implement
- Create transactions and send configuration details to the devices you are managing

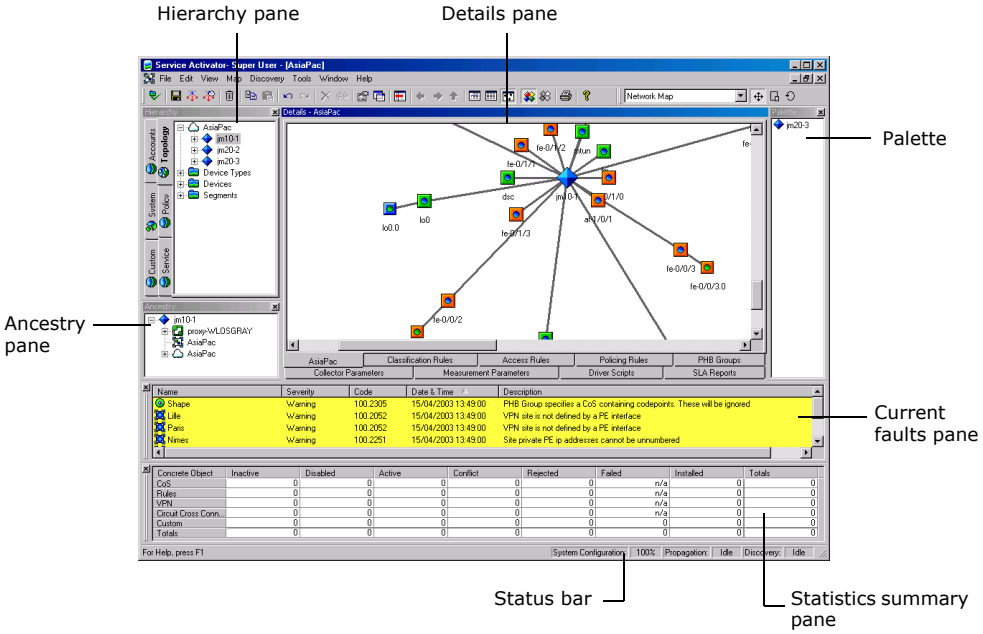
The following shows a domain management window with the topology map displayed:



When you run Service Activator for the first time, there are no domains created and you cannot display a domain management window. For information on creating a domain, see *Setting up domains* on page 104.

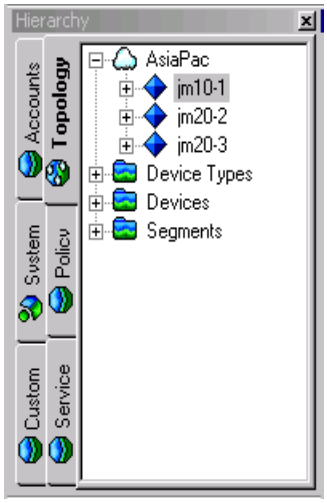
Panes in the domain management window

The domain management window is divided into sections.



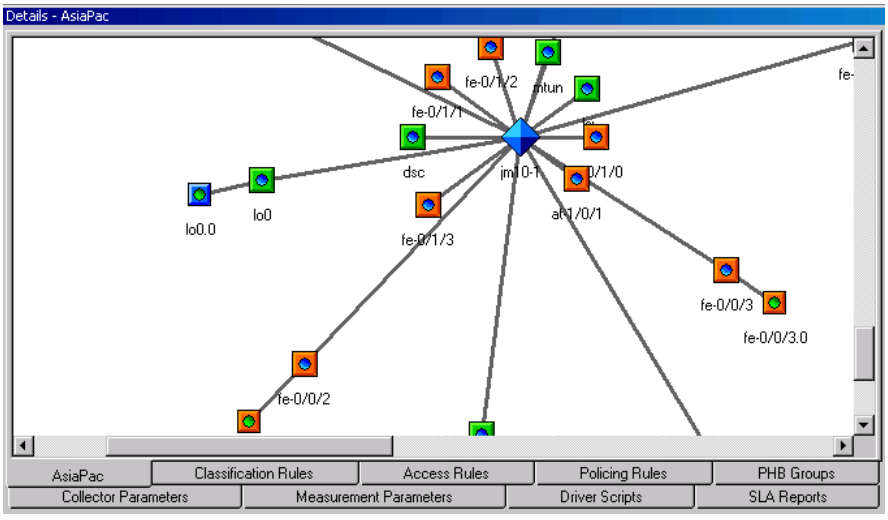
For information on hiding and showing panes in the domain management window, see *Hiding and showing windows and window elements* on page 38.

- The hierarchy tree pane displays Service Activator's object model as a tree structure, showing the major container (parent/child) relationships between the objects. Associated objects are arranged into tabbed groups. You can access all of the objects within the domain by selecting tabs and browsing through the hierarchy.



For more information about the hierarchy tree's tabs, see *Tabs in the domain management window* on page 20. For information about Service Activator's object model, see *How Service Activator models the system* on page 22.

- The **Details** pane displays information about the object that you have selected in the hierarchy pane. Information is displayed in map, list or report form, depending on the selected object's type.



For example:

— If you have selected a network object, its details can be viewed as a topology map, a list of the devices within that network or a report list showing details of each device's state, role, IP address and so on.

— If you have selected a VPN object, the sites within the VPN can be viewed in map, list or report form.

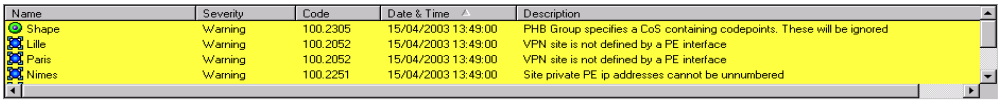
— If you have selected a device, the interfaces available on the device can be displayed in list or report form.

Service Activator remembers the current view when you close the User Interface. On restarting the GUI, the same view is used as the preferred view.

You can also display information about the configuration that applies to an object in the **Details** pane or the content of any of the system's log files. For more information, see the *Configuring Policy Services* and *Configuring VPN Services* guides. For information on Service Activator's log files, see the *Administrator's Guide*.

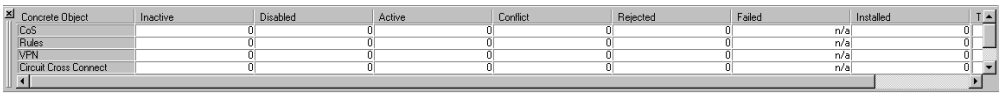
For information on printing the content of the **Details** pane, see *Printing* on page 44.

- The current faults pane displays information, warning and error messages reported by the system. Each message type is color coded for quick reference.



- The ancestry pane displays the object model as a reverse tree, with the object that you have currently selected in the hierarchy pane shown at the top level and its parent objects beneath it. It allows you to trace all the links between an object and its parent objects. The ancestry tree is not a navigation tool, that is, selecting items in the tree does not affect the information that is displayed in the **Details** pane. For more information about the object model, see *How Service Activator models the system* on page 22.

- The statistics summary pane summarizes status information for PHB groups, policy rules, CCCs, VPNs, Layer 2 Martini VPNs, and driver scripts.



- The status bar displays information about the state of Service Activator when you discover devices, propagate configuration data or retrieve device capabilities.

- The palette displays a list of discovered network objects that can be added to a network map or sites that can be added to a VPN map. It is only displayed when you are viewing a map. In the case of network maps, you can specify which objects are displayed on the palette. For more information, see *Recalculating a map's layout* on page 195.

You can hide or show many of these window sections. For more information, see *Hiding and showing windows and window elements* on page 38.

Tabs in the domain management window

Within the domain management window, sets of related objects are grouped and displayed on separate tabs. The tabs are actually part of the hierarchy pane. If the hierarchy pane is not displayed, the tabs are not visible. On each tab, objects are further grouped into folders and, in the case of the **Topology** tab, networks.

You can change the position of the hierarchy pane's tabs. For more information, see *Changing the hierarchy pane's tab position and style* on page 39.

Objects are grouped under the following tabs:

- **Topology**: network elements, including networks, devices and interfaces. Work on this tab to discover the network, view and create maps of the network to be managed and assign services and policies to the network elements.

- **Policy**: Class of Service (CoS) and policy building blocks. Work on the **Policy** tab to create roles, define the levels of service that can be applied to traffic and set up standard and MQC PHB groups and to define the following elements:
  - IP Protocols
  - Packet markings
  - Traffic types
  - Date and time templates
  - Classifications

  Note that you can create some basic rule component data by loading policy configuration files. For more information, see *Loading policy configuration data on page 107*.
- **Service**: information relating to VPN services. Work on this tab to create and maintain customers, VPNs and sites, TLSs and layer 2 sites, and point-to-point connections.
- **Accounts**: group, subnet, host and user accounts. Work on this tab to create the accounts that act as source or destination points to which rules can be applied.
- **System**: system users, components and associated system information. Work on this tab to view information about Service Activator system components and their host machines, add and edit user details, access the system's log files, view information about pending, scheduled and committed transactions, define external systems and create SAA templates.
- **Custom**: driver scripts that currently exist in Service Activator. Work on this tab to import, run and create driver scripts.

The global setup window

The Global Setup window lists the policy domains that are managed by the system and displays general information about each one.

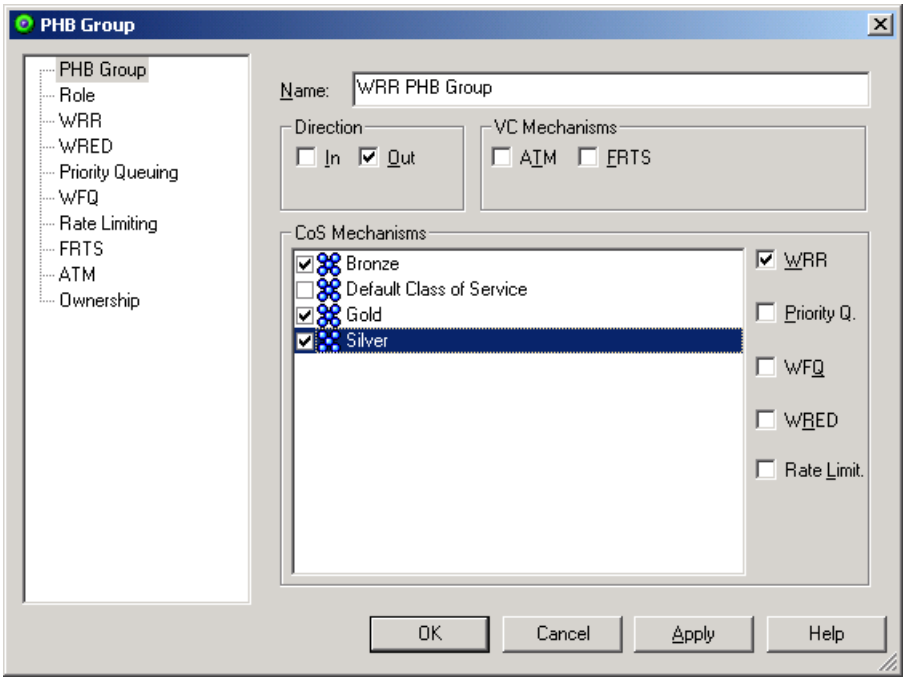
The window has three tabs:

- **Domains**: lists the domains that have been created and displays summary information for each one.
- **System**: this tab is identical to that displayed in the domain management window.
- **Custom**: this tab is identical to that displayed in the domain management window.

How Service Activator models the system

Service Activator regards anything that it can manipulate as an object. An object may be a representation of a physical element, such as a device or an interface, or a logical element, such as an account or a rule. Information about the objects that exist in the system is held in the database.

Every object has a set of attributes associated with it that define, at the minimum, the name of the object or more complex characteristics, such as detailed information about queuing mechanisms. The number and type of attributes vary between different types of object. For example, a device object has attributes that include its name and IP address, its role, and the interfaces available on the device. A packet marking object, by contrast, has only two attributes associated with it that define its name and the marking associated with it. The combined attributes of an object are referred to as object properties. You edit an object's properties in its dialog box.



Information about objects, the relationships that can exist between them and the actions you can perform on them, together make up the object model. Service Activator's object model represents three main areas:

- The underlying network topology
- The policies and services applied to the network

- The Service Activator system configuration itself

You can view the objects that are currently defined in the system in the domain management windows. You can also view information about system and domain-related objects in the global setup window. For more information, see *Tabs in the domain management window* on page 20.

You can view information about the relationships that exist between the objects in Service Activator using the hierarchy and ancestry panes:

- The hierarchy pane shows an object's relationship to objects that exist at a lower level in the object model. You can use the hierarchy tree to drill down through the hierarchy.
- The ancestry pane shows how an object is related to objects that exist at a higher level in the object model.

You can view the properties that are associated with an object by selecting **Properties** from the object's pop-up menu or from the **View** menu. For more information, see *Viewing and editing an object's properties* on page 25.

Modelling the network, services and policies

As you work in the user interface, you model the network and the policies or services to be applied to it by:

- Creating objects such as VPNs, classes of service and sites.
- Defining each object's properties or attributes, such as the connectivity type of a VPN or the name of a site
- Creating relationships between objects. Objects exist within a hierarchy of parent/child relationships, with the policy, system or domain objects existing at the highest level. Every new object created is a child of an existing object. You can also make parent/child relationships between some objects by linking them. For more information about linking objects, see *Linking objects* on page 29.

Service Activator maintains a central version of the object model called the common object model which is updated when you commit or save a transaction from a remote user interface. For further information see *Local and common object models* on page 51.

Inheritance between objects

There is a hierarchy of inheritance between objects, where objects at a lower level in the hierarchy inherit parameters that have been applied to objects at a higher level. Inheritance applies to the following:

- Rules
- PHB groups

- Driver scripts
- Collector parameters
- Measurement parameters

Note that devices and interfaces must be tagged with the appropriate role or roles for policy or parameters to be inherited. Service Activator's inheritance model is described in the *Configuring Policy Services* guide.

Working with objects

You carry out many of the tasks that are associated with objects using object pop-up menus. The options on each object's menu depend on the object type.

> Folders also have pop-up menus associated with them and some object types are created from these menus. For more information, see *Creating new objects* on page 25.

To view an object's pop-up menu

- Select the object and click the secondary mouse button.

  The right mouse button is generally configured as the secondary mouse button.

Selecting objects

You select an object by single clicking on it. A selected object's icon appears shaded.

The **Details** pane will only display object details if you double-click on an object.

You can also select multiple objects by:

- Selecting several objects by clicking and dragging over the objects.

  As you drag, the selection area is indicated by a dotted selection box. Selected objects appear shaded on the screen.
- Selecting objects one by one by holding down the Control key and clicking on each object.
- Where objects are listed in the **Details** pane and the palette, you can select a range of objects by holding down the Shift key and clicking on the first and last items in the range.

Note that you can use the multi-select technique to:

- Create a new map or network view that includes the selected network objects. For more information, see *Creating subsidiary networks and maps* on page 200.

- Set the properties for a number of classification, policing or access rules. For more information, see the *Configuring Policy Services* guide.

Creating new objects

You create new objects using an existing object's pop-up menu or the pop-up menu associated with a folder.

An object's pop-up menu provides options to create new objects of a type that the object model allows for that object.

In addition, you can generally add new objects from a folder's pop-up menu. For example, you can add new device types using the **Device Type** folder's pop-up menu. The exceptions to this rule are the **Devices** and **Segments** folders on the **Topology** tab and the **System Hosts** folder on the **System** tab.

Any new object is linked as a child of the object from which it was created. It may also be automatically linked to other objects. For example, if you create a new user in a user group, the user is a child of the user group object and also the system object.

You can view an object's parentage in the ancestry pane.

To create an object

1. Select the relevant folder or object in the hierarchy, ancestry or **Details** pane and select the appropriate **Add** command from the pop-up menu.

   For example, to create a new classification select the **Classifications** folder on the **Policy** tab, then select **Add Classification** from the pop-up menu.

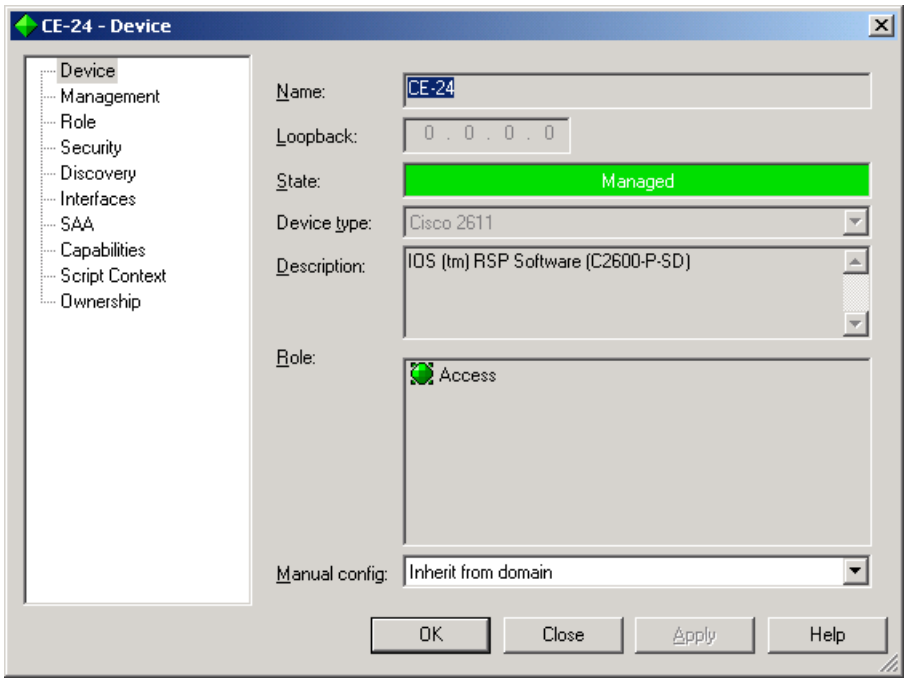
   Service Activator displays the properties dialog box for that object.

2. Complete the necessary details in the dialog box.

   If you need help on completing a particular property page, press F1 or click the **Help** button on the dialog box.

Viewing and editing an object's properties

You view and edit the attributes or properties that are associated with an object in the object's properties dialog box. Depending on the object's type, the dialog box may contain any number of property pages, which are accessed by clicking the appropriate entry in the navigation tree on the left side of the dialog box. The following illustration shows the properties dialog box for a device:



Properties dialog boxes are modeless windows, which means that you can leave them open while you select other objects from the interface. As you select objects, the properties dialog box changes to display the selected object's details. However, if you edit any of the details displayed in the dialog, the window becomes modal and you must apply the changes before you can select another object. You can only have one properties dialog box open at a time.

Every properties dialog box includes the following command buttons:

OK Validates the input, applies any changes that have been made and closes the dialog box.

Cancel Closes the dialog box without applying any changes. This button is only displayed once changes are made in the dialog box but have not yet been saved by clicking on the **Apply** button.

Close Closes the dialog box. This button is only displayed if changes made in the dialog box have been saved by clicking on the **Apply** button.

Apply Validates the input, applies any changes that have been made but leaves the dialog box open. This button is grayed out initially and only becomes active when changes are made.

Help Displays help relevant to the currently selected property page.

In addition, a validation failure message box may appear if you enter incorrect values. The message will indicate why the value cannot be accepted.

To display an object's properties

- Select the object and do one of the following:
  - — Select **Properties** from the object's pop-up menu.
  - — Select **Properties** from the **View** menu.
  - — Click on the **Properties** button on the toolbar.

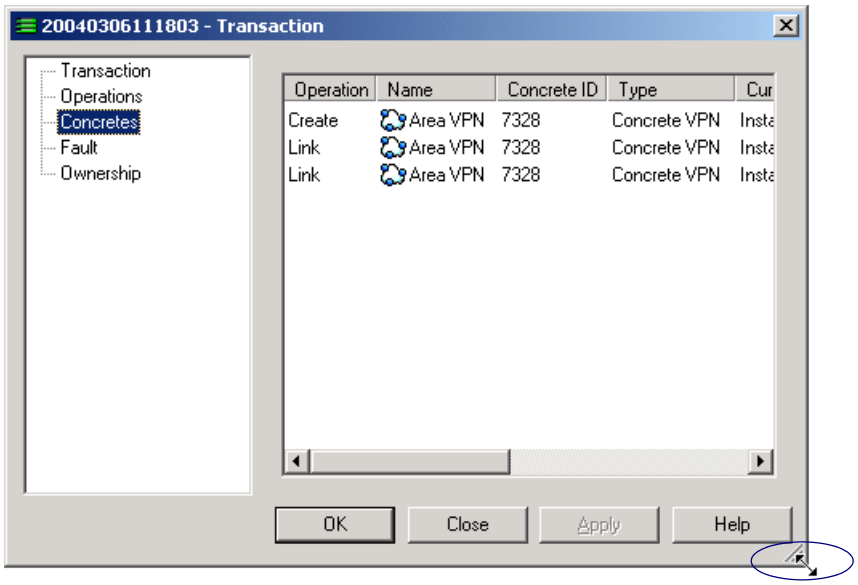
    

    If the object has no children, you can also double-click on the object in the hierarchy or **Details** pane to open the properties dialog box.

Resizing an object's property dialog box

You can resize the property dialog box for an object by clicking and dragging the resize handle in the lower right-hand corner of the dialog box. This is very useful if a dialog box contains information that is not immediately viewable.

Changes to the default dialog box size are persistent — once you set it, Service Activator remembers your preference.



Viewing an object's configuration

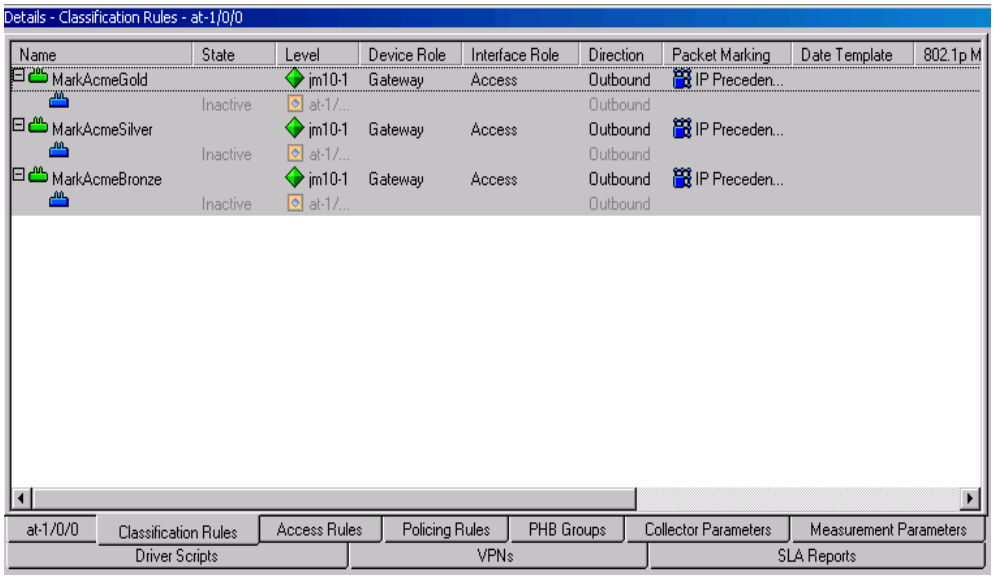
The following objects are referred to as 'policy targets':

- Networks
- Customers
- VPNs
- Sites
- Domains
- Devices
- Interfaces
- Sub-interfaces
- VC endpoints

These objects may have some or all of the following policy types applied:

- QoS and access control policy is defined by rules and PHB groups. For more information, see the *Configuring Policy Services* guide.
- A VPN service is defined by a VPN, TLS, CCC, or Layer 2 Martini VPN object. For more information, see the *Configuring VPN Services* guide.
- Measurement and collector parameters define the measurement type that applies to a policy target and the destination for the exported measurement data. For more information, see the *Network and SLA Monitoring Guide* guide.
- If you are using an integrated reporting tool with Service Activator, reports may also be available. The reports that are generated for a policy target depend on the target's type, the measurement that has been applied to it and the reporting tool used. For more information, see the *Network and SLA Monitoring Guide* guide.
- If a policy target has been affected by a driver script, information about this script is displayed. For more information, see the *Configuration Development Kit Guide*.

The policy elements, measurement and collector parameters associated with an object can be listed in the **Details** pane by double-clicking on the object and selecting the relevant tab.



Domains may have role assignment rules associated with them and these are also listed in the **Details** pane on the global setup window.

Tabbed options at the bottom of the pane enable you to select the policy details you want to view.

To view the configuration associated with an object

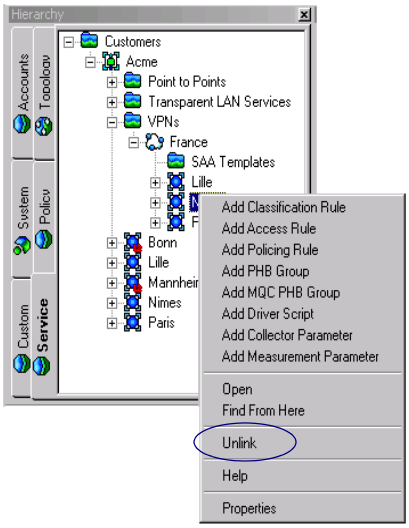
- Double-click on the object in the hierarchy, ancestry or **Details** pane.

  Depending on the object's type, the rules, PHB groups, CCCs, VPNs, driver scripts, measurement and collector parameters and reports that are associated with the object are listed in the **Details** pane. You can view different elements by selecting from the tabs at the bottom of the **Details** pane.

Linking objects

You set up associations between objects by linking them. For example, you link sites to a VPN object to create a VPN. Service Activator shows the link between two objects as a link object, which appears in the hierarchy pane as a copy of the object

you have linked. Unlike a 'standard' object, a link object's pop-up menu includes an **Unlink** option.



This section describes how to link objects using the drag and drop and copy and paste link techniques. You can drag and drop objects to create links between them working in the hierarchy and **Details** panes, or by opening new domain management windows and dragging between windows.

The copy and paste link technique uses options from the **Edit** menu. This involves copying the object and selecting the paste link option over the object to which you want to link the first object.

You cannot link all object types – valid links are dictated by Service Activator's object model. If you are using drag and drop linking, an invalid link is shown by the mouse pointer, which changes over an invalid link target. If you are using copy and paste link, the **Edit** menu's **Paste Link** option is not enabled.

Note that there are other indirect methods of linking objects. You can create links between objects when you select options on an object's property page. For example, when you select a class of service on a PHB group's property page, you effectively create a link between the PHB group and the class of service. Service Activator also creates some links automatically – for example, when you drag discovered devices on to a topology map.

To create a link using drag and drop

1. Organize the display so that the objects to be linked are both visible.

   You can simultaneously display objects in the hierarchy and **Details** panes or by opening and arranging two domain management windows.

2. Drag the source object and drop it on the destination object.

   The mouse pointer changes as you drag the item:

   This pointer appears when you are over a valid destination, that is, one to which the selected object can be linked.

   This pointer appears when you are over an invalid destination, that is, one to which the selected object cannot be linked.

   A new link object appears in the new position in the hierarchy tree.

To create a link using the Copy and Paste Link commands

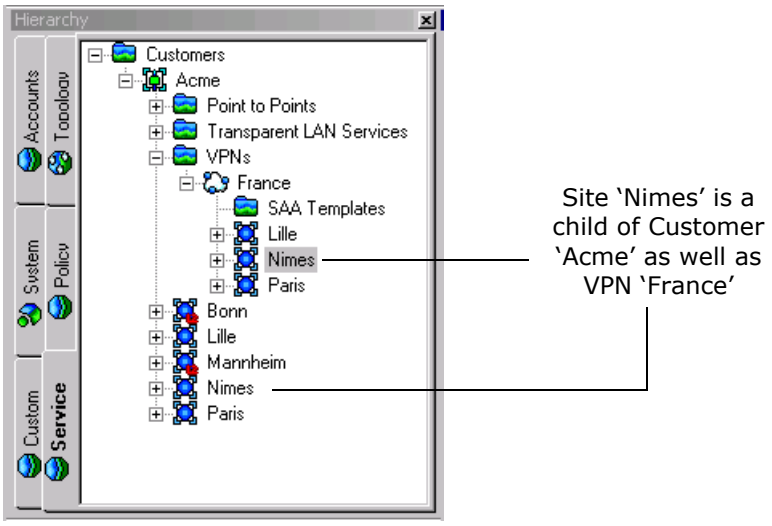
1. Select the source object and select **Copy** from the **Edit** menu, or click the Copy toolbar button.
2. Select the object to which you want to link the copied object and select **Paste Link** from the **Edit** menu.

   If the destination is not valid, the **Paste Link** option is not available.

   When a link has been successfully created, a new link object appears in the new position in the hierarchy.

Unlinking objects

An object may be linked to other objects at multiple points in Service Activator's object model. Where an object is linked to another object, Service Activator displays the linked object as a child of the object to which it is linked. Therefore, an object may appear in multiple locations within Service Activator's user interface.



You can remove links between objects – for example if you have linked two objects by mistake or if you are making changes to the data.

> If an object has been linked to one or more parent objects, you must remove those links before you can delete the object. For example, if you have created a host account named 'zeus' and linked it to an account group, you cannot delete the zeus account until you have removed its link to the account group.

To check whether an object is a child of a parent object

- Right-click on the object to display the object's pop-up menu.

  If the **Unlink** option is listed, the object is linked.

To unlink an object

1. In a domain management window's hierarchy pane, locate the parent object and select the link object listed below it.

   For example, to delete the link between a proxy agent and a device, locate the proxy agent in the hierarchy tree and select the device link object, listed beneath the proxy agent.

2. Do one of the following:

   — From the **Edit** menu or the object's pop-up menu, select **Unlink**.

   — Click on the Unlink button on the toolbar.

   The link is deleted and the link object is deleted from the hierarchy.

   Alternatively, you can drag the object onto a new object – this removes the existing link and links the object to a new parent.

> Note that if the **Unlink** option does not appear in the object's pop-up menu, the object is a real object rather than a link object.

Deleting objects

You can delete objects from the object model, for example a router that does not exist, or a rule component that is no longer used.

Note that you cannot delete system-defined (top-level) folders – these provide a means of grouping objects but are not objects themselves. To delete a user-defined folder, see *To delete a user-defined folder* on page 35.

> Take care when deleting objects – Service Activator does not always display a confirmation message prior to deletion. If you delete something by mistake you can use the **Edit** menu's **Undo** command to restore it provided that you have not saved the current transaction since the object was deleted. You may also be able to rollback the transaction that performed the deletion.
>
> For more information on undoing commands during a transaction see *Creating transactions – the current transaction* on page 55. For information on rolling back a transaction, see *Unmerging or rolling back a transaction* on page 65.
>
> You cannot delete any object currently in use, for example, a Date and Time template that is assigned to a policy rule.

To delete an object

- Select the object in the hierarchy, ancestry or **Details** pane, and do one of the following:
  - — Select **Delete** from the pop-up menu.
  - — Select **Delete** from the **Edit** menu.
  - — Press the Delete key on your keyboard.
  - — Click on the Delete button on the toolbar:

    The object is deleted from the object hierarchy.

Organizing objects into user-defined folders

The following types of objects can be organized into user-defined folders:

- Customers - customer folders are created under **Customer** objects in the **Service** tab
- Sites - customer sites are created under **Site** objectes in the **Service** tab
- Driver Scripts - driver script folders are created under the **Driver Scripts** folder in the **Custom** tab
- PHB Groups - PHBs and MQC PHBs can be organized into folders and sub-folders under the **PHB Groups** folder in the **Policy** tab.

- Classifications - classifications and classification groups can be organized into folders and sub-folders under the **Classifications** folder in the **Policy** tab.
- Classes of service can be organized into folders and sub-folders under the **Classes of Service** folder in the **Policy** tab.
- Roles can be organized into folders and sub-folders under the **Roles** folder in the **Policy** tab.
- Traffic Groups are effectively folders organized under the **Traffic Types** folder in the **Policy** tab.

User-defined folders can contain sub-folders. Drag and drop operations can be used to move folders, or their contents. You can also define folder permissions to restrict access to a subset of users for these folders.

To create a user-defined folder

1. **For Customers:** On a **Domain** detail window or **Network** detail window, select the **Service** tab.

   For Sites: On a **Domain** detail window, select the **Service** tab.

   For Driver Scripts: On a **Domain** detail, **Network** detail window or **Global View** window, select the **Custom** tab.

   For PHB Groups, Classifications, Classes of Service, Roles, and Traffic Groups: On a **Domain** detail window, select the **Policy** tab.
2. Right-click on the parent object for the new folder. This can be the main system-supplied container folder (such as Customers, Driver Scripts or PHB Groups) or an existing user-defined folder.
3. From the pop-up menu, select **Add Folder**. (For Traffic Types, select **Add Traffic Group**.)

   The appropriate Folder dialog box opens, one of the following:

   - Customer Folder
   - Site Folder
   - Driver Script Folder
   - PHB Group Folder
   - Classification Folder
   - Classes of Service Folder
   - Roles Folder
   - Traffic Group
4. Enter an identifying **Name** for the folder and, optionally, accompanying **Remarks**.

5. If you wish to restrict access to the folder, select its **Ownership** property page and specify permissions.
6. Click **OK** and commit the transaction.

To add and remove objects in user-defined folders

To add an object to a user-defined folder, simply drag the object to the target folder. Note that an object can be included in only one folder.

To move an object out of a user-defined folder, drag the object to the new folder, or to the root folder for the objects.

To move a user-defined folder

To move a user-defined folder, click on the folder and drag it to its new parent. The parent can be the root folder for the objects or another user-defined folder of the same object type.

Note: When you move a user-defined folder, its contents (objects and sub-folders) are moved with it.

To delete a user-defined folder

To delete a user-defined folder, click on the folder and either right-click and select **Delete** from the pop-up menu, or click the Delete button in the Service Activator toolbar.

Note: When you delete a user-defined folder, all of its contents (objects and sub-folders) are deleted with it.

To rename a user-defined folder

To rename a user-defined folder, right-click on it and select **Properties** from the pop-up menu. Enter the new name for the folder in the **Name** field, and click **OK**. Commit the transaction.

Navigation

Service Activator offers a range of techniques for navigating the system. Basic navigation techniques include:

- Viewing different object groups by selecting tabs in the domain management window.
- Navigating the hierarchy tree by clicking on the ⊞ and ⊟ icons in the hierarchy or ancestry panes.
- Viewing more information about an object in the **Details** pane by double-clicking on the object in the hierarchy pane.
- Creating different views of the policy domain simultaneously by opening and arranging new windows. For more information about working with windows, see *Service Activator's windows* on page 13.

Synchronizing panes

When you double-click on an object in the hierarchy pane the **Details** pane synchronizes automatically and the selected object's details are displayed in the **Details** pane. However, you can also single-click on objects to browse the hierarchy tree but leave the view in the **Details** pane static.

The synchronize panes feature enables you to resynchronize the view shown in the hierarchy pane with the **Details** pane. It returns you to the tab whose details are shown in the **Details** pane, with the tab in the same state as last viewed.

To synchronize the hierarchy pane with the Details pane

- Click on the Synchronize Panes button on the toolbar:

  Service Activator synchronizes the hierarchy pane with the **Details** pane.

  Synchronization only occurs if you have navigated objects in the hierarchy pane using the single-click technique.

Using the navigation buttons

Whenever you open a new window and browse through the object model, Service Activator keeps a record of the object views you have visited, that is, your browse sequence. You can move through the recorded browse sequence using the toolbar's navigation buttons.

The Up button is only enabled when you drill down through a map on the **Topology** tab. When selected, it takes you up one level in the topology hierarchy and displays details of the selected object's parent. However, if you have a segment selected, the Up button is not enabled. This is because a segment may have a number of connected interfaces and therefore has multiple parents.

To navigate a recorded browse sequence

- Use the Back and Forward browse buttons to step backwards and forwards through the browse sequence.

- Use the Up button to return to a previous map view after you have drilled down a level.

Changing views

The **Details** pane can display details of selected objects in several formats. You can switch between views using toolbar buttons:

Report View button – displays the contents of the **Details** pane in the form of a list of objects, together with additional information about the object's properties. You can sort on any column by clicking on its title.

Compact List View button – displays the contents of the **Details** pane in the form of a multi-column list of objects.

Map View button – displays the network topology map or the VPN map in the **Details** pane. This button is only available when you have selected an object that has an associated map, that is, a network object or a VPN object.

Status Context button (multi-colored) – displays the map view showing the status of all network elements

Policy Context button (gray) – displays the map view showing the policy role of all network elements

Service Activator remembers the current view when you close the User Interface. On restarting the GUI, the same view is used as the preferred view.

You can also change the type of information that is displayed on the topology map, viewing information about the status of the map's devices or the policy implemented at each device. For more information, see *How objects are represented* on page 186.

Changing the appearance of the user interface

Service Activator gives you flexibility over the appearance of the user interface. You can hide and show window panes, arrange panes or move elements around the window. The only window element that you cannot hide is the **Details** pane.

Hiding and showing windows and window elements

You can hide and show windows and window elements by selecting options from the **View** menu. You can hide or show the following elements:

- Toolbar
- Status bar
- Hierarchy tree pane
- Ancestry tree pane
- Map palette
- Global setup window
- Current faults pane
- Statistics summary pane
- System statistics window

To hide or show a window element

- From the **View** menu, select the element you want to hide or show.

  An element will be displayed if its name is checked.

Changing the size and position of a pane

You can change the size of any of the displayed panes and, for selected panes, drag them to a new position.

To change the relative size of panes

- Click on and drag the vertical and horizontal bars separating the relevant panes.

To move and dock a pane

- Do one of the following:

— On the hierarchy, ancestry or palette pane, click on the pane's title bar and drag it to a new position.

— On the current faults pane, click on the pane's docking bar and drag the pane to a new position.

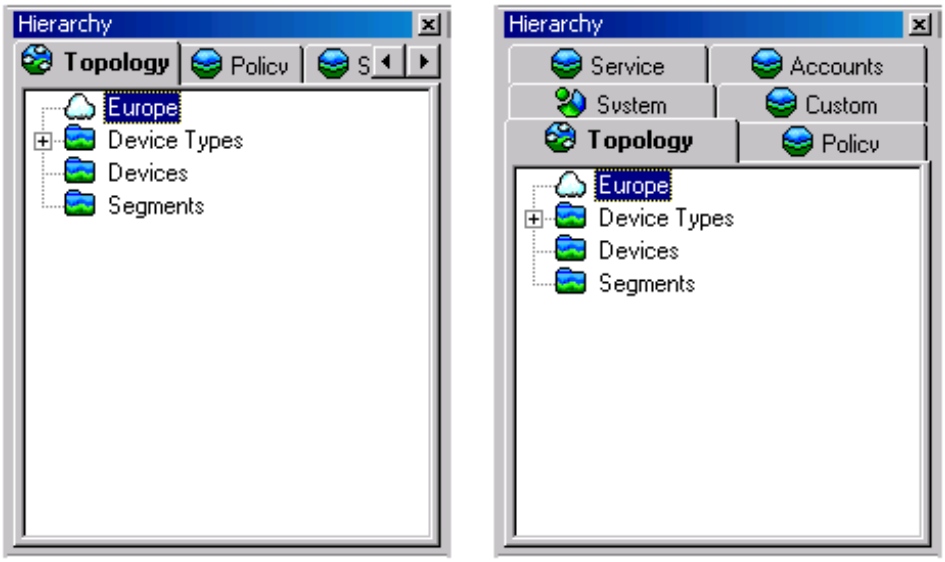
Depending on the pane's new location, Service Activator may change the pane's orientation and size.

If you drag the pane outside the domain management window, it becomes a separate, floating window. To return the pane to the window, position the pane over the window's edge.

Changing the hierarchy pane's tab position and style

You can customize the appearance of the hierarchy pane's tabs by selecting their location on the pane. If tabs are displayed at the top or bottom of the pane, you can also specify whether they appear in a single line or multiple lines. The default is multiple lines.

The following illustrate single lines and multiple lines:



To change tab position

1. Right-click on the hierarchy pane's title bar and select **Display Tabs** to display the pane's pop-up menu.
2. Select **Top**, **Bottom**, **Left** or **Right**.

   The tabs move to the position you selected.

To change the tab display style

- Right-click on the hierarchy pane's title bar and select **Multilined**.

  The **Multilined** option is only displayed if tabs are displayed at the top or the bottom of the pane.

  The tab style changes depending on whether the **Multilined** option is selected or deselected.

Searching Service Activator

Service Activator provides the following options for searching:

- Search for objects based on a text string

  The text string may be in the form of a regular expression. You can choose whether to expand or restrict the search to policy objects, including inherited policy objects. You can also restrict the search by object type, state and/or role.

- Search for concrete configuration objects

  This search locates concrete rules, PHB groups, driver scripts, VPNs and CCCs. You can restrict the search by state.

Conducting a text-based search

A text-based search is based on a string that may include a regular expression. The search operates from a user-selectable point in the hierarchy. Service Activator attempts to match the search string against all attributes of an object. This means, for example, that you can search for a device by name, role or IP address, or a rule by its name, status or associated traffic type.

The search text may include any alphanumeric or punctuation characters up to a maximum of 64 characters.

By default, Service Activator attempts to match the search text at any point in a text string. For example, the search text '165' will be matched against 168.165.0.4. Alternatively, you can specify that the search text is matched against whole words only. You can specify whether the search is case sensitive.

Service Activator remembers previous search start points and displays them in a pull-down list in the **Find** dialog box. This list is cleared if the user interface is stopped and restarted.

Note the following:

- Service Activator searches the contents of the Object Model. Note that the System Logs are not held in the Object Model, and you therefore cannot search their content from the user interface.

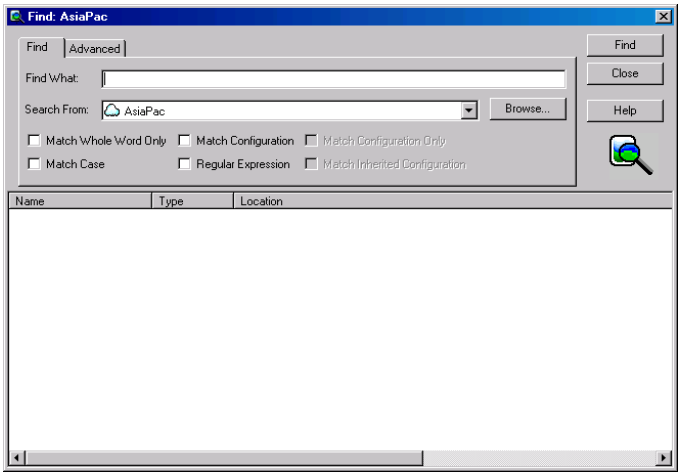
- You can search within a single domain only.

Note: For complete dialog box and property page descriptions, refer to the *Online Help*.

To conduct a text-based search

1. From the **Tools** menu, select **Find**.

   The **Find** dialog box opens.

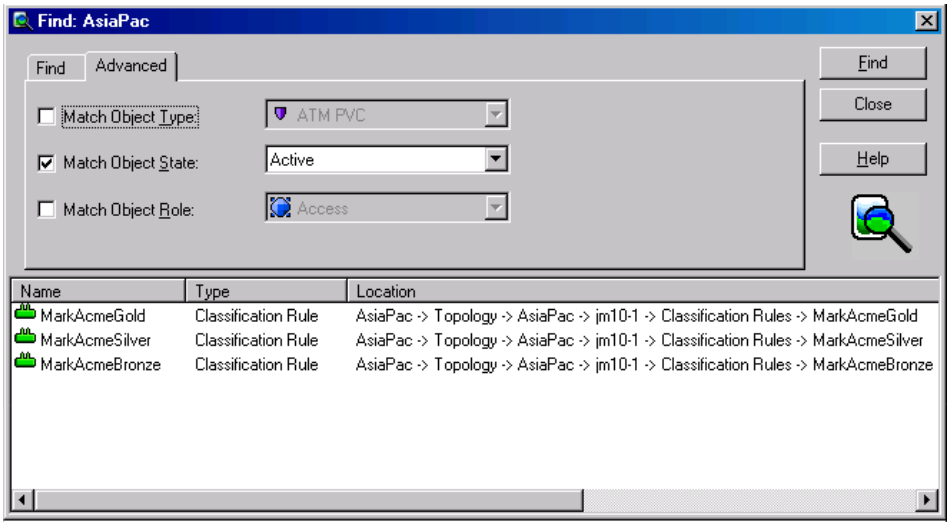


2. In the **Find What** field, specify the search string.
3. Optionally, select options including **Search From, Match Whole Word Only, Match Case, Regular Expression, Match Configuration, Match Configuration Only, Match Inherited Configuration**.
4. If you wish to restrict the search by object type, state or role, select the **Advanced** tab and select from options, including **Match Object Type, Match Object State** and **Match Object Role**.
5. Click **Find**.

   Service Activator searches for the specified search criteria. As matches are found, the search results are added to the list in the lower half of the **Find** dialog box.

Depending on the amount of information held by Service Activator, a search may take some time to complete. You can stop the search by clicking the **Cancel** button that is displayed while the search is in progress.

When the search is complete the results are shown in the lower half of the **Find** dialog box.

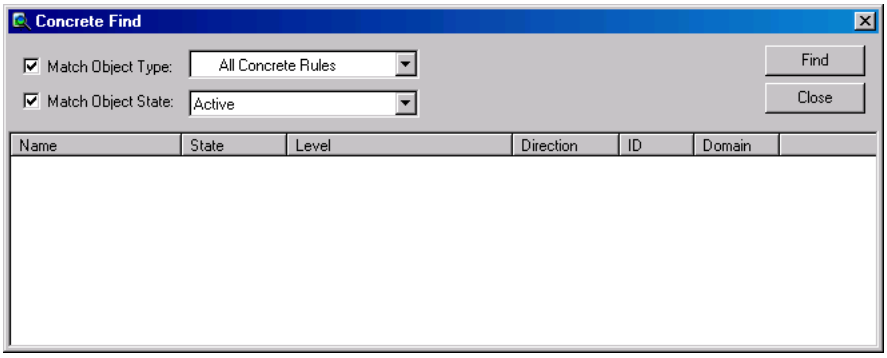


By default, found objects are sorted by object name. You can:

- Change the sort order criteria by single-clicking on a column title.
- Display a found object in the hierarchy pane by double-clicking on it. Service Activator highlights the object in the hierarchy pane and, where relevant lists children of the found object in the **Details** pane.
- Open a new window on a found object by selecting **Open** from the object's pop-up menu.
- View an object's properties by selecting **Properties** from the object's pop-up menu.

Searching for concrete objects

A concrete object search attempts to locate concrete objects that match the specified state. This means, for example, that you can search for all concrete access rules whose state is 'Installed' or search for VPNs whose state is 'Active'.



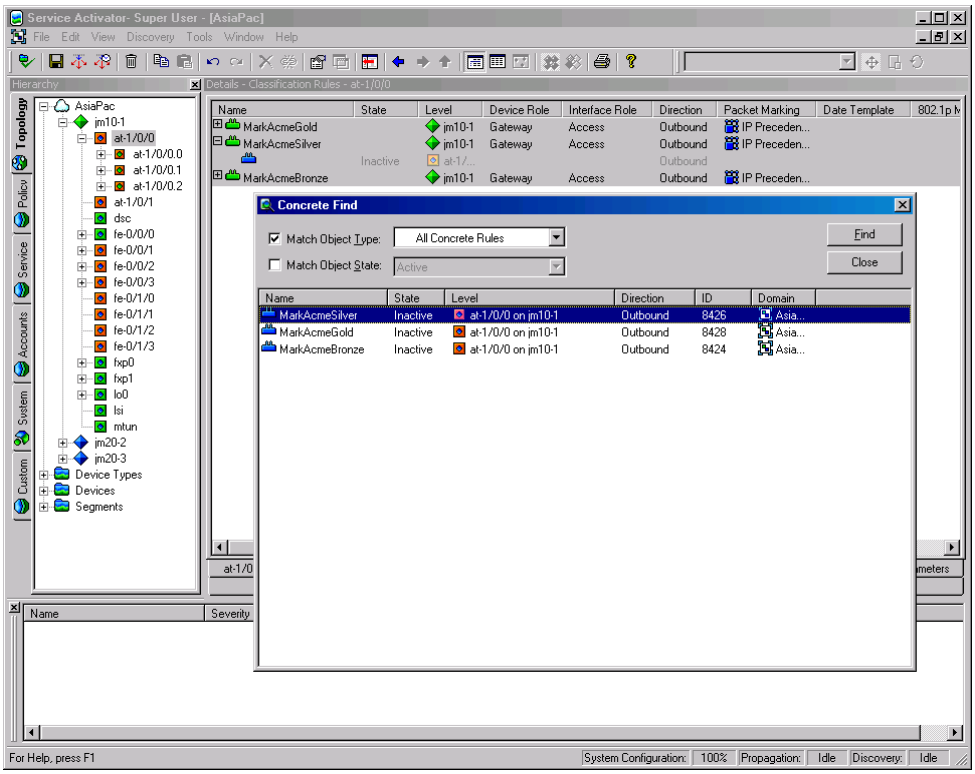
Note: For complete dialog box and property page descriptions, refer to the *Online Help*.

To search for concrete objects

1. Do one of the following:
   - Choose **Concrete Find** from the **Tools** menu.
   - In the Statistics Summary pane, double-click on a cell. (For information on viewing the Statistics Summary pane, see *Hiding and showing windows and window elements* on page 38.)

     The **Concrete Find** dialog box opens.
2. Select the **Match Object Type** checkbox and select a concrete object type from the pull-down list.
3. If you wish to restrict the search by status, select the **Match Object State** checkbox and select a state from the pull-down list.
4. Click **Find** to start the search.

   Double-click on a found concrete object to view the point at which it applies in the hierarchy pane and details of the concrete object in the **Details** pane.

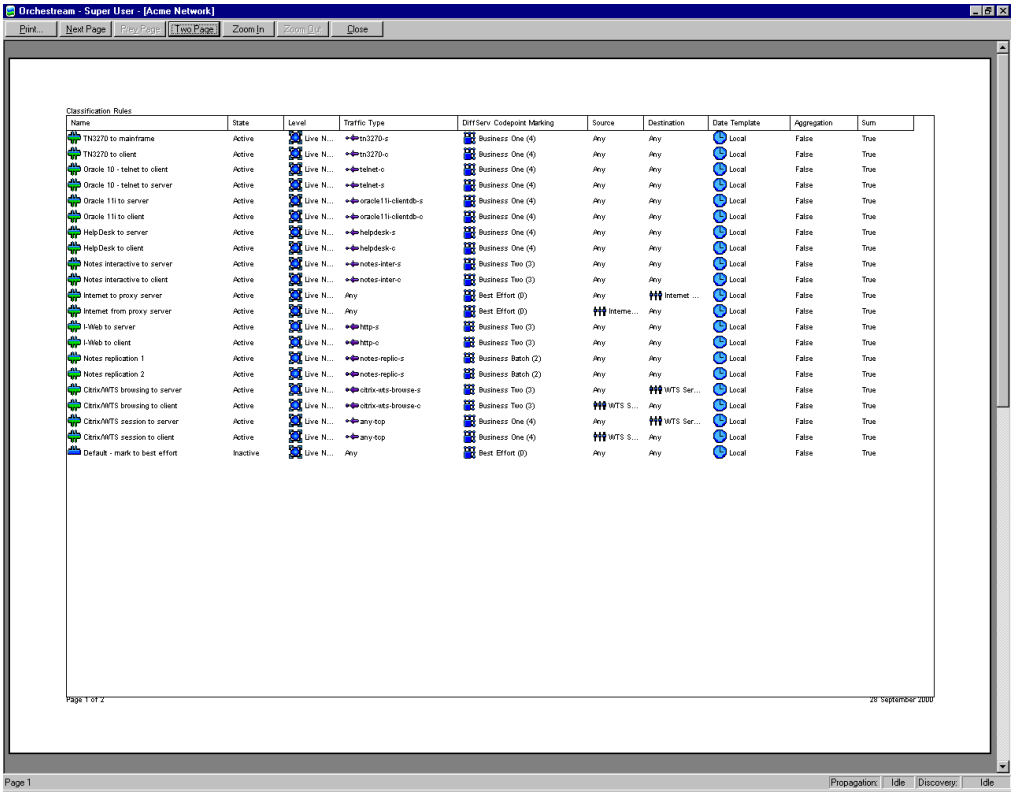


Searching committed transactions

Service Activator supports the ability to search through committed transactions. For details, refer to *Searching committed transactions* on page 68

Printing

You can print the contents of the **Details** pane to produce, for example, a printout of the system log file or topology map. Each page has a ruled border with information about the page content printed round the border edge. Details of the page layout, including margin size, can be specified and the printed layout previewed.



When printing a topology map that occupies multiple pages, there is some overlap between pages.

For information on printing reports, see the *Network and SLA Monitoring Guide* guide.

To print the Details pane

1. From the **File** menu, select **Print**.

   The **Print** dialog box opens.

2. Adjust the print settings if required.

3. Click **OK**.

To define the print page setup

1. From the **File** menu, select **Page Setup**.

The **Page Setup** dialog box opens.

2. Select the **Paper**, **Orientation** and **Margins** as appropriate.
3. If you want to select a printer, click the **Printer** button, select the printer and click **OK** to confirm the printer selection.
4. Click **OK**.

To display a print preview of the Details pane

1. From the **File** menu, select **Print Preview**.

   Service Activator displays a preview of the pane printout. You can move between pages using the **Next Page** and **Prev Page** buttons. You can view the image in more or less detail using the **Zoom In** and **Zoom Out** buttons.
2. If you want to print the previewed image, click **Print** to display the **Print** dialog box and **OK** to confirm the print request.
3. If you want to close the previewed image, click **Close**.

Chapter 3

Transactions

Service Activator uses a transaction-based model for updating the system and implementing configuration changes. This means that the changes you make through the user interface can be implemented immediately or saved in a pending state for future implementation. Configuration changes can be broken down into a number of transactions and implemented in a controlled manner.

This chapter:

- Provides a definition of transactions in Service Activator
- Suggests alternative workflows for working with transactions
- Describes the local and common object models that are central to Service Activator's transaction handling
- Provides 'how to' information for working with transactions, including viewing, creating and committing transactions
- Describes how to roll back the configuration changes made by a transaction

About transactions

A transaction is a set of changes made through Service Activator's user interface or via the OSS Integration Manager (OIM). These changes may include logical changes, such as creating a new domain, user group or users, as well as changes that affect device configuration, such as setting up a VPN or security policy.

As you make changes through the user interface, Service Activator adds those changes to a transaction referred to as the 'current transaction'. You can choose when to stop the current transaction and how to handle it:

- Saving the transaction in a pending state
- Committing the transaction's changes immediately and configuring the network
- Scheduling the transaction to be implemented at a specified date and time
- Discarding or aborting the transaction

These options support two methods for working with transactions. You can follow a one-stage update model, in which changes are made through the user interface and implemented immediately. Alternatively, you can follow a more secure two-stage update model, in which transactions are created and saved in a pending state, for checking and committing at a later date. You may wish to use the two-stage update model in combination with user security options – allowing all users to create transactions and a subset of users to check and commit their work. For information on setting permissions on Read Write groups, see *Setting up Read Write group permissions* on page 95.

Whether you choose to follow a one or two-stage update model, it may be possible to roll back a transaction's changes and remove any associated configuration from the network.

> If other changes made in Service Activator are dependent on the modifications made by a transaction, you may be unable to perform a roll back.

Note that Service Activator always attempts to reflect the current state of the network through the user interface. The user interface only differs from the network's current state when a user is in the middle of the current transaction. As soon as he or she saves or schedules the transaction, the user interface reverts to the state of the network as it is currently configured. For more information, see *Local and common object models* on page 51.

Transaction workflows

There are two potential workflows for transactions:

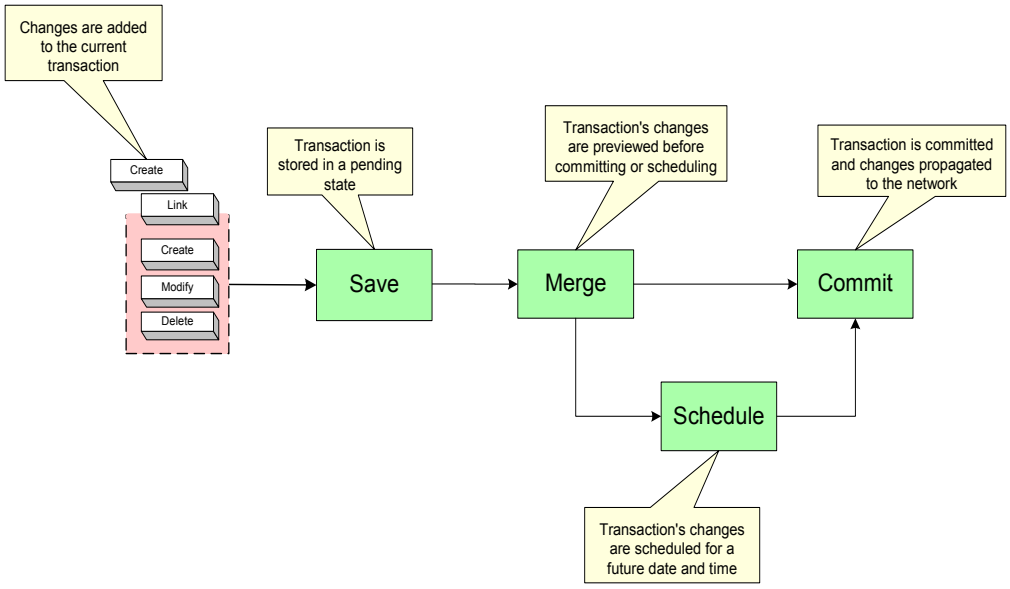
- One-stage update model, where a user makes some changes through the user interface and immediately commits the changes.
- Two-stage update model, where one user saves the changes as a transaction and another merges the transaction to check its changes and finally commits it.

The one-stage commit model

In the one-stage commit model, changes made in the current transaction are committed immediately. This model is useful for initial system setup and demonstration purposes.

The two-stage commit model

This model is illustrated in the following diagram.



In this model, changes made through the user interface are added to the current transaction, which is saved and stored in a pending state. After saving, the transaction's changes must be checked by merging (see *Merging or previewing a transaction* on page 63) before it can be committed or scheduled for automatic commitment at a specified date and time. Merging performs a validation check and enables you to preview changes before committing them. For information on merging, see *Merging or previewing a transaction* on page 63.

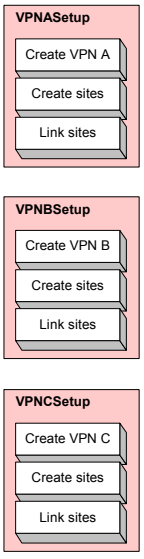
This model provides a granular and secure method for making updates and configuring the network. Changes can be broken down into a number of transactions and implemented in a controlled manner. For example, a series of policy rules can be held in a number of transactions and their implementation phased in over a period of time.

The action of creating and saving the transaction and committing the transaction may be performed by different users. In a distributed system, once a user has saved a transaction, other users working on remote user interfaces can see the transaction and potentially implement it. We therefore recommend you give the transactions you create meaningful names.

To ensure that only trusted users can implement a transaction, you can set permissions on the actions associated with transactions. So, for example, you can allow all users to save transactions in a pending state but restrict the ability to implement transactions to a subset of users. For information on setting user access levels, see Chapter 4, *Setting Up Users*.

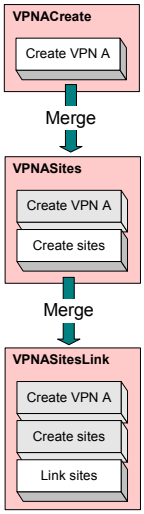
There are two options for creating and saving transactions:

- Create discrete transactions with no overlap – each transaction is self-contained and performs distinct operations. For example, VPNASetup creates VPN A and the sites associated with it and links the sites with the VPN, VPNBSetup creates VPN B and the sites associated with it, and so on.



- Alternatively, you can create a transaction and save it in a pending state and then extend the transaction by merging it into the current transaction before performing additional tasks. This creates a series of transactions, where each transaction builds on the previous one. For example, the first transaction creates

a VPN (VPNACreate), the second transaction merges VPNACreate and creates the sites associated with it (VPNASites), the third transaction merges VPNASites and links the sites with the VPN (VPNASitesLink), and so on.



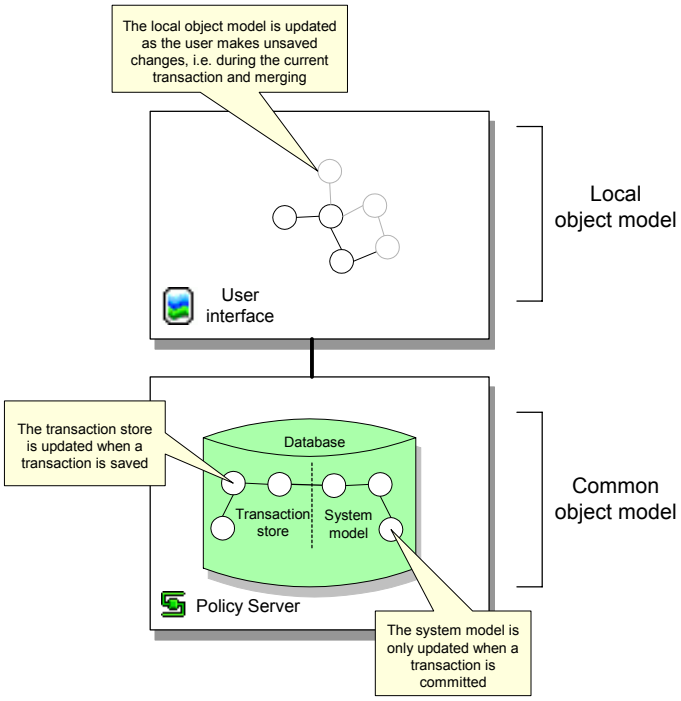
We recommend you adopt a naming convention when using the two-stage commit model, for example, by using the same prefix for related transactions.

Local and common object models

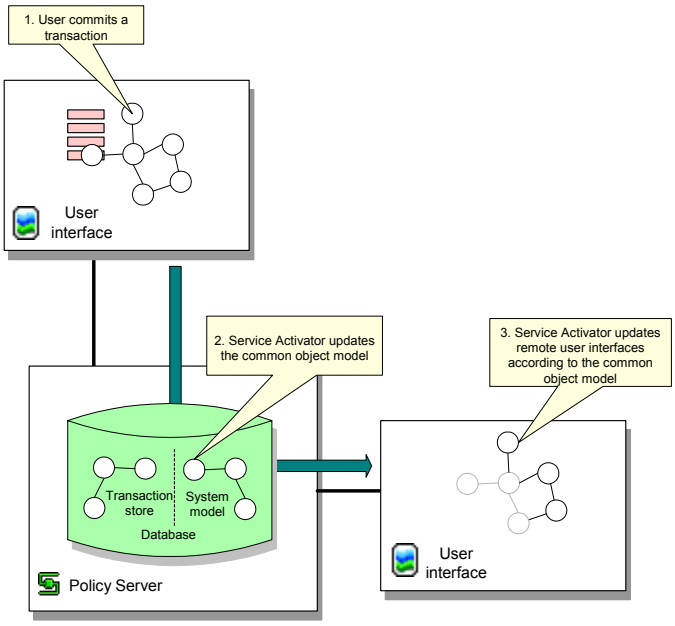
The changes made through the user interface and held in a transaction represent changes to Service Activator's object model. The common object model is maintained by the policy server and stored in the database, often located on the policy server host machine. The object model can be divided into two parts:

- The transaction store is updated when a transaction is saved – a new transaction object is created in the transaction store and is viewable by all user interfaces. For information on the transaction store, see *The transaction store on page 53*.
- The system model is updated with a transaction's changes when the transaction is committed.

As changes are made on a user interface host machine, Service Activator makes the changes locally. We refer to this 'version' of the object model as the local object model.



Each user interface's local object model is updated according to the common object model whenever a transaction is committed. The transaction may have been committed on the local host machine or on a remote user interface host.

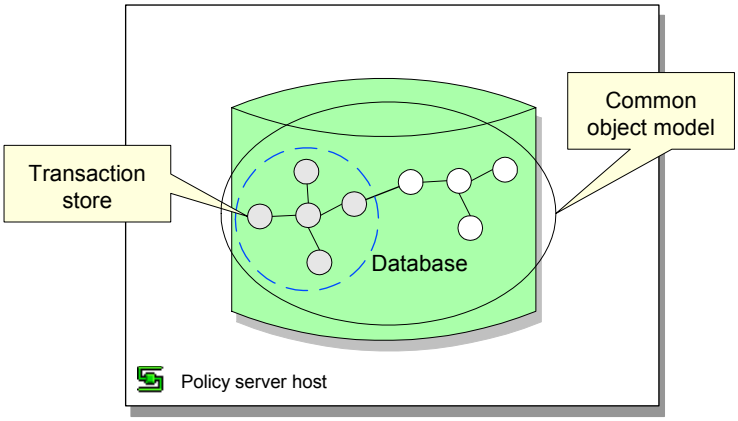


When a transaction is committed, Service Activator compares the common object model to each user interface's local object model and resolves any inconsistencies by removing them from the local object model.

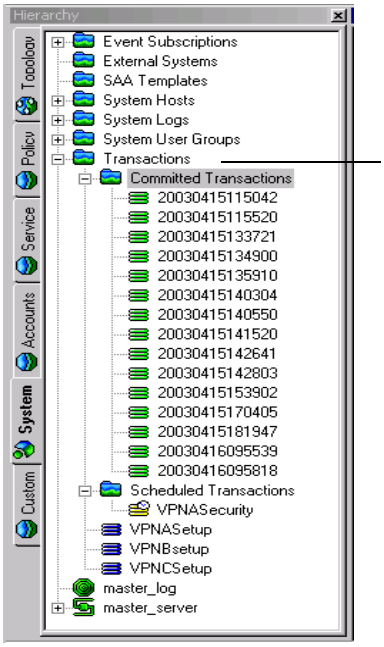
> A user with unsaved changes may lose those changes if they conflict with changes that have been committed to the common object model.

The transaction store

As part of the common object model, Service Activator maintains a transaction store which holds pending, scheduled and committed transactions. The store is updated whenever a transaction is saved or its status changes.

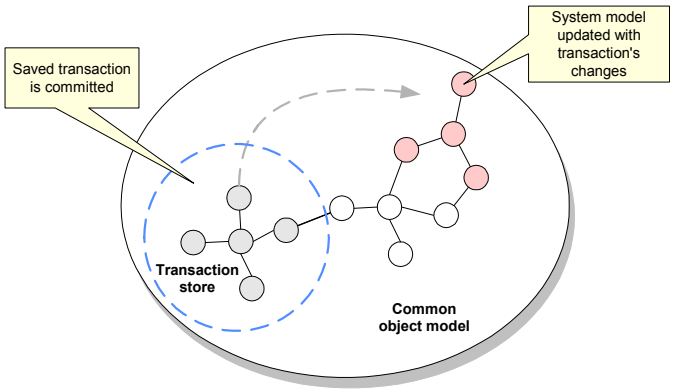


You can view the content of the transaction store on the **System** tab in the **Transactions** folder.



The Transactions folder displays the contents of the transaction store.

The system model section of the common object model is only updated with a transaction's changes when the transaction is committed.



Working with transactions

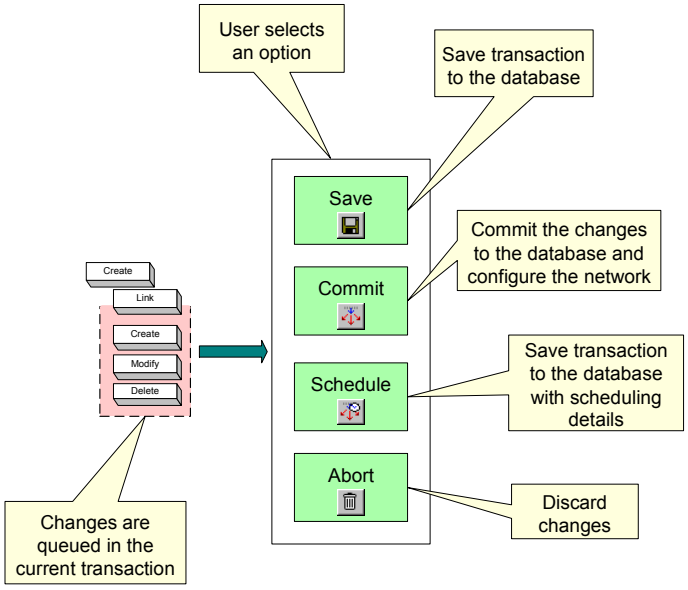
This section describes how to work with transactions. It explains how to create, commit, save and schedule, merge and unmerge, and delete transaction.

Creating transactions – the current transaction

As you work in the user interface the changes you make to the object model are held in the current transaction. This transaction acts as a cache, queuing the changes you make until the transaction is stopped. The options available for stopping the transaction may include some or all of the following:

- Saving the transaction – details of the transaction are added to the database but the transaction's changes to the object model and device configuration are not implemented
- Committing the transaction – the common object model is updated with the transaction's changes and any associated configuration is propagated to the network
- Scheduling the transaction – similar to saving the transaction, but a date and time are set for implementing the transaction's changes

- Aborting the transaction – the changes held in the current transaction are discarded

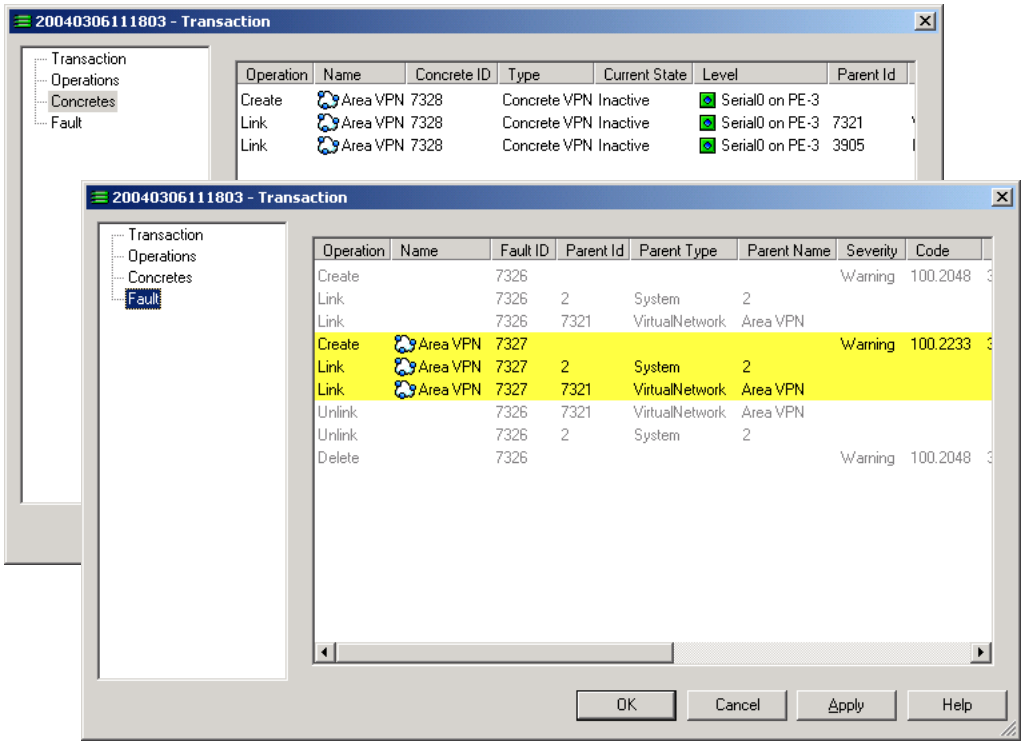


Some or all of these options may be available depending on your security rights. For example, some users may be able only to save the current transaction while users with a higher security level can commit or schedule transactions. For information on defining user security levels, see *Setting Up Users* on page 77.

> You can undo commands during the current transaction up to the start of the transaction.
>
> Network discovery and related tasks, such as fetching capabilities, cannot be carried out part way through the current transaction. These tasks can only be performed immediately after a transaction has been saved or committed and before a new transaction is started.

Committing a transaction

When you commit a transaction, Service Activator validates the transaction's changes. The **Transaction** dialog box opens, showing details of the transaction including any concrete objects or faults that would be generated by confirming the commit.



After reviewing the potential concrete objects and faults, you can choose whether to proceed with the commit or cancel.

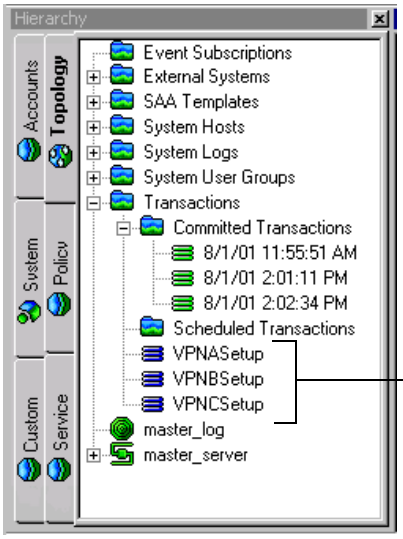
If you proceed with the commit, Service Activator updates the common object model with the transaction's changes and any configuration changes are propagated to the network. If there are other users working on remote user interfaces, Service Activator updates the object model on each remote user interface according to the updated common object model. For information on local and common object models, see *Local and common object models* on page 51.

If you cancel the commit, you can revise details of the policy or the points at which it is applied before trying to commit the transaction again.

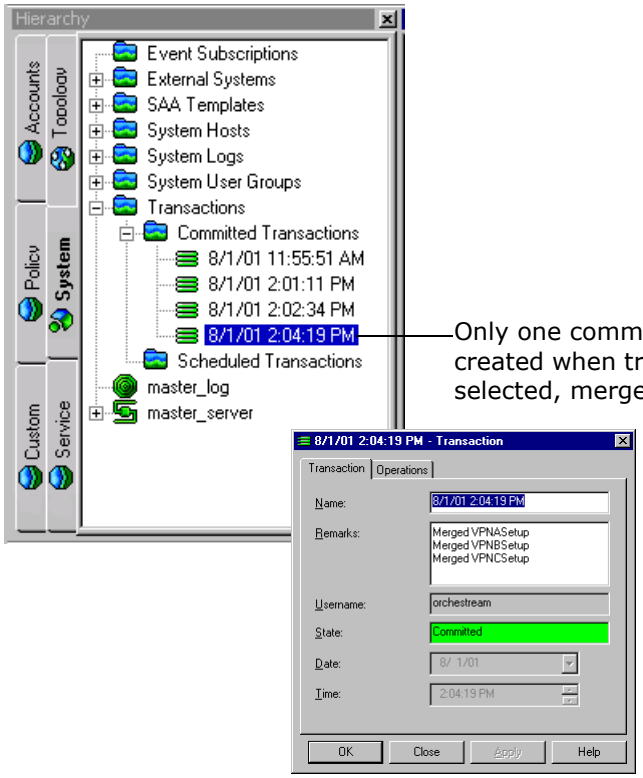
> Before you can commit a saved transaction you must first merge it. For information on merging, see *Merging or previewing a transaction* on page 63.
>
> If a user on a remote machine is creating a transaction that conflicts with the updated object model, a warning message is displayed and any unsaved changes made by that user are discarded.

Whenever you commit a transaction, Service Activator creates a new transaction object in the **Committed Transactions** folder. The transaction bundles the actions that have been committed.

Note that a single committed transaction object may correspond to a number of previously saved and merged transactions. For example, if you select a number of transactions, merge them and commit them, only one committed transaction object is created for those transactions.



Transactions VPNASetup, VPNBSetup and VPNCSetup are listed separately as pending transactions.



Only one committed transaction is created when transactions are multi-selected, merged and committed.

Immediately after committing a transaction the transaction buttons on the toolbar are disabled. When the next change is made in the user interface, Service Activator starts a new current transaction and the toolbar buttons become enabled once again.

If necessary, you can remove a committed transaction's changes from the object model and any related configuration from the network by unmerging the transaction – see *Unmerging or rolling back a transaction* on page 65.

To commit a transaction

1. From the toolbar, select the **Commit** button.

   The **Commit** button is only enabled if a current transaction exists – that is, you have made unsaved changes in Service Activator – and your user security level permits you to commit transactions. The **Transaction** dialog box opens.

   Service Activator applies a default name that indicates the date and time at which the transaction was committed and populates the **Remarks** field with

information about the committed transaction, listing and naming any merged transactions. Changes that have not been saved as a transaction before being committed are listed as 'unidentified user action(s)'.

2. Select the **Concretes** property page and review the concrete objects that will be created if you proceed with the commit.

   Double-clicking on a listed concrete object displays details of the relevant policy target in the **Details** pane.

3. Select the **Fault** property page and review the faults that would result from committing the transaction.

4. If you wish to cancel the commit as a result of your review, select **Cancel**.

5. If you wish to proceed with the commit, optionally change the transaction's default details on the **Transaction** property page before selecting **OK**.

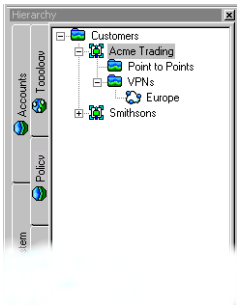
   The transaction is saved to the database, the common object model is updated and any related configuration changes are propagated to the network. The common object model is pushed to remote user interfaces.

Saving a transaction

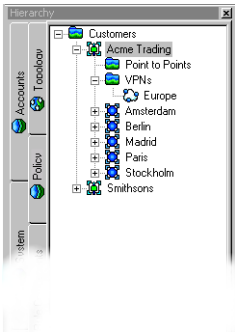
Saving a transaction puts the changes the transaction makes 'on hold' for implementation at some future date or time.

When you save the current transaction, Service Activator creates a transaction object and adds it to the common object model's transaction store. The object model changes associated with that transaction are not executed and no configuration changes are propagated to the network.

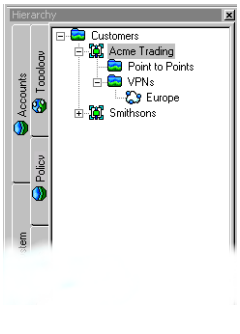
For the user who saves or schedules the current transaction, the user interface reverts to show the state of the network as it is currently configured – that is, the state of the system held by the common object model. In the following example, five sites are created through the user interface and the current transaction is then saved and held in a pending state.



1. Before the transaction is started.



2. After starting a new transaction and creating five sites – Amsterdam, Berlin, Madrid, Paris and Stockholm. The user interface shows the changes made in the current transaction.



3. After saving the transaction, the user interface reflects the network as it is currently configured. Sites created in the saved transaction therefore do not feature in the user interface.

When you save the current transaction, its status changes to Pending.

After saving a transaction you can:

- Merge it into the current model – this is an essential step before committing the transaction. For information on merging, see *Merging or previewing a transaction* on page 63.
- Schedule the transaction – see *Scheduling a transaction* on page 62.
- Delete the transaction – see *Deleting a transaction* on page 66.

To save a transaction

1. On the toolbar, select the **Save** button.

   The button is only enabled if there are unsaved changes in the user interface. The **Transaction** dialog box opens.
2. Enter the **Name** and **Remarks**.
3. Select **OK**.

The transaction is saved to the common object model's transaction store. The user interface reverts to show the state of the network as it is currently configured – that is, the state of the system held by the common object model.

> If a transaction incorporates a merged transaction – for example, a pending transaction was merged during the current transaction – icons representing both the merged transaction and the saved transaction are displayed on the **System** tab when the 'parent' transaction is saved.

Scheduling a transaction

As for saved transactions, a scheduled transaction's changes are put on hold but the date and time is set for implementing the transaction's changes. Depending on your user security level, you may be able to:

- Schedule the current transaction
- Schedule a pending, merged or unmerged transaction

When you schedule a transaction, its status changes to Scheduled.

If a scheduled transaction runs successfully, Service Activator updates the common object model and propagates the changes to the network. A new committed transaction object is added to the transaction store and displayed in the **Committed Transactions** folder.

If a scheduled transaction fails, the transaction is moved to the list of pending transactions and its status changes to Failed.

After scheduling a transaction you can:

- Leave the transaction to be committed at the allocated date and time
- Merge and commit the transaction manually – see *Merging or previewing a transaction* on page 63 and *Committing a transaction* on page 56
- Unschedule the transaction – see page 63

To schedule the current transaction

1. From the toolbar, select the **Schedule** button.

   The **Schedule** button is only enabled if you have started a transaction. The **Transaction** dialog box opens.

2. Enter the details including **Name, Remarks, Date** and **Time**.
3. Click **OK**.

   The current transaction is saved to the database. The user interface reverts to show the state of the network as it is currently configured – that is, the state of the system held by the common object model.

To schedule a pending transaction

1. On the **System** tab, open the **Transactions** folder and select a pending transaction.

2. From the transaction's pop-up menu, select **Schedule**.

   The **Transaction** dialog box opens.
3. Use the **Date** and **Time** fields to specify when to commit the transaction.
4. Select **OK**.

   Service Activator moves the transaction to the **Scheduled Transactions** folder.

To unschedule a transaction

1. On the **System** tab, open the **Transactions** folder and select a scheduled transaction.
2. From the transaction's pop-up menu, select **Unschedule**.

   Service Activator moves the transaction out of the **Scheduled Transactions** folder and changes its status to Pending.

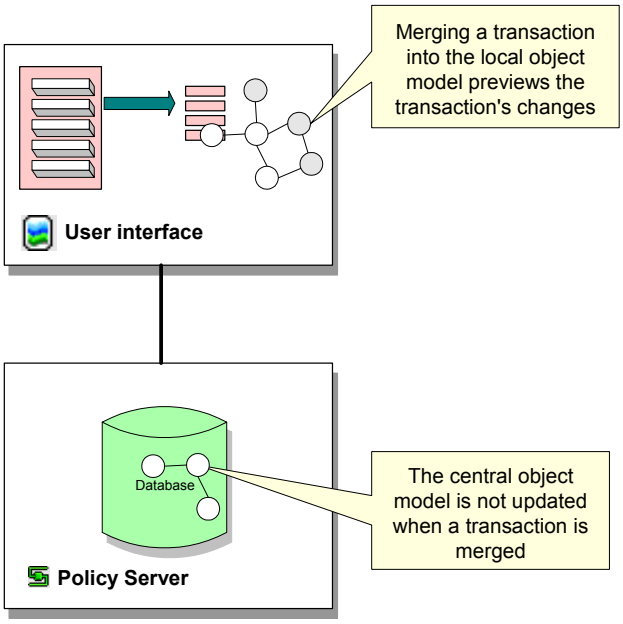
> You must select the **Commit** button after scheduling or unscheduling a transaction in order to implement the scheduling details.

Merging or previewing a transaction

If a transaction has been saved or scheduled, it is possible to preview its changes before committing the transaction.

> For saved transactions, merging is a required step before the transaction can be committed.

Previewing a transaction merges its changes into the local object model – that is, the object model maintained by the local user interface. The merge does not affect the common object model, and the results are therefore not reflected in remote user interfaces. For information on local and common object models, see page 51.



During the merge, Service Activator tests the validity of the transaction against the local object model. If there is a conflict, Service Activator abandons the merge and reports an error – this occurs, for example, if you attempt to merge a transaction that creates an object that already exists.

You can merge pending or scheduled transactions. If you merge a scheduled transaction, Service Activator removes its associated schedule details – the transaction can be committed manually or rescheduled if required.

You can multi-select a number of transactions and merge them in one step. Service Activator merges the transactions in the order in which they were selected. For information see *Selecting transactions* on page 74.

After merging a transaction you can do one of the following:

- Commit the merged transaction's configuration changes – see *Committing a transaction* on page 56
- Perform additional tasks and save or commit the new transaction – see *Saving a transaction* on page 60 and *Committing a transaction* on page 56

- Abort the current transaction – this effectively unmerges the merged transaction and removes its changes from the local object model – see *Discarding the current transaction* on page 66

To merge a transaction

1. On the **System** tab, open the **Transactions** folder and select a pending transaction.
2. From the transaction's pop-up menu, select **Merge Changes**. The changes associated with the transaction are merged into the local object model. The merged transaction's icon is no longer displayed on the **System** tab.

Unmerging or rolling back a transaction

When you roll back a transaction, its changes are removed from the object model and, where configuration has been installed on network devices, it is removed.

The ability to roll back a committed transaction depends on whether subsequent changes are dependent on that transaction's changes. For example, a transaction that creates a VPN to which sites and interfaces have subsequently been linked cannot be rolled back. In this case, the user interface displays an error message.

Rolling back a transaction is a two-step process that consists of:

1. Unmerging the transaction from the local object model
2. Committing the unmerge to update the common object model and remove any associated configuration details from the network. For information on committing transactions, see *Committing a transaction* on page 56.

To unmerge a transaction

1. On the **System** tab, open the **Transactions** folder and select a committed transaction.
2. From the transaction's pop-up menu, select **Unmerge Changes**.

   Service Activator attempts to remove the transaction's changes from the local object model. A message indicates whether the action was successful.

Discarding the current transaction

Aborting a transaction stops the current transaction and discards its changes.

To abort the current transaction

1. On the toolbar, select the **Abort** button.

   Service Activator asks you whether you want to lose your changes.

2. Select **Yes**.

   The current transaction is stopped and its changes discarded. The user interface reverts to show the state of the network as it is currently configured – that is, the state of the system held by the common object model.

Deleting a transaction

You can delete a Pending, Failed or Scheduled transaction.

> If you want to undo the changes made by a transaction, unmerge the transaction and commit the change. For information, see *Unmerging or rolling back a transaction* on page 65.

To delete a transaction

1. On the **System** tab, open the **Transactions** folder and select a transaction.
2. From the transaction's pop-up menu, select **Delete**.

> You must select the **Commit** button after deleting a transaction in order to implement the deletion.

Managing transactions

This section describes broad management tasks associated with transactions. It describes how to:

- Run a user interface in confirmed commit mode
- View a list of transactions
- Search committed transactions
- View transaction details
- Check which user committed a transaction
- Specify the transaction archive limit

Running in confirmed transaction mode

By default, a transaction that has been committed is queued and processed by the policy server as a background task. This means that a user can continue working in the user interface and commit further transactions while the policy server saves data to the database. However, if the policy server fails any transactions that it had not yet written to the database will be lost.

In confirmed transaction mode, a user who has committed a transaction cannot make any further changes in the user interface until the transaction has been successfully saved to the database.

> If your deployment integrates Service Activator with other OSS systems, it is particularly important that these systems are guaranteed persistence of data. This ensures synchronization between systems. You may therefore wish to consider running Service Activator user interfaces in confirmed commit mode in this scenario.

To run a user interface in confirmed transaction mode

1. From the **Tools** menu, select **Options**.

   The **Options** dialog box opens.

2. Select the **Transactions** property page and select the **Confirmed transaction mode** checkbox.

3. Click **OK** or **Apply**.

Searching committed transactions

Service Activator supports the ability to search through committed transactions.

To search committed transactions directly from the hierarchy pane

1. Click the **System** tab in the hierarchy pane.
2. Expand the **Transactions** folder.
3. Right click on the **Committed Transactions** folder.
4. Select **Find From Here** from the popup menu.

   The **Find** dialog is displayed. The **Search From** field is populated with the **Committed Transactions** folder.
5. Enter your search criteria and click **Find**.

To search committed transactions from the Find dialog

1. From the **Tools** menu, select **Find** (or use **Ctrl+F**).
2. Click **Browse**.
3. Navigate to the **Committed Transactions** folder and select it.
4. Enter your search criteria and click **Find**.

Exporting transactions

To export all transactions:

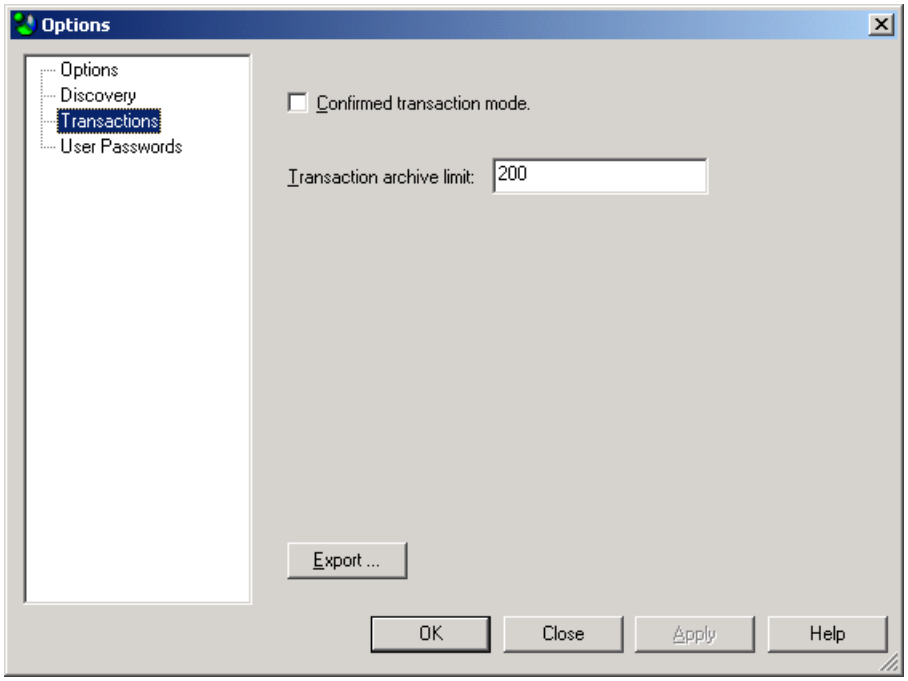
1. Select **Options**... from the **Tools** menu.

   The **Transactions** dialog box is displayed.

2. Select the **Transactions** property page.
3. Click **Export**... to export transactions to a text file.

   The **Save As** dialog is displayed.

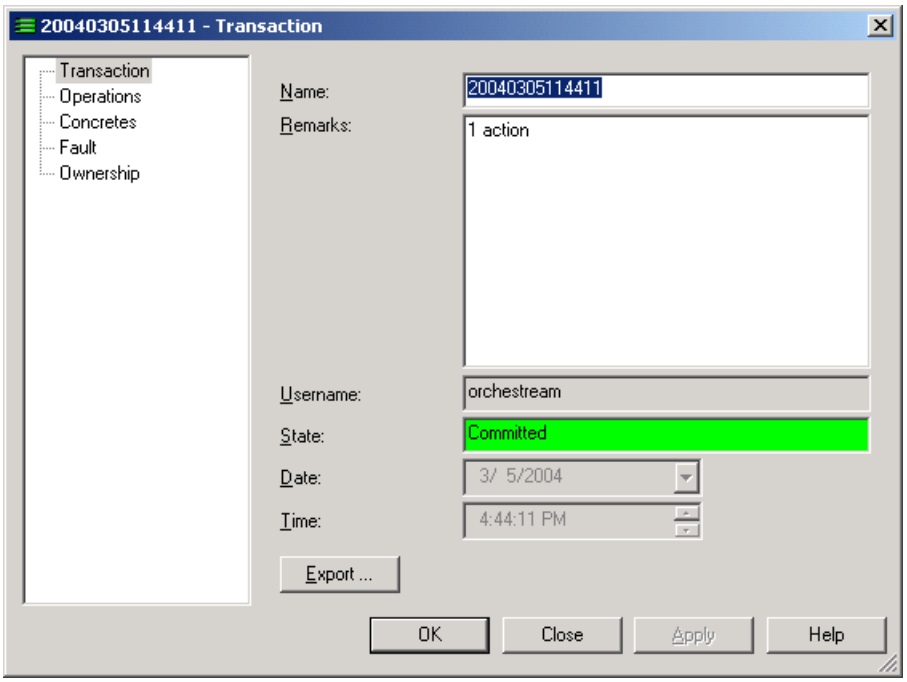
4. Enter the filename for the destination text file and click **Save**.



To export a single transaction:

1. Locate the single transaction you wish to export.
2. Right click on the transaction and select **Properties** from the popup menu.

   The **Transaction** dialog box is displayed.

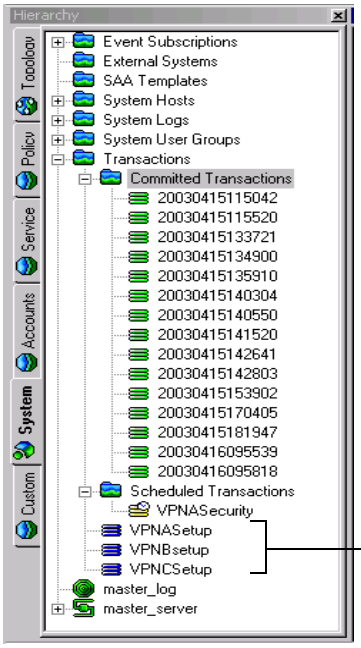


3. Select the **Transaction** property page.
4. Click **Export**... to export the transaction to a text file.

   The **Save As** dialog is displayed.
5. Enter the filename for the destination text file and click **Save**.

Viewing a list of transactions

Service Activator lists the transactions that are currently held in the transaction store on the **System** tab under the **Transactions** folder.



Depending on your organization, the transactions that are listed may include:

- Transactions created by you
- Transactions created by another user working on a remote user interface
- Transactions created by a third-party tool through the OSS Integration Manager (OIM)

Note that the number of transactions that are listed in the **Committed Transactions** folder is configurable – see *Setting the archive limit for transactions* on page 75.

To view a list of transactions

- On the **System** tab, open the **Transactions** folder.

  Pending transactions are listed under the **Transactions** folder, scheduled and committed transactions are grouped into subfolders.

Viewing transaction details

You can view detailed information about all transactions in the **Details** pane and sort on any of the displayed columns. If you have adopted a naming convention for transactions based on the first letters of the transaction name, this is a way of sorting related transactions. For information on sorting on a column, see *Changing views* on page 37.

To view transaction details

- On the **System** tab, double-click the **Transactions** folder.

  The **Details** pane lists transaction information under the following headings:

  - **Name**: Identifier for the transaction
  - **Description**: Additional transaction information (optional)
  - **State**: Transaction status (see *Transaction status* on page 72)
  - **Schedule**: Time at which the transaction was or will be committed
  - **Username**: Name of the user who created the transaction, entered automatically by Service Activator.
  - **Size**: Size of the transaction in bytes
  - **System Configuration**: Number of concrete objects created by the transaction and their status.
  - **Fault Level**: Number and type of faults created by the transaction.
  - **Owner**: If ownership of the transaction has been specified, the owner's username.
  - **OwnerGroup**: If ownership of the transaction has been specified, the group to which the owner belongs.

Transaction status

Every transaction has a status, or state, that indicates where it is in the transaction lifecycle.

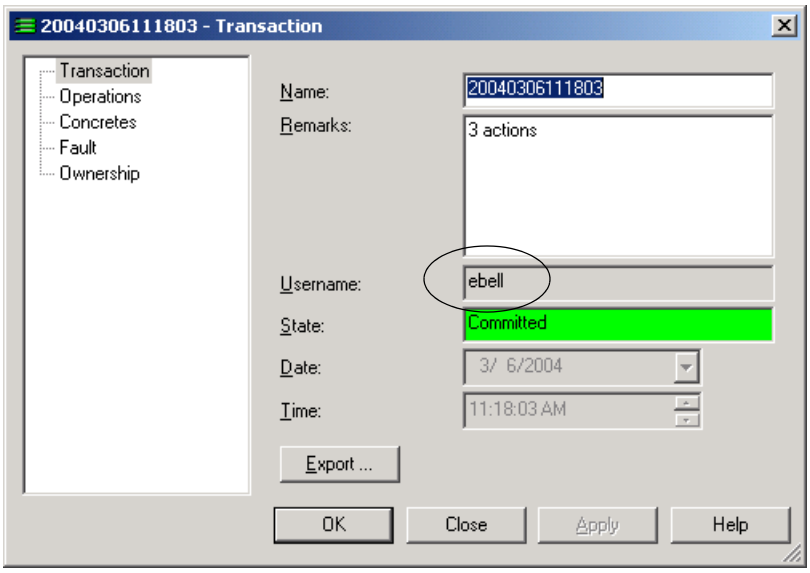
State	Description	Icon
Pending	A transaction that has been saved to the transaction store but whose changes have not been propagated to the network. Pending transactions are visible in all user interfaces.	
Scheduled	A transaction that will be committed at a future date and time.	

State	Description	Icon
Failed	A scheduled transaction that Service Activator was unable to commit due to conflicts with the common object model.	
Committed	A transaction whose changes have been saved to the database and related configuration changes propagated to the network.	

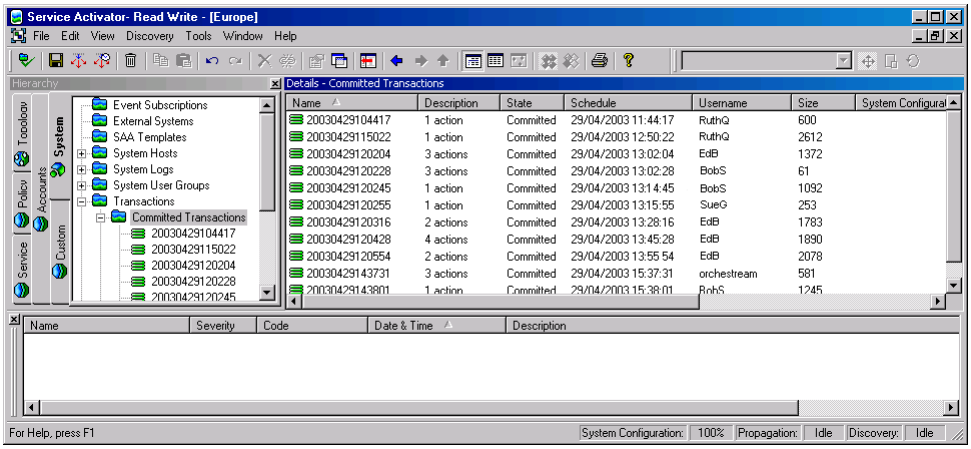
Checking the origin of a transaction

There are a number of methods for checking which user created a transaction. You can:

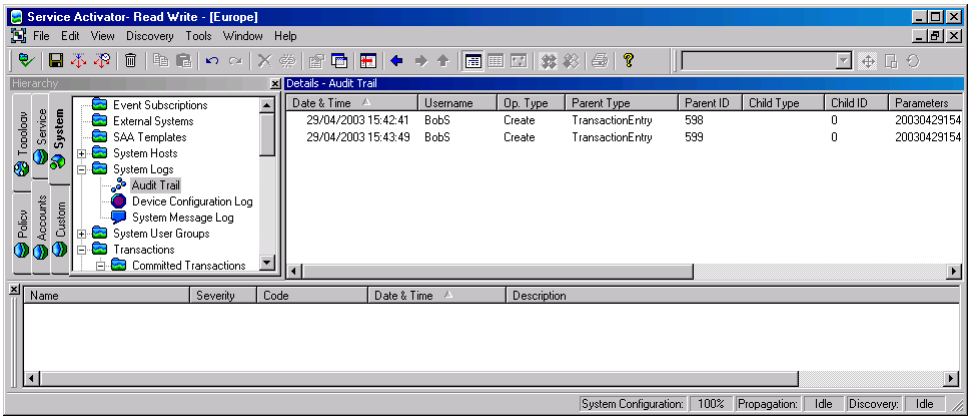
- Open the relevant transaction's properties dialog box and check the **Username** field on the **Transaction** property page.



- List all transaction details in the **Details** pane (see *Viewing transaction details on page 72*)



- Check the audit trail (viewable under the **System** tab's **System Logs** folder).



Selecting transactions

You can select a single transaction or multiple transactions listed in the **Details** pane and select an action to perform on them. For example, you can merge a number of transactions in one step by multi-selecting them and selecting **Merge** from the pop-up menu. Where several transactions are selected, Service Activator processes them in the order in which they were selected.

For information on selecting objects in the **Details** pane, see *Selecting objects on page 24*.

Setting the archive limit for transactions

You can view the list of committed transactions on the **System** tab under the **Transactions** folder and specify the number of committed transactions that Service Activator maintains in its transactions archive.

If you specify an archive limit, Service Activator also applies an upper threshold of 25 for committed transactions over and above the specified limit. The total number of archived transactions is therefore the combined value of the defined archive limit and the hard-coded upper threshold. For example, if you specify an archive limit of 100, Service Activator deletes committed transactions only when their number exceeds 125.

> When you set the archive limit, if more than 25 committed transactions are in the archive, none of these transactions will be deleted. This is due to Service Activator's application of a hard-coded upper threshold of 25 above the specified archive limit.

By default, Service Activator maintains information about the last 200 committed transactions. If you lower the limit, any committed transactions above the limit (plus the hard-coded threshold) are permanently deleted from Service Activator.

To set the archive limit for transactions

1. From the **Tools** menu, select **Options**.

   The **Options** dialog box opens.

2. Select the **Transactions** property page.

3. In the **Transaction archive limit** field, specify a new archive limit.

   The default is 200.

Chapter 4

Setting Up Users

This chapter explains the tasks you need to do immediately after installation in order to set up user groups and users.

This includes the following:

- Setting up user groups and users, including setting permissions for Read Write groups
- Disabling and re-enabling a user's access to the system
- Setting up global rules for the use of passwords
- Viewing user and user group information
- Owning and setting permissions on objects

About users and security

Service Activator allows you to control who has access to the system, and at what level. Access to Service Activator's user interface is controlled via user names and passwords. Every user who wishes to gain access must be set up as a user with a user name and password. The ability to create new users and assign passwords is restricted to users with Super User access, the highest access level in Service Activator.

The actions a user can perform in Service Activator are dictated by the group to which he or she belongs. Every user is a member of a user group and the access rights specified for the group apply to the group's members. The following group access levels are available:

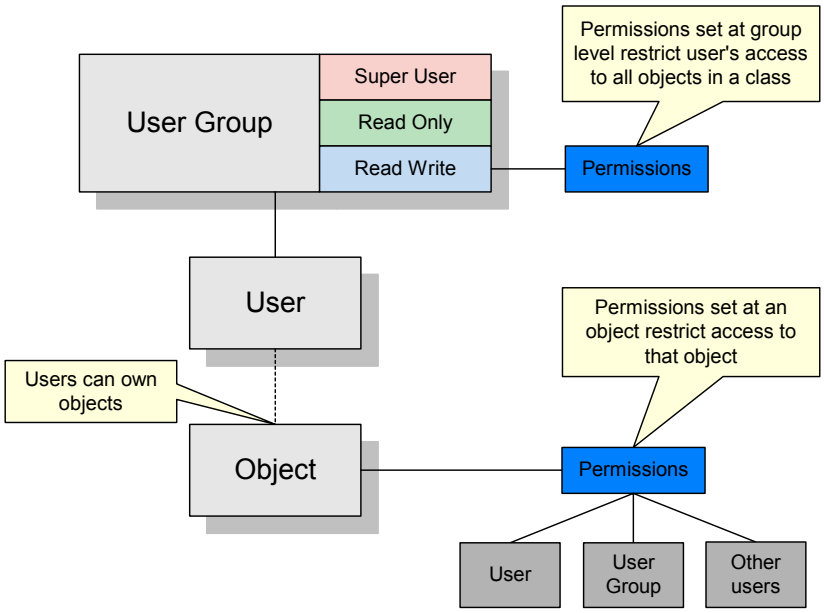
- Super User
- Read Write
- Read Only

For Read Write groups, you can specify exactly which elements of Service Activator its members can view and modify using permissions. This enables you to limit access according to the role a user performs, or the customer accounts he or she is managing. When you first install Service Activator you may need to create a number of Read Write groups to apply the necessary access levels to all users.

Note that a user's access level affects the appearance of the user interface to that user. For example, a Read Write user with access barred to policy cannot see the **Policy** tab or access menu items that relate to policy.

Service Activator also includes a concept of object ownership that allows a user to own specific objects and set permissions that apply to themselves, the group to which they belong, and all other users.

The following diagram illustrates the relationship between user groups, users and permissions.



User groups

You can set up user groups with the following access rights:

- **Read Only** – Users can view all parts of Service Activator's object model but they cannot make any changes. Permissions cannot be set for a Read Only group.
- **Read Write** – Users can view all or selected parts of Service Activator's object model. Group permissions may be defined that restrict the actions that group members can perform.
- **Super User** – Users can view all parts of Service Activator's object model and perform unlimited actions. Only members of a Super User group can create or amend user groups and users. Permissions cannot be set for a Super User group.

The access rights that are set for a user group apply to all members of the group.

Access rights associated with a Read Write user group can be fine tuned by setting detailed permissions on specific classes of object. These permissions may vary between Read Write groups. For example, one group may restrict its members to

Read Only permissions for classes of service while another group may give access to classes of service but restrict access to VPN creation.

> User groups can only be created by users who have Super User access.

Users

Every user in Service Activator is a member of a single user group. The level of access a user has to Service Activator's commands depends on the group to which he or she belongs. Members of a group that has Read Write access are more restricted in the actions they can perform than members of a group that has Super User access. This difference in access level is reflected in the user interface. Service Activator elements to which a user does not have access are hidden from the user. For example, users with restricted access may be unable to view selected tabs and menu options.

It is possible to create a user account that can be used by several users concurrently. So, for example, you can create a guest account that can be used by multiple users to log in to Service Activator at the same time.

> New users can only be created by users who have Super User access.
>
> There must always be at least one user who has Super User access. Service Activator enforces this rule by preventing deletion of the last user with Super User access.

Passwords

All users access Service Activator by username and password. Initial set up of passwords can only be performed by users with Super User access.

A user can be forced to change his or her password at the next login to Service Activator and the password set to expire after a set number of days.

Password options are set in two different places.

Global password options are set on the **User Passwords** property page of the **Options** dialog box. To access this dialog box, select **Tools > Options** from the main menu of the **Service Activator** User Interface.

See *Creating rules for passwords* on page 93

Password options for particular users are set by accessing the **Password** property page of the **System User** dialog box for the user. To access this dialog box, right-click on the user in the **System User Groups** folder, and select **Properties** from the pop-up menu.

See *Select the Password property page.* on page 92.

It is also possible to create rules for passwords. Users with Super User access can:

- Specify the minimum and maximum length for passwords.
- Set an expiry period that applies to all passwords.
- Specify the number of passwords that Service Activator remembers for each user.
- Specify the number of login attempts Service Activator allows before barring a user. A user with Super User access can reinstate barred users.

Users with Super User access are never barred after successive failed logins, but Service Activator records the number of failed logins in the system log file (see the *Administrator's Guide* for more information) and displays an error message in the current faults pane (only visible to Super Users).

Passwords and password rules can only be created by users who have Super User access.

Permissions

Permissions define the actions that can be performed on a class of objects or an instance of an object or the ability to perform an operation, such as discovering devices. Permissions can be set at the following levels:

- Group level – Super Users can set permissions for groups that have Read Write access rights.

- Object level – users with Read Write or Super User access can set permissions on the objects they own. Different permissions can be set for themselves, the group to which they belong and to all other users.

Additional attributes of permissions include:

- A permission that is set for a Read Write group applies to a class of object. A permission that is set for a specific object applies to that object instance only.
- A user who assigns permissions to a previously unowned object assumes ownership of the object by default, or can assign ownership to another user.
- To calculate an object's actual permission for a user, the system combines the permission assigned to the object with the permission assigned to its class of objects, for that user.

Permission levels

In the following descriptions, "class of objects" is the same as "object type". Also, "objects of the class" is the same as "instances of an object type". The following permission levels are available:

- Denied – the user cannot access the object or class of objects
- Read – the user can view the object or class of objects
- Link (Reference Only) – objects or a class of objects can be linked to (referenced by) other objects, but cannot have other objects linked to them
- Link – the user can link the object/class of objects to appropriate objects, and appropriate objects can be linked to the object/class of objects
- Modify – the user can modify the object/class of object's properties
- Write – combines the Modify and Link permissions
- Create – the user has Write permission plus the user can create the class of objects
- Execute – the user can perform the operation

For a more detailed description, refer to the Service Activator Online Help topic on object permissions. More detailed topics on Link, Link (Reference Only) and Execute are provided later in this section.

The following table summarizes how each permission level restricts the possible operations on objects:

Permission level	View	Link		Unlink	Modify	Create	Delete	Own
		From	To					
Denied	×	×	×	×	×	×	×	×
Read Only	✓	×	×	×	×	×	×	×
Link (Reference Only)	✓	✓	×	×	×	×	×	×
Link	✓	✓	✓	✓	×	×	×	×
Modify	✓	×	×	×	✓*	×	×	×
Write	✓	✓	✓	✓	✓*	×	×	×
Create	✓	✓	✓	✓	✓	✓	✓	✓

*Except the **Ownership** property page cannot be modified.

Instances of most object types can be owned and permissions can be set, except for the following object types:

- User groups and users
- Some rule components – 802.1p User Priority, IP Protocols, Packet Markings and Traffic Types
- Device types
- System logs
- The policy server

Permission examples

- A user who has been granted Read Only permission on a VPN can view the VPN but cannot link other objects to the VPN or unlink objects from it, modify, create, delete or own the VPN.
- A user who is a member of a Read Write group that has Modify permission on the policy area of the object model can view classes of service and modify their property values. However, the user cannot link the CoS to other objects, unlink existing associations, create, delete or own classes of service.

Link permission

Link permission refers to linking the target object to another object and also to linking another object to the target object. This means, for example, that if you have link access for a domain or a site, you can link to these objects without necessarily having permission to modify them.

Users can view objects of the selected class and folders and tabs that contain objects of the class. Objects of the class can be linked to other objects and can have other objects linked to them. Property pages are read-only. Options to create objects of the class are not shown and options to delete objects with this permission are disabled.

Link (Reference Only) permission

Link (Reference Only) is a more restrictive version of the "Link" permission:

- Users can view objects of the selected class and folders and tabs that contain objects of the class.
- Objects or classes of objects can be linked to (referenced by) other objects, but cannot have other objects linked to them. The referenced object can be unlinked from the other objects.
- Property pages are read-only. Options to create objects of the class are not shown and options to delete objects with this permission are disabled.

When Link (Reference Only) is assigned to a parent object, this permission is not automatically enforced to all children in the hierarchy. The user is responsible for setting and controlling permissions down the hierarchy of objects.

On Ownership property pages of classification groups and compound traffic objects, the "Objects below this level inherit the permissions" checkbox does not apply. The children of these objects do not inherit their parents' permission level. This restriction prevents unintended overwriting of the permission on an object [with multiple parents] that inherits different permissions from each parent.

Primary purpose of Link (Reference Only) permission

The unidirectional linkage of this permission maintains the definition of a policy object while using it to create new policy objects. In a common application, users need to use globally defined policy objects to define customer-specific policy data. Link (Reference Only) permission allows users to create a new policy object by referring to an existing policy object, but prevents accidentally changing the existing policy object that could result from linking another object to it.

Follow the recommended usage guidelines to ensure secure access to classification group and traffic type elements.

Usage guidelines for classification groups

Use these Link (Reference Only) permission guidelines to secure the definition of your classification group and classification objects in the GUI:

- Create two folders, Folder1 and Folder2.
- Set Folder1 permission to Link (Reference Only) and select its "inherit" checkbox. (The actual checkbox label is "Objects below this level inherit the restrictions".)
- Move all existing top-level classification groups under Folder1.
- Set Folder2 permission to Denied.
- Under Folder2, create subfolder Subfolder2 with Link (Reference Only) permission and select its "inherit" checkbox.
- For each top-level classification group contained in Folder1, link every element of the classifiction group into Subfolder2. (That is, link every classification, sub-classification group, and sub-classification.)

These guidelines ensure that all elements in any top-level Link (Reference Only) classification group maintain the same permissions. Because inheritance is not supported within classification groups, other users will be unable to overwrite these permissions.

Note that a Link (Reference Only) classification or classification group that either does not have a parent, or does not have a parent with restricted permissions, is in danger of being unlinked from its original location. Placing the classification and classification group objects within restricted folders (according to the above guidelines) provides a secure location for the objects.

Usage guidelines for traffic types

Use these Link (Reference Only) permission guidelines to secure the definition of your traffic type policy objects in the GUI. Note that Traffic Groups behave like traffic folders.

- Create 2 traffic groups: Group1 and Group2.
- Set Group1 permission to Link (Reference Only) and select its "inherit" checkbox. (The actual checkbox label is "Objects below this level inherit the restrictions".)
- Move all existing top-level traffic types and compound traffic under Group1.
- Set Group2 permission to Denied.
- Under Group2, create subgroup SubGroup2 with Link (Reference Only) permission and select its "inherit" checkbox.

- For each compound traffic object in Group1, link every traffic type contained in the compound traffic object into SubGroup2 (if the traffic type does not already exist directly under Group1).

These guidelines ensure that all traffic types in any top-level Link (Reference Only) compound traffic object maintain the same permissions.

Execute permission

The Execute permission level can only be set for Read Write groups and applies to:

- Loading configuration data (policy files)
- Telnetting to a device
- Managing a device
- Discovering the network
- Validating data
- Creating site accounts
- Finding system components
- Saving, merging and committing user transactions

Folder permissions

- The object permissions assigned to a folder are inherited down to its sub-folders when the “Inherit” flag ("Objects below this level inherit the restrictions" checkbox) is selected. The inherited permission can be overwritten by assigning independent ownership to a sub-folder.
- Permissions assigned to a Policy object folder are inherited by the top-level policy objects contained in that folder when the inherit flag is selected. Note that folder permissions are not inherited by classifications contained in a classification group within the folder. Similarly, the permission assigned to a traffic group is not inherited by traffic types contained in a traffic compound within the traffic group.
- An object in a folder cannot be moved to another folder unless the user has "Create" permissions for the object, or the object is un-owned.

Impact of permissions in the GUI view

The user interface displays information relevant to the permissions that apply to the user group.

If access to an object class or instance of an object is denied, the object and any folders and tabs that exclusively contain that object are not displayed to the relevant users. In addition, objects do not appear on property pages, menus and drop-down menus.

Multiple user interfaces

Because changes to permissions only take place the next time that a user logs in to Service Activator, it is possible for different permissions, and thus different views of the system, to apply to members of the same user group on two user interfaces.

> Permissions set for Read Write groups are calculated and fixed for the duration of a user's login session. Changes to group permissions are not apparent to the group's members until the next time they log into Service Activator.
>
> Permissions set for an instance of an object are apparent to other users immediately. For information on owning and setting permissions on an object, see page 99.

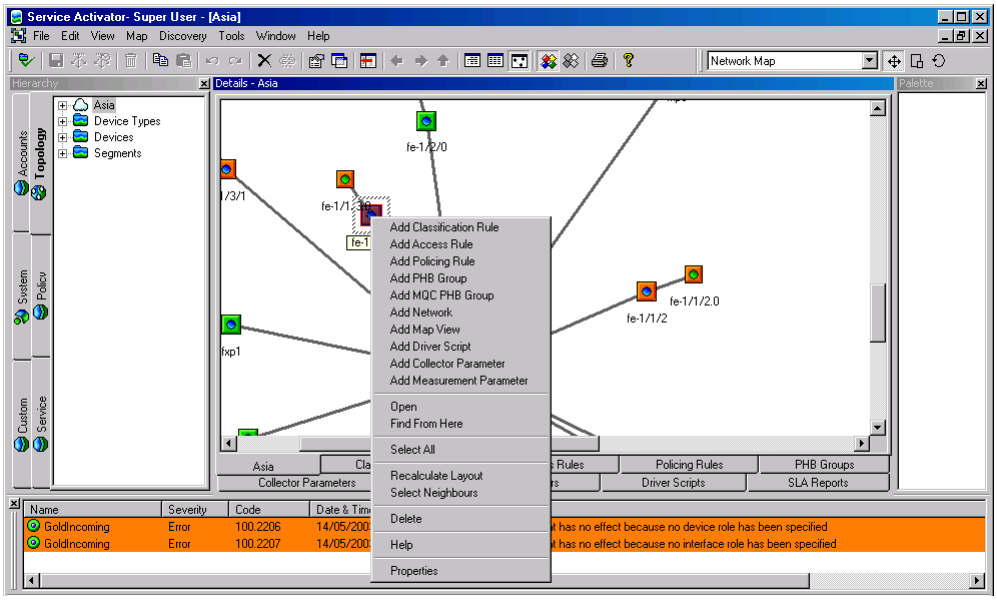
We recommend that a system Super User makes changes to permissions only when no other users are logged in. If necessary, the Super User should disable access to the relevant user group while changes are being made (see *Disabling or re-enabling a user's access* on page 96).

Inheritance of permissions on faults

Note that faults inherit object permissions from the object on which the fault is reported. This ensures that users do not see faults that are reported on objects that they do not have access to.

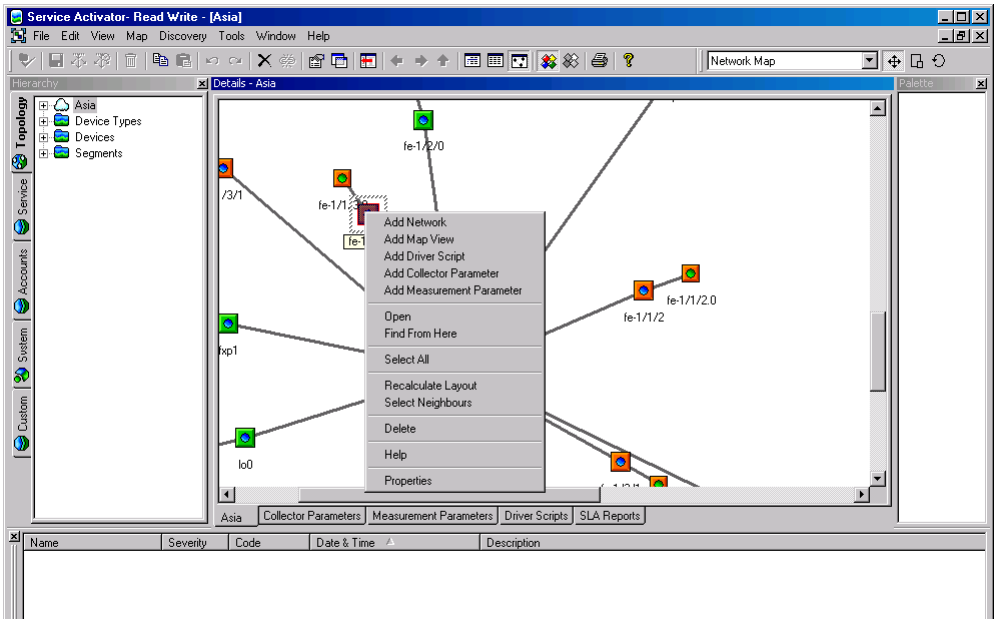
GUI Example

For a Read Write user who has access to all parts of the system, the Service Activator user interface appears as shown in the following illustration.



Note that all tabs are visible and options to add policy and measurement elements and driver scripts are available on the pop-up menu.

For a user who has Read Write access with access denied to policy objects, the user interface appears as follows:



Note that the **Policy** tab is not displayed and the options to create PHB groups and rules are not available on the pop-up menu.

Note also that you cannot see faults associated with object types to which you do not have access. In this example, the warnings that relate to a PHB group with no roles associated with it cannot be seen in the second illustration.

Inheritance of permissions

The permissions set on a class of object for a group or on a specific object by its owner can be inherited by lower-level objects. For example, applying Read permission at device level can be inherited so that a group's users also have Read permission for interfaces, sub-interfaces and VC endpoints.

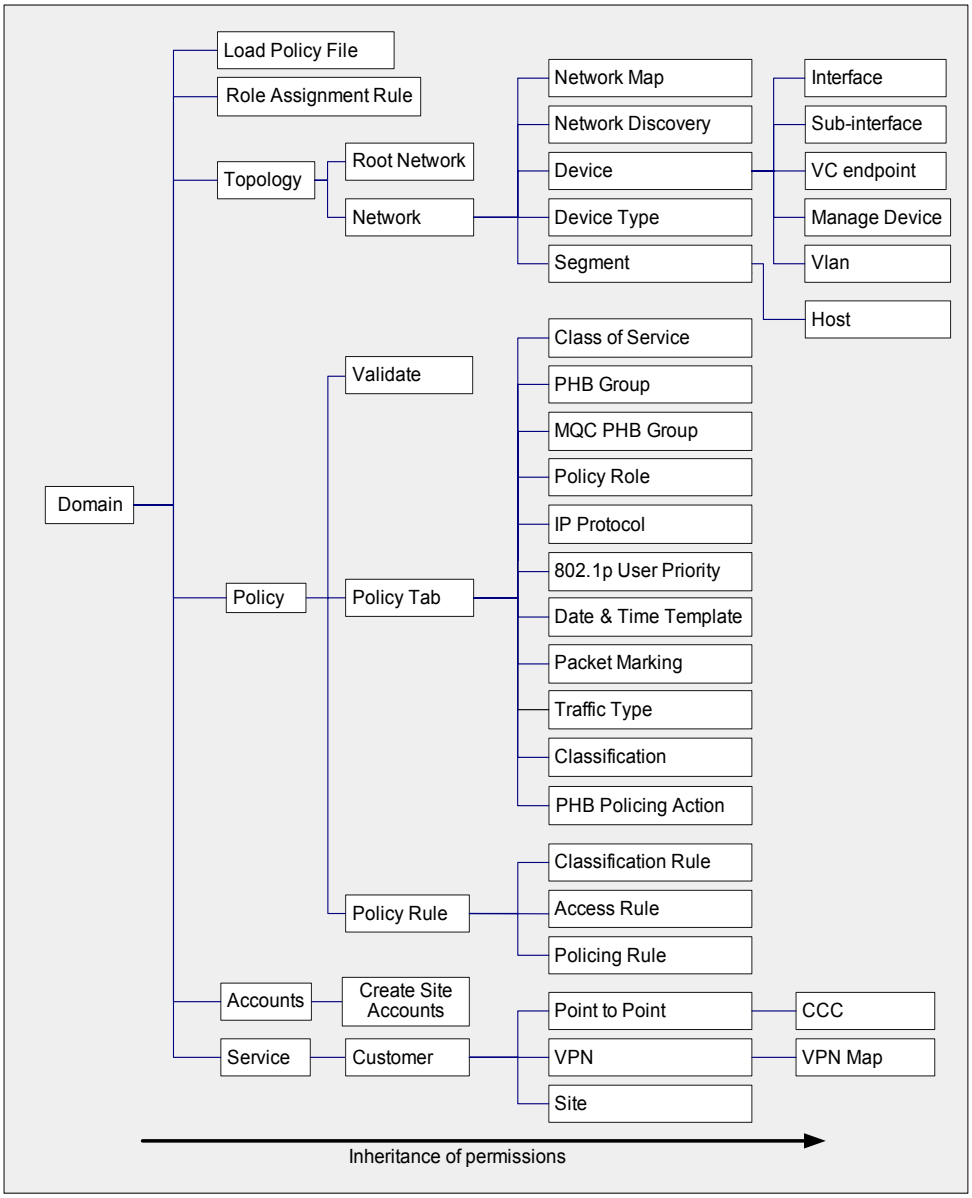
Service Activator's permission hierarchy is divided into three main areas: Custom, Domain, and System.

Custom

Within the Custom area, inheritance applies to driver scripts and script context.

Domain

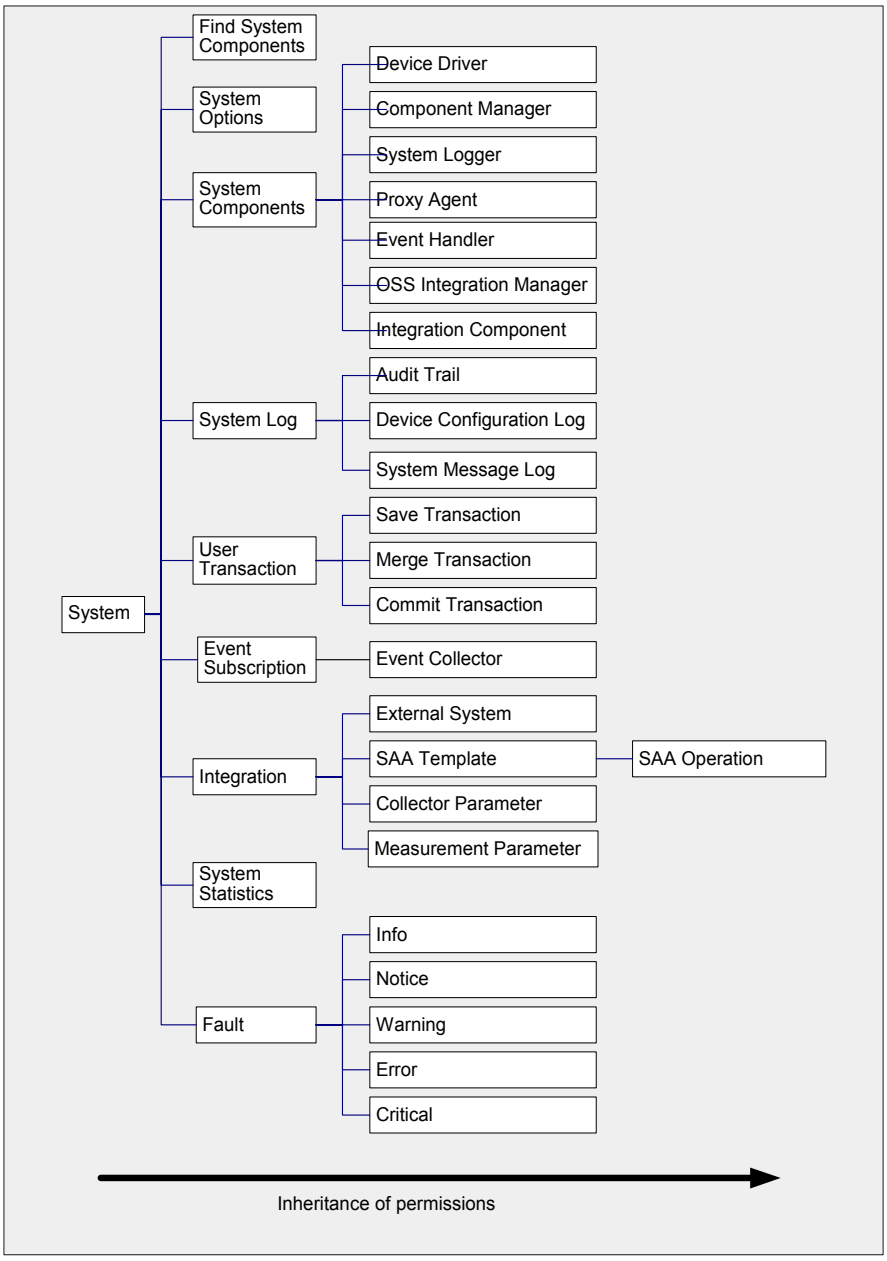
The following diagram shows the hierarchy of permissions for classes of objects within the Domain area.



Note 1: Traffic Type box: object inheritance does not occur below Compound Traffic objects.
Note 2: Classification box: object inheritance does not occur below Classification Group objects.

System

The hierarchy of permissions within the System area is as follows:



Inheritance of permissions on sites and VPNs

Take care when setting up permissions for sites and VPNs to avoid sites being hidden from users. This can occur when sites that are a child of one customer are linked to multiple VPNs with different security permissions.

- When sites that are already included in a VPN are subsequently added to a management VPN, to ensure that sites in the management VPN can be seen by the customer when logged in, do not select the **Objects below this level inherit the restrictions** checkbox on the **Ownership** property page of the management VPN object.
- When sites owned by one customer are included in a VPN owned by another customer, to ensure that sites can be seen by any other customer, ensure that **Access for other users** for all Site objects is set to at least Read.

Changing the default user

When the policy server is started for the first time after installation, it creates a default user with Super User access rights. Any user interface components that are subsequently started automatically use the default user's username and password details to provide access – the Service Activator main screen appears immediately and no username and password details are required.

We strongly recommend you change the default user group and default Super User as soon as possible. Once the default user has been changed, or a new user created, all users must enter a name and password each time they start the Service Activator user interface.

To change the default user

1. Open the **System User Groups** folder on the **Systems** tab in a global setup or domain management window.
2. Select the **Super Users** group. Initially there is a single user named ***admin***.
3. Right click on the ***admin*** user, then select **Properties** from the pop-up menu. The **System User** dialog box opens.
4. Change the **Username** to an appropriate unique entry.
5. Select the **Password** property page.
6. Change the **Password** to an appropriate unique password, and confirm the password by re-entering it in the **Confirm Password** field.

We also recommend you set up additional users with different levels of access. To do this, you must first create additional user groups with the relevant access levels. You can then add users to these groups. For more information, see *Setting up user groups and users* on page 93 and *Creating users* on page 96.

Creating rules for passwords

Users with Super User access can set rules for passwords, such as the minimum and maximum length a password must be and the period after which passwords expire.

> Password options can also be set for an individual user account. See *Select the Password property page.* on page 92.

When a user's password expires, he or she is prompted for a new password on the next login to Service Activator's user interface:

Note: For complete dialog box and property page descriptions, refer to the *Online Help*.

To edit global password rules

1. From the **Tools** menu, select **Options**.

   The **Options** dialog box opens.

2. Select the **User Passwords** property page.
3. Enter values including **Minimum Length, Maximum Length, Passwords expire after, Password Memory** and **Disable account after login failures.**
4. Click **OK** to close the dialog box.

Setting up user groups and users

Users with Super User access can define access to the system by setting up new user groups and user logins.

Creating a user group

A user group defines the access level that its members have to Service Activator. Every user is a member of a user group.

Service Activator automatically creates one user group on installation. The group has Super User access and contains a single user (see *Changing the default user* on page 92). To create users with Read Write and Read Only access, you must set up additional user groups. For Read Write groups, you can also define the permissions that apply to the group – that is, which parts of Service Activator the group's members can view and/or modify.

Some example Read Write user groups might include:

- Network engineers – group members can import topology files, define roles, create maps and create driver scripts
- Policy engineers – group members can create traffic types and PHB groups
- Customer service engineers – group members can manage customers and their associated VPN services
- Demonstration users – group members can make changes but cannot save them

A user can only be a member of a single group.

To set up a user group

1. Select the **System User Groups** folder on the **Systems** tab in a global setup or domain management window.
2. Select **Add System User Group** from the pop-up menu.

   The **System User Group** dialog box opens.
3. Enter details including **Name, Access Rights, Remarks, Disable all group users.**

   See also *Setting up Read Write group permissions* on page 95.
4. Click **OK** to close the dialog box.

Setting up Read Write group permissions

For Read Write user groups, group permissions define the actions that group members can perform on classes of object. Group permissions allow groups to be restricted according to the roles and responsibilities of the group's members.

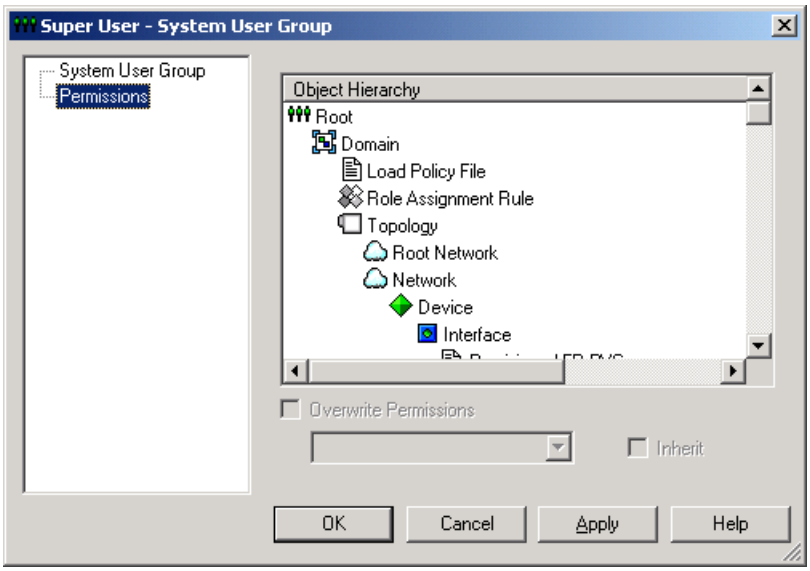
Note: For complete dialog box and property page descriptions, refer to the *Online Help*.

To set or change a permission for a user group

1. On the **System** tab, open the **System User Group** folder and select **Properties** from the relevant user group's pop-up menu.

   The **System User Group** dialog box opens.

2. Select the **Permissions** property page.



   Group permissions are displayed in a fixed hierarchy, and the appropriate permissions are displayed for each level within the hierarchy. For more information on permissions and the permissions hierarchy, see *Permissions* on page 81.

3. Select the object class whose permissions you want to set.
4. Click the **Overwrite Permissions** checkbox, and then select the permission level you require from the drop-down menu.
5. If you want object classes at a lower level in the hierarchy to inherit the permission, select the **Inherit** checkbox.

6. Click **OK** to close the dialog box.

> You can select individual object classes at a lower level within the hierarchy and change the individual permissions as required.
>
> When you change permissions for a user group, the changes are only apparent to group members the next time they log into Service Activator.

Creating users

Users with Super User access can set up new users, specifying their username and password details. It is possible to force users to change their password when they first log into Service Activator.

A user must be set up within a user group, and can be a member of one group only.

Note: For complete dialog box and property page descriptions, refer to the *Online Help***.**

To set up a user

1. Select the **System User Groups** folder on the **Systems** tab in a global setup or domain management window.
2. Select the group to which you want to add a new user.
3. Select **Add System User** from the pop-up menu.

   The **System User** dialog box opens.

4. Enter details including **Username, Remarks** and **Allow concurrent logins**.
5. Select the **Password** property page.
6. Enter details including **Password, Confirm Password**, **User must change password at next login,** and **Expires after**.
7. Click **OK** to close the dialog box.

Resetting a user's password

If a user with Super User access changes a user's password, the user must enter this new password on re-entry to Service Activator.

Disabling or re-enabling a user's access

You can disable access for all members of a group or for an individual user. Disabling access for a user who is logged in forces the user out of Service Activator.

To disable access for all members of a group

1. Select the **System User Groups** folder on the **Systems** tab in a global setup or domain management window.
2. Click on the relevant group.
3. Select **Properties** from the pop-up menu.

   The **System User Group** dialog box opens.
4. Select the **Disable all group users** checkbox.
5. Click **OK** to close the dialog box.

To disable access for an individual user

1. Select the **System User Groups** folder on the **Systems** tab in a global setup or domain management window.
2. Select the group that contains the user whose access you want to disable, then click on the relevant user.
3. Select **Properties** from the pop-up menu.

   The **System User** dialog box opens.
4. Select the **Disable user** checkbox.
5. Click **OK** to close the dialog box.

Re-enabling users

If a user is locked out of Service Activator as the result of too many failed login attempts, a user with Super User access can re-enable the user by deselecting the **Disable user** checkbox on the user's properties pages. If necessary, you can increase the number of permitted login attempts – for more information, see *Creating rules for passwords* on page 93.

Viewing user group and user information

You can view user group and user information and see which users are logged into Service Activator.

Users with Super User access can see all user group and user information. Users with Read Write or Read Only access can only view the group to which they belong and their own user details.

To view user group information

- Double-click on the **System User Groups** folder.

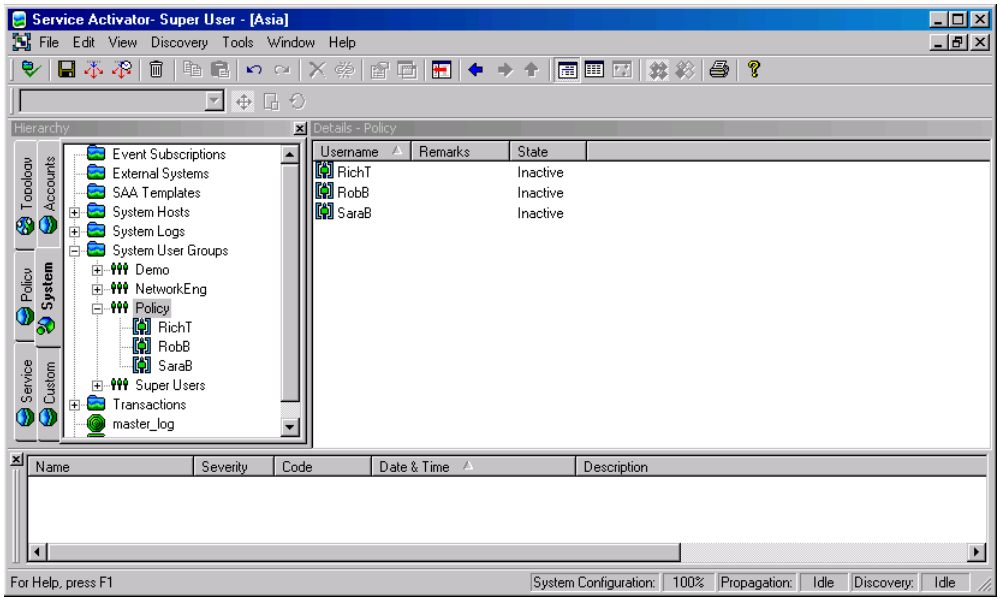
The information is displayed in the **Details** pane as follows:

- **Name** – The name of the user group
- **Access Rights** – The access rights of the user group
- **Remarks** – Any remarks that were entered when the user group was set up.
- **Active Users** – For each user group, lists the number of users currently logged in to Service Activator

To view user group member details

- Open the **System User Groups** folder and double-click on the user group whose membership details you wish to view.

  The information is displayed in the **Details** pane:



The details are as follows:

- **Username** – The name of the user.
- **Remarks** – Remarks that have been added about the user.
- **State** – The state of the user:
  - **Inactive**: the user is enabled with no active sessions.
  - **Active**: the user has active sessions (if more than one session is active, the number of sessions appears in parentheses)

- **Disabled**: user's access has been disabled by a Super User.
- **Denied**: user has been denied access because the prescribed number of login attempts has been exceeded.

> You can also view the status of a user by selecting the required user, then select **Properties** from the pop-up menu. The status field shows the current status of the user.

Owning and setting permissions on an object

If you create an object you can assume ownership of that object and specify which users can view and modify the object. You can set different permissions for yourself, other members of your user group and members of other groups. Ownership of an object can also be passed to another user.

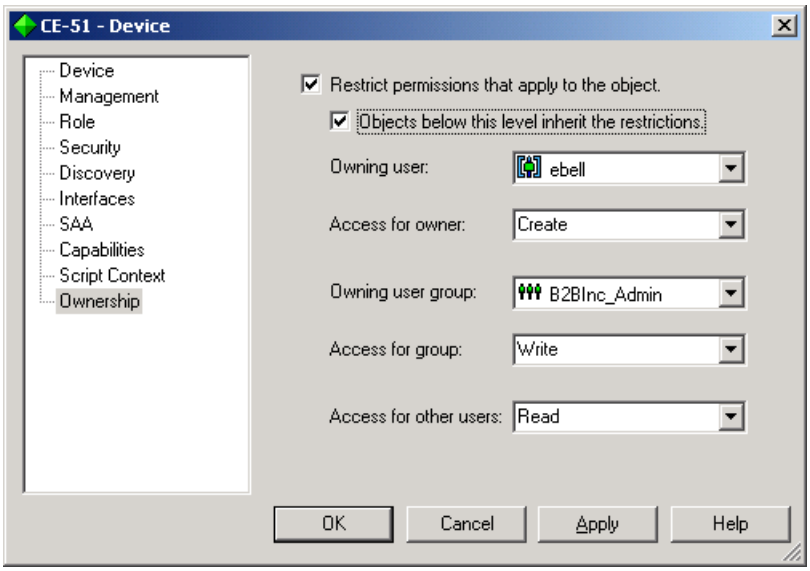
Permissions can be set on most objects except user groups and users, some rule components, device types, system logs and the policy server. For full details about permissions refer to *Permissions* on page 81.

To set permissions on an object

1. Select the object you want to set permissions on, then select **Properties** from the pop-up menu.

   The object's properties dialog box opens.

2. Select the **Ownership** property page:



3. Select the **Restrict permissions that apply to the object** checkbox to set ownership requirements.
4. Set the **Objects below this level inherit the restrictions** checkbox if you want permissions to be inherited by objects below the selected object.
5. Set the following details by selecting from the drop-down lists:
   - **Owning user**: the name of the user who will own this object.
   - **Access for owner**: the type of access permitted for the owner.
   - **Owning user group**: the name of the user group who will own this object.
   - **Access for group**: the type of access permitted for members of the user group.
   - **Access for other users**: the type of access permitted for all other users.

   For details of the actions permitted by each type of access, see *Permissions* on page 81.

6. Click **OK** to close the dialog box.

If a user is deleted, any objects owned by that user are left ownerless.

A user who assigns permissions to an unowned object assumes ownership of the object by default.

Ownership of an object can be changed by any user with Create permission on the object.

Chapter 5

Setting Up Domain Information

This chapter explains the tasks you need to do immediately after installation in order to set up domains.

The chapter describes how to:

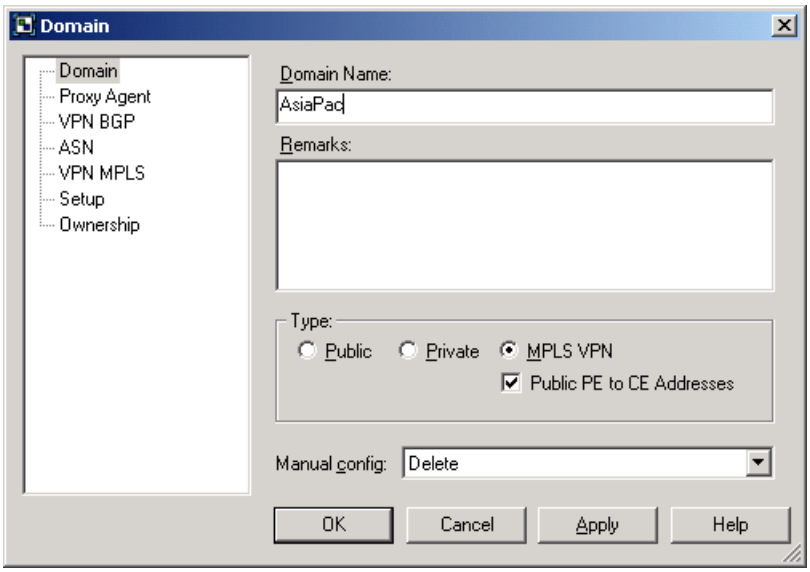
- Create domains and set up proxy agent assignment for the devices within the domain
- Load files containing basic configuration data
- Open domains

Setting up domains

Service Activator uses a concept of domains to define the logical networks to be managed. You can create multiple domains to represent managed networks – one domain for each AS region to be managed. No domains exist when you first install Service Activator, so the first step is to create one or more domains.

To create a new domain

1. Select **New Domain** from the **File** menu. The **Domain** dialog box opens:



2. Enter an identifying name for the domain.
3. Specify the type of domain:
   - **Public** if the network only uses public IP addresses.
   - **Private** if you plan to manage independent networks with overlapping address space.
   - **MPLS VPN** if you are going to set up MPLS-based virtual private networks.
4. For MPLS VPN domains, specify whether interfaces on Provider Edge routers that connect to the Customer Edge routers use public or private addresses.

   By default, they are assumed to use unique, public IP addresses. If they will use private IP addresses, clear the **Public PE to CE Addresses** checkbox.
5. In the **Manual Config** field, specify the default action within this domain for detecting manual configuration. Note that the system only checks to see if

changes have been made to the commands that may be configured by Service Activator, as these may affect the operation of the system:

- **Delete**: If manual configuration is detected, the device drivers will issue commands to reconfigure the device. No warning is output regarding the detected manual configuration.
- **Warn and delete**: If manual configuration is detected, a warning message (Message 3203) is output and the device drivers will issue commands to reconfigure the device.
- **Fail and don't delete**: If manual configuration is detected, a critical fault (Message 3494) is raised. The device status is set to Intervention Required but is not reconfigured.

Note that the domain-level manual configuration settings can be overridden for specific devices (see *Setting manual configuration detection* on page 166).

Note that Service Activator never deletes manually pre-configured VRF tables even if you select the **Delete** or **Warn and delete** options.

6. If you wish to own the domain and set permissions on it for yourself, members of your user group and other users, select the **Ownership** property page.

   For more information on setting permissions on an object, see *Owning and setting permissions on an object* on page 99.

If you are setting up VPNs you also need to set parameters on the **VPN BGP**, **VPN MPLS** and **ASN** property pages. For full details of how to set up VPNs, see *Configuring VPN Services*.

Setting the default loopback ID value for discovery

On the **Domain** dialog box, **VPN BGP** property page, you can specify a value for in the **Loopback ID** field. The Loopback ID value is used to create a loopback interface name by appending it to the name 'loopback'. For example, if the Loopback ID is 0, the loopback interface name created is 'loopback0'. When a device in this domain is discovered, a check is made to see if a loopback interface matching this text string exists. If it does, the IP address of the loopback interface is stored with the device information. The Loopback ID value can be overridden on a per-device basis (on the **Device** dialog box) and on a per-discovery basis (on the **Topology** dialog box).

Setting up proxy agent assignment

All devices in the domain that are to be managed by Service Activator must be assigned to a proxy agent. It is the proxy agent that controls when and what type of configuration is to be applied to a specific interface.

Although it is possible to assign devices to proxy agents manually, it is generally performed automatically during device discovery.

Assigning a proxy agent to a domain

If you have a distributed installation with multiple proxy agents and you are creating multiple domains, you can assign specific proxy agents to each domain.

A device can only be assigned to a proxy agent that is either global or has been assigned to the domain that the device is in.

To assign a proxy agent to a domain

1. Select the appropriate domain from the global setup window and select **Properties** from the pop-up menu.
2. Select the **Proxy Agent** property page. The list shows all the proxy agents currently installed.
3. Select one or more proxy agents to be used within the domain by clicking on the checkbox associated with the proxy agent.

Repeat these steps to assign specific proxy agents to specific domains.

> If you do not explicitly define proxy agents for each domain, all proxy agents remain global and devices within any domain can be assigned to them.

Setting up proxy agents for automatic assignment

Devices can be assigned to a proxy agent automatically whenever devices are discovered or rediscovered. You can configure Service Activator either to assign all devices to one proxy agent or to assign the devices equally to a number of proxy agents.

Devices are only assigned to proxy agents that are:

- Specifically assigned to the domain that the device is in, or are global (that is, not assigned to any domain) and
- Defined as active for auto-assignment

To check that a proxy agent is active, display the proxy agent properties by selecting the relevant proxy agent on the **System** tab in the **System Hosts** folder and choosing **Properties** from the pop-up menu. Ensure the **Auto device assignment** is set to **On** (this is the default setting).

To configure automatic proxy agent assignment within a domain

1. From the **Tools** menu, select **Options**.

   The **Options** dialog box opens.

2. Under **Auto proxy assignment**, select either **First** or **Load balance**:

   - **First** assigns each new device discovered to the first active proxy agent. This is the default setting.
   - **Load balance** allocates devices to multiple proxy agents with an equal allocation to all active proxy agents in order to balance the processing load between them.
   - If **Off** is selected, devices are not assigned to proxy agents automatically. Before they can be managed you will need to link them manually.

Only supported devices, that is, those that are included in the **DeviceTypes.cfg** file in the **Config** directory, can be automatically assigned.

Loading policy configuration data

The next step is to install one or more set-up files that create standard policy configuration data in the Service Activator database – basic policy components and, if required, sample rules and PHB groups.

> Although loading policy configuration data is optional, it is strongly recommended.

The following configuration files are supplied in the **SamplePolicy** directory:

- **default.policy** creates some basic policy data, including:
  - Packet markings representing IP Precedence values 0-7
  - Gold, Silver and Bronze classes of service
  - Packet marking based traffic types representing the Gold, Silver and Bronze classes of service and port-based traffic types representing the most common TCP and UDP port numbers
  - Classifications based on the traffic types

We strongly recommend that you load this default data. If you do not, you will have to create all the basic component data yourself.

- **advanced.policy** creates additional policy data:
  - Packet markings based on the full range of DiffServ codepoints and MPLS experimental bits
  - Packet marking based traffic types representing DiffServ codepoints and MPLS experimental bits
  - Port-based traffic types representing the most common IP protocols
  - Classifications based on the packet marking traffic types

  This data will be useful if your routers support the full range of DiffServ codepoints and/or MPLS experimental bits.
- **Rule_and_PHB.policy** includes some example policy rules, standard PHB groups and role assignment rules. If you wish, you can base your own rules and PHB groups on these examples.

The files must be loaded in the order in which they are listed above.

The following files may also be loaded:

- **juniper.policy** defines MPLS packet markings, classes of service and an example PHB group for configuring WRR on Juniper M-series devices. For more information, see the *Juniper M-series Driver Support Guide*.
- **Role_Assignment_Rules.policy** defines a set of role assignment rules that allocate system-defined roles to devices and interfaces. For more information, see *Pre-defined role assignment rules* on page 123.

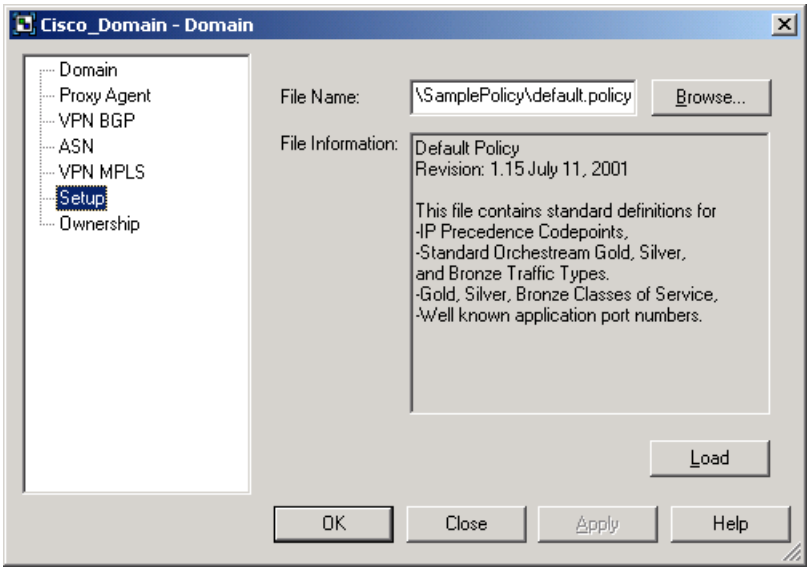
The **juniper.policy** and **Role_Assignment_Rules.policy** files are not dependent on any other policy files being loaded.

The **SharedPolicyData.policy** file is loaded automatically at system start-up. It defines a set of commonly-used IP protocols which are available in any domain you create. You only need to load this file if the IP protocols are deleted or edited incorrectly.

All of the above files are domain-specific – that is, they must be loaded into each domain in which you wish to use their information.

To load a policy configuration file

1. Select the appropriate domain from the global setup window and select **Properties** from the pop-up menu.
2. On the **Domain** dialog box, select the **Setup** property page.



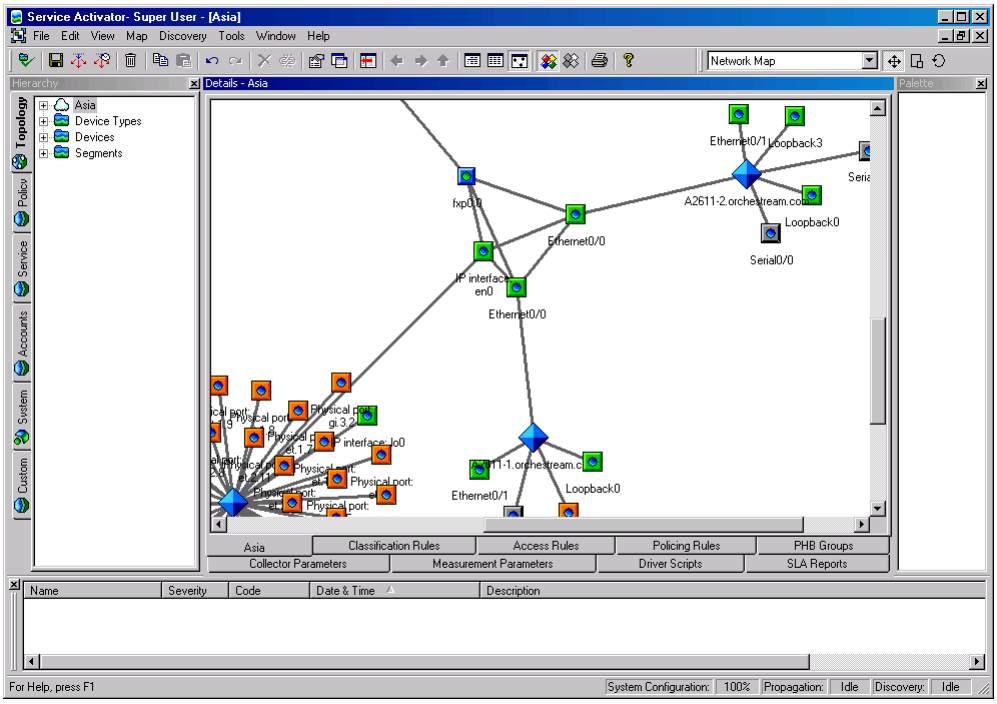
3. Click **Browse** to view the available configuration files in the SamplePolicy folder.
4. Select the file to load and click **Open**. A brief explanation of the file appears in the File Information box.
5. Click **Load** to load the selected file and create the data. Note that files must be loaded in the following order:
   1. **default.policy**
   2. **advanced.policy**
   3. **Rule_and_PHB.policy**

> It is important to load the files in order. Always load **default.policy** first, and do not load **Rule_and_PHB.policy** if you have already created PHB groups.

Opening the domain

To open the domain on which you are going to work, double-click on the relevant domain on the **Domains** tab, or select the domain and select **Open** from the pop-up menu.

A new domain management window is opened.



Chapter 6

Defining and Applying Roles

This chapter introduces the concept of roles and describes their function in applying policy and measurement to the network.

The chapter:

- Provides an overview of roles
- Describes the role types that feature in Service Activator – system and user-defined roles
- Explains how to associate a role with a policy target

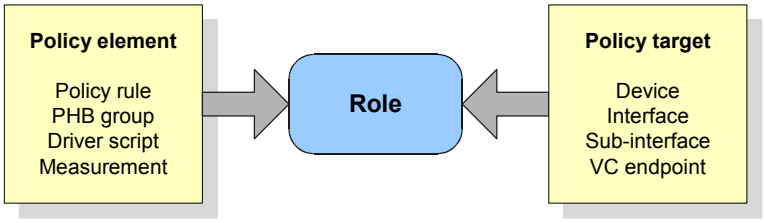
About roles

A role is a means of grouping a set of policy targets that should 'attract' the same policy-based configuration.

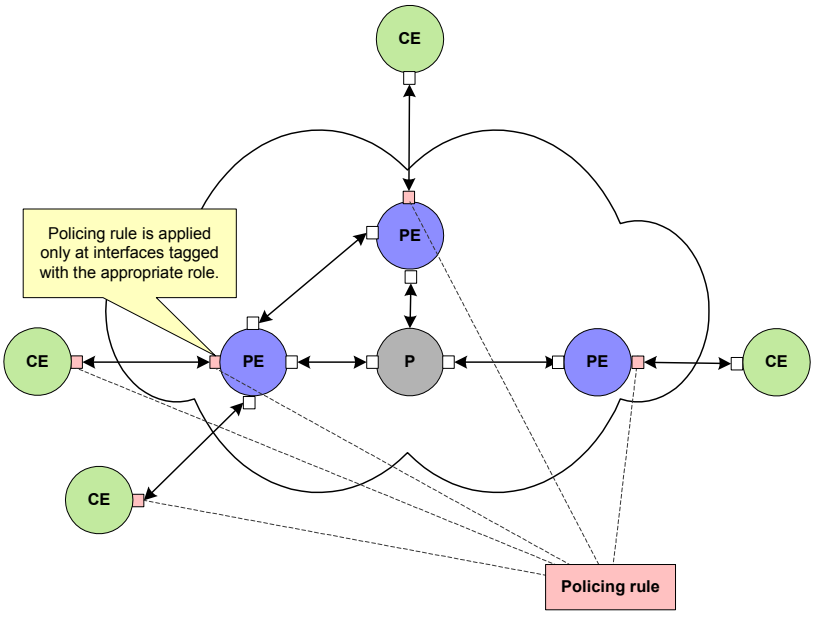
You can associate roles with:

- Some policy targets – a device, interface, sub-interface or VC endpoint
- Policy elements – a rule, PHB group, driver script or measurement policy element (measurement or collector parameters)

Service Activator applies QoS and measurement policy elements to policy targets only where their roles match.



This means that you can target QoS or measurement policy to specific points in the network. The following illustration illustrates this concept using a policing rule.



The ability to associate a role with any policy target down to the VC endpoint level provides a fine degree of control. For example, you can tag a group of sub-interfaces with a role to apply policy at the sub-interface level, independent of their parent interfaces.

Service Activator provides a set of system-defined roles that follow the DiffServ model but you can create additional user-defined roles. For example, it is possible to create roles that group interfaces according to their bandwidth and apply different policy to each set of interfaces in accordance with their bandwidth capacity.

Roles can be assigned to policy targets manually or using role assignment rules. If you intend to use role assignment rules, we recommend you define them before discovering the network.

For information on associating roles with policy elements, see the *Configuring Policy Services* guide. For information on associating roles with collector and measurement parameters, see the *Network and SLA Monitoring Guide*.

Both system and user-defined roles are further subdivided into device and interface roles:

- A device role can be assigned to devices.
- An interface role can be assigned to interfaces, sub-interfaces or VC endpoints.

System and user-defined roles

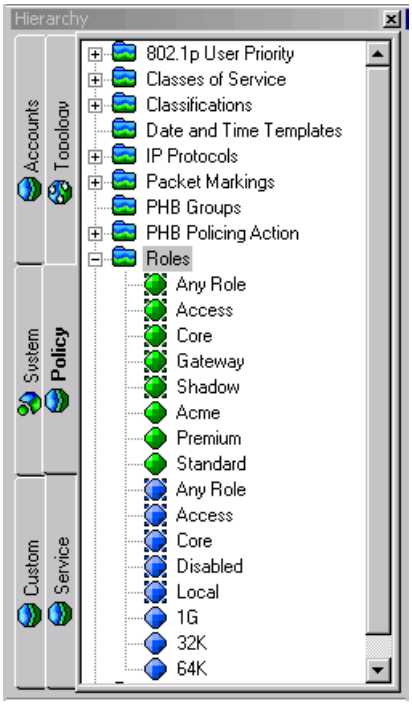
Service Activator recognizes two types of role:

- System-defined – a set of device and interface roles that support the DiffServ policy model.
- User-defined – a role that has been created by a user.

Both system and user-defined roles are available across all domains.

Viewing the roles that are defined in Service Activator

You can list the device and interface roles that are currently defined in Service Activator on the **Policy** tab beneath the **Roles** folder.



The icon associated with a role indicates its type and origin:

- A system-defined device role
- A system-defined interface role
- A user-defined device role
- A user-defined interface role

System-defined roles

System-defined roles follow the DiffServ model – see the *Product Overview* for details of the policy model.

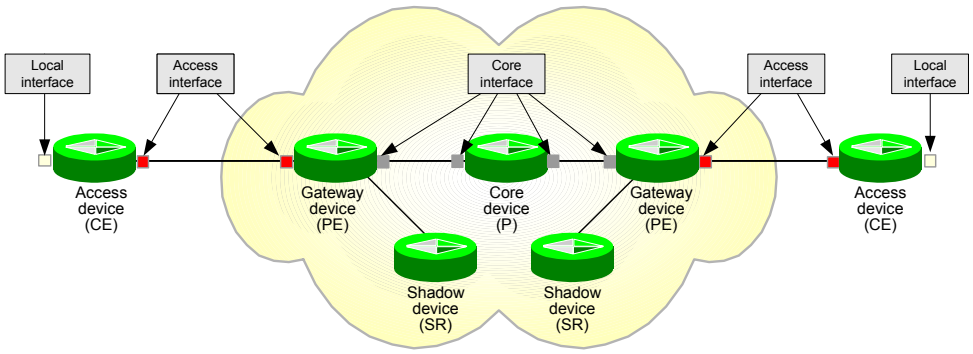
You can assign one system-defined role and multiple user-defined roles to a policy target. Policy elements can be associated with only one system-defined role and one user-defined role.

> If you are implementing MPLS VPNs or measurement-only VPNs with Service Activator you must use system-defined roles. For information on how roles are used within MPLS VPNs, see *Configuring VPN Services*.

The following system-defined roles are provided in Service Activator.

Role		Function
Device	Access	Devices that provide access to the core network or WAN from an external subnet or customer LAN. Equivalent to the Customer Edge (CE) role in MPLS VPNs.
	Gateway	Devices on the edge of the core network or WAN that directly connect to the local or customer access device. Equivalent to the Provider Edge (PE) role in MPLS VPNs.
	Core	Devices within the core network. Equivalent to the Provider (P) role in MPLS VPNs.
	Shadow	Devices dedicated to Service Assurance Agent (SAA) measurements in a Service Provider's Points of Presence (POP).
Interface	Local	An inbound interface from a customer site.
	Access	An interface that connects Gateway and Access devices.
	Core	An interface that connects two Core devices, or a Core and a Gateway devices.
	Disabled	A non-managed interface.

The points in the network at which these roles apply in the DiffServ model are shown in the following diagram:



For information on the Shadow role and SAA measurement, see the *Network and SLA Monitoring Guide*.

> You cannot delete system-defined roles.

User-defined roles

You can create user-defined device and interface roles and assign any number of user-defined roles to a policy target. By assigning multiple roles to a policy target, it becomes a member of several role groups.

You can assign user-defined roles to policy targets in combination with system-defined roles.

The number of roles you need to create will depend on the network set-up and the number of variables that affect policy. For example, you may need to create roles that classify policy targets by link capacity, device function, customer and service package. For an example based on user-defined roles, see *Configuring Policy Services*.

> The ability to create roles depends on your user security level. For more information, see *Setting Up Users* on page 77.

To create a role

1. On the **Policy** tab, select the **Roles** folder.
2. Select **Add Device Role** or **Add Interface Role** from the pop-up menu.

   The **Device** or **Interface Role** dialog box opens.
3. Enter details including **Name** and **Remarks**.

Deleting user-defined roles

You cannot delete a role that is linked to a policy element or to a policy target – unlink the role before deleting it. You cannot delete system-defined roles.

To delete a role

1. On the **Policy** tab, select the **Roles** folder.
2. Select the device or interface role you want to delete.
3. From the role's pop-up menu, select **Delete**.

Assigning a role to a policy target

You must assign one or more roles to each device and interface, sub-interface or VC endpoint to be managed in order to define the points in the network at which services will be configured and policy applied.

You can assign one system-defined role to a device, interface, sub-interface or VC endpoint, and any number of user-defined roles.

You can apply a role by:

- Automatically assigning the role using role assignment rules
- Manually assigning the role to an object

Role assignment rules provide a more maintainable method of applying roles and their use is recommended.

When defining a role assignment rule for an interface, sub-interface or VC endpoint, you can specify whether the role is also applied to policy targets that are lower in the hierarchy. For example, you can assign a role to an interface and specify that the role also applies to any attached sub-interfaces and/or VC endpoints.

You can also specify that the assignment of a role to an interface, sub-interface or VC endpoint is dependent on the role of the attached device. For example, you can apply the system-defined 'Local' interface role only if the role of the attached device is 'Access'.

When you apply a role manually, the role applies to the selected policy target only.

About role assignment rules

Assigning roles to the devices and interfaces in the domain is an essential part of setting up Service Activator.

You can set up the system to apply role assignment rules automatically whenever you discover the network or apply the rules as a separate standalone task. If new devices have been added to the network, Service Activator classifies these devices according to the defined role assignment rules. Role assignment rules therefore provide a more maintainable solution to classifying devices than manual classification. Classifying devices and interfaces manually is suitable only for very small networks, for testing purposes or for overriding the classification applied by a role assignment rule during discovery.

> If you have manually assigned a role to a policy target and then apply role assignment rules, the manually-applied role is overridden where a policy target matches the criteria specified by a role assignment rule. The only exception is the system-defined Disabled interface role, which is never overridden. For information on turning off the application of role assignment rules during discovery, see *Specifying when role assignment rules are applied* on page 131.
>
> A set of pre-defined role assignment rules can be created by loading a Service Activator configuration file. For information, see *Pre-defined role assignment rules* on page 123.

You define role assignment rules on a domain-wide basis. You can view the list of rules defined for a domain by selecting the domain object and selecting the **Role Assignment Rules** tab.

Depending on the domain to be managed and the service or policy you intend to implement you need to define, at minimum, role assignment rules for devices and interfaces (interfaces, sub-interfaces or VC endpoints). If applicable to your system, you can specify within interface role assignment rules that the role also applies to attached sub-interfaces and VC endpoints.

Role assignment criteria

Role assignment rules can be set up for devices, interfaces, sub-interfaces and VC endpoints.

At each "level" the system matches criteria from higher levels. For example, when setting a device role, the system can match device type and name, when matching an interface the device role is also taken into account, and when matching a sub-interface the device role, interface type and interface roles can be considered.

The following table details the criteria each type of role assignment rule uses to match a role to a policy target.

Match criteria	Rule type			
	Device	**Interface**	**Sub-interface**	**VC endpoint**
Device IP address/ mask	✓	✓	✓	✓
DNS name	✓	✓	✓	✓
Device type	✓	✓	✓	✓
Device role	×	✓	✓	✓
Interface type	×	✓	✓	✓
Interface role	×	×	✓	✓
Sub-interface role	×	×	×	✓
Interface parent	×	×	×	✓
Sub-interface parent	×	×	×	✓
Connected device role (optional)	×	✓	✓	✓

All match criteria are mandatory with the exception of connected device role, which is optional for interface, sub-interface and VC endpoint role assignment rules.

Options for interfaces, sub-interfaces and VC endpoints

By default, a role assignment rule applies the specified role to the specified type of policy target only. For example, a rule that applies to interfaces applies the specified role to all interfaces that match the rule's criteria.

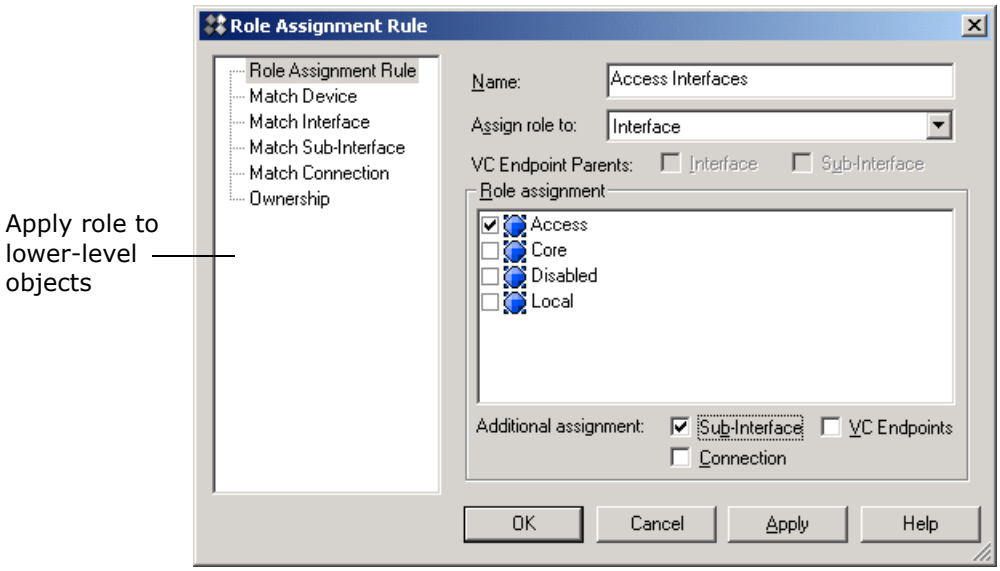
For interfaces, sub-interfaces and VC endpoints, further options are available:

- You can apply the role to attached policy targets at a lower point in the hierarchy, for example, you can set a role on an interface and apply it to all sub-interfaces and VC endpoints as well.

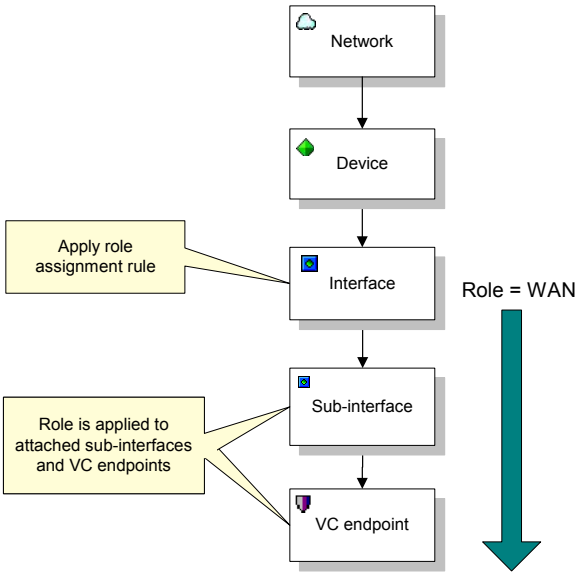
- You can apply the role only when the connected device matches the specified role.

  When you select this option, the role is automatically applied to attached policy targets at a higher point in the hierarchy.

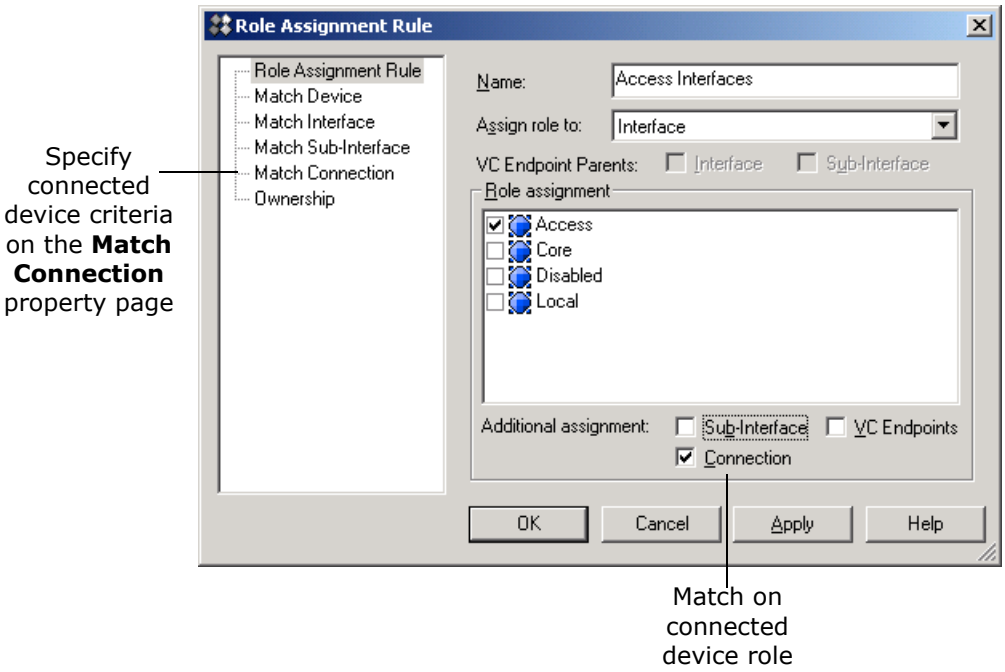
Using **Additional assignment** options, a rule that assigns a role to an interface can also assign the same role to the sub-interfaces and/or VC endpoints. A rule that assigns a role to a sub-interface can also assign the same role to the VC endpoints.



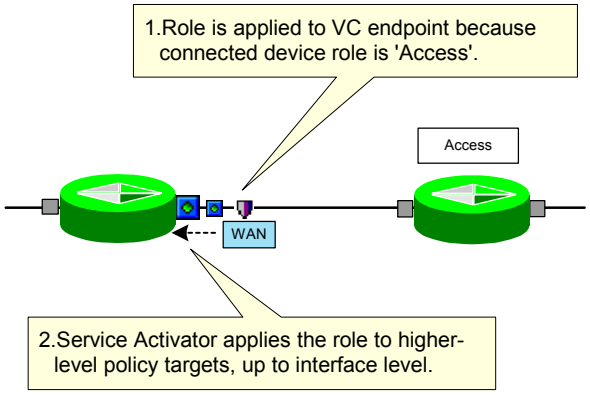
.



Roles can also be applied depending on the role of the connected device. Select the **Connection** option and specify the connected device criteria on the rule's **Match Connection** property page. When a match is dependent on the connected device role, Service Activator also applies the specified role to parent objects up to interface level.



For example, you can specify that the role 'WAN' is applied to a VC endpoint only if the role of the device at the other end of the connection is 'Access'. Service Activator also applies to the 'WAN' role to the VC endpoint's parent sub-interface and interface.



Note that, in this situation, if one of the higher level policy targets already has a role assigned (manually or by another role assignment rule), that role is overridden.

> Service Activator cannot match on the connected device role if a device is connected via a network cloud which connects more than two devices.

If a rule applies to VC endpoints, you must also specify the role of the parent interface and/or sub-interface role, depending on whether the VC endpoint is connected direct to the interface or via a sub-interface.

Pre-defined role assignment rules

A set of pre-defined role assignment rules based on the DiffServ model can be created by loading the supplied **Role_Assignment_Rules.policy** file. For information on loading configuration data, see *Loading policy configuration data* on page 107.

The rules in this file can only be applied if you are using system-defined device and interface roles. Rules allocate a role to an interface according to the devices to which it is connected. For information on system-defined roles, see *System-defined roles* on page 115.

If devices are not directly connected, no role assignment is made to their intervening interfaces. For example, if a device is defined as an Access device and a Gateway device is directly connected to one of its interfaces, then both connecting interfaces will be assigned an Access policy role setting. However, if the connection is not direct, that is, there are undiscovered devices or a segment is between the discovered devices, no role assignment will occur. An indirect connection between two devices is shown by a dotted line on the topology map.

The following table lists the interface assignments that the pre-defined role assignment rules apply, depending on the role of the device to which an interface belongs:

Device 1 – classified by rule	Device 2 – direct connection	Interface on device 1	Interface on device 2
Access	Access	Local	Local
	Gateway	Access	Access
	Core	Disabled	Disabled
	Unknown	Unchanged	Unchanged

Device 1 – classified by rule	Device 2 – direct connection	Interface on device 1	Interface on device 2
Gateway	Access	Access	Access
	Gateway	Core	Core
	Core	Core	Core
	Unknown	Unchanged	Unchanged
Core	Access	Disabled	Disabled
	Gateway	Core	Core
	Core	Core	Core
	Unknown	Unchanged	Unchanged
Unknown	Access	Unchanged	Unchanged
	Gateway	Unchanged	Unchanged
	Core	Unchanged	Unchanged
	Unknown	Unchanged	Unchanged

Creating role assignment rules

You create role assignment rules at the domain level. The order in which rules are listed in the **Details** pane is significant. If a policy target matches the criteria specified in a rule that appears at the top of the list, Service Activator does not check the object against any other rules. You therefore need to ensure that rules are defined in an order that will result in the intended classification.

This section divides role assignment rule creation into the following categories:

- Creating role assignment rules for devices: see *Setting up a device role assignment rule* on page 125.
- Creating role assignment rules for interfaces: see *Setting up an interface role assignment rule* on page 127.
- Creating role assignment rules for sub-interfaces: see *Setting up a sub-interface role assignment rule* on page 128.
- Creating role assignment rules for VC endpoints: see *Setting up a VC endpoint role assignment rule* on page 130.

For more information on rule order for role assignment rules, see *Viewing and managing role assignment rules* on page 132.

> Before creating role assignment rules, ensure you have set up all the device and interface roles you need to use.
>
> You can load the **Role_Assignment_Rules.policy** configuration file to create a set of example role assignment rules. For more information, see *Pre-defined role assignment rules* on page 123. For information on loading policy files, see *Loading policy configuration data* on page 107.

Setting up a device role assignment rule

Set up device role assignment rules to assign system-defined and user-defined device roles to the various devices in your network.

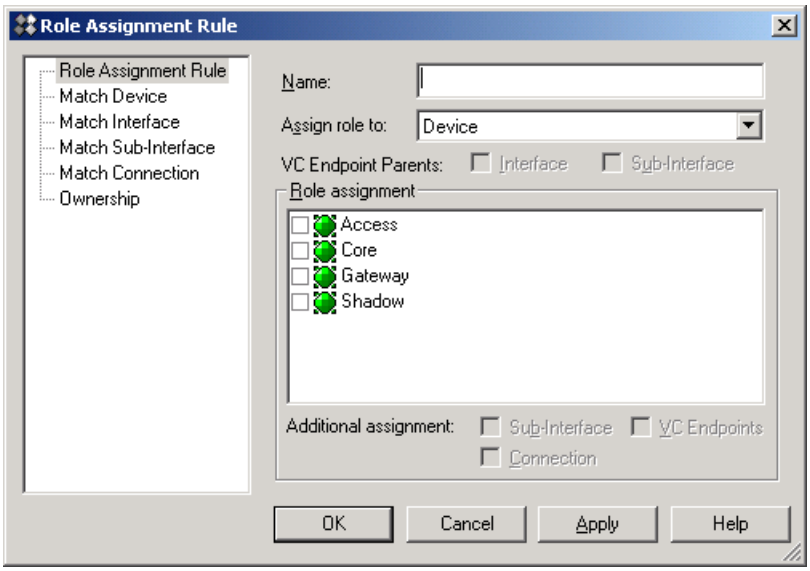
To create a device role assignment rule

1. Select the **Domains** tab on the appropriate global setup window.
2. If you want to display the list of role assignment rules as you work on them, select the domain object and click on the Configuration button on the toolbar:

3. Right-click the relevant domain object and select **Add Role Assignment Rule** from the pop-up menu.

The **Role Assignment Rule** dialog box opens.



4. On the **Role Assignment Rule** property page, define the rule and specify the role(s) to be assigned using the **Name** and **Role assignment** fields. For **Assign role to**: select **Device**. Note that icons for system-defined roles have a square background.
5. Select the **Match Device** property page and set the criteria for devices to match:
   - **Device Type**: Select from any device type or set the field to Any. The dialog box lists all valid device types, that is, those in the **DeviceTypes.cfg** file.
   - **DNS Name**: Specify the DNS name or include the **.*** wildcard to represent any sequence of characters.

> If you want to specify that the rule may be matched against any DNS name, the **.*** wildcard must appear in the **DNS Name** field. If this field is blank, the rule will not match against any device.

   - **Address Range/Mask**: By default, all IP addresses are matched. To specify a particular address or range, clear the **All** checkbox and set the IP address and mask values.

   Note that the device must meet **all** the conditions for the role to be assigned.
6. Click **Apply** or **OK** to set up the rule.

Other property pages are not required when setting device roles.

Setting up an interface role assignment rule

Set up interface role assignment rules to assign system-defined and user-defined interface roles to the interfaces to be managed in your network, and optionally, to apply the same roles to sub-interfaces and/or VC endpoints on those interfaces. Interface roles can be dependent on any combination of the device type, DNS name, address or role, and on the interface type. Ensure you have set up all the device role assignments first.

Note: For complete dialog box and property page descriptions, refer to the *Online Help*.

To create an interface role assignment rule

1. Right-click the relevant domain object and select **Add Role Assignment Rule** from the pop-up menu.

   The **Role Assignment Rule** dialog box opens.

2. On the **Role Assignment Rule** property page, define the rule and specify the role(s) to be assigned to interfaces. Specify values for **Name**, **Assign role to** (select **Interface**), **Role assignment, Additional assignment, Connection.**

   Also see *Options for interfaces, sub-interfaces and VC endpoints* on page 119.

3. Optionally, select the **Match Device** property page and set the criteria for devices to match against.

> If you want to specify that the rule may be matched against any DNS name, the .* wildcard must appear in the **DNS Name** field. If this field is blank, the rule will not match against any device.

4. Select the **Match Interface** property page to specify the interface type to be matched. Note that the rest of this page is greyed out when applying a role to an interface.

5. If you selected the **Connection** checkbox on the **Role Assignment Rule** property page, select the **Match Connection** property page to make the interface role assignment dependent on the role of an attached device:

   - Select one system-defined role and/or one user-defined role and select **Add** to add them to the rule criteria.

     By default, the system-defined 'Any Role' is selected, which means that the role is assigned regardless of the role of the connected device.

   Note that the interface must meet **all** the conditions for the role to be assigned.

6. Click **Apply** or **OK** to set up the rule.

   Other property pages are not required when setting interface roles.

Setting up a sub-interface role assignment rule

Set up sub-interface role assignment rules to assign system-defined and user-defined interface roles to the sub-interfaces to be managed in your network, and optionally, to apply the same roles to VC endpoints on those sub-interfaces. Sub-interface roles can be dependent on any combination of the device type, DNS name, address and role, and the interface type and role. Ensure you have set up all the device and interface role assignments first.

To create a sub-interface role assignment rule

1. Right-click the relevant domain object and select **Add Role Assignment Rule** from the pop-up menu.

   The **Role Assignment Rule** dialog box opens.

2. On the **Role Assignment Rule** property page, define the rule and specify the role(s) to be assigned to sub-interfaces. For **Assign role to**: select **Sub-Interface**.

   Also see *Options for interfaces, sub-interfaces and VC endpoints* on page 119.

3. Select the **Match Device** property page and set the criteria for devices to match against.

> If you want to specify that the rule may be matched against any DNS name, the .* wildcard must appear in the **DNS Name** field. If this field is blank, the rule will not match against any device.

4. Select the **Match Interface** property page and set the criteria for interfaces to match against.

5. If you selected the **Connection** checkbox on the **Role Assignment Rule** property page, select the **Match Connection** property page to make the sub-interface role assignment dependent on the role of an attached device:

   - Select one system-defined role and/or one user-defined role and select **Add** to add them to the rule criteria.

   By default, the system-defined 'Any Role' is selected, which means that the role is assigned regardless of the role of the connected device.

6. Click **Apply** or **OK** to set up the rule.

> Note that the sub-interface must meet all the conditions for the role to be applied.

Setting up a VC endpoint role assignment rule

Set up VC endpoint role assignment rules to assign system-defined and user-defined interface roles to the VC endpoints to be managed in your network. VC endpoint roles can be dependent on any combination of the device type, DNS name, address and role, the interface type and role, and the sub-interface role. Ensure you have set up all the device, interface and sub-interface role assignments first.

To create a VC endpoint role assignment rule

1. Right-click the relevant domain object and select **Add Role Assignment Rule** from the pop-up menu.

   The **Role Assignment Rule** dialog box opens.

2. On the **Role Assignment Rule** property page, define the rule and specify the role(s) to be assigned to VC endpoints. For **Assign role to**: select **ATM VC End Point** or **Frame Relay End Point**.

> Role assignment will not work if roles are not assigned to all the objects identified as VC endpoint parents.

   Also See *Options for interfaces, sub-interfaces and VC endpoints* on page 119.

   Note that you must select the **Connection** option if you wish to include the connected device role in the rule's criteria.

3. Select the **Match Device** property page and set the criteria for devices to match against.
4. Select the **Match Interface** property page and set the criteria for interfaces to match against.
5. If you specified that the VC endpoint had a sub-interface as a parent, select the **Match Sub-Interface** property page to set the sub-interface match criteria.
6. If you selected the **Connection** checkbox on the **Role Assignment Rule** property page, select the **Match Connection** property page to make the VC endpoint role assignment dependent on the role of an attached device:
   - Select one system-defined role and/or one user-defined role and select **Add** to add them to the rule criteria.

   By default, the system-defined 'Any Role' is selected, which means that the role is assigned regardless of the role of the connected device.

7. Click **Apply** or **OK** to set up the rule.

> Note that the VC endpoint must meet all the conditions for the role to be applied.

Example of role assignment rules

Using role assignment rules allows fine control over policy management. The following example explains how to set up role assignment rules if you want to enable sub-interfaces to be managed without configuring the interfaces they are on. The example assigns all Fast Ethernet interfaces on Catalyst 6509 devices a role of Disabled, and assigns all sub-interfaces on these interfaces the role of Access.

Follow these steps:

1. Ensure suitable device and interface roles are set up.
2. Create a *device role assignment rule* to assign appropriate devices (for example Cisco Catalyst 6509 devices) a role of "Catalyst". Set the match criteria on the **Match Device** property page to match on Device Type.
3. Create an *interface role assignment* rule to assign all appropriate interfaces (such as all Ethernet interfaces) a role of "Disabled". This ensures the interfaces themselves are not configured by Service Activator. Set the match criteria to match the **Device Role** of Catalyst (on the **Match Device** property page) and the **Interface Type** of fastEther(62), for example, on the **Match Interface** property page.
4. Create a *sub-interface role assignment rule* to assign all sub-interfaces on those Disabled interfaces a role of "Access". Set the match criteria to match the **Device Role** of Catalyst (on the **Match Device** property page), the **Interface Type** of fastEther(62) and the **Interface Role** of Disabled (on the **Match Interface** property page).
5. Apply all role assignment rules.

Specifying when role assignment rules are applied

You can apply role assignment rules automatically whenever the network is discovered or as a one-off task. By default, rules are automatically applied on discovery.

To switch automatic application of role assignment rules on and off

1. From the **Tools** menu, select **Options**.

   The **Options** dialog box opens.
2. Select the **Discovery** property page.

3. Set the **Automatic Role Assignment on Discovery** option:
   - When the option is selected, rules are automatically applied after every discovery
   - When deselected, rules are not applied automatically

To make a one-off application of role assignment rules

1. Do one of the following:
   - In the **Hierarchy** pane, select a network object
   - In the **Details** pane, select a point on the topology map
2. Select **Apply Role Assignment Rules** from the pop-up menu.

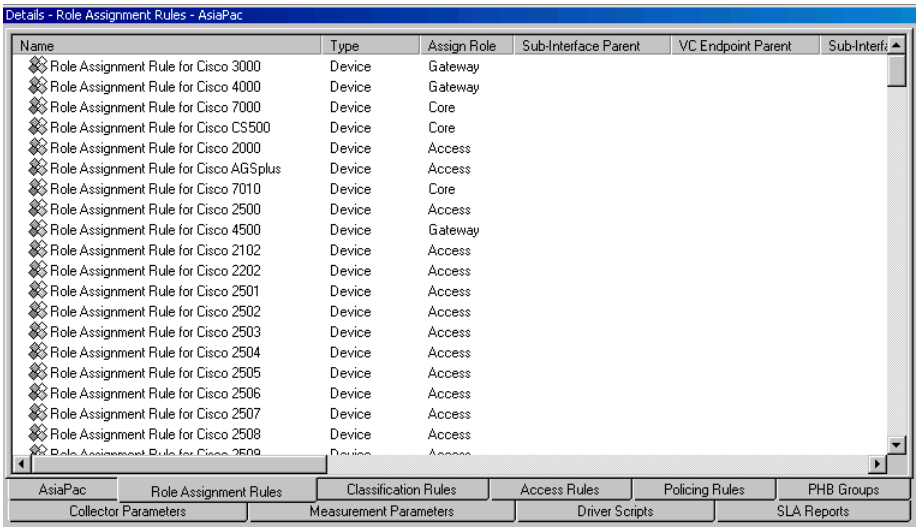
Viewing and managing role assignment rules

Role assignment rules are applied in strictly sequential order. If a policy target matches the criteria defined in a rule at the top of the list, Service Activator does not check any further rules. Therefore, as you create role assignment rules, you may need to adjust the order in which they are applied or delete unnecessary rules.

You can see a list of rules that have been created for a domain in the **Details** pane. Service Activator lists rules in the following order:

- Device
- Interface
- Sub-interface
- VC endpoint

If you have loaded the **Role_Assignment_Rules.policy** file and created your own role assignment rules, Service Activator lists example rules above user-defined rules within each rule type.



When you create a new rule, Service Activator places the rule at the appropriate point in the list – that is, a rule that classifies interfaces is placed with other interface rules.

Whenever role assignment roles are applied to a newly-discovered policy target, Service Activator checks the role assignment rules in the order in which they are listed in the **Details** pane. If an object matches a rule's criteria, Service Activator allocates the role specified by that rule and no other rules are checked.

Because Service Activator stops checking rules after a match is made, rule order is critical. More specific rules should appear above general rules in the list. You can amend the order in which rules are listed by selecting and dragging a rule to a higher or lower position in the list. You cannot move a rule out of its position in the hierarchy, however – that is, you cannot move an interface rule above a device rule, and so on.

Developing suitable role assignment rules and a correct list order is likely to be an iterative process.

To view a domain's role assignment rules

1. In the global setup window, select the domain object.
2. Select the **Role Assignment Rules** tab.

To change the list order of rules

- With the domain's role assignment rules displayed, click on a rule and drag it to a new list position.

You can re-order rules within sub-categories. However, you cannot drag a rule out of the appropriate category – for example, you cannot place an interface rule above a device rule. For more information, see *Viewing and managing role assignment rules* on page 132.

To edit a rule

1. With the domain's role assignment rules displayed, double-click on the relevant rule or select the rule and select **Properties** from the rule's pop-up menu.

   The rule's properties dialog box opens.

2. Edit the rule as necessary and select **OK** when complete.

Assigning roles to a policy target manually

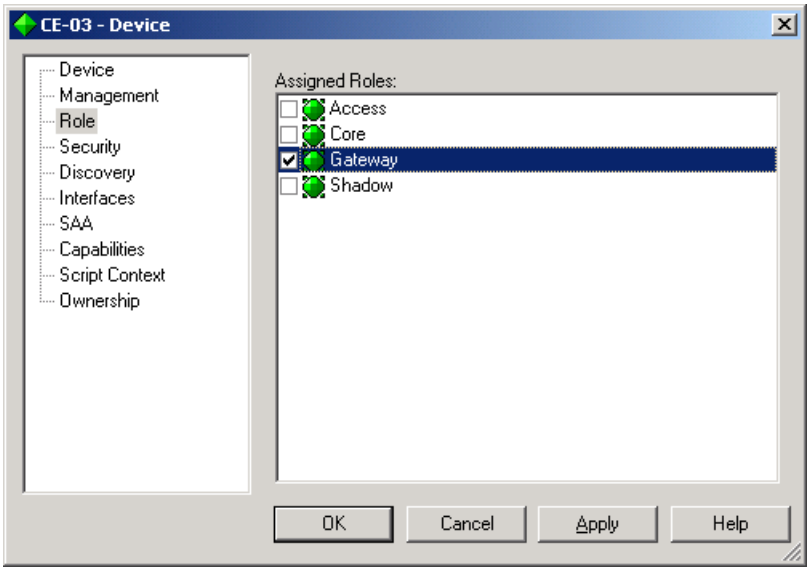
You can apply a role to a policy target manually using the object's property pages. Note that, if role assignment rules are subsequently applied, where a policy target matches the parameters specified by a role assignment rule, a manually-assigned role is overridden. The exception to this is the system-defined interface role Disabled, which is never overridden.

You can switch off automatic override of manually-assigned roles. For more information, see *Specifying when role assignment rules are applied* on page 131.

To assign/unassign a role manually

1. Open the **Topology** tab and select the device, interface, sub-interface or VC endpoint to which you want to assign a role.
2. From the device or interface's pop-up menu, select **Properties** and select the **Role** property page.

   The property page lists the currently-defined device or interface roles.



3. If you want to assign a role, select the checkbox associated with the role name.

   Note that you can only assign one system-defined role. Selecting another pre-defined role to the list replaces the previously-assigned role. You can assign any number of user-defined roles.

4. If you want to unassign a role, deselect the checkbox associated with the role name.

Chapter 7

Discovering and Setting Up the Network

Before you can apply services or set up policies, you need to set up an accurate representation of the network to be managed. Service Activator can automatically discover all IP addressable network elements within your network, that is, devices, hosts and network segments, and set up appropriate information. Once you have discovered the devices to be managed, you can set up maps showing the network topology.

This chapter includes the following:

- Introduction to network discovery
- Steps you need to take before running device discovery, including setting the correct domain information
- The initial discovery of the network, including different methods of network discovery
- Steps you need to take after device discovery, to ensure that the information is complete

Introduction to the discovery process

During network discovery, Service Activator finds out details of devices, hosts and network segments within the domain that you are managing. It also retrieves each device's interface capabilities and, where possible, sets up the information that Service Activator needs in order to manage the devices. The discovery process includes the following stages:

- Discovering devices, segments and hosts
- Assigning devices to proxy agents
- Assigning policy roles to devices and interfaces if role assignment rules have been set up
- Setting up the security parameters that Service Activator needs to configure devices
- Discovering the capabilities of devices and interfaces – the VPN, QoS, security and measurement options that are available

Discovering the network

The topology discovery process finds out information about the network topology by interrogating network nodes using SNMP. Given an IP address or DNS name, Service Activator discovers details of the node. If the node is classified as a device (that is, it forwards IP packets – for example, a router or a Layer 2 switch) details of connected network segments, interfaces and any PVC endpoints are obtained. From a segment, Service Activator can find further directly-connected nodes, that is, hosts and devices.

The discovery process is controlled by a set of SNMP parameters; default values are assumed by Service Activator, but these can be overwritten if required.

The way in which discovery works depends on whether public or private IP addresses are used within the domain. See *Before running device discovery* on page 141.

Assigning devices to proxy agents

All devices in the domain that are to be managed must be assigned to a proxy agent. The proxy agents are the Service Activator components that manage the low-level device driver commands for each device type.

Assigning devices to proxy agents is generally performed automatically during device discovery. You can set up Service Activator either to assign all devices to one proxy agent or to assign the devices equally to all active proxy agents. (For more information, see *Setting up proxy agent assignment* on page 106.)

Assigning roles to devices and interfaces

All devices and interfaces to be managed must be assigned a role in order to define the points in the network at which service configuration or policy will be applied. Devices can be classified with any system-defined and/or user-defined roles that have been set up within the domain.

If role assignment rules have been defined they are applied by default after discovery. Role assignment rules allow you to specify that particular routers in the network should be classified as Access, Core, and so on, depending on the type of device, DNS name or range of IP addresses.

> Service Activator provides a set of example role assignment rules in the **Role_Assignment_Rules.policy** file. These rules assign roles according to device type.
>
> If role assignment rules are not set up before discovery, devices and interfaces are not automatically assigned roles. In this case you need to assign the appropriate policy role to each device and interface manually. For a large network, we recommend that you spend some time devising suitable role assignment rules and then rediscover the network.

For more information on defining role assignment rules, see *Creating role assignment rules* on page 124.

Setting up device security parameters

For each device, appropriate security parameters, such as user IDs and passwords must be specified to allow Service Activator to configure the device. This information must be set up prior to discovery as Service Activator requires write access to routers before it can obtain the device and interface capabilities.

By setting up standard security information it will be applied automatically to all discovered devices. This can save time if you use the same access methods and passwords for a number of devices in your network. For more details, see *Defining default security options for discovery* on page 159.

Discovery capabilities

At the end of the discovery process, Service Activator attempts to discover the capabilities of each device and its interfaces. These capabilities dictate the VPN, QoS, security and measurement options available. For more information, see *Checking capabilities* on page 210. If Service Activator is unable to discover capabilities, for example because security parameters are incorrectly set up, you can discover the device capabilities at a later point in time. See *Refetching device capabilities* on page 213.

The discovery process can open a large number of file descriptors on Solaris systems. The size of the fd_set used is FD_SETSIZE, which is hard-coded to be 1024 on Solaris (in **/usr/include/sys/select.h**). By default, Service Activator can use half of this maximum, i.e. 512. An alternative value can be set by specifying the following command-line parameter to the policy server:

```
-SnmpMaxSessions <xxx>
```

where *xxx* is the maximum number of file descriptors, between 1 and 1024, with the default value being 512.

BGP Autonomous System discovery

Each router managed by Service Activator has a BGP Autonomous System (AS) number. Service Activator supports a default AS at the domain level but individual routers within this domain may have a different AS than the domain default. A domain that contains devices belonging to different Autonomous Systems is referred to as a Multi-AS domain.

Service Activator supports Multi-AS domains by actively discovering the BGP AS number configured into each device as it is discovered, or re-discovered, in the system. All BGP provisioning on a device will use the last discovered BGP AS number. If a device changes its Autonomous System membership, it must be re-discovered to ensure Service Activator is aware of the change.

During device discovery, if the discovered AS number differs from the default Domain AS number, Service Activator raises a warning against the device (Code 1627): "<DeviceName> (<DeviceManagementIPAddress>) ASN on device does not match Domain ASN".

This warning is for information purposes only and will not inhibit service provisioning using the discovered AS number. If the device being discovered already has roles, due to re-discovery or role assignment rules, this behavior is seen only for devices assigned the role of Gateway or Core. For devices assigned the role of Access, such as CE devices, AS number validation is not performed against the default domain AS. Instead, if the device is found to belong to a VPN site and if the site connectivity includes EBGP, the discovered AS number is matched against the peer BGP AS number specified on the VPN site. If they are found to differ, the peer BGP AS number in the VPN site is automatically modified to match the discovered CE-device AS number.

If a managed device is re-discovered, and the newly discovered AS number does not match the previously discovered AS number, a warning is raised against the device (Code 1631): "The device <DeviceName> (<DeviceManagementIPAddress>) has a non-zero ASN in the OM that is different from the discovered ASN".

The pre-existing AS number for the device with Service Activator is not changed. To actually change the AS number for the device, the device must first be unmanaged and then re-discovered.

Service Activator displays the discovered device AS number (ASN) in the Device dialog box on the Device property page.

Before running device discovery

This section explains the steps that you need to take before starting the device discovery process:

- Ensure domain details are correctly set up.
- Ensure proxy agents are set up for automatic assignment.
- Specify the system's use of IP addresses for device management.
- Set up role assignment rules.

Setting up the domain details

The way in which discovery works depends on whether public or private IP addresses are used within the domain. For the discovery process to work correctly, ensure that the domain type is set correctly on the **Domain** property page before running the discovery process. This can be **Public**, **Private** or **MPLS VPN**.

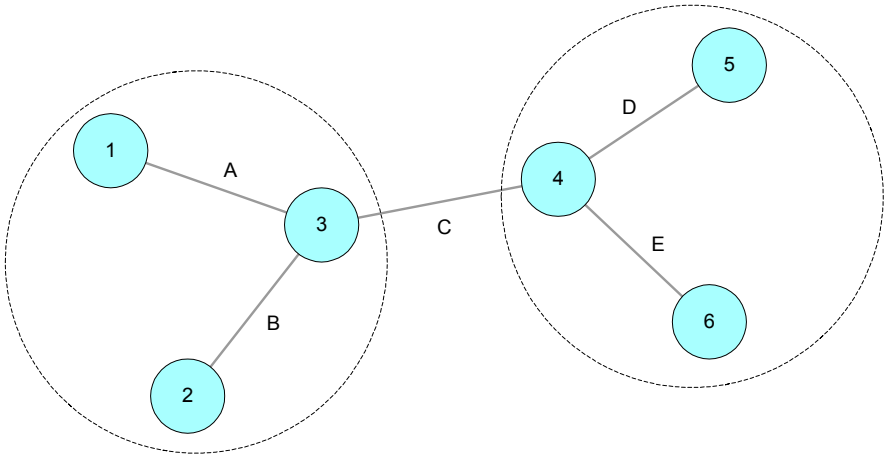
> You must create one domain per AS region.

For information on setting up domain details, see *Setting up domains* on page 104.

Public domains

If the domain is defined as **Public**, Service Activator can discover an entire network starting from a single device, since all IP addresses found are unique. The extent of the discovery can be controlled by specifying the number of hops to go from the original device. Alternatively, if no IP addresses or DNS names are known, a discovery can be started from the segment local to the policy server to find details of all connected devices, hosts and further segments.

A device can only be in one domain, but where a segment links two devices that appear in two different domains, the segment will appear in both domains.



If devices 1, 2 and 3 are in one domain and devices 4, 5 and 6 in another, a representation of segment C will appear in both domains.

In a public domain, the IP address of a device specified for the discovery process may be changed. For example, if a Cisco device has a loopback address defined, this will be used to identify the device in preference to any IP address previously used. The default loopback address used is 0 but may have been configured for the domain.

See *Setting the default loopback ID value for discovery* on page 105 for details.

Private domains

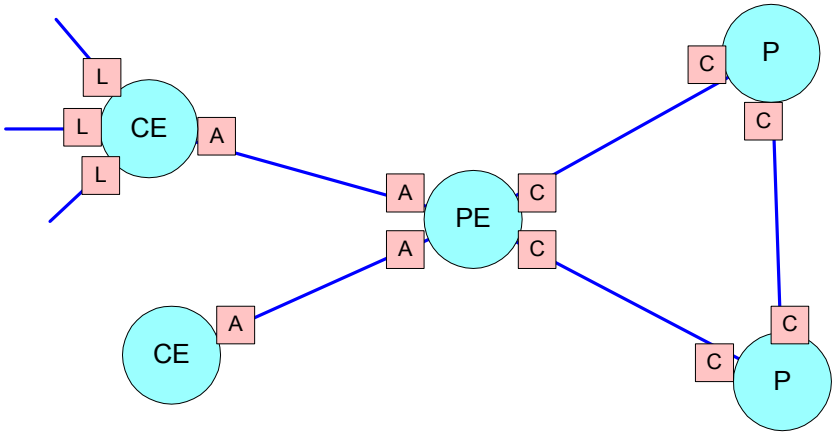
If the domain is defined as Private, each device must be identifiable by a unique IP address, which may be a public address or a Network Address Translation (NAT) address. However, some or all of the interface IP addresses are likely to be non-unique. Therefore the discovery process requires a unique IP address or DNS name to be specified, and is not able to find further devices. The **Hops** field on the **Discovery** dialog box is therefore ignored. Note that connections between discovered segments are not automatically shown in the user interface, though these connections can be created manually by dragging one segment on to another.

MPLS VPN domains

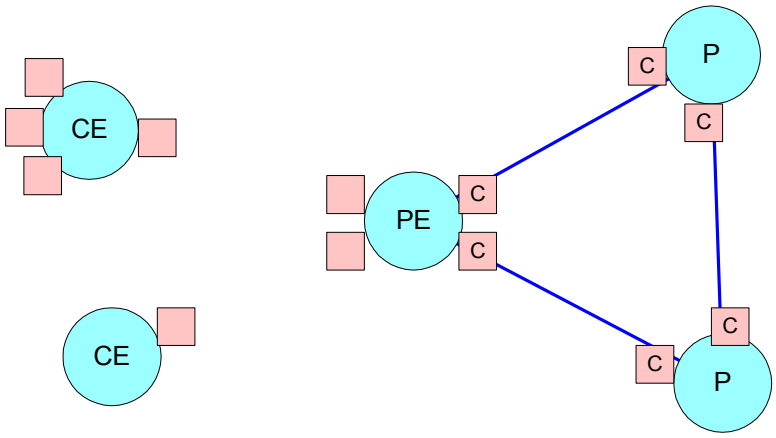
For MPLS VPNs the core provider network is assumed to use public addresses, and Service Activator can discover an entire network starting from a single device. All CE routers are assumed to use private addresses and an IP address or DNS name must be specified in order to discover them.

If PE interfaces that are connected to CE devices have public addresses, the PE and CE devices will be automatically connected to segments during discovery.

Depending on the role assignment rules applied, devices and interfaces will be assigned appropriate roles.



If PE interfaces that are connected to CE devices have private addresses, connectivity cannot be determined. Therefore CE devices and access interfaces on PE devices will not automatically be connected to segments. In addition, because the devices are not directly connected, interfaces will not automatically be assigned roles.



The connections between the PE and CE devices can be applied manually, by dragging one interface on to another on the topology map.

> Note that if a role is assigned to the CE device – for example, by a role assignment rule during discovery – Service Activator does not show the CE device's connection to the PE device. If there is no role assigned to the CE device, however, Service Activator shows the connection.

Setting up proxy agents for automatic assignment

All devices in the domain that are to be managed by Service Activator must be assigned to a proxy agent. It is the proxy agent that controls when and what type of configuration is to be applied to a specific interface.

Although it is possible to assign devices to proxy agents manually, it is generally performed automatically during device discovery.

For information on setting up proxy agents for automatic assignment, see page 106.

Defining the way in which IP addresses are used

Service Activator uses a specific IP address when managing devices. Commonly this is the loopback address, but if this is not a unique address it is possible to use an alternative. On the **Options** dialog box you can set up a global option to define the standard way in which IP addresses are used. If necessary you can override this for individual devices (see *Setting specific IP addresses for device management* on page 165).

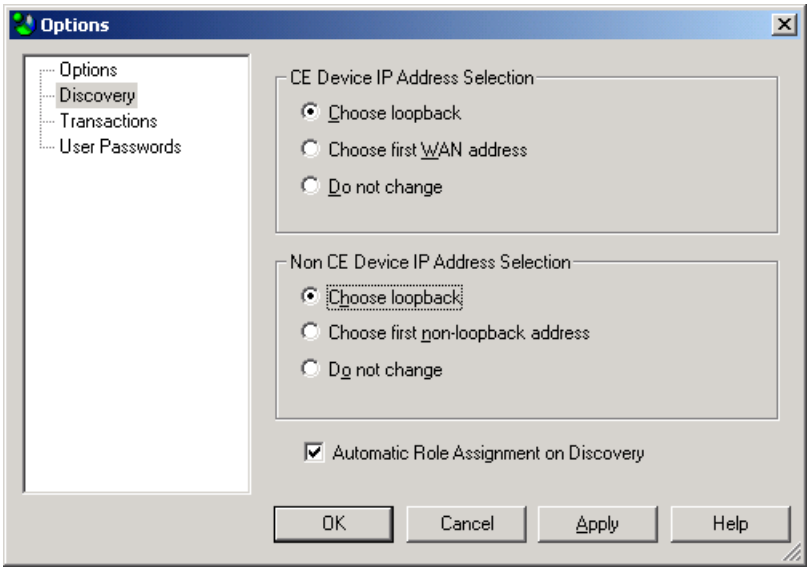
Note: For complete dialog box and property page descriptions, refer to the *Online Help*.

To set global IP address usage

1. From the **Tools** menu, select **Options**.

   The **Options** dialog box opens.

2. Select the **Discovery** property page.



3. Under **CE Device IP Address Selection**, select the global option used to set the IP address for CE devices. Choose from **Choose loopback**, **Choose first WAN address**, and **Do not change.**

   Also see *Setting the default loopback ID value for discovery* on page 105.)

4. Under **Non CE Device IP Address Selection**, select the global option used to set the IP address for all devices that are not CE devices. Choose from **Choose loopback, Choose first non-loopback address**, and **Do not change**.

   Also see *Setting the default loopback ID value for discovery* on page 105.)

5. Click **OK**.

Setting up roles and role assignment rules

A role is a label that can be applied to one or more policy targets. Roles define:

- The policy or service configuration that is applied to that object
- For external collector systems, from which devices data is collected

A set of system-defined roles is supplied and user-defined roles can be set up in line with any policy model. For information on defining roles, see *Defining and Applying Roles* on page 111.

The recommended method for applying roles is using role assignment rules which are applied during device discovery. Therefore, before discovering the domain, you should identify the roles that feature in the domain and create suitable role assignment rules.

You can also apply role assignment rules independent of discovery. For more information, see *Specifying when role assignment rules are applied* on page 131.

For information on defining roles and role assignment rules, see *Defining and Applying Roles* on page 111.

Customizing discovery using Autodiscovery.cfg

The **Autodiscovery.cfg** file is used to specify how Service Activator interprets discovery information from devices in terms of translating the SNMP MIB-2 data into objects. It does this through a number of rules matching the formats shown below. Each rule defines a set of matching conditions. If a discovered item matches the conditions, it is reported into Service Activator as the type of object the rule specifies.

The **Autodiscovery.cfg** file is located in `opt/OracleCommunications/Service Activator/Config`.

Note: Be extremely careful when editing the **Autodiscovery.cfg** file. Do not change the settings unless you fully understand the impact on discovery. You should fully understand SNMP, ifTable, ifStack, and core MIB-2 concepts before creating changes to the **Autodiscovery.cfg** file. It is important to construct these statements with extreme care, because conflicts are possible between statements and particularly, regex statements.

The Policy Server reads the **Autodiscovery.cfg** file on startup. In order for changes to the **Autodiscovery.cfg** file to take effect, the Policy Server must be restarted.

In general, the Boost regex (regular expression) is used.

The **Autodiscovery.cfg** file is made up of the following types of entries. The basic mechanism is to evaluate each returned MIB-2 object from SNMP against the statements in the **Autodiscover.cfg** file. The discovery information is matched against the statement parameters to determine how Service Activator should handle the object.

The possible statement types in the **Autodiscovery.cfg** are:

- Enterprise:<enterpriseNumber>;<enterpriseName>;<deviceDriver>;<supported>;<accessType>;<configLevel>;
- Subinterface:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<Invert>;
- Sublayer:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<Invert>;
- Vlan:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<Invert>;

- Vlanport:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<invert>;
- Main:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;
- Ignore:<enterpriseNumber>;<highIfType>;<regexp>;
- Ifname:<enterpriseNumber1>;<enterpriseNumber2>;<ifType>
- Ifdesc:<enterpriseNumber1>;<enterpriseNumber2>;<ifType>
- Persistant:<enterpriseNumber1>;<enterpriseNumber2>;
- Volatile:<enterpriseNumber1>;<enterpriseNumber2>;
- Rename:<enterpriseNumber>;<ifType>;<pattern to match>;<pattern to substitute>;
- Icmp:[on | off]
- AtmVcInterfaceSource:<enterpriseNumber>;<aal5VccTable/atmVclTable>
- Host:<enterpriseNumber1>;<enterpriseNumber2>;
- Device:<enterpriseNumber1>;<enterpriseNumber2>;
- Controller:<enterpriseNumber>;<ifType>;<regexp>;

Most of the statements function in a similar way. Generally for statement X, if the MIB-2 information for an object matches the parameters in the statement, the object is treated as an object of type X in the Service Activator object model.

Enterprise

The Enterprise type defines a vendor to the discovery system. The Enterprise type identifies the device as belonging to a particular vendor and defines how Service Activator can communicate with the device, and what paradigm Service Activator uses when we configure the device.

The syntax is as follows:

```
Enterprise:<enterpriseNumber>;<enterpriseName>;<deviceDriver>;<supported>;<accessType>;<configLevel>;
```

where:

- <enterpriseNumber> - registered enterprise number to match on (e.g. Cisco = 9). This comes from the sysObjectId in the MIB2 report from the device for this vendor.
- <enterpriseName> - text string used to visually represent the vendor
- <deviceDriver> - the device driver / cartridge type that must be used to deal with devices from this vendor (if not supported, leave empty).

- <supported> - tells Service Activator whether any configuration other than discovery can be performed with this vendor's devices (yes/no).
- <accessType> - the default mechanism to get into the device for the purpose of command delivery (TACACS, NamedUser, anonymous, SNMPv1, SNMPv2c, none, PasswordOnly) - the details of the access mechanisms are documented in the Discovery section of the *Online Help.*
- <configLevel> - defines whether the device is to be configured as a whole or if it can be treated as a number of individual interfaces. Specify either Device or Interface.

For example:

```
Enterprise:9;Cisco;cisco;yes;TACACS;Interface;
```

This statement indicates that for devices with enterprise number 9, name the vendor as 'Cisco', use the cisco device driver, access the devices using TACACS, and treat the device as though it has a number of interfaces below the device in hierarchy.

Subinterface

The syntax for Subinterface is as follow:

```
Subinterface:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<I
nvert>;
```

- <enterprise_number> - registered enterprise number to match on.
- <highIfType> - Interface IfType for higher layer in ifStack.
- <lowerIfType> - Interface IfType for lower layer in ifStack.
- <regexp> - regular expression to match the reported name on.
- <Invert> - invert flag. The optional Invert value indicates that Service Activator should invert whatever relationship was reported by SNMP. For example, if SNMP reports that 'B' is lower than 'A', we might actually want 'A' to be a Subinterface not 'B'. The invert flag compensates for this situation.

The use of the Subinterface, Sublayer, and Main statements is to enforce the hierarchy of interfaces based on type (and regex matching on the name).

For example:

```
Subinterface:9;32;32;.*;
```

This statement indicates that an interface discovered that has an enterprise number of 9 and if ifStack reports a higher ifType of 32 and a lower ifType of 32, and the object is named anything, treat it as a subinterface'.

Sublayer

The parameters for Sublayer are similar to Subinterface.

Sometimes devices deal with interfaces at a level of abstraction or granularity that is not reflected in the available discovery types in Service Activator. Objects can be reported as interfaces that are not actually configurable interfaces. For example, some artifacts of configuration are reported through SNMP as interfaces.

For example:

```
Sublayer:9;134;0;.*\.0;
```

The Sublayer statement is used to keep the reported object matching the statement in the Service Activator object model without storing it as an interface or subinterface.

Vlan and Vlanport

The syntax for Vlan and Vlanport are as follows:

```
Vlan:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<Invert>;
Vlanport:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;<inver
t>;
```

The parameters for Vlan and Vlan port are similar to Subinterface.

VLAN and VLAN Port identify the discovered interfaces as VLAN or VLAN Port interfaces respectively as opposed to general use interfaces for devices for which these are not reported correctly through SNMP.

Main

The syntax for Main is as follows:

```
Main:<enterpriseNumber>;<highIfType>;<lowerIfType>;<regexp>;
```

The parameters for Main are similar to Subinterface.

For this an interface, if the enterprise number, interface pair, and optionally the regex match, treat it as a main interface in Service Activator.

For example:

```
Main:9;32;39;^Serial.*;
```

This example would map Cisco Frame Relay interfaces on a SONET controller as main interfaces in Service Activator if the description matches the regular expression ^Serial.*.

Ignore

The syntax for Ignore is as follows:

```
Ignore:<enterpriseNumber>;<highIfType>;<regexp>;
```

This means for the device matching the enterprise number, interfaces matching the high interface type number (and optionally matching the regex) are ignored. Discovery of matching interfaces will not be reported to Service Activator.

For example:

```
Ignore:9;94;.*-adsl;
```

This statement causes discovery to ignore ASDL interfaces on Cisco devices.

Ifname and Ifdesc

The syntax are as follow:

```
Ifname:<enterpriseNumber1>;<enterpriseNumber2>;<ifType>
Ifdesc:<enterpriseNumber1>;<enterpriseNumber2>;<ifType>
```

SNMP supports two different text representations of the interface - name and description. Some vendors put the canonical name of the interface in the 'name' field, while other put it in the 'desc' field. These statements allow you to account for these variations. By default, Service Activator uses the description to map to the description of the interface in the object model. However, by supplying an enterprise identifier and interface type in an Ifdesc statement, you specify that the discovered interface's name will be supplied from the description reported by SNMP.

For example:

```
ifname:9;5;0;
```

CatOS interfaces discovered store their name in 'name', not 'desc'.

Icmp

The syntax is as follows:

```
Icmp:[on | off]
```

Icmp controls whether Service Activator should ping a destination first before attempting SNMP discovery. This is used when a range of IP addresses (as specified by an address/mask subnet identification) are being discovered, or during segment-based discovery. All viable IP addresses are contacted and addresses which respond are discovered. This setting is 'on' by default.

Persistant

The syntax is as follows:

```
Persistant:<enterpriseNumber1>;<enterpriseNumber2>;
```

Volatile

The syntax is as follows:

```
Volatile:<enterpriseNumber1>;<enterpriseNumber2>;
```

For the given device (where its sysObjectId matches <enterpriseNumber1>.<enterpriseNumber2>), assume that the ifIndex values assigned to each interface are either preserved (i.e. Persistant) or recomputed on restarts or similar situations, and therefore can change (i.e. Volatile). This statement indicates to Service Activator whether the ifIndex value of an interface can be used to find matches and uniqueness. (When Service Activator re-discovers a device, it needs to map the interfaces coming out of SNMP discovery to the already-discovered interfaces in the object model).

Rename

The syntax is as follows:

```
Rename:<enterpriseNumber>;<ifType>;<string-to-match>;<string-to-
replace>;
```

For this interface number, for this interface type, if you find an interface name that matches this pattern, use the substitute pattern to modify the name that is reported to Service Activator.

For example:

```
Rename:9;134;-atm subif;;
```

In this example, ATM subinterfaces are renamed so that the text "-atm subif" is removed from the name (i.e. is replaced with nothing).

AtmVcInterfaceSource

The syntax is as follows:

```
AtmVcInterfaceSource:<enterpriseNumber>;<aal5VccTable/atmVclTable>
```

There are two different SNMP MIBs that can supply virtual circuit information to Service Activator - the AAL5 table, and the ATM/VCL table. In some cases, SNMP reports data from both, even though only one is accurate. This statement lets you

indicate, for the given enterprise number, which type of reporting of ATM VC Interfaces should be used by Service Activator.

For example:

```
AtmVcInterfaceSource:9;aal5VccTable;
```

Cisco devices will be have the aal5VccTable information reported to Service Activator for Cisco ATM VCs.

Host and Device

The syntax are as follows:

```
Host:<enterpriseNumber1>;<enterpriseNumber2>;
Device:<enterpriseNumber1>;<enterpriseNumber2>;
```

These statements let you indicate which enterprise number pairs indicate hosts, and which indicate devices, in terms of the Service Activator object model.

For example:

```
Device:3224;0;
```

Juniper Netscreen devices will be discovered as devices, not hosts.

Controller

The syntax is as follows:

```
Controller:<enterpriseNumber>;<ifType>;<regexp>;
```

Devices sometimes report hardware controllers as interfaces, even though they are not. This statement lets you indicate that reported items matching the values given in the statement are hardware controllers.

Order of Evaluation

When comparing discovered information against the **Autodiscovery.cfg** file, the Policy Server considers statements in this order.

1. Enterprise ID
2. Device or Host
3. AtmVcInterfaceSource
4. Interfaces: Main, Sub, Lay, Vlan, Vlan Port, or Controller
5. Ifname and Ifdef

6. Rename

Tip: Use an SNMP MIB Browser to determine what information is coming from SNMP about your devices. Then you can determine which exceptions in the returned information you need to create statements for.

Running device discovery

Once you have set up the domain parameters and role assignment rules you can set up the network topology. There are several alternative methods of doing this:

- You can initiate a discovery process from one or more device IP addresses or DNS names. This is the recommended method.
- If you don't know the names and addresses of any devices, you can initiate a discovery from the host system running the policy server.

Network discovery and related tasks, such as fetching capabilities, cannot be carried out part way through the current transaction. These tasks can only be performed immediately after a transaction has been saved or committed and before a new transaction is started.

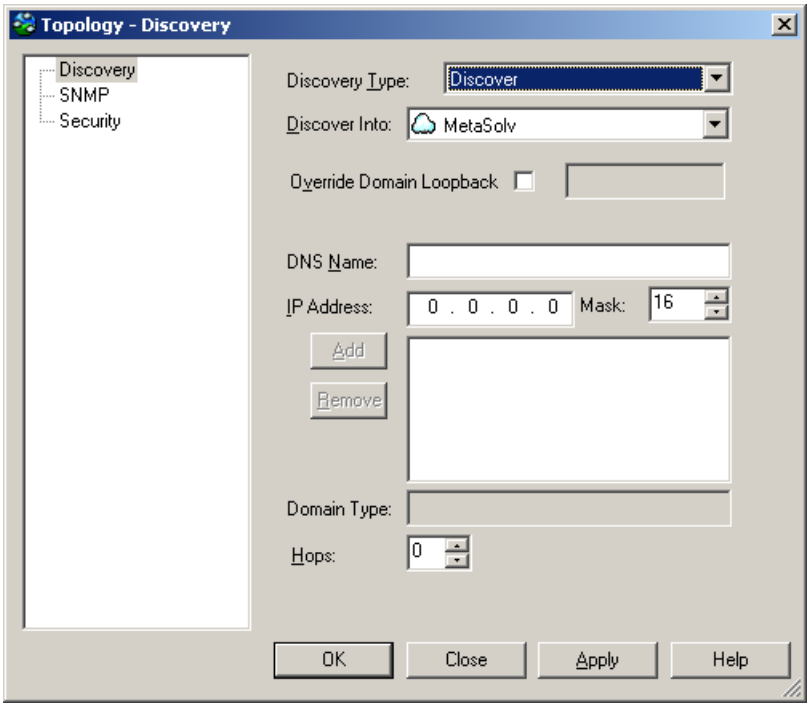
To open the Topology Discovery dialog box

- From the **Discovery** menu, select **Discover**.

  The **Topology Discovery** dialog box opens.

Note that the **Discover** menu option is only available when network objects appear in the **Details** pane (either a map view or a details list).

The **Discover** menu option is not available if there are unsaved changes in the user interface.



Set up parameters on the three property pages of this dialog box:

- **Discovery** parameters specify the type of discovery
- **SNMP** parameters control the way SNMP works
- **Security** parameters specify access settings that apply to all discovered devices

Setting up the discovery parameters

The settings you select on the **Discovery** property page depend on the type of discovery you want to perform.

Discovering one or more specific devices

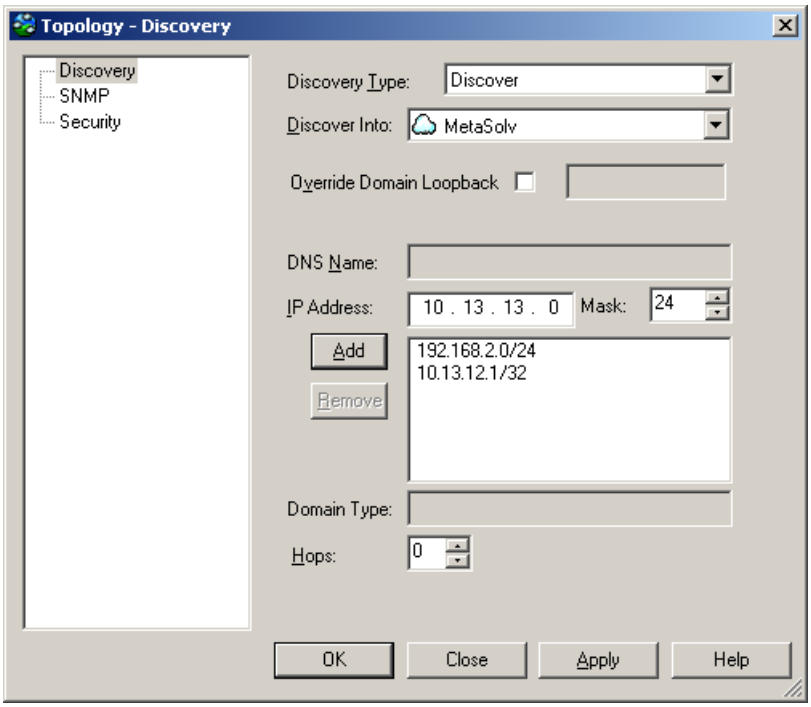
If you know the IP address or DNS name of one or more of the devices in your network you can use the **Discover** option in the **Topology Discovery** dialog box to discover its details and the network nodes to which it is connected.

> A device can only be discovered into a single domain – a device cannot feature in two domains.

1. On the **Topology Discovery** dialog box, select **Discover** from the **Discovery Type** drop-down list.
2. In the **Discover Into** field, select the network to which you want discovered devices to be assigned.
3. If desired, override the default loopback ID value set for the Domain. See *Setting the default loopback ID value for discovery* on page 105 for more.
4. Enter the **DNS Nam**e or the **IP Address** and **Mask** for the device and click **Add** to add it to the list of devices to be discovered.

   The mask defaults to 32, unless an IP address in the format X.X.X.0 is entered in which it is defaults to 24. It can be set to any value from 24 through 32.

   You can discover a number of devices at once, either by listing them individually or by specifying an IP address and mask.



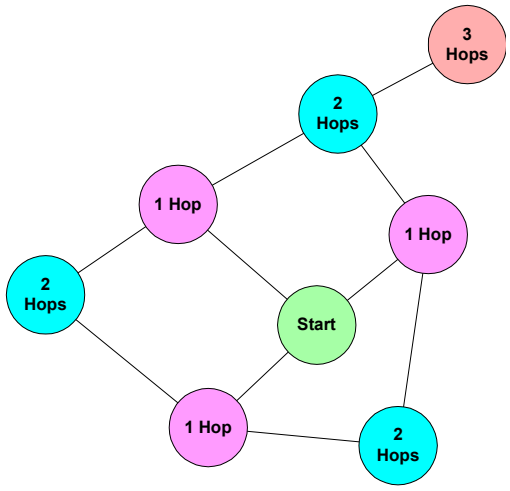
5. In a domain defined as Public or MPLS VPN, you can set the **Hops** field to a value between 1 and 10 to specify that the discovery process is to search for connected devices within a certain number of hops from the original device.

   The default hop number is zero, that is, just the specified devices will be found. Increasing the hop count value broadens the discovery process and results in

more of the network topology being discovered in a single discovery operation. Note that the more hops, the longer the process will take.

> In a domain defined as Private the **Hops** field is ignored, as the discovery process requires a unique IP address or DNS name in order to find a device.
>
> It is not possible to discover Juniper M-series devices using the hop count method. These devices do not make their routing tables available via SNMP, the protocol used by Service Activator to interrogate devices and network segments.

An example hop pattern from a starting device is illustrated in the following diagram:



Discovering the local segment

If you do not know the IP addresses or DNS names of any devices in your network you can use the Local Segment option to start a discovery from the host system running the policy server.

> This method cannot be used in a domain defined as Private. Note also that the policy server workstation must be running an SNMP agent.

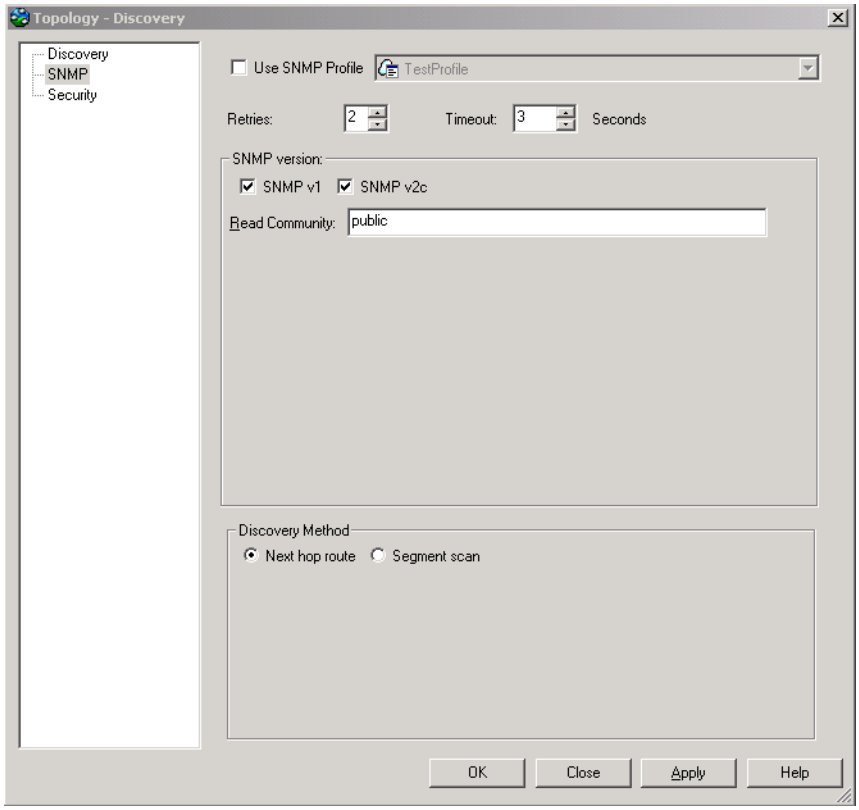
To discover devices using the Local Segment option

1. On the **Topology Discovery** dialog box, select **Local Segment** from the **Discovery Type** drop-down.

2. In the **Discover Into** field, select the network to which you want discovered devices to be assigned.
3. You can set the **Hops** field to a value between 1 and 10 to specify that the discovery process is to search for connected devices the specified number of hops from the original device. Because the segment on which this discovery process is performed includes the policy server, it is likely that the segment is a local subnet outside the enterprise or service provider WAN. Therefore, it is a good idea to set the hop value to at least three so that some nodes in the WAN are discovered.

Defining the SNMP options for discovery

The parameters controlling the way in which the discovery process operates are set up on the **SNMP** property page of the **Topology Discovery** dialog box, illustrated following.



On this property page, you can select SNMP v1 and/or SNMP v2c to be used for device discovery. If you also want to use SNMP v3 for device discovery, you must select an SNMP Profile on this property page. More details follow:

- If you are using an **SNMP Profile** to set SNMP for discovery, select the checkbox and select a profile in the drop-down menu. An SNMP Profile allows you to use **SNMP v3** for discovery, as well as **SNMP v1** and **SNMP v2C**. Details for using SNMP profiles are provided in the Online Help.

  Service Activator follows an SNMP "fallback" sequence until successful communication is established with a device. For example, if you have selected three SNMP versions for discovery, Service Activator starts the communication using SNMP v3, then falls back to SNMP v2c, and then to SNMP v1 if required, to establish the communication.

- The default number of **Retries** is 2 and the default **Timeout** period is 3 seconds. You may need to increase these values if your network is slow, or if you set a high hop count for the discovery process and select the **Next hop route** discovery method.

- By default both **SNMP v1** and **SNMP v2c** checkboxes are selected. This means that the discovery process will try SNMP v2c first and, if this fails, will try SNMP v1. If you specify SNMP v2c only, there is a possibility that you will not discover some devices in your network. In particular, Cisco devices that are running an IOS earlier than version 12.0 do not support SNMP v2c. Selecting both versions of SNMP for the discovery process ensures that current and older router models can be discovered. If you are sure that only one version is in use you can deselect one version in order to speed up the discovery.

- The default **Read Community** setting is **public**; you will need to change this if an alternative SNMP community has been set up on the devices.

- The **Discovery Method** setting is only applied if the **Hops** value on the **Discovery** property page is greater than zero. The default setting is **Next hop route**, which queries the device's routing tables – this is normally faster, but if you have Internet routers with very large routing tables, it can be slow. If you select **Segment** scan the discovery process scans each newly-discovered segment checking each address in turn. This process is comprehensive but can be slow.

When a device has been discovered, the specific SNMP settings used are saved as device-specific parameters. For example, if SNMP v1 was used, only the SNMP v1 checkbox is selected on the device-specific SNMP settings. The device-specific settings always override the global settings when devices are rediscovered.

If SNMP compression is configured on a device, discovery of that device will not work. Ensure SNMP compression is not configured on the device. Note that Juniper M-series devices have SNMP compression turned on by default.

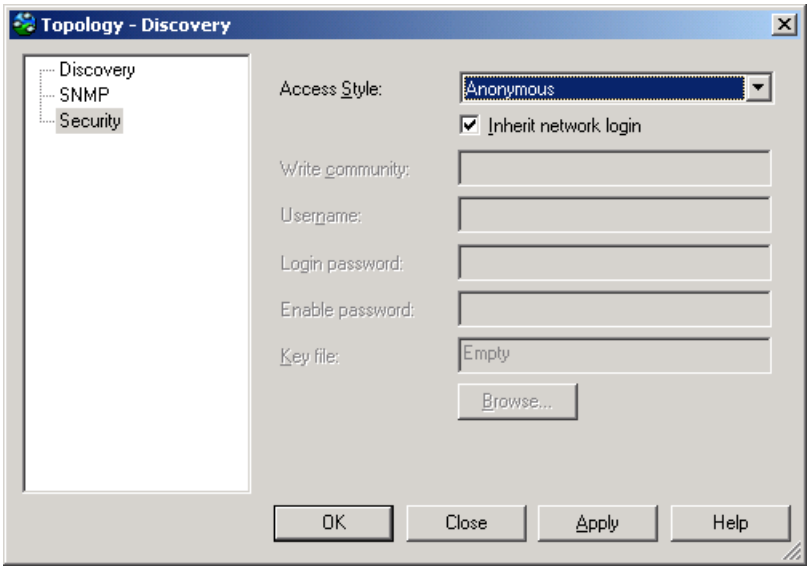
Defining default security options for discovery

You are recommended to set up default security parameters before running the device discovery. This will ensure:

- That security settings are automatically applied to all subsequently discovered devices.
- That QoS, security and VPN capabilities are retrieved from all devices (write access to devices is required in order to ascertain the capabilities).

If you do not set these values, you will need to set security settings for each device individually. In addition, Service Activator will be unable to obtain device capabilities.

The access parameters are set on the **Security** property page of the **Topology Discovery** dialog box:



Access Styles

The **Access Style** is the mode in which Service Activator will access the discovered devices for configuration. Access Styles include:

- Named User
- Anonymous
- TACACS+
- SNMP v1
- SNMP v2c
- SSH with password authentication
- SSH with keyed authentication
- Password Only
- None

The selection determines which additional fields are displayed on the screen. Access methods are device-specific; at present, Service Activator configures devices as follows:

- Cisco devices are configured via the command-line interface using an anonymous user login with password authentication, via SSH with password authentication, or via a TACACS+ server if configured.
- Juniper M-series devices are configured via the command-line interface using Named User, via a TACACS+ server if configured, or via SSH with password authentication. (SSH with keyed authentication is not supported.)
- Juniper E-series devices are configured via the command-line interface using an anonymous login with password authentication.
- Alcatel devices are configured via the command-line interface using an anonymous login with password authentication, via a Named User login, or via SSH with password authentication. (SSH with keyed authentication is not supported.)

The access styles are as follows:

- For login as a **Named User**, you need to specify the SNMP **Write Community** and the command-line interface login details (**Username** and device **Login** and **Enable** passwords). Note that this method is not supported at present.
- For login as an **Anonymous** user, you just need to set up the **Login** and **Enable** passwords.
- **TACACS+** indicates that a TACACS+ server is used to control access. The appropriate passwords must be set up at network level (see *Setting specific security settings* on page 167).

- For **SNMP v1** or **v2c** you need to specify the SNMP **Write Community** for write access to the device. This must match the community set up on the device(s).

> SNMP v1 and v2c are not supported as access styles on any currently-supported devices.

- For **SSH with password authentication**, specify the **Username** and **Login Password**.
- For **SSH with keyed authentication**, specify the **Username** and select a **Key file**. Specify the private key file – the public key file must be present on the device.

Note: For details on creating SSH keys for devices supporting this access style, refer to the appropriate Device Driver Support guide.

For the purposes of discovery, you can leave the access style as **None**. In this case, Service Activator applies a default access style according to the device type. The access styles are defined in the **AutoDiscovery.cfg** configuration file, which is installed in the **Config** subdirectory of the host system running the policy server. However, login and password information cannot be set and therefore must be set manually for each device.

> Note that the specified access settings apply as defaults. You can amend them for specific discovered devices, in which case the device-specific settings will override the global settings when devices are rediscovered.

Cartridge support for Access Styles

The table below displays the Access Style supported for each cartridge, using the Network Processor.

Access Style	Huawei	Cisco	Juniper	Foundry
Named User			x	
Anonymous	x	x		
TACACS+	x	x		x
SNMP v1				
SNMP v2c				

Access Style	Huawei	Cisco	Juniper	Foundry
SSH with password authentication	x	x	x	x
SSH with keyed authentication				
Password Only		x		
None				

The discovery process

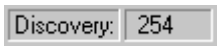
As soon as the changes are committed to the database, the network discovery starts. Service Activator initially creates a list of nodes to be discovered. When discovering a segment, each additional node found is added to the end of the list. If a timeout or failure occurs, the process moves on to the next node on the list.

Discovered devices are linked to the network object that was selected when the discovery process was started.

As part of the discovery, devices and their interfaces are assigned roles according to the role assignment rules (see *Assigning a role to a policy target* on page 117) and devices are allocated to a proxy agent (see *Setting up proxy agent assignment* on page 106). Where possible, device and interface capabilities are ascertained.

Monitoring the discovery process

The length of time that the discovery process takes varies depending on the number of devices and segments found, the type of discovery and the speed of the network. If you have set a high hop count, a large number of devices may be found. You can monitor the progress of the discovery from the status bar at the bottom of the screen.



The number indicates the number of nodes currently on the discovery list. This can initially increase as new devices and segments are found and interrogated, and will return to Idle when the discovery is complete.

When discovery is complete, Service Activator interrogates each device for its interface capabilities. The status bar indicates that this phase is in progress.

If a device's capabilities could not be obtained, or Service Activator obtained capabilities for one but not all of its interfaces, an error is generated.

The topology section of the object model is updated with details of the discovered devices, segments and hosts.

Automatic mapping

By default, you create a network map by arranging devices manually. However, if you have selected automatic layout, the discovered network objects are added to the network map automatically while the discovery process runs.

In automatic mapping, you create a layout filter that defines which network objects are displayed on the map. Up to 200 network objects can be mapped automatically with minimal delay in display time. For greater numbers of network objects, there may be some delay in mapping and displaying the objects.

For information on automatic and manual mapping options, see *How objects are represented* on page 186.

Stopping the discovery process

If necessary, you can stop the discovery process.

To stop discovery of the current object only

- Select **Stop** from the **Discovery** menu.

  The discovery process continues with the next node on the list, if there is one.

To terminate the discovery process completely

- Select **Stop all** from the **Discovery** menu or select **Stop All** from the **Discovery Type** drop-down on the Topology Discovery dialog.

  All devices and segments discovered prior to this command are retained in the object model, but discovery queries in progress at the time are stopped and the rest of the items on the discovery node list ignored.

After discovery is complete

This section explains the steps you need to take after discovery is complete to ensure your network is set up satisfactorily. You need to check the following:

- Ensure devices are assigned policy roles, and apply roles manually if necessary.
- Ensure devices are assigned to proxy agents.

- Set any specific IP addresses used to manage devices.
- Define how the system is to deal with manually-applied configuration.
- Set any specific security settings required for Service Activator to configure devices.
- Set devices to Managed to ensure that they will be configured by Service Activator.

You should also check any errors listed in the current faults pane and correct any problems. Occasionally, a new device type may be reported.

Ensuring devices are assigned roles

If you have applied role assignment rules, you should check the device and interface roles and set any that have not been assigned by the rules. On a small network or test installation, you may decide to set all roles manually.

For information on applying roles manually, see *Assigning roles to a policy target manually* on page 134.

> Note that if you change the role of a device or interface manually, by default, the role will be reset where a role assignment rule is applicable when you next run a device discovery. To avoid this, you can switch off application of role assignment rules (see *Specifying when role assignment rules are applied* on page 131) or amend the role assignment rule.

Ensuring devices are assigned to proxy agents

All devices that are to be managed by Service Activator must be assigned to a proxy agent. This will normally be done automatically during device discovery, but if devices are not assigned to the correct proxy agents you need to assign them manually.

To check which devices are assigned to a proxy agent

1. On the **System** tab, open the **System Hosts** folder.
2. Open the component manager and the relevant proxy agent. Service Activator lists the devices that are assigned to the proxy agent in the hierarchy pane.

To check whether a device is assigned to a proxy agent

1. Display the ancestry pane by selecting **Ancestry Tree** from the **View** menu.
2. On the **Topology** tab, select the relevant device.

If the device is assigned to a proxy agent, Service Activator lists the proxy agent in the ancestry pane.

To assign a device to a proxy agent manually

1. Display the devices in the **Details** pane by double-clicking on the relevant network cloud (display devices in Map view or Details view), select the **System** tab and display the proxy agent in the **System Hosts** folder.
2. Drag the appropriate device from the **Details** pane and drop it on to the proxy agent that will control it.

   Alternatively, select the device and select **Copy** from the **Edit** menu (or click the **Copy** button). Then select the proxy agent on the **System** tab and select **Paste Link** from the **Edit** menu.

To unassign a device from a proxy agent

1. Select the **System** tab and display the proxy agent in the **System Hosts** folder.
2. Select the device under its parent proxy agent and click on the Unlink button on the toolbar or select **Unlink** from the pop-up menu.

Setting specific IP addresses for device management

Service Activator requires a unique IP address when communicating with and managing devices. This is frequently a loopback address, but an alternative can be specified.

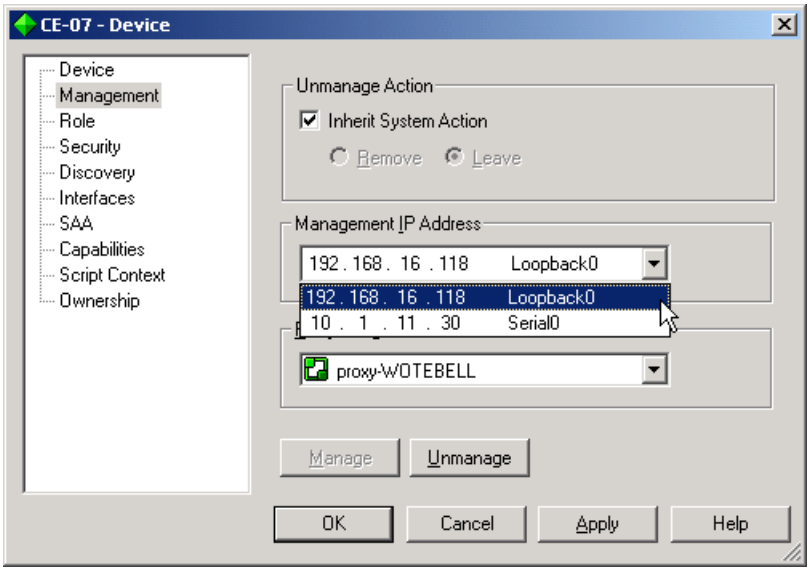
You can set up global standards for IP address usage on the **Options** dialog box (see *Defining the way in which IP addresses are used* on page 144). However, if necessary you can override these defaults with device-specific settings.

> Note that for Juniper E-series devices, the address used to manage the device must not be changed from that used for discovery. This is because the IP address used must be the one valid for the default virtual router. If an alternative address is used, the device is discovered but the driver cannot communicate with it.

To set a specific IP address for device management

1. Right-click on the device and choose **Properties** from the pop-up menu.

   The **Device** dialog box appears.
2. Select the **Management** property page.
3. Select the IP address to be used for managing the device by choosing from one of the known addresses listed in the **Management IP Address** drop-down, or by entering the management IP address in the edit box.



4. Click **OK**.

Setting manual configuration detection

You can specify the action that Service Activator takes if it discovers that a device has been configured manually. A domain-level setting specifies the default behavior (see *Setting up domains* on page 104), but you can override this for particular devices.

To set up manual configuration behavior for a device

1. Right-click on the device and choose **Properties** from the pop-up menu.
2. Select the required behavior from the **Manual config** drop down:
   - **Inherit from domain**: the setting for the device is inherited from the domain-level setting. See *Setting up domains* on page 104.
   - **Delete**: if manual configuration is detected, it is deleted and the device reset to the original configuration.
   - **Warn and delete**: if manual configuration is detected, a warning message is output and the device is reset to the original Service Activator configuration.

- **Fail and don't delete**: if manual configuration is detected, a critical fault is raised. The device status is set to Intervention Required and no configuration is applied.

> For VPN-related configuration, Service Activator always preserves manually pre-configured VRF tables, even if a delete option is selected in the **Manual config** field. However, BGP configuration is deleted unless the **Fail and don't delete** option is selected.

Setting specific security settings

Before you can configure the network, you need to set up appropriate security and authentication settings to allow Service Activator to access the managed devices.

If you set up security values before you run a device discovery, (see *Defining default security options for discovery* on page 159) these settings apply to all discovered devices within the domain. However, you can overwrite these with device-specific settings. If you haven't set default security settings you will need to set suitable values for all devices to be managed.

The security settings you need to define depend partly on:

- The requirements of the device driver. Service Activator writes to all currently-supported vendor devices using Telnet and a command-line interface.
- The set-up of your network – for example, whether you are logging on as a named user or authenticating via a TACACS+ security server.

If you are using the same authentication details for all devices, you can set the security parameters at network level. If you have created subsidiary networks within the domain, you can specify that these networks inherit access parameters from the parent network.

To set security parameters for a network

1. Select the network object, either on the network map or from the **Topology** tab.
2. Select **Properties** from the network's pop-up menu and select the **Security** property page.
3. For a sub-network, select the **Inherit from parent network** checkbox if the security settings are to be inherited from the network to which the sub-network belongs.
4. Enter details including **Write Community, Username, Login Password, Enable Password**.
5. **Key file**: for SSH with keyed authentication, specify the private key file.

A public key file must be present on each device in the network.

> Note that if the **Inherit from parent network** checkbox is selected, these field values are inherited from the parent network object and are therefore read only.

6. Click **OK** to close the dialog box.

To check or set the security parameters for a device

1. Select the device, either on the network map or from the **Topology** tab.
2. Select **Properties** from the device's pop-up menu and select the **Security** property page.
3. To set individual security parameters for this device, deselect the **Inherit network login** checkbox.
4. Select the appropriate **Access Style** and set appropriate parameters.
5. Click **OK** to close the dialog box.

Managing devices

Managing and unmanaging a device controls whether or not Service Activator configures the device.

A device's status is automatically set to Unmanaged when it is first discovered. Before a device can be configured, its status must be set to Managed.

> Note that a device must also be assigned to a proxy agent before it can be configured by Service Activator, even if its status is set to Managed. For more information, see *Assigning devices to proxy agents* on page 138.

You can stop a device from being managed by setting its status to Unmanaged. An Unmanaged device's configuration is not updated when you commit a transaction. You should unmanage devices that are no longer within the policy domain. You may also need to unmanage a device temporarily if attempts to configure it with Service Activator result in a fault that cannot be resolved.

You can manage or unmanage devices on a domain-wide basis or device-by-device. When you manage or unmanage a device, its color changes to reflect its new status. A managed device is represented by a green icon, an unmanaged device is represented by a blue icon.

To manage or unmanage all devices within a domain

- Select the network object that represents the entire domain and select **Manage All Devices** or **Unmanage All Devices** from the object's pop-up menu.

To manage or unmanage an individual device

1. Select **Properties** from the device's pop-up menu.
2. On the **Management** property page, click the **Manage** or **Unmanage** button.

Retaining or removing Service Activator configuration

By default, when you unmanage a device any configuration that Service Activator writes to the device remains configured. However, you can specify that the configuration is removed when a device is unmanaged. You can do this on a system-wide or device-by-device basis.

To specify the unmanage action globally

1. From the **Tools** menu, select **Options**. The **Options** dialog box opens.
2. Under **Unmanaged Action**, select:
   - **Remove** to remove the Service Activator configuration when devices are unmanaged.
   - **Leave** to leave the Service Activator configuration on devices when they are unmanaged.

To specify the unmanage action for a device

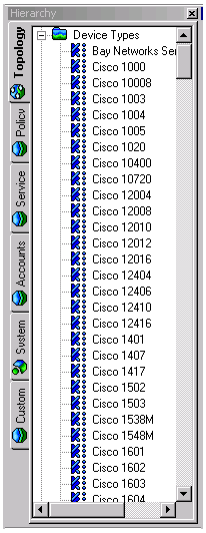
1. From the device's pop-up menu, select **Properties**. The **Device** dialog box opens.
2. Select the **Management** property page and select one of the following:
   - **Inherit System Action** to apply the global unmanage action to this device (this is the default setting).
   - **Remove** to remove the Service Activator configuration from this device when it is unmanaged.
   - **Leave** to leave the Service Activator configuration on the device when it is unmanaged

   Selecting **Remove** or **Leave** overrides any global setting.

Discovering a new device type

A device type defines the appropriate device driver to use to control the device. Every router in the network is classified according to type. The association between a router type and the device driver that will be used to control it is defined in the

DeviceTypes.cfg configuration file. The information from this file populates the **Device Types** folder on the **Topology** tab.



Some of the routers in the network may fall outside the classification types recognized by Service Activator. Where this is the case, Service Activator classifies the router according to its SNMP SysObjectId parameter during the discovery process.

For details of the devices currently supported by Service Activator, see the relevant device driver guide.

When the discovery process is run, Service Activator checks the device type of each discovered device against the **DeviceTypes.cfg** file. If it discovers a new device type, Service Activator automatically adds its details to the database and displays notice 1609 - *New device type has been created* in the current faults pane. This indicates that you need to check and amend the type's details and set up the appropriate device driver in the **Device Type** dialog box

Note that you may discover some devices that cannot be managed by Service Activator. The discovery process discovers most SNMP-enabled devices and hosts that support MIB-II. If you discover a device type that is not supported by any device driver, you can configure the new device type's details but attempts to configure the device are likely to result in error messages.

To set up a new device type

1. Double-click on the relevant message 1609 in the current faults pane.

   The **Device Type** dialog box opens.

2. In the **Device driver** field, enter the name of the device driver that will be used to manage this device. This must match exactly the name specified in the appropriate **Driver** properties dialog box. You can access a device driver's properties dialog box via the **System** tab and the **System Hosts** folder.

3. Amend the **Vendor** and **Product** fields if you wish, to provide more meaningful information. Note that these fields are for information only.

4. The **Model, s/w Vn** field identifies the device type, and appears on the **Device** property page, so it can be useful to provide more meaningful information. To edit this field, select the **Edit fields** checkbox.

> Do not edit the SysObject ID and Configure level fields.

5. Click OK. The data is saved when the transaction is committed

> Note that if you set up a new device type in this way, it is at your own risk. If you require this device to be certified as tested by Oracle Communications, you must contact Oracle Communications Technical Support.

Reducing the number of listed device types

The **Device Types** folder lists a large number of device types. If you delete a device type from the folder, it is recreated the next time the policy server is started because the **DeviceTypes.cfg** file is reloaded at this point. If you want to reduce the number of listed device types, you can edit the **DeviceTypes.cfg** file using a text editor.

> We strongly recommend that you create a backup copy of the file before editing.

Managing Configuration Thresholding

Use configuration thresholding to restrict the number of commands provisioned on devices in a single transaction.

About Configuration Thresholding

Configuration Thresholding provides a safety mechanism that blocks any device configuration action by Service Activator that exceeds certain user-specified parameters. The threshold is configured by means of two values - a regular expression (regex) against which to match commands, and the threshold value itself.

The regular expression uses a syntax based on the Boost regex library. See http://www.boost.org/libs/regex/doc for details.

Setting the Configuration Threshold

Note: This feature is turned off by default. In other words, all Network and Device settings for the threshold are set to **No Limit**. The feature starts working when you change this setting for a Network or Device.

The regular expression against which to match commands is not accessible through the GUI. The threshold value can be set at the **Network** or the **Device** level.

Configuration Threshold settings applied at the network level (on the **Network Properties** dialog box, **Device Management** property page) are automatically inherited by the other networks and devices inside that network.

These settings can be overridden for individual devices (or networks under a parent network for that matter.) Use the **Device Properties** dialog box, **Management** property page to set values for a device. In most cases, you will configure devices to inherit values from their parent network. Only the most sensitive devices (such as PE devices) will require specific settings.

If you do make changes, you must wait for propagation to occur before they are enforced.

See the *Setting up Configuration Thresholding* on page 173 for step-by-step instructions.

What happens when the threshold is exceeded

If a transaction causes more configuration commands than specified by the threshold value, which match the regular expression, the telnet session to that device is aborted and everything done in that telnet session is undone. No commands are sent to the device, and the device is placed into Intervention Required mode.

As well, a Critical fault is raised against the router (visible in both the GUI and the OIM). This fault provides the threshold value as well as the current match count in the fault details. The aborted commands are logged to the Audit Trail file for that day with the prefix "max-transaction-exceeded".

The threshold parameters are applied only to commands generated by the Service Activator device driver. Specifically, it is not applied to commands generated by CDK scripts and modules.

Recovering from an exceeded threshold

Once a device enters the intervention required state, no further configuration is sent to the device. To recover, you must examine the aborted commands in the audit trail. If these commands are legitimate (i.e. it was actually your intent to issue such commands), then you must increase the threshold value for that router, delete the fault and commit.

If the commands are incorrect due to operator error, you must undo the actions that led to the behaviour (for example, taking a role off a device), delete the fault and commit.

If the commands are incorrect due to an Service Activator internal problem or if the you are not able to make a determination, contact Global Customer Care.

It is highly recommended that the device recovery take place during a maintenance window.

Notes

The command pattern to match against (i.e. the regex) can be viewed, but not modified, in the GUI. It can only be modified through the OIM.

Some features with deficient implementations unnecessarily remove and re-install configuration on devices. Such removal commands may be counted towards the maximum transaction size, if they match the pattern.

If the threshold value is modified and the role removed from a device in the same transaction, the new threshold does not take effect until a new role is added to the device.

If the threshold value is modified and the device unmanaged (with the unmanage action set to remove) in the same transaction, the new threshold does not take effect until the device is re-managed.

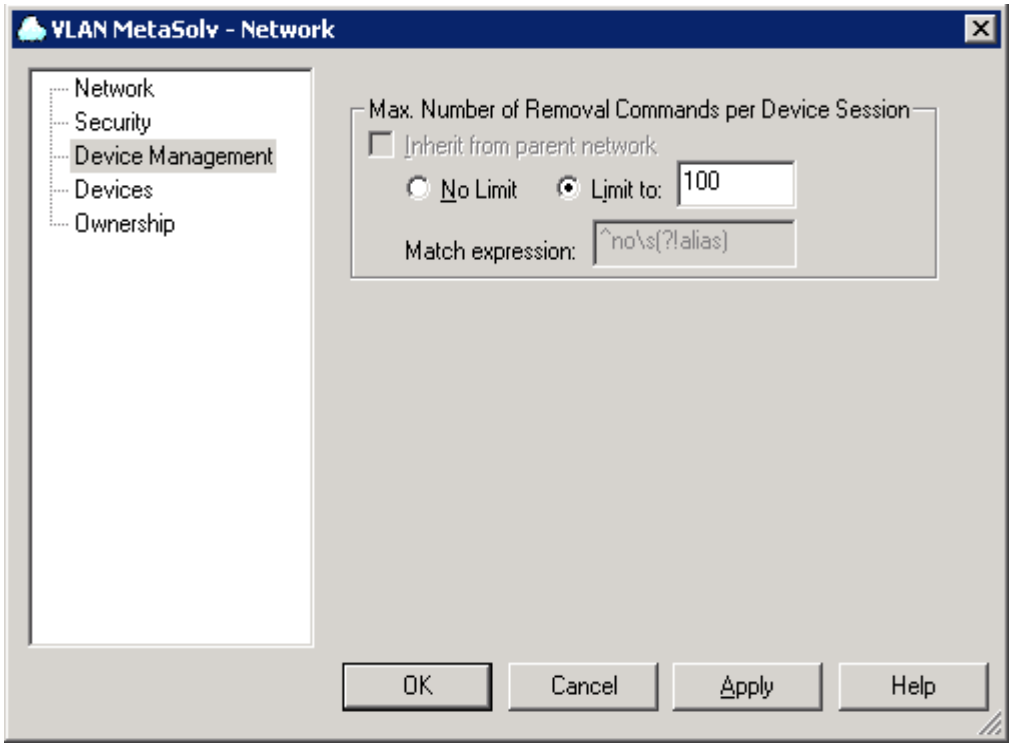
Setting up Configuration Thresholding

This topic contains a sample procedure for setting up Configuration Thresholding.

Tip: It is recommended that you group devices which are at the same functional category. For example, PE routers which behave the same way can be grouped within sub-networks.

To set Configuration Thresholding values:

1. Right-click on a network, select **Properties**, and set the maximum number of removal commands per device session on the **Network Properties** dialog box, **Device Management** property page. By default, devices and sub-networks inherit the configuration thresholding settings from their parent network. See above for guidelines on selecting the threshold level.



2. Set appropriate values for your network configuration including **Inherit from Parent Network, No Limit, Limit To,** and **Match Expression**.
3. If there are other sub-networks, repeat the previous step to set the threshold limit for each of them.
4. Customize the settings for any devices as required by right-clicking on the, selecting Properties, and accessing the Device properties - Management property page . Uncheck the Inherit from network checkbox, and provide device-level settings.

To set the Configuration Thresholding regex match expression:

Note: The commands included in the threshold count can be set only using Integration Manager or using OIM python scripts.

To use the Integration Manager, launch its CLI and log in.

If you know the name of the network to set up, skip this step. To find all the networks and their IDs, use the command:

```
find / network:"*"
```

To set the regular expression to match configuration statements when counting for Configuration Thresholding for a particular network:

```
find / network:"<network-name>"
getattributes <network id>
modify <network id> MatchesPatternTransactionSize="<regex>"
commit
```

A sample regex expression is: `no\\s(?!alias|auto-summary|synchronization)`

This would count any configuration statements starting with the word "no" except for those that start with "no alias", "no auto-summary" or "no synchronization".

Note: The threshold value can be set using `MaxTransactionSize` or set using the GUI.

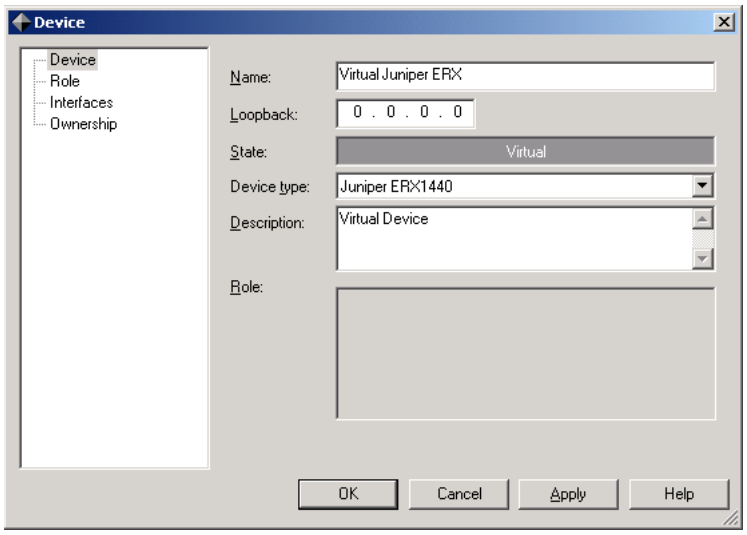
Creating virtual devices and interfaces

As well as discovering existing devices, you can manually create device and interface objects to represent those that Service Activator has not discovered. This allows devices not managed by Service Activator, such as those in the core network, to be modeled in the system in the form of virtual devices.

> Virtual devices and their interfaces cannot be discovered or managed.

To create a virtual device

1. Right-click the appropriate network object and select **Add Device** from the pop-up menu. The **Device** dialog box opens:



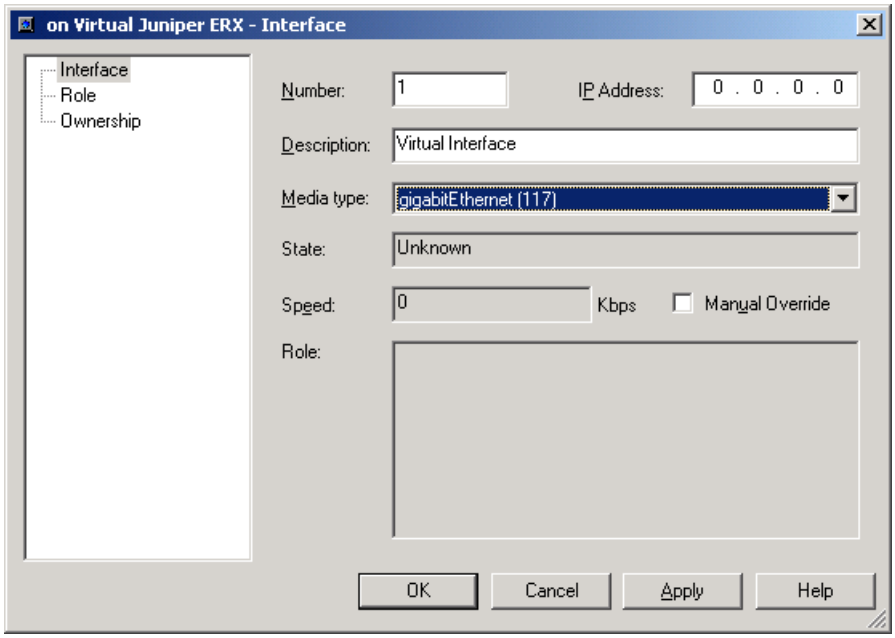
Note that **Management**, **Security**, **Script Context**, **Capabilities** and **Discovery** property pages are not present.

2. Enter identifying details of the device. You can set the **Name**, **IP Address**, **Device Type** and **Description** fields; other information is read-only.
3. Click **Apply** or **OK** to create the device.

Devices created in this way always have a status of Virtual, and the device icon appears in dark gray.

To create an interface on a virtual device

1. Right-click the appropriate virtual device object and select **Add Interface** from the pop-up menu. The **Interface** dialog box opens:



Note that **Capabilities**, **Script Context** and **Details** property pages are not displayed.

2. Enter identifying details of the interface. You can set the **Number**, **IP Address**, **Description** and **Media Type** fields; other information is read-only.
3. Click **Apply** or **OK** to create the interface.

Interfaces created in this way always have a status of Unknown, and the interface icon appears in dark gray.

> You can set roles for virtual devices and their interfaces, but this is entirely for identification purposes as they are not managed or configured by Service Activator.

Maintaining the network topology

You need to ensure that the network you are managing is accurately represented within Service Activator. For example, if new routers are added or existing ones reconfigured, you must ensure that Service Activator is updated accordingly. You can rediscover the entire domain, update individual devices and if required, set up the discovery process to run automatically on a regular basis.

Refreshing the entire domain

You can rerun the discovery process at any time if any changes are made to the network configuration, or just to ensure that the information you have is accurate.

> The rediscovery of an entire domain may take some time. You can monitor its progress from the status bar at the bottom of the screen.

To update the topology of a domain

1. Within the appropriate domain, select **Discover** from the **Discovery** menu. The **Topology** dialog box opens.
2. On the **Discovery** property page, select **Domain Refresh** from the **Discovery Type** drop-down list.
3. Set the **Hops** field if required.

> Note that the Hops setting only applies to new devices and segments found.

4. Click **OK**.

> You cannot change the SNMP or Security settings on a domain refresh. The settings that were saved in the database from the previous discovery for each device or host are used.

Rediscovering individual devices

You can re-run the discovery process at any time for a selected device, segment or host.

To re-run device discovery for an existing device, segment or host

1. Select the device, segment or host (either on the map or anywhere within a domain management window).

2. Select **Refresh** from the **Discovery** menu or select **Discover** from the pop-up menu on the object. Service Activator uses SNMP to interrogate the device to discover details of its interfaces and any further devices or segments connected to it.
3. If new nodes are discovered, they will appear in the palette. You can drag them on to the map and run the discovery process on each new node in turn, if required.

Deleting missing interfaces

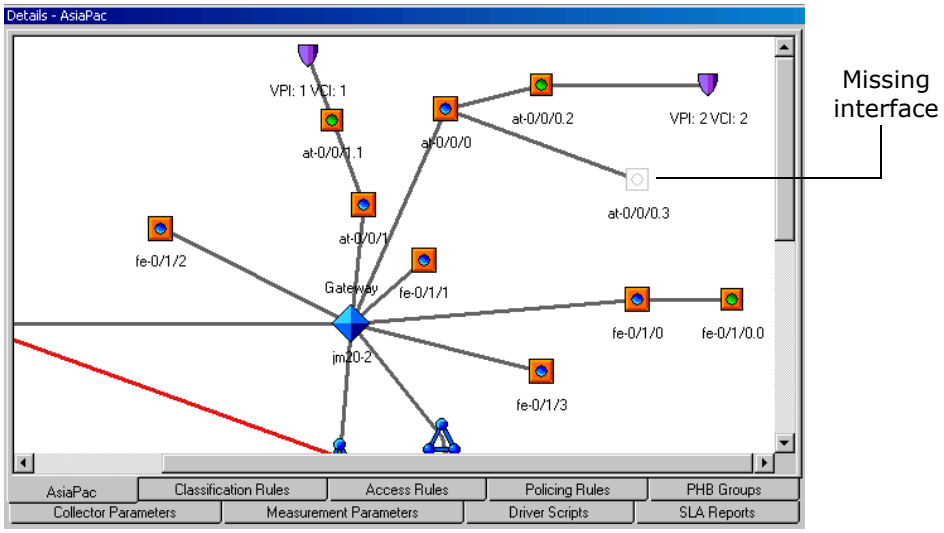
Service Activator's discovery process uses SNMP to identify devices and interfaces and stores a representation of the discovered network topology in its internal object model. Interfaces are discovered by their ifIndex values, but Service Activator also records the value of each interface's ifName and/or ifDescription variable.

On a rediscovery of the network, previously discovered interfaces are primarily matched against interfaces encountered in the current discovery using the interface name (based on the ifName/ifDescription). However, if no match can be made, Service Activator compares the interfaces' ifIndex values and, where values coincide, matches the interfaces.

If Service Activator cannot match details held for an interface in its object model against any ifName/ifDescription or ifIndex values encountered during a new discovery, it regards the interface as 'missing'. A missing interface is assigned a status of Not Found and its ifIndex value is set to zero in Service Activator's object model.

Note that an interface is only classifed as missing if it has been assigned a role or is linked to a site in a VPN or a layer 2 site in a TLS. If neither of these conditions apply to a missing interface, the interface is automatically deleted.

An interface whose status is **Not Found** is represented in the following way in the user interface.



Possible reasons for an interface to be assigned the Not Found status are:

- The interface has been deleted from the device, either deliberately or accidentally.
- The interface has been relabelled on the device – that is, both the interface's ifName/ifDescription and ifIndex details have changed. This sometimes occurs, for example, when a device's operating system is upgraded.

> You must investigate the cause of an interface being reported as Not Found before you consider deleting the interface.

If the interface has been deleted

The interface may have been deleted deliberately or by accident:

- If the interface has been deleted deliberately, delete the Not Found interface in Service Activator.
- If the interface has been deleted accidentally, re-create the interface on the device (you can Telnet to the device from the user interface) before rediscovering the device.

When a device that has 'not found' interfaces is unmanaged, the GUI displays a popup warning about possible loss of configuration.

Neither the GUI nor the OSS Integration Manager interface allows you to manage a device with 'not found' interfaces. These have to either be discovered to be found or must be deleted individually.

Interfaces which are 'not found' can be restored to 'found' state by using the preserve missing interfaces command:

To preserve all missing interfaces on a device

- Select **Preserve Missing Interface(s)** from the relevant device's pop-up menu.

  Service Activator restores the interfaces to its representation of the network.

To delete missing sub-interfaces on an interface

- Select **Preserve Missing Interface(s)** from the relevant interface's pop-up menu.

  Service Activator restores the missing sub-interfaces to its representation of the network.

To preserve a specific missing interface or sub-interface

- Select the interface or sub-interface and select **Delete** from the object's pop-up menu.

Interfaces can also be preserved from the OSS Integration Manager interface. Issue the command:

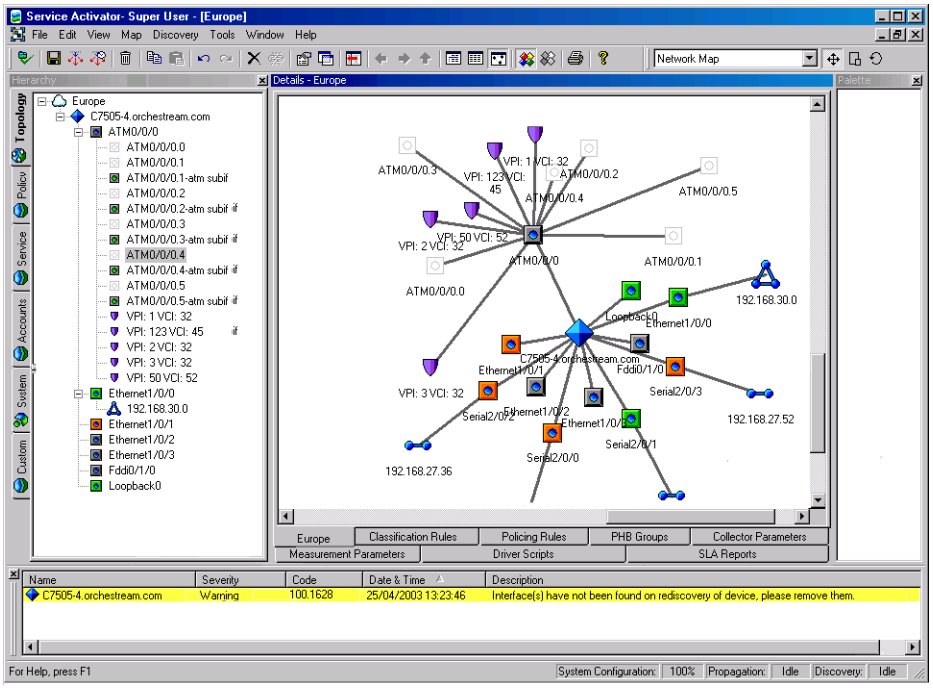
```
preservemissinginterfaces <object_id>
```

where <object_id> is the identifier for the device, interface, or sub-interface.

If an Interface or Sub-Interface is deleted when it is in Offline maintenance mode (either because the parent device, parent interface, sub-interface, or network processor is in Offline maintenance mode) the configuration on that device will not be removed from the device, but IPSA will lose all knowledge of that configuration. In addition, if the deleted interface or sub-interface was created by IPSA, it also will not be removed from the device, but IPSA will lose ownership of that interface/sub-interface. The interface/sub-interface will reappear in the GUI during the subsequent re-discovery of that device.

If the ifName/ifDescription and the ifIndex have changed

If a number of Not Found interfaces are displayed for a device, and these interfaces appear to duplicate found interfaces, it may indicate that the device's operating system has been upgraded. This means that both the ifName/ifDescription and the ifIndex have changed for all interfaces on the device.



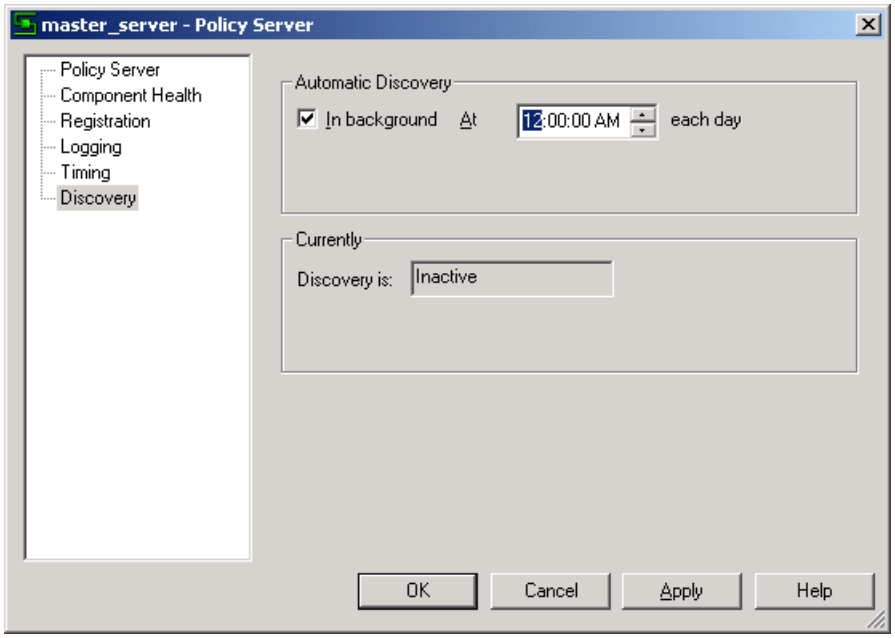
Contact Oracle Global Customer Care for information on the procedure for handling a change to an interface's ifName/ifDescription and the ifIndex.

Setting up discovery to run automatically

As a full rediscovery of a large domain can take some time, you can set up Service Activator so that the device discovery process is run automatically every day, for example as an overnight process.

To set up discovery to run automatically

1. Select the **System** tab on a global or domain level window.
2. Right-click the policy server object (master_server) and select **Properties** from the pop-up menu.
3. Select the **Discovery** property page on the **Policy Server** dialog box.



4. Select the **In background** checkbox and enter the time at which you want the discovery process to start each day, and click **OK**.

The discovery process is controlled by the SNMP parameters on the **Discovery** property page for each node. You should ensure these are set up correctly.

After the network discovery process has run each day, you should check the network topology and make any necessary corrections. For example:

- If a device is not found, it is not deleted, but its status is changed to Not Found. You will need to check it.
- If you are using manual mapping, any newly-discovered devices and segments are listed in the palette. You can drag them on to the appropriate map and ensure they are set up correctly.
- Messages relating to new or incorrectly-configured devices may be reported in the current faults pane. You should fix any problems before proceeding.

Chapter 8

Representing and Mapping Objects

The user interface provides a graphical and hierarchical representation of the network objects included within your policy domain. You can create multiple maps that allow you to customize how information is displayed.

This chapter:

- Describes how objects are represented in the user interface, including VLANs and virtual routers
- Explains how to create a topology map manually or automatically
- Provides guidelines for working with maps, including recalculating map layout and configuring the palette
- Explains how to create subsidiary maps and alternative map views
- Describes how to display background images and change a map's scale

How objects are represented

Once a network has been discovered, details of devices, interfaces, segments and so on appear in the **Topology** tab of the hierarchy tree pane and in the **Details** pane. For details of how this information is displayed, see *Service Activator's windows* on page 13.

Alternative device views

You can choose two alternative views of devices and interfaces on the user interface using toolbar buttons:

Status Context View button (multi-colored) – all devices and interfaces are shown color-coded according to their status.

Policy Context View button (gray) – all devices are shown in shades of gray according to their policy role.

Status Context view

If you select the Status Context button, different colored icons are used to represent the different status values that a device or interface may have.

Unreachable (orange) – Service Activator cannot communicate with the device.

Unmanaged (blue) – the device has been discovered but is not managed by Service Activator.

Not Found (light gray) – the discovery process failed to find the device.

Managed (green) – the device is being managed by Service Activator.

Virtual (dark gray) – the device has been manually created within Service Activator and cannot be managed.

Intervention Required (red) – Service Activator has failed to resynchronize correctly with the device after a failure, or the device's physical configuration has changed.

Policy Context view

If you click the **Status Context** button, different shadings are used to represent different system-defined roles of devices and interfaces.

Access (CE) or Shadow policy role

Gateway (PE) policy role

Core (P) policy role

Unknown policy role. This icon is displayed for virtual devices and for network components that have not been assigned a system-defined role.

In the policy context, different colors are used to represent the different policy roles of links (that is, the connected interfaces):

- Access links are shown in Blue
- Core links are shown in Dark Blue
- Local links are shown in Light Blue
- Disabled links are shown in Dark Gray
- Links classified as None are shown in Light Gray
- Links in conflict, where the roles assigned to either end of the connection do not match, are shown in Purple

Object labels

By default, each network component displayed on the map is labeled with the object's name, description or net address. The label used depends on the object type. If you have assigned system-defined roles to devices, each device's role is also displayed.

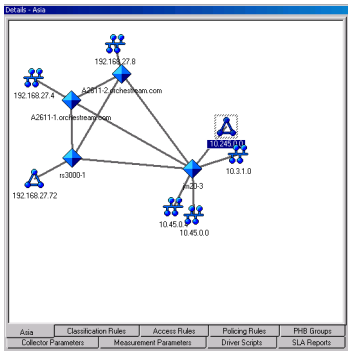
To switch object labels on or off

- From the **Map** menu, select **View Labels**.

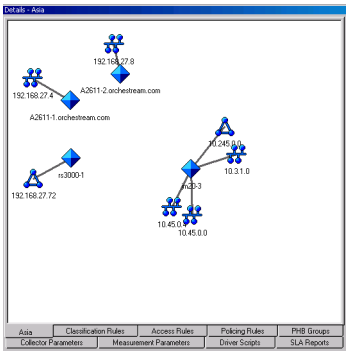
  Object labels are displayed on the map when a check is displayed next to the **View Labels** menu option.

Network segments

You can choose to hide the connectivity of a network segment if it is not shown on the map. This is controlled by the **Hide connectivity when not on map** checkbox on the Segment property page. By default this checkbox is cleared, and objects connected via a segment are always shown as connected, whether or not the segment appears on the map. If this checkbox is selected, no connectivity will be shown between objects that are indirectly connected via the segment.



With **Hide connectivity when not on map** checkbox cleared



With **Hide connectivity when not on map** checkbox selected

VLAN representation

Service Activator discovers VLANs on Cisco Catalyst switches running CatOS and represents the VLAN in the user interface.

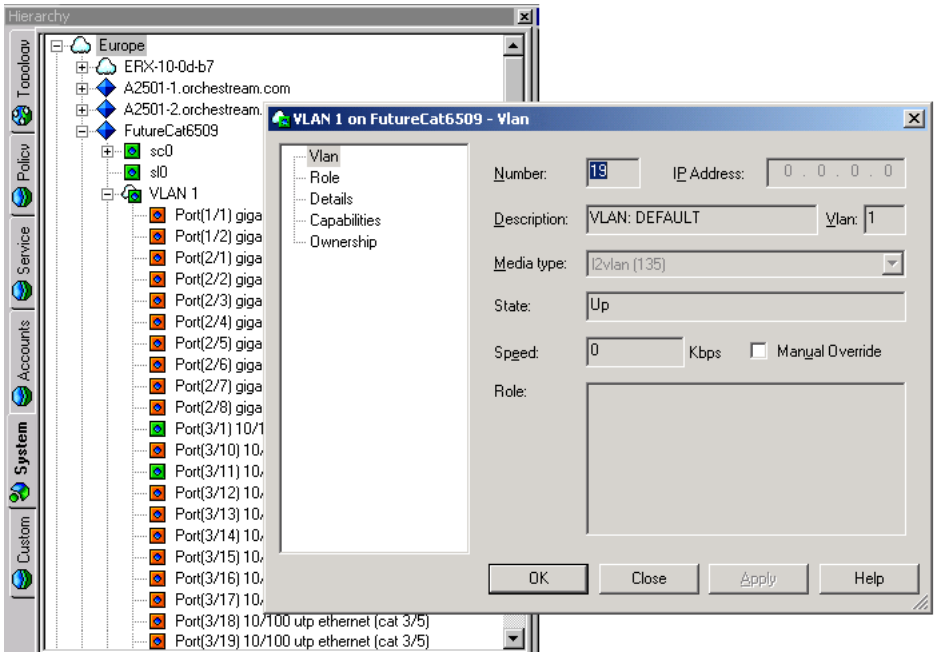
A Catalyst switch may run CatOS (operating at Layer 2) or IOS (operating at Layer 3) or have an MSFC card installed that runs IOS. Service Activator supports the discovery of VLANs and port assignment to VLANs on Catalyst switches running CatOS. If a Catalyst switch has an MSFC card that runs IOS, the switch's Layer 2 manageable entity and the MSFC card (Layer 3) are represented as two separate network devices, the Layer 3 entity (IOS), which can be managed by the Cisco device driver, and the Layer 2 entity (CatOS), which can be managed by the CatOS script driver.

VLANs are represented by different icon types on the Layer 2 manageable entity and the MSFC card.

VLAN representation on a CatOS device

On a Layer 2 CatOS device , a VLAN is represented as a VLAN interface connected directly to the device. The icon that represents the VLAN interface is shown below:

The physical ports that are currently assigned to a given VLAN appear as children of the VLAN interface. The property pages for the VLAN are identical to those of an interface object with the addition of a Vlan ID field.



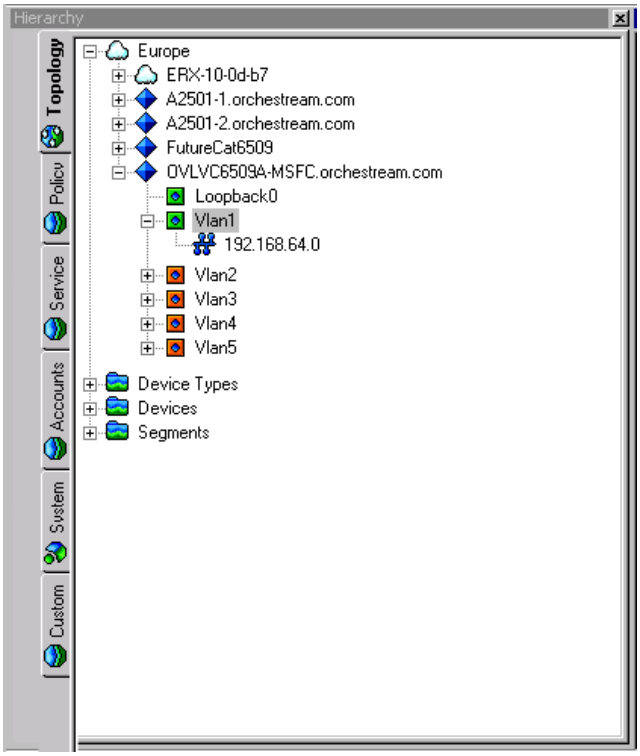
Capabilities are not returned for a VLAN interface on a Layer 2 manageable entity or for the physical ports that participate in the VLAN.

You may apply CDK scripts to the ports that participate in a VLAN by assigning a role to the relevant ports and associating the relevant scripts with that role. You cannot apply CDK scripts to the VLAN interface itself.

VLAN representation on an MSFC (IOS) device

On an MSFC (IOS) device, a VLAN is represented as a VLAN interface connected directly to the device. As the VLAN interface represents an IP-addressable Layer 3 entity, a standard interface icon is used to represent the VLAN interface:

The physical ports associated with the VLAN are not listed as children of the VLAN interface. A VLAN interface on an MSFC card has a network address associated with it and the segment attached to it is therefore displayed as well.



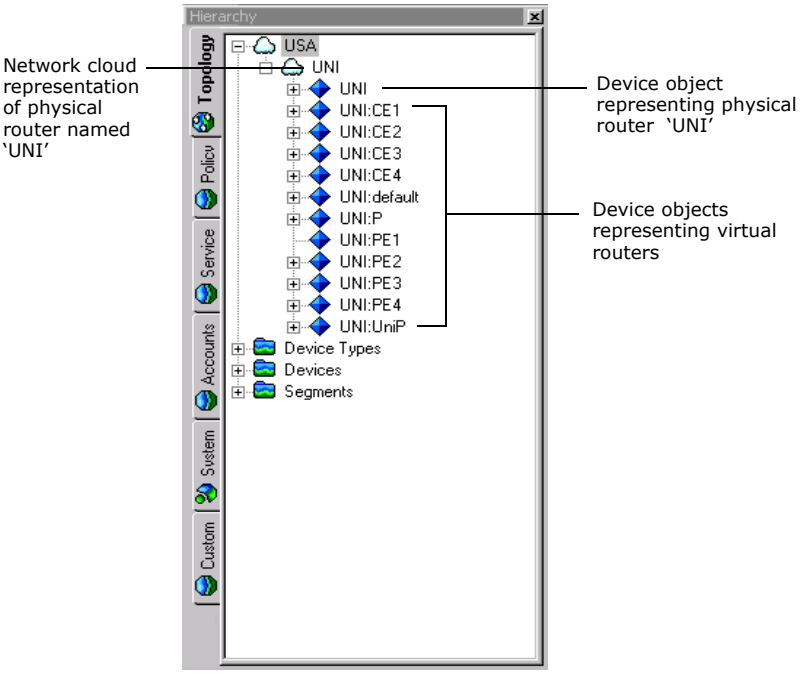
Capabilities are returned for the VLAN interface on an MSFC card, and roles can be assigned in exactly the same way as for physical interfaces.

The VLAN interface on an MSFC card can be associated with a site for inclusion in an MPLS VPN.

Juniper E-series virtual router representation

Juniper E-series virtual routers are displayed on the topology map in a similar way to real devices.

When Service Activator discovers a device containing virtual routers it creates a network cloud to 'hold' the routers. Beneath this cloud, Service Activator creates device objects that represent both the physical device and its virtual routers. The physical and the virtual routers exist at the same hierarchy level.

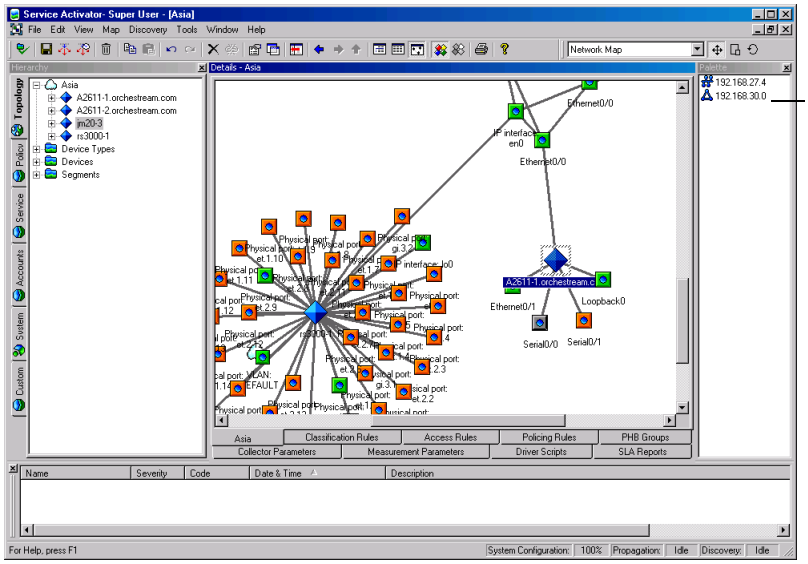


You can move virtual routers to other networks in the same way as physical devices.

> Because virtual routers can never be independent of the physical router, you should never move virtual routers on the user interface, or drag and drop them between networks.

Creating and viewing a topology map

The map view for a domain is displayed in the **Details** pane when you double-click on the root network object that represents the entire domain. You can either create the map manually or allow Service Activator to populate the map with network objects automatically



The palette lists unmapped network components

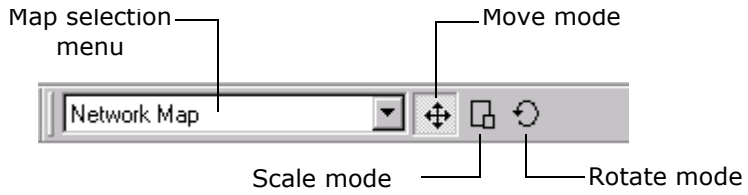
By default, maps are called **Network Map**, but you can change the name to something more descriptive. The map name is displayed on the tab at the bottom of the **Details** pane. Right-click on the tab and select **Properties** from the pop-up menu to edit the map's properties.

The palette on the right of the screen displays network components that have not yet been added to the map. It can be configured to list all unmapped network components or list components appropriate to the current context. For more information about the palette, see *Panes in the domain management window* on page 17.

> If you are laying out the map manually, you can remove an object from the map by dragging it onto the palette. If you delete an object on the map, it is deleted from Service Activator and must be rediscovered.

The map toolbar

The map toolbar provides options for working with the topology map.



- Map selection menu – lists all the network maps defined within the domain
- Move mode – enables you to move selected nodes to a new location while retaining the layout of the selected nodes
- Scale mode – allows you to enlarge or reduce the area occupied by selected nodes
- Rotate mode – enables you to rotate the selected nodes

For information on using the map toolbar, see *Selecting and laying out objects on a map* on page 196 and *Creating additional map views* on page 202.

Creating the map manually

This is the default setting for map creation. In this setup, the map is blank during the discovery process and Service Activator adds devices to the palette as they are discovered (the palette is normally displayed to the right of the screen). You can populate the map with network objects by dragging them from the palette or hierarchy pane on to the map. A snap to grid option is available when mapping objects manually. Connections between objects appear automatically.

Creating the map automatically

When automatic layout is selected, Service Activator adds network objects to the map as they are discovered.

By default, Service Activator adds networks, sites, segments and devices to the map. You can override the default to restrict or expand the network components that are mapped automatically – for example, you can map networks, sites, devices and interfaces automatically. Network components that are not mapped automatically are displayed in the palette.

> You cannot drag objects from the palette onto the map while the automatic layout option is selected.

Setting automatic layout parameters

Depending on which object types you select to appear on the map, the number of network objects may be quite large. This may lead to display problems if the number of objects is too high. For example, objects may overlap and affect the map's clarity or there may be an unacceptable delay time when displaying the map.

Service Activator can automatically lay out up to 200 network objects with minimal delay in display time. For greater numbers, there may be some delay in mapping and displaying objects. You can limit the delay by setting the values on the **Map Properties** dialog box, **Layout** property page including **Max nodes**, **Max fan**, **Max devices**, **Max devices, Max fan** and **Max nodes.**

Note: For complete dialog box and property page descriptions, refer to the *Online Help***.**

> If the number of network objects exceeds the defined limit, Service Activator omits some objects from the map. In this scenario, we recommend you lay out the map manually, incorporating the maximum number of nodes to be included in the map, and use the **Recalculate Layout** map option to regulate the map's layout. This option is not affected by the **Max nodes** setting. For information on the **Recalculate Layout** option, see *Recalculating a map's layout* on page 195.

Guidelines for working with maps

You may wish to work by initially selecting the automatic layout option to create the basic map structure and then change to the manual layout option to add other network objects to the map.

- If you are using a background image with a map, we recommend you use the manual mapping option so that you can position objects accurately on the image. For information on using a background image, see *Displaying a background image* on page 198.
- If your network includes ATM or Frame Relay VCs, you can drag one VC endpoint on to another to link the endpoints and create a VC link object. Note that you can also link VC objects in the hierarchy pane. The link between the VC endpoints and the VC link object is shown by a solid connection line.

To specify layout options for a map

1. With the Map View displayed, right-click anywhere in the **Details** pane and select **Properties** from the map's pop-up menu.

   Alternatively, click on the map and select Alt + Enter.

The **Map View** dialog box opens.

2. Select the **Layout** property page.
3. If you want to map network objects manually, select **Manually lay out items on map**.

   To specify that objects snap to a grid select the **Snap to grid** option and specify the granularity of the **Grid** in millimeters.
4. If you want to map objects automatically, select **Automatically lay out items on map** and specify the objects to be included on the map including **Networks, Sites, Devices, Interfaces, VCs, VC End Points**, and **Segments.**
5. Specify values for the map's maximum limits (**Max nodes, Max fan,** and **Max devices**).
6. Click **OK**.

To rename a network map

1. From the map's pop-up menu, select **Properties**.

   Alternatively, click on the map and select Alt + Enter.

   The **Map** dialog box opens.
2. Specify the **Map name** and click **OK**.

Recalculating a map's layout

If you have the manual layout option selected, the map's pop-up menu includes an option to recalculate the map layout, that is, arrange objects according to an automatic layout scheme.

Selecting the recalculate option does not switch the automatic layout option on but simply applies an automatic layout as a one-off operation.

This feature is useful if you have dragged objects on to the map from the palette or moved multiple objects.

To recalculate the map layout

1. With the Map View displayed in the **Details** pane and the map's manual layout option selected, select **Recalculate Layout** from the map's pop-up menu.

   The **Recalculate Layout Options** dialog box opens.
2. Select a layout option from **Hierarchical, Circular,** or **Symmetric.**
3. Click **OK**.

Selecting and laying out objects on a map

Service Activator provides options for moving, scaling or rotating a set of selected nodes on a map.

To select objects on a map

- Do one of the following:
  - Click on the network map, hold down the mouse button and drag the cursor over the nodes to be selected.

    As you drag the cursor, a dotted selection box is drawn around the selected nodes.
  - Select one or more objects and choose **Select Neighbours** from the objects' pop-up menu.
  - Press the Ctrl button while single-clicking each node in turn.

To lay out objects on a map

1. Select the relevant objects.
2. On the map toolbar, select a mode:
   - If you wish to move the node set to a new location on the map, select Move mode  and drag the nodes to the new location
   - If you wish to reduce or enlarge the area occupied by a set of nodes, select Scale mode  and drag the cursor up (increase area) or down (reduce area)
   - If you wish to rotate the selected nodes around a user-selected central node, select Rotate mode  , click on the node that will act as the rotation point and drag the cursor up (clockwise) or down (anti-clockwise)

Configuring the palette

By default, the palette is context-sensitive and the objects it lists reflect which type of network object is selected in the map. For example, if you select a device, any unmapped interfaces are displayed in the palette. If you select an interface, the palette lists any unmapped segments, sub-interfaces, etc. You can turn off this context-sensitive behavior and specify which object types are listed in the palette.

If you drag a displayed object into the palette area, it is removed from the map but not deleted from Service Activator.

> You can only drag objects between the palette and the map if you have the manual layout option selected.

By default, the palette is always displayed when viewing a map but you can switch off palette display for a map on a temporary or permanent basis.

For information on moving and docking the palette, see *Changing the size and position of a pane* on page 38.

To specify palette preferences

- From the **Map** menu, select **Palette**.

  The palette sub-menu is displayed.

  — To make the palette context-sensitive, select **Context Sensitive**.

  — To turn off context-sensitivity, deselect the **Context Sensitive** option and select the network objects you want to list in the palette.

To set a permanent palette display option for a map

1. From the map's pop-up menu, select **Properties**.

   The **Map** dialog box opens.

> If a map has a background image, you may need to right-click on the map's tab at the bottom of the **Details** pane to display its pop-up menu.

2. Select or deselect the **Default Palette Visible** checkbox as appropriate.
3. Click **OK**.

To set a temporary palette display option for a map

- With a topology map displayed, from the **View** menu, select **Map Palette**.

  The palette is displayed when the **Map Palette** option is checked. Selecting this option shows or hides the palette only for the current map viewing session. If you exit and return to the map, the palette's display is set according to the map's permanent map display setting.

Displaying a background image

If you wish, you can display one or more selected bitmap images as a background to a network topology or VPN map, for example, an outline of a country or region.

If you create your own images, avoid using background colors that are the same as those used by Service Activator icons. Background images must be stored in the **ServiceActivator\Backgrounds** directory and must be in Bitmap (.bmp) format.

> If there are multiple user interface components running on a number of host machines, the relevant background image must be accessible to each user interface component for the image to be displayed.

To add or remove a background image

1. Right-click anywhere within the map area and select **Properties** from the pop-up menu.

   The **Network Map** dialog box opens.

2. Select the **Background** property page.

   The **Background Image** pull-down list displays all the bitmap files (*.bmp format) that are in the **Backgrounds** folder. If files have already been selected for the map, they are displayed in the list below this field.

3. If you want to add a bitmap file to the map, select the file from the **Background Image** pull-down list and click **Add**.

   The file name is added to the list. Every file in this list will be added to the network map. As you click on each file name, Service Activator displays an image **Preview**.

4. If you want to remove a bitmap file from the map, select the file name from the list and click **Remove**.

5. Click **OK** when you are satisfied with the selection.

   The files you selected are added to the network map and positioned at point 0,0 – that is, the top left-hand corner of the map.

> If you selected more than one background image, the files are overlaid on the map.

To move an image

Either:

- Click on the image to select it and drag it to a new location.

  A selected image is surrounded by a handled selection box.

Or:

- Click on the image to display the **Background** dialog box and specify the image's new co-ordinates using the **X Axis** and **Y Axis** fields.

> Note that the map's 0, 0 co-ordinates are initially at the top left-hand corner of the map. If you move objects beyond this point, they will have a negative co-ordinate value. If you subsequently set a background image's x and y co-ordinates to 0, 0, this may not correspond to the top left-hand corner of the visible map.

To resize a background image by dragging

1. Click on the image to select it.

   A selected image is surrounded by a handled selection box.

2. Click a corner handle on the selection box and drag the handle to resize the image.

To resize a background image using the properties dialog box

1. From the background image's pop-up menu, select **Properties**.
2. Under **Size**, specify the image's **Width** and **Height**.
3. Click **OK**.

Changing the scale of the map

The scale of the network map can be varied to enable you to zoom in on a particular section or zoom out to view more of the map.

When a network or VPN map is displayed, a **Map** drop-down menu appears on the menu bar. The menu provides temporary zoom values for the map, enabling you to change the scale of the map between 25% and 200%. Alternatively, you can select the auto zoom feature to scale the map according to the value of its length or width, whichever is the greatest value, and retain the aspect ratio.

The temporary zoom setting is not retained if you exit and review the map. To make the zoom setting more permanent, set the map's default zoom setting. By default, this is 100%.

To set a map's temporary zoom setting

- From the **Map** menu, select **Zoom** then the magnification you require.

  You can change the scale of the map between 25% and 200%. Selecting **Auto Zoom** scales the map according to its length or width, whichever is the greatest value.

To define a map's default zoom setting

1. From the map's pop-up menu, select **Properties**.

   The **Map** dialog box opens.

> If a map has a background image, you may need to click on the map's tab at the bottom of the **Details** pane to display its pop-up menu.

2. In the **Default Zoom** field, select a new zoom value.
3. Click **OK**.

   The map is resized to the new zoom setting.

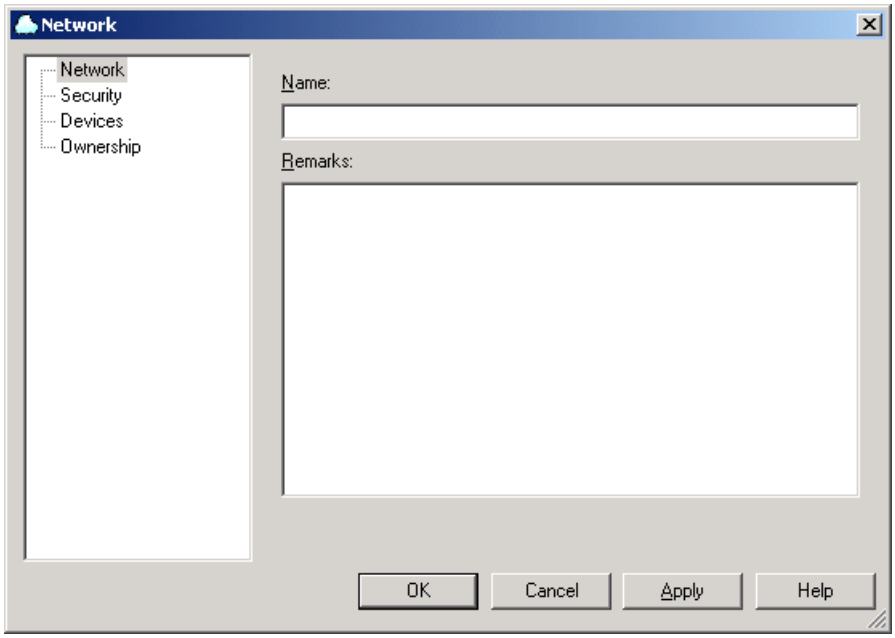
Creating subsidiary networks and maps

To simplify the management of large and complex networks you can divide the network into a number of subsidiary networks, each with its own topology map. Depending on the complexity of your network and the information you want to represent, you can create a multi-level hierarchy of subsidiary networks. Each device within the domain can only appear in one network.

To create a subsidiary network

1. Do one of the following:
   - — If you want to create a new network map that initially contains no devices, right-click on a blank area in the map display window for the domain and select **Add Network** from the pop-up menu.
   - — If you want to create a new network map that contains a copy of part of an existing network map, click and drag over the relevant devices to select them and select **Add Network** from the pop-up menu.

   The **Network** dialog box opens.



2. Enter a name for the subsidiary network and a brief description of the network and click **OK**. (You can enter the rest of the network properties after you have created the map and decided what devices are to be included in the network.)

A network cloud object appears on the map representing the subsidiary network. The network cloud object also appears in the hierarchy pane.

> Alternatively, you can select the top-level network in the hierarchy pane and select **Add Network** from the pop-up menu. In this case the network cloud will appear in the palette but will not initially appear on the map.

To move devices between networks

- To move a device into a lower-level network, select the device and drag it onto the appropriate network cloud. When you drop the device icon, it disappears into the new network cloud leaving only its links to other network objects visible.
- To move a mapped device into a higher-level network, select the device in the **Details** pane and select **Promote Upward** from the pop-up menu. It is removed from the current map and appears in the higher-level map's palette.

  Any links you made between devices in the lower-level network are not retained.

- Alternatively, you can move devices between networks by dragging and dropping devices within the hierarchy pane.

To set up and view maps for subsidiary networks

A map is automatically created for each subsidiary network.

- To view a subsidiary map, double-click on the new network cloud (either on the hierarchy pane or on an existing map).

  Initially subsidiary maps are blank, and you need to create the map by dragging devices from the palette.

- To return to a higher level map, click the **Up** toolbar button 

Creating additional map views

In addition to creating subsidiary networks and maps, you can set up several map views of each network. These appear as separate tabs on the **Details** pane when the map is displayed. For example, you can create one map that just displays the network devices and segments and another that also displays device interfaces and sub-interfaces. Devices and other network objects can appear on more than one map view of the network.

To create an additional network map for the domain

1. Do one of the following:
   - — If you want to create a new network map that initially contains no devices, right-click on a blank area on the map and select **Add Map View** from the pop-up menu.
   - — If you want to create a new network map that contains selected devices, click and drag over the devices to select them and select **Add Map View** from the pop-up menu.

   The **Map View** dialog box opens.
2. Enter a name for the map and a brief description of the map view and click **OK**. A new tab appears at the bottom of the Map display window.
3. Click on the tab to display the new map view. To start with, the map area is blank and all the network objects that have been discovered in the domain are listed in the palette.
4. Create the map in the standard way by dragging devices from the palette.

To view another network map

- On the map toolbar, select a map from the network map selection menu.

Network map
selection menu

Chapter 9

Checking Device Status and Capabilities

This chapter describes how to:

- View the status of a device or interface
- View the list of interfaces on a device
- Check a device or interface's capabilities
- Define device/interface capabilities for devices configured by a Network Processor cartridge

Viewing the status of a device or interface

While a device is managed by Service Activator, the proxy agent polls it regularly to check that it is still running. A device's status, as reported in the user interface, may change as a result of polling. For example, if a device fails, its status is changed to Unreachable. The device is still polled and will be reset when it recovers. If the problem fails to reset, however, or if a critical fault is raised on a device, the status is changed to Intervention Required until the problem is resolved.

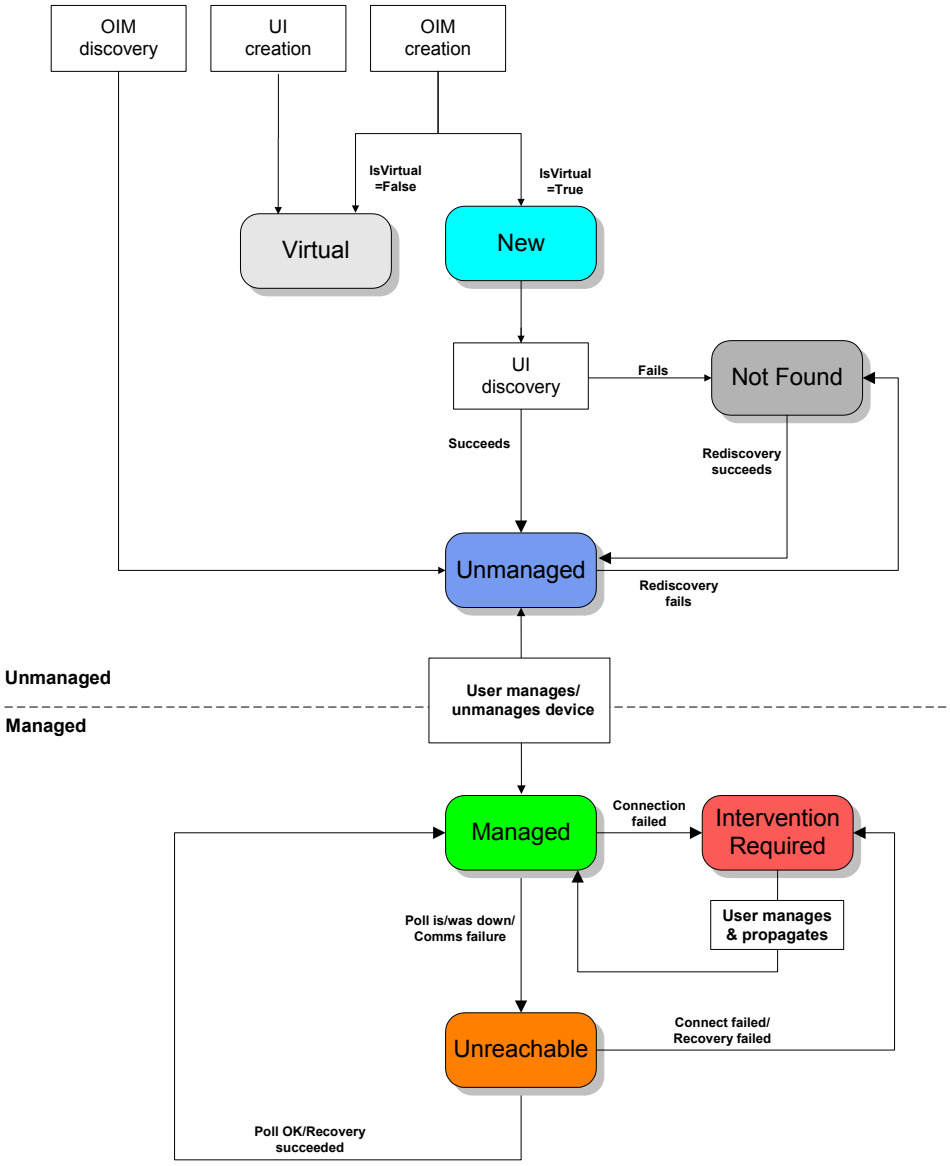
The diagram on page 207 shows the range of device states and the transitions between them.

Note that initial discovery of a device may be by any of the following routes:

- Discovering or creating the device through the user interface. For information, see *Running device discovery* on page 153.
- Discovering or creating the device via the OSS Integration Manager (OIM) – see the *OSS Integration Manager Guide*.

You can check the status of the devices and interfaces within a domain by viewing the topology map in status context. Every device and interface state is represented by a display color in the map.

For more information, see *How objects are represented* on page 186.



To access a device or interface's properties dialog box

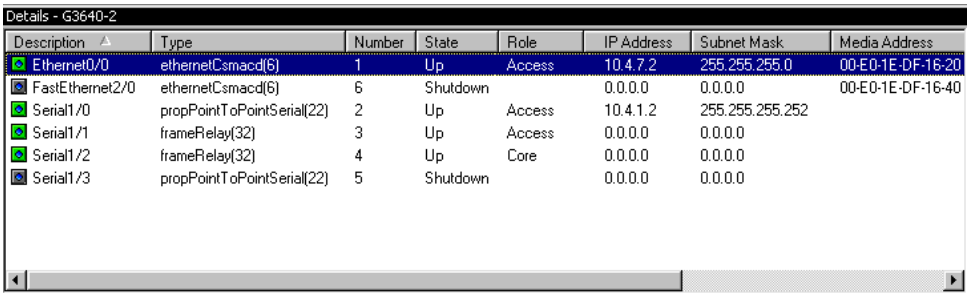
1. Select the object, either on the network map or **Topology** tab.
2. From the object's pop-up menu select **Properties**.

   If the selected object is a device or an interface, the information displayed in the **Capabilities** property page is particularly important. It summarizes the support

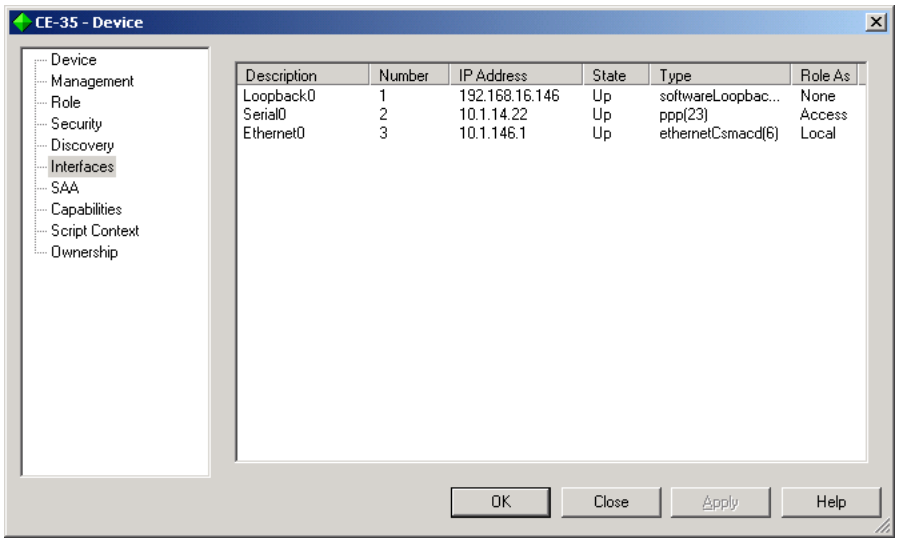
that the interface provides for QoS mechanisms, VPN capabilities, policy rules and SLA monitoring. You should be aware of these capabilities when planning services and policies. For more information on interface capabilities, see *Checking capabilities* on page 210.

Viewing a list of interfaces on a device

You can list the interfaces that are available on a device by double-clicking the device in the hierarchy pane. Service Activator lists the selected device's interfaces in the **Details** pane.



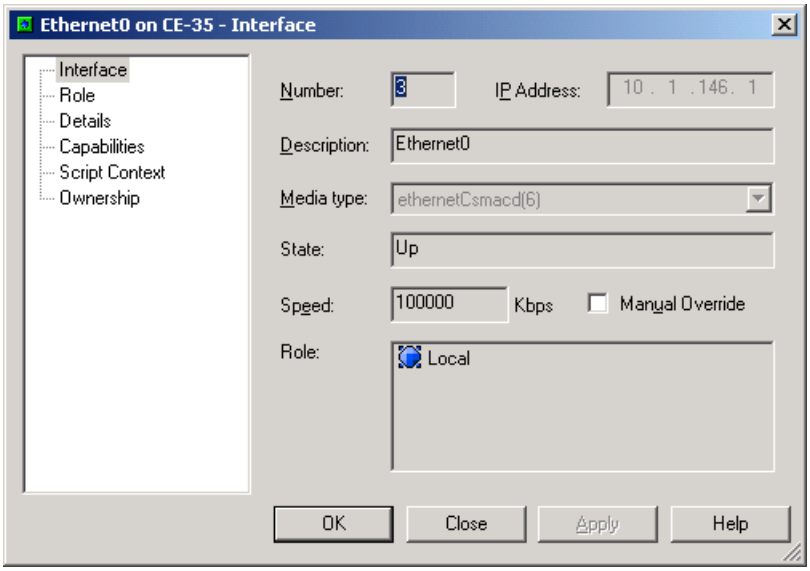
You can also list a device's interfaces on the **Interfaces** property page on the **Device** dialog box.



For each interface on the device, the **Interfaces** property page displays the following information:

Description	A description of the connection. This is the SNMP **ifDescription** parameter.
Number	The number of the interface.
IP Address	IP address of the interface. (If the interface has multiple IP addresses, this is the first address.)
State	Status of the interface (**Up**, **Down**, **Shutdown** or **Unknown**). For virtual interfaces, the state is always **Unknown**.
Type	The type of interface, such as Ethernet, point-to-point. This is the SNMP **ifType** parameter.
Role As	The policy role assigned to this interface, such as **Core**, **Local**, **Access**, **None** or **Disabled**.

Double-clicking on an interface in the **Description** column opens the properties dialog box for that interface.



Checking capabilities

Service Activator displays information about the capabilities of devices, interfaces, sub-interfaces and VC endpoints. Being aware of each device and interface's capabilities is an important factor in developing and applying policies and services as they indicate the VPN service, policy and measurement types that are supported. In addition, if capabilities are not known, rules, PHB groups and some measurement types cannot be applied to that network component.

Capabilities are obtained as part of the device discovery process. However, if Service Activator is unable to obtain the capabilities, for example because the device's security settings are incorrect, you can request the capabilities later.

Service Activator derives interface and sub-interface capabilities from the device's operating system, device type and interface name. VC endpoint capabilities also take account of the interface type.

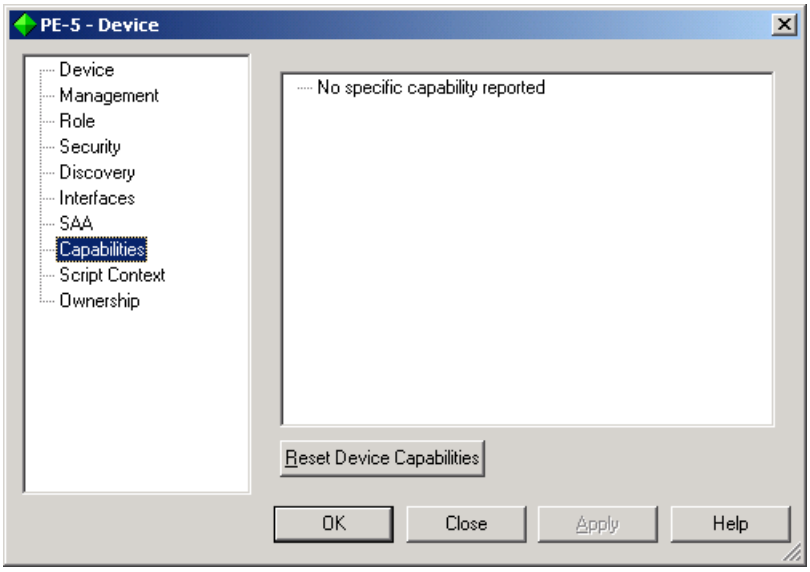
Service Activator stores capability details for devices, interfaces, sub-interfaces and VC endpoints.

Device-level capability categories

A device may have any of the following capabilities:

- SAA support: indicates the maximum number of SAA probes (operations) supported and which SAA operations are supported – ICMP Echo, UDP Echo, Jitter or TCP Connect
- NetFlow support: indicates in which versions of UDP format NetFlow data may be exported from the device – Version 1, Version 5 or Version 8

These capabilities are listed on the **Capabilities** property page of the device's properties dialog box, for example:



If the words 'Capabilities not obtained' are displayed on this page, Service Activator has not obtained the relevant information from the device. This may be because Service Activator could not communicate with the device, because the security parameters are incorrect (for Cisco or Juniper E-series devices) or because the device's operating system is not supported.

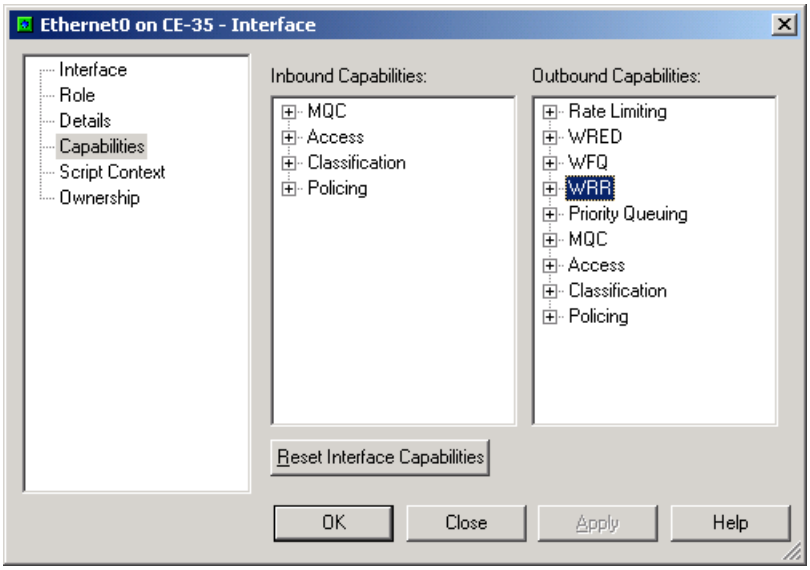
Interface-level capability categories

An interface, sub-interface or VC endpoint may have any of the following capabilities:

- **Access**: indicates that you can implement access rules.
- **ATM**: indicates that you can implement PHB groups using the ATM mechanism.
- **CCC**: indicates support for CCCs (Circuit Cross Connects).
- **Class-based queuing**: indicates support for CB-WFQ.
- **Classification**: indicates that you can implement classification rules.
- **FRTS**: indicates that you can implement PHB groups using the Frame Relay Traffic Shaping mechanism.
- **Martini**: indicates support for Layer 2 Martini VPNs
- **MQC**: indicates support for Modular QoS CLI on Cisco devices.
- **Policing**: indicates that you can implement policing rules.

- **Priority Queuing**: indicates that you can implement PHB groups using the Priority Queuing mechanism.
- **Rate Limiting**: indicates that you can implement PHB groups using rate limiting.
- **Virtual Circuits**: indicates a Frame Relay or ATM VC.
- **VPN**: indicates that this interface supports VPNs.
- **WFQ**: indicates that you can implement PHB groups using the WFQ queuing mechanism.
- **WRED**: indicates that you can implement PHB groups using the WRED queuing mechanism.
- **WRR**: indicates that you can implement PHB groups using the WRR queuing mechanism.

These capabilities are listed on the **Capabilities** property page of the **Interface** dialog box.

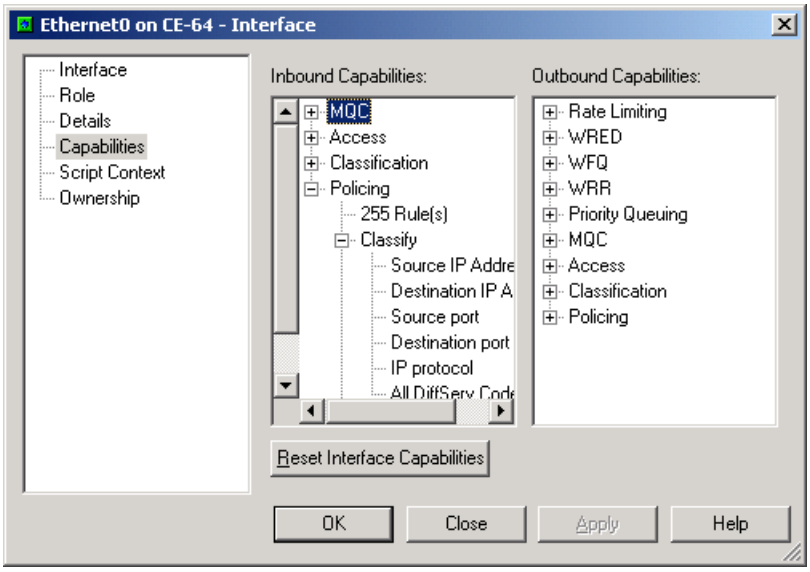


If the words 'Capabilities not obtained' are displayed on this page, Service Activator has not obtained the relevant information from the device. This may be because Service Activator could not communicate with the device, because the security parameters are incorrect (for Cisco or Juniper E-series devices) or because the device's operating system is not supported.

If no text is displayed in a field, it indicates that the interface does not support any Service Activator features or the device or interface are not supported. If the device

or interface are not supported, an error message is displayed in the current faults pane.

You can find out more detailed information about most of the capabilities listed on the **Capabilities** property page by double-clicking on the ⊞ icon next to the capability. The details are displayed beneath the capability. For example, the following details are displayed if you double-click on Policing:



The details displayed vary depending on the capability type. The ways in which the interface can identify and classify traffic is specified for rule and queuing capabilities, while for queuing mechanisms, the number of queues that can be handled by the interface is specified.

Refetching device capabilities

If Service Activator is unable to discover capabilities during the discovery process – for example, because the device's security settings are incorrect – you can request the capabilities at a later point.

In addition, if the device capabilities have changed – for example, as a result of an operating system upgrade – you can refresh capabilities by resetting and remanaging the device.

Changes made to the capabilities files will not take effect in the system until the fast start process has completed. Erratic behaviour may be seen if there is any attempt to re-fetch capabilities while the proxy agent and/or device driver is in the start-up process. Wait until the message **Driver is up and running (fast start complete)** appears in the Current Faults Pane before rediscovering capabilities.

Although the network processor does not include a fast-start process, you must wait for startup to complete before re-fetching capabilities. Network processor start is indicated in an information message in the Current Faults Pane.

To discover unknown device capabilities

1. From the **Discover** menu, select **Discovery**.

   The **Discovery** dialog box opens.

2. Select **Fetch Capabilities** from the **Discovery Type** drop-down.

3. Click **OK** or **Apply**.

Service Activator attempts to discover capabilities for all devices that have not had capabilities returned. It does not rediscover capabilities where they have already been reported.

To update previously found device capabilities

1. Unmanage the device ensuring Service Activator configuration is left on the device:
   - Open the device's property pages and select the **Management** property page
   - Ensure that **Inherit System Action** is set to **Leave**
   - Click the **Unmanage** button
   - Click **OK** to close the dialog box
   - Commit the transaction
2. Reset the device capabilities:
   - Open the device's properties dialog box and select the **Discovery** property page.
   - Click on the **Reset Device Capabilities** button
   - Click **OK** to close the dialog box
   - Commit the transaction

3. Fetch the updated capabilities:
   - From the **Discovery** menu, select **Discover**. The **Topology Discovery** dialog box opens.
   - In the **Discovery Type** field, choose **Fetch Capabilities**
   - Click **OK** to close the dialog box
   - Commit the transaction
4. Remanage the device:
   - Open the device's properties dialog box and select the **Management** property page:
   - Click the **Manage** button
   - Click **OK** to close the dialog box
   - Commit the transaction

Modifying device/interface capabilities

This procedure applies only to devices configured by a network processor cartridge.

You can modify the interface capabilities of devices configured by a specific cartridge unit, by modifying the capabilities file registered to that cartridge unit.

The `MIPSA_registry.xml` file lists information about the cartridge units in a vendor-specific cartridge family. It names the cartridge unit name, driver type, device type, OS type, some service model/device model/CLI transforms, and the applicable capabilities file. A sample entry follows:

```
<!-- Cisco 3640 devices with IOS 12.2(8)T4 -->
<cartridgeUnit>
    <name>com.metasolv.serviceactivator.cartridges.cisco.units.cu1.3640.
         12.2(8)T4</name>
    <driverType>cisco</driverType>
    <deviceType>Cisco 3640</deviceType>
    <osRegex>.*12\.2\(8\)T4.*</osRegex>
    <smToDmQuery>com/metasolv/serviceactivator/cartridges/cisco/units/cu1/
         sm2dm.xq</smToDmQuery>
    <dmValidation>com/metasolv/serviceactivator/cartridges/cisco/units/cu1/
         dmValidation.xq</dmValidation>
    <dmToCliQuery>com/metasolv/serviceactivator/cartridges/cisco/units/cu1/
         annotatedDm2Cli.xq</dmToCliQuery>
    <capabilities>com/metasolv/serviceactivator/cartridges/cisco/capabilities/
         cisco_3640_12.xml</capabilities>
</cartridgeUnit>
```

In the above sample, `MIPSA_registry.xml` defines a cartridge unit used to configure Cisco 3640 devices running the 12.2(8)T4 IOS. It names the capabilities file, `cisco_3640_12.xml`.

Each capabilities file consists of sections, including the following:

- device capabilities
- interface capabilities, inbound and outbound sections
- sub-interface capabilities, inbound and outbound sections
- virtual circuit capabilities, inbound and outbound sections

In file `cisco_3640_12.xml`, for example, you can reference the following enumerated types to add Classification and Marking capabilities on interfaces:

SRC_IP	Source IP address
DST_IP	Destination IP address
SRC_PORT	Source port
DST_PORT	Destination port
IP_PROTO	Internet Protocol
DIFFSERV	Differentiated Services
IPv4PRECEDENCE	IP version 4 precedence bits
IPv4TOS	IP version 4 type-of-service byte (octet)
URL	Universal Resource Locator
MIME	Multipurpose Internet Mail Extensions
APP_PROTO	Application protocol
APP_NAME	Application name
DOMAIN_NAME	Domain name
IEEE_802_1_PBITS	IEEE 802.1P priority field
MPLS_EXP	Multi-Protocol Label Switching experimental value
FR_DE	Frame Relay discard eligibility bit
ATM_CLP	Asynchronous Transfer Mode cell loss priority bit
CLASS_MAP	Class map
MATCH_ANY	Match any
ALCATEL_INTERNAL_QUEUE	Alcatel internal queue
TCP_HEADER_OPTIONS	Transmission Control Protocol header options
TCP_ESTABLISHED	Transmission Control Protocol established
EXCLUDE_CLASSIFICATION	Exclude classification

Code sample

Sample code excerpts follow. (Entries removed are implied by colons, for the sake of brevity.)

```
<caps:capabilities xmlns:caps="http://www.metasolv.com/serviceactivator/
                                                                capabilities"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <caps:device>
        :
        :
        <caps:ipsec_support>
            <caps:esp_algorithms_supported>0</caps:esp_algorithms_supported>
            <caps:ah_algorithms_supported>0</caps:ah_algorithms_supported>
            <caps:compression_algorithms_supported>0</caps:compression_
                 algorithms_supported>
            <caps:ipsec_modes_supported>0</caps:ipsec_modes_supported>
        </caps:ipsec_support>
        :
        :

    </caps:device>
    <caps:interface>
        <caps:ifType>32</caps:ifType>
        <caps:inbound>
            :
            :
            <caps:service_rule_support>
                <caps:rules_supported>0</caps:rules_supported>
                <caps:mark_codepoints_supported/>
                <caps:limits_supported>0</caps:limits_supported>
                <caps:guarantees_supported>0</caps:guarantees_supported>
                <caps:limits_and_guarantees_supported>0</caps:limits_and_
                     guarantees_supported>
                <caps:classification_supported/>
            </caps:service_rule_support>
            <caps:policing_rule_support>
                <caps:policing_supported>7</caps:policing_supported>
                <caps:classification_supported>
                    <caps:supported>SRC_IP</caps:supported>
                    <caps:supported>DST_IP</caps:supported>
                    <caps:supported>SRC_PORT</caps:supported>
                    <caps:supported>DST_PORT</caps:supported>
                    <caps:supported>IP_PROTO</caps:supported>
                    <caps:supported>APP_PROTO</caps:supported>
```

```
            </caps:classification_supported>
            <caps:classification_queues_supported>7</caps:classification_
                 queues_supported>
            <caps:mark_codepoints_supported>
                <caps:supported>SRC_IP</caps:supported>
                <caps:supported>DST_IP</caps:supported>
                <caps:supported>SRC_PORT</caps:supported>
            </caps:mark_codepoints_supported>
        </caps:policing_rule_support>
        :
        :
    </caps:inbound>
    <caps:outbound>
        :
        :
        <caps:service_rule_support>
            <caps:rules_supported>0</caps:rules_supported>
            <caps:mark_codepoints_supported/>
            <caps:limits_supported>0</caps:limits_supported>
            <caps:guarantees_supported>0</caps:guarantees_supported>
            <caps:limits_and_guarantees_supported>0</caps:limits_and_
                 guarantees_supported>
            <caps:classification_supported/>
        </caps:service_rule_support>
        <caps:policing_rule_support>
            <caps:policing_supported>7</caps:policing_supported>
            <caps:classification_supported>
                <caps:supported>SRC_IP</caps:supported>
                <caps:supported>DST_IP</caps:supported>
                <caps:supported>SRC_PORT</caps:supported>
                <caps:supported>DST_PORT</caps:supported>
                <caps:supported>IP_PROTO</caps:supported>
                <caps:supported>APP_PROTO</caps:supported>
            </caps:classification_supported>
            <caps:classification_queues_supported>7</caps:classification_
                 queues_supported>
            <caps:mark_codepoints_supported>
                <caps:supported>SRC_IP</caps:supported>
                <caps:supported>DST_IP</caps:supported>
                <caps:supported>SRC_PORT</caps:supported>
            </caps:mark_codepoints_supported>
        </caps:policing_rule_support>
        :
        :
```

```
            </caps:outbound>
            <caps:subInterface>
                <caps:inbound>
                :
                :
                </caps:inbound>
                <caps:outbound>
                :
                :
                </caps:outbound>
            </caps:subInterface>
        </caps:interface>
    </caps:capabilities>
```

Index