Copyright © 1997, 2008, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee’s responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
Contents

Preface ........................................................................................................ vii

Chapter 1 Overview ..................................................................................... 1

Chapter 2 Service Assurance Agent (SAA) measurement .................. 7

Service Assurance Agent publications .................................................... ix

Chapter 1 Overview ..................................................................................... 1

Introduction ............................................................................................... 2

Measurement types ...................................................................................... 2

External systems ......................................................................................... 3

Configurable XML Exporter ....................................................................... 3

Integration with InfoVista .......................................................................... 3

Integration with MicroMuse ....................................................................... 4

Getting started ........................................................................................... 5

Chapter 2 Service Assurance Agent (SAA) measurement .................. 7

What is Service Assurance Agent? ............................................................. 8

SAA operations .......................................................................................... 8

Configuration considerations .................................................................... 8

Supported operations ................................................................................ 15

SAA Responder ......................................................................................... 16

SAA templates .......................................................................................... 16

Applications in Service Activator ............................................................. 17

MPLS VPN ............................................................................................... 17

Measurement-only VPN .......................................................................... 17
Contents

Chapter 3 MIB-based and NetFlow measurements ....................... 31
  Overview of MIB and NetFlow-based measurements ....................... 32
  CAR MIB ................................................................................................ 32
  MIB2 ..................................................................................................... 32
  NetFlow ................................................................................................. 33
    Architecture and components ........................................................... 33
    What is a flow? .................................................................................. 34
    UDP formats ...................................................................................... 34
    Aggregation ....................................................................................... 35
  Configuring measurement types in Service Activator ..................... 37
    Considerations ................................................................................... 37
    Applying NetFlow or MIB-based measurements to a policy target ....... 39

Chapter 4 Setting Up Service Activator for Integration ................. 43
  External systems .................................................................................. 44
    Types ............................................................................................... 44
    Creating an external system .............................................................. 45
  Collectors ............................................................................................... 45
    Roles and measurement parameters ................................................. 46
    Creating a collector in Service Activator .......................................... 48

Chapter 5 Generic Exporter ............................................................. 49
  Integration with Service Activator ...................................................... 50
    Key components ................................................................................ 51
    TopologyExporter.xml Filtering ....................................................... 52
  Customizing integration ...................................................................... 52
    TopologyExporterConfig.xml .......................................................... 52
    TopologyExporter.xsl ........................................................................ 52
Contents

Invoking the Micromuse Integration Module .................................................. 78
Micromuse object mapping ........................................................................... 79
Specifying additional object information ..................................................... 80
  MicromuseExport.xml Filtering .............................................................. 81
Customizing Micromuse integration .............................................................. 82
  MicromuseExportConfig.xml ................................................................. 82
  Micromuse.xsl .................................................................................... 83
Deployment ............................................................................................. 84
Error reporting .......................................................................................... 84

Appendix A TopologyExporterConfig.xml Fields ..................................... 85
  Directory locations .............................................................................. 86
  Configuration fields .......................................................................... 86
  Output rules ....................................................................................... 87

Index ........................................................................................................ 1
Preface

About this document

This guide is for users of third-party SLA monitoring and reporting tools. It explains the different measurement techniques available and how to configure Service Activator to measure service levels, for monitoring Service Level Agreements (SLAs).

It consists of the following chapters:

- Chapter 1: Overview provides an overview of Service Activator’s SLA monitoring capabilities, including how Service Activator integrates with third-party reporting tools.
- Chapter 2: Service Assurance Agent (SAA) measurement provides an overview of the Service Assurance Agent (SAA), describes which point-to-point connections may be monitored and explains how to apply SAA in Service Activator.
- Chapter 3: MIB-based and NetFlow measurements gives an overview of NetFlow and MIB-based measurements and describes how to configure these measurement parameters in Service Activator.
- Chapter 4: Setting Up Service Activator for Integration describes how third party reporting tools are integrated with Service Activator.
- Chapter 5: Generic Exporter describes the Service Activator Generic Exporter.
- Chapter 6: InfoVista Integration describes InfoVista’s architecture and how it is integrated with Service Activator.
- Chapter 7: Micromuse Integration describes Micromuse’s architecture and how it is integrated with Service Activator.

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](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](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](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](http://www.oracle.com/technology/documentation).
**Downloading a media pack**

To download a media pack from Oracle E-Delivery

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

Overview

This chapter provides a brief overview to the use of Network and SLA Monitoring with Service Activator. It includes the following topics:

- Introduction
- Measurement types
- External systems
- Configurable XML Exporter
- Integration with InfoVista
- Integration with MicroMuse
- Getting started
Introduction

Service Activator integrates with a wide range of third-party reporting systems and applications to offer network and SLA monitoring. To do this, network measurement parameters are first configured in Service Activator to generate network statistics from the devices. A third-party reporting tool is then set up to collect and process the performance data.

Measurement and SLA monitoring are currently supported on Cisco IOS-based devices only.

Measurement types

Service Activator supports a range of measurement types that enable the measurement of point-to-point performance, or performance at a specific point in the network. The following measurement types are supported:

- **Service Assurance Agent (SAA)** – a Cisco technology that performs point-to-point measurement based on key SLA metrics such as response time, network resources, availability, jitter, connect time and packet loss.
- **CAR MIB statistics** - report on performance of CAR-enabled interfaces. Used to monitor bandwidth usage.
- **MIB2 statistics** – report on Management Information Bases (MIBs) in MIB2 format. Their variables indicate the basic state of the network – for example, load, availability, discards, broadcast rate.
- **NetFlow** – a Cisco technology that enables you to characterize and analyze an IP flow on an interface. It is often used as a metering base for other applications, including accounting/billing, network planning, and marketing.

SAA measurements can be collected for pairs of peer devices within a VPN – either an MPLS VPN or a measurement-only VPN, with no MPLS configuration applied. Measurement is applied by defining an SAA template that holds one or more SAA operations and applying the template to a VPN.

NetFlow and MIB-based measurements can be collected for any Service Activator policy target – that is, a domain, network, customer, VPN, site, device, interface, sub-interface or VC endpoint. Measurement is applied using a measurement template which can be selectively inherited to lower level policy targets using device and interface roles. This means that you can apply measurement at a high level, such as the network, and implement measurement only at selected points in the network.
External systems

Service Activator uses external systems to model third-party applications in the user interface and object model. When setting up reporting and measurement, one or more external systems must be modelled through the Service Activator user interface.

The type and number of external systems that must be modelled depends on the reporting software used.

For complete information on modelling external systems, see External systems on page 44.

Configurable XML Exporter

The Configurable XML Exporter runs as a client of the OSS Integration Manager (OIM). Based on information contained in a local configuration file, the Configurable XML Exporter creates an export file that can be used by a third-party reporting tool to collect data for report purposes.

The local configuration file specifies the following information:

- Service Activator IP address
- which data, including type and scope, to extract from the object model
- the number of XSL stylesheets to apply to the XML data
- name of the final export file

Integration with InfoVista

Service Activator’s InfoVista Integration module enables InfoVista to monitor and report on Service Activator’s IP services. The solution offers a range of monitoring techniques that support SLA monitoring of point-to-point services, such as VPNs, as well as performance of specific interfaces.

Integration between Service Activator and InfoVista occurs in the following way:

- Service Activator’s Configurable XML Exporter creates an export file that identifies which data to collect for the InfoVista reports
- InfoVista uses Vista Provisioner to import the export file and generate the necessary reports
As the diagram shows, Service Activator uses the InfoVista Configurable XML Exporter utility to export object model data in XML format. The exported data is then transformed by one or more XSL stylesheets into a format that can be imported by Vista Provisioner.

The export of data between Service Activator and InfoVista can occur on demand or can be scripted to occur on a scheduled basis using a cron task.

**Integration with MicroMuse**

Service Activator’s Micromuse Integration Module enables Micromuse Netcool to monitor and report on Service Activator’s IP services. The solution offers a range of monitoring techniques that support SLA monitoring of point-to-point services, such as VPNs, as well as performance of specific interfaces.

Integration between Service Activator and Micromuse occurs in the following way:

- Service Activator’s Configurable XML Exporter creates an export file that identifies which data to collect for the Micromuse reports.
Micromuse Netcool Impact Server imports the exported XML file and generates the necessary reports.

As the diagram shows, Service Activator uses the Micromuse Configurable XML Exporter utility to export object model data in XML format. The exported data is then transformed by one or more XSL stylesheets into a format that can be imported by the Micromuse Netcool Impact Server.

The export of data between Service Activator and Micromuse can occur on demand or can be scripted to occur on a scheduled basis using a cron task.

**Getting started**

To set up network and SLA monitoring in Service Activator, you must first configure the devices in Service Activator to generate statistics for the various measurement types.
To do this, refer to the following chapters:

- **Chapter 2:** Service Assurance Agent (SAA) measurement
- **Chapter 3:** MIB-based and NetFlow measurements

Once the devices are configured to generate statistics, you should then set up the third-party reporting tool to collect the statistics and generate reports.

Procedures on how to do this are covered in the following chapters:

- **Chapter 4:** Setting Up Service Activator for Integration
- **Chapter 6:** InfoVista Integration
- **Chapter 7:** Micromuse Integration
Chapter 2

Service Assurance Agent (SAA) measurement

This chapter covers the following topics:

- What is Service Assurance Agent?
- SAA operations
- SAA Responder
- SAA templates
- Applications in Service Activator
- Configuring SAA
What is Service Assurance Agent?

Service Assurance Agent (SAA) is a Cisco technology that measures key SLA metrics such as response time, network resources, availability, jitter, connect time and packet loss between two devices. SAA is also referred to as Response Time Reporter (RTR).

SAA’s measurement features are built into the Cisco IOS software, supporting response time monitoring without the need to purchase and deploy additional equipment and software in the network. This feature may represent significant cost and management savings.

For information on device and IOS support, see the Cisco IOS Device Support Guide.

SAA operations

When you apply SAA to a VPN, Service Activator tests the connection between pairs of sites in the VPN. The SAA operation that performs the test is configured at the device level.

Also known as probes, SAA operations collect response time and availability information. An operation is configured on a device and tests the connection to a target device by placing synthetic packets in the network. The packets simulate other forms of network traffic, depending on the type of operation that has been configured. Each SAA operation has a unique ID number that enables tracking of the operation.

This section details how different VPN configurations can affect the implementation of SAA operations. The SAA operations supported by Service Activator are also covered.

Configuration considerations

The number of operations that are configured when you apply SAA to a VPN depends on the topology of the VPN and whether you choose to configure one or two-way tests.
Also, unmanaged devices can also be configured for SAA measurement by modelling the device as a virtual device in Service Activator.

**Topology**

The number of tests performed depends on the VPN’s topology and whether the probe properties are set to half or full duplex:

- In a fully-meshed VPN, Service Activator tests the connection between each pair of sites
In a hub and spoke VPN, Service Activator tests the connection between the hub site and each spoke site. If there are multiple hub sites and they are meshed, Service Activator also tests the connection between each pair of hub sites.

**One-way and two-way probes**

When you apply SAA to a VPN, you can specify whether the probe configured for each pair of sites is:

- A one-way (half duplex) – an operation is configured on only one device in the connection
- A two-way (full duplex) – an operation is configured on both devices in the connection

Note that for TCP Connect, UDP Echo and Jitter operations, Service Activator also configures SAA Responder on target devices probed by one of these operations.
**One-way probe**

For a one-way probe, Service Activator configures the operation on only one device for each A to B pair.

For hub and spoke VPNs, a probe is configured on each spoke, testing the connection from the spoke to the hub.
For fully-meshed VPNs, Service Activator chooses one of the devices in each pair and configures the probe on that device only, ensuring that the load imposed by SAA is evenly distributed between devices.
Two-way probe

For a two-way probe, Service Activator configures the operation on both devices for each A to B pair.

For hub and spoke VPNs, an operation is configured on each spoke, with the hub as the target, and on the hub an operation is configured to probe each spoke connection.
If there are multiple hub sites and they are meshed, Service Activator probes the connection between each pair of hub sites.

For fully-meshed VPNs, probes are performed between each pair of sites.

Unmanaged devices

An unmanaged device is a device that is not managed by Service Activator. It is possible to test the connection from a managed device to an unmanaged device by modelling the unmanaged device as a virtual router. The virtual device is set up as the destination device in a one-way probe. One or more managed devices can then send test packets to the virtual device.
For information on modelling a virtual device, see the *Network Discovery and Basic Setup* guide.

Note that when testing the connection to a virtual device, Jitter, UDP Echo and TCP Connect operations require the manual configuration of rtr responder on the target device. Service Activator cannot configure rtr responder on an unmanaged (virtual) device.

The connection to a virtual device can be tested in one direction only – that is, from the managed to the unmanaged device.

**Supported operations**

The following table summarizes Service Activator’s support for SAA operations.

<table>
<thead>
<tr>
<th>Supported operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP Echo</td>
<td>Measures end-to-end response time between a Cisco router and devices using IP.</td>
</tr>
<tr>
<td>UDP Echo</td>
<td>Calculates UDP response times between a Cisco router and an IP-enabled device.</td>
</tr>
<tr>
<td>TCP Connect</td>
<td>Discovers the time it takes to connect to a target device.</td>
</tr>
<tr>
<td>Jitter</td>
<td>A superset of the UDP Echo operation. The Jitter operation additionally measures per-direction packet-loss and jitter (inter-packet delay variance).</td>
</tr>
</tbody>
</table>

For detailed information about these operations, consult the Cisco documentation.

The number of SAA operations that can be configured on a Cisco device is limited by the router’s IOS and hardware specification. Devices running IOS 12.1 or later support a maximum of 500 operations.

For UDP Echo, ICMP and Jitter operations, the target device may not natively provide the service. You can specify whether the operation sends a control message to the target device to enable the destination port prior to sending a packet.
For all operations, you can define the rising and/or falling threshold for response times to the operation’s test packets. A variety of algorithms are available for calculating a violation. You can also specify how long the operation waits for a response, whether error checking is performed and whether loss of connection is reported for connection-oriented protocols. You can specify what happens when a threshold is violated or a timeout or error condition occurs – sending an SNMP trap or an SNA NMVT alert. An SNA NMVT alert is an SNA message that conveys network management specific information.

Service Activator does not configure the target address for an SNMP trap or SNA NMVT alert. If you wish to send an SNMP trap or SNA NMVT alert, you must manually configure the target address details on the relevant devices.

If you are generating reports using InfoVista, note that InfoVista does not respond to SNMP traps sent as a result of a violated threshold, timeout or loss of connection.

**SAA Responder**

UDP Echo, Jitter and TCP Connect operations use non-native services to test the connection to a target device. The SAA Responder enables a router to respond to these operation types and must be configured on target devices. Service Activator automatically configures an SAA Responder on the relevant devices.

**SAA templates**

In Service Activator, an SAA template acts as a container for one or more SAA operations. A template may contain configuration details for different operation types, such as UDP and Jitter, or TCP Connect and Jitter. Once defined, a template can be applied to any number of VPNs.

For information on the limits that operate when defining an SAA template and applying SAA to a VPN, see *Creating an SAA template on page 26.*
Applications in Service Activator

Service Activator can apply SAA measurements to both MPLS and measurement-only VPNs.

MPLS VPN

When you apply SAA to a VPN, Service Activator configures operations between pairs of devices within the VPN. The number of operations configured depends on the VPN's topology and whether connections are tested in one or both directions. For more information, refer to the section Configuration considerations on page 8.

Applying SAA to an MPLS VPN is suitable only where the VPN has a small number of sites or has a hub and spoke topology. If the VPN topology is fully-meshed, Service Activator configures a full mesh of SAA operations. The number of operations configured on each device will therefore be one less than the number of sites in the VPN.

SAA is processor intensive and significantly affects device performance. The greater the number of sites in the VPN, the greater the number of operations configured on each device.

If you choose to deploy more than one SAA operation – for example, Jitter and TCP Connect – operations are configured for each type.

Measurement-only VPN

A measurement-only VPN allows you to apply SAA to a subset of devices in an MPLS VPN or to apply SAA without configuring an MPLS VPN. A measurement-only VPN can be used to:

- Group a subset of devices that belong to an MPLS VPN and apply SAA to those devices.

  This enables you to measure the connection between selected devices in an MPLS VPN, where the size of the VPN or its topology makes measuring all connections unfeasible.

  The following diagram illustrates this idea. All CE devices are members of an MPLS VPN. To reduce the number of operations configured, however, only the connection between CE1 and CE2 is measured. This is achieved by overlaying the MPLS VPN with a measurement-only VPN and applying SAA to the measurement-only VPN.
Group devices that do not participate in an MPLS VPN and apply SAA measurement.

This enables you to measure a connection between any pair of devices without the need to configure MPLS.
Configuring SAA

SAA Measurement is configured in Service Activator by defining an SAA template that holds one or more SAA operations and applying the template to a VPN.

This section covers the following topics:

- **Prerequisites**
- **Considerations**
- **Configuring SAA measurements for different VPN connections**
- **Configuring SAA measurement in Service Activator**

For more information on the Service Assurance Agent, refer to the Cisco documentation.

**Prerequisites**

Before configuring SAA in Service Activator, ensure that the following pre-requisites are implemented:

- The RTT-MON MIB v2.1.0 or later must be present on the devices that you want to configure an SAA operation.
- Given that memory problems are common, the following command should be run on CE devices before configuring the SAA probes:

  ```
  Global config mode
  rtr low-memory <#>
  ```

  The variable `<#>` should be set to 25% of the memory available in the system. For more information, refer to the Cisco documentation.

  - Also, if you intend to generate reports based on the SAA measurement, there may be additional pre-requisites associated with the reporting tool. For more information on the reporting tool’s measurement requirements, refer to the chapter for the reporting tool in this guide.
Considerations

This section highlights some of the deployment considerations associated with applying SAA using Service Activator.

- Check the device’s capabilities to determine which operation types are supported.
  
  To do this, right-click on the device object and select Properties from the pop-up menu. Then select the Capabilities property page.

- Make sure that you specify the correct device roles when defining data collection points – see Collectors on page 45.

- Ensure that you apply SAA templates only to the VPNs that you want to measure.

Configuring SAA measurements for different VPN connections

SAA measurement can be set up for the following VPN connections:

- Configuring SAA for a CE to CE connection
- Configuring SAA for a PE to PE connection using shadow routers
**Configuring SAA for a PE to CE connection using a shadow router**

It is possible to test the connection to an unmanaged device by modelling it as a virtual router. For more information, see *Unmanaged devices on page 14.*

A shadow router is a device dedicated to SAA measurements in a service provider’s Points of Presence (POP). Deploying shadow routers enables realistic PE to PE or PE to CE metrics to be gathered and protects measurement routers from customers. A PE device may have any number of shadow routers attached to it.

When configuring SAA measurements for different VPN connections, you may need to refer to the following procedures:

- For information on setting up a VPN, refer to the *Configuring VPN Services* guide.
- Refer to the section *Configuring SAA measurement in Service Activator on page 26* for procedures on how to:
  - Create an SAA template and apply it to a VPN.
  - Create a measurement-only VPN.
Configuring SAA for a CE to CE connection

By deploying SAA on CE devices, you can capture information about the complete end-to-end performance of a VPN service.

There are two alternatives for monitoring a CE to CE connection:

- Apply SAA to the MPLS VPN – this setup is feasible only in some scenarios (see sections Configuration considerations on page 8 and MPLS VPN on page 17)
- Overlap an MPLS VPN with a measurement-only VPN populated by a subset of the CE devices and apply SAA to the measurement-only VPN

You can only test a CE to CE connection where one or both CE devices are managed. An unmanaged device may be modelled as a virtual router and a one-way test performed. See Unmanaged devices on page 14.

To apply SAA to a CE to CE connection in an MPLS VPN

1. Create a Management VPN populated by the CE devices and the management station.
2. Create an MPLS VPN where each site consists of a CE device and a PE interface.
3. Apply an SAA template to the MPLS VPN.

**To apply SAA to a CE to CE connection in a measurement-only VPN**

References to the procedure listed below are provided on page 21.

1. Create a Management VPN populated by the CE devices and the management station.
2. Create an MPLS VPN where each site consists of a CE device and a PE interface.
3. Create a measurement-only VPN, populated by a subset of the MPLS VPN’s CE devices.

You do not need to associate the PE interface with the measurement-only VPN’s sites.

4. Apply an SAA template to the measurement-only VPN.

**Configuring SAA for a PE to PE connection using shadow routers**

PE to PE measurement provides information about the performance across the network backbone.

If you wish to collect PE to PE metrics, you must deploy shadow routers.

In Service Activator, a shadow router behaves in a similar way to a CE device – it may be associated with a site and have a VRF table associated with it, a protocol may be specified for connection to the PE device, and so on.

Service Activator includes a system-defined Shadow role which must be assigned to the shadow devices. For information on assigning a role to a device, see the Network Discovery and Basic Setup guide.
The PE device to which a shadow router is attached may participate in more than one MPLS VPN, where each VPN belongs to a different customer. If you wish to view performance metrics per customer in Service Activator using an integrated reporting tool, you must create a measurement-only VPN for each customer. Populate the measurement-only VPN with the relevant shadow routers and apply SAA to the VPN.

**To apply SAA to a PE to PE connection using shadow routers**

References to the procedure listed below are provided on page 21.

1. Create a Management VPN populated by the shadow routers and the management station.
2. Create an MPLS VPN where each site is a shadow router and a PE interface.
3. If you wish to group a subset of shadow routers by customer, create a measurement-only VPN populated by the relevant shadow routers.
4. Apply an SAA template to the MPLS VPN and, if created, each measurement-only VPN.
Configuring SAA for a PE to CE connection using a shadow router

PE-CE metrics provide information about the connection between the service provider's POP and a customer site. A shadow router must be deployed at the PE device.

If the CE device is not managed it is possible to test the connection to it by modelling the device as a virtual router. See Unmanaged devices on page 14.

To apply SAA to a PE to CE connection using a shadow router

For information on setting up a VPN, refer to the Configuring VPN Services guide.
1. Create a Management VPN that contains the shadow routers and the management station and, optionally, the CE devices. Alternatively, you can create two Management VPNs – one for the shadow routers and one for the CE devices.

2. Create an MPLS VPN where each site is a CE device and a PE interface.

3. Create a measurement-only VPN populated by a shadow router attached to one of the MPLS VPN’s PE devices and one or more of the CE devices attached to the PE device.

4. Apply an SAA template to the measurement-only VPN.

**Configuring SAA measurement in Service Activator**

This section covers the following procedures:

- *Creating an SAA template*
- *Adding an SAA operation to a template*
- *Creating a measurement-only VPN*
- *Applying an SAA template to a VPN*

### Creating an SAA template

By default, an SAA template has:

- A maximum of 1024 devices may participate in the VPN
- A maximum of 4 Type of Service (ToS) values may be used within the VPN
- A maximum of 4 operation types may be applied to the VPN

However, you can adjust these VPN-specific limits by specifying the amount of space that Service Activator allocates to store device, ToS value or operation type data.
Note: For complete dialog box and property page descriptions, refer to the Online Help.

By adjusting these values you can increase the number of devices within the VPN, or the number of operations or operation types that can be applied.

Note that the values specified for Device ID Bits, TOS Bits, and SAA Bits must add up to exactly 14. Therefore, if you choose to increase the number of bits allocated to the device ID, you must reduce either the number of bits allocated to the ToS value or the SAA operation type.

If you wish to apply more than one operation to a single peer-to-peer connection in a VPN, you must ensure that either the ToS value or the operation type differs between the operations. This is because Service Activator bases each operation’s unique RTR (Round Time Response) number on the ToS value and operation type (in combination with other types).
**Note:** For complete dialog box and property page descriptions, refer to the Online Help.

**To create an SAA template**

1. On the **System** tab, right-click on the **SAA Templates** folder and select **Add SAA Template** from the pop-up menu.
   
   The **SAA Template** dialog box opens.

2. Specify values including **Name**, **Details**, **Device ID Bits**, **TOS Bits**, and **SAA Bits**.

3. Click **OK** and commit the transaction.

**Adding an SAA operation to a template**

**Note:** You can only configure SAA operations to test the connection between two devices that are in the same customers VPNs.

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

**To add an SAA operation to a template**

1. On the **System** tab, open the **SAA Templates** folder. Right-click on the SAA template and select **Add SAA Operation** from the pop-up menu.

   The **SAA Operation** dialog box opens.

2. On the **SAA Operation** property page, specify values including **Type**, **Source port**, **Destination Port**, **Request size**, **ToS**, **Control**, **Packets in sequence**, **Inter-packet interval**, **History lives kept**, **History filter**, and **History buckets**.

3. Select the **Threshold** property page.

4. Specify values including **Type**, **Rising**, **Falling**, **Repetitions**, **X**, **Y**, **Average**, and **Action**.

5. Select the **Other** property page.

6. Specify values including **Period**, **Timeout**, **Lifetime**, **Half duplex**, **Full duplex**, **Error checking**, **Connect checking**, and **Timeout checking**.
Creating a measurement-only VPN

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

You can use SAA to measure the point-to-point connection between any pair of devices without configuring an MPLS VPN.

**To apply SAA without configuring an MPLS VPN**

1. Apply the appropriate roles to the relevant devices.
   You can associate devices tagged with the Access or Shadow role with sites in a VPN.
2. Associate each device with a site.
   Note that you do not need to associate an interface with the site.
3. If needed, specify the device’s SAA destination and source measurement address:
   - Open the **Device** dialog box.
   - Select the **SAA** property page.
   - To configure Measurement Probe Destination Address, select **Use device management address** to use the device management IP address as set up on the **Device property** page, as the SAA destination measurement address. Alternatively, select **Use** and from the dropdown list, choose one of the interfaces previously discovered on the device.
   - To configure Measurement Probe Source Address, Select **Use Measurement Probe Destination Address** to use the Measurement Probe Destination Address as the Measurement probe source address in the device. Alternatively, Select **Use Device Management Address** to use the Device Management Address as the Measurement probe source address in the device. Alternatively, select **Use** and from the drop-down menu select an address from the other available addresses in the device as the Measurement probe source address. Select **Not configured** to avoid configuring the Measurement probe source address in the device.
   - Select **Use** and select an IP address from the drop-down list of interfaces that have previously been discovered by the device.
4. For each site:
   - Right-click on the site and select **Properties** from the pop-up menu to open the **Site** dialog box.
   - Select the **Connectivity** property page.
   - In the **Routing Type** field, select **None**.
5. Create a VPN by right-clicking on the VPN folder and selecting **Add MPLS VPN** from the pop-up menu.

6. Ensure that the **Use MPLS** checkbox on the **VPN** property page of the **VPN** dialog box is **cleared** to effectively create a measurement-only VPN.

7. Add the relevant sites to the VPN.

**Applying an SAA template to a VPN**

**To apply an SAA template to a VPN**

1. On the **Service** tab, select the relevant VPN and select **Properties** from the VPN’s pop-up menu.

   **VPNs** are listed in the **VPNs** folder beneath the relevant customer.

2. Select the **Measurement** property page.

3. In the **SAA Measurement** field, select an SAA template from the pull-down list.

---

If a site is a member of more than one VPN to which SAA measurement is applied, there may potentially be one operation per VPN configured on the device, depending on the VPN’s topology and whether one or two-way tests are performed. For information on the factors that affect the number of operations configured within a VPN, refer to the section *SAA operations on page 8.*
Chapter 3

MIB-based and NetFlow measurements

This chapter covers the following topics:

- Overview of MIB and NetFlow-based measurements
- CAR MIB
- NetFlow
- Configuring measurement types in Service Activator
Overview of MIB and NetFlow-based measurements

Service Activator allows you to measure performance based on various information sources other than SAA probes. You can apply CAR MIB, MIB2 and NetFlow measurement types to any policy target, including a customer, VPN, device or VC endpoint.

Measurement is set up by applying a measurement parameter to a policy target. Measurement types are inherited in the same way as rules and PHB groups. Device and interface roles provide fine-grained control over the application of NetFlow and MIB-based measurements.

Once the measurement parameters are configured, a third party reporting tool can then set up to collect and process the performance data. For more information, refer to the following chapters:

- Chapter 4, Setting Up Service Activator for Integration
- Chapter 6, InfoVista Integration
- Chapter 7, Micromuse Integration

A third party reporting tool, such as InfoVista, can then be used to view measurement-based reports.

CAR MIB

Committed Access Rate (CAR) is configured on router interfaces. It is used to limit the input and output traffic rates, control bandwidth usage and implement a selective IP entry policy.

CAR MIB reports provide a clear picture of the performance of each interface channel based on its CAR configuration. CAR MIB polls the following MIBs:

- CISCO-CAR-MIB
- CISCO-SMI-MIB

MIB2

MIB2 measurement polls variables defined in any MIB2 format MIB. The MIB and the variables polled depend on the reporting tool used to generate reports. For example, if InfoVista is used for reporting, a range of MIBs are queried, including the Cisco Memory Pool MIB, Cisco AAL5 MIB and the IF MIB.

Applying MIB2 measurement provides information about the basic state of the network – for example, load, availability, discards, broadcast rate. Monitoring MIB2
values can provide useful information for detecting escalating error conditions and to determine trends that aid in capacity planning.

There is no device configuration associated with MIB2 measurement.

**NetFlow**

NetFlow is a Cisco-specific feature that enables you to characterize and analyze an IP flow on an interface with minimal impact on router performance. Often used as a metering base for other applications, including accounting/billing, network planning, and marketing, NetFlow generates flow-based statistics per interface and the information produced is highly granular. NetFlow is often deployed at the PE interface within an MPLS VPN but can be applied at any point in the network.

Flow-based statistics are gathered on the router and stored in a cache. At intervals, the router exports its stored information in the form of NetFlow UDP datagrams to collector software – the software varies, depending on which reporting tool you wish to use (see Collectors on page 45. A range of UDP formats is supported – later versions minimize bandwidth usage by aggregating data before export from the device.

The NetFlow measurement parameter specifies which version of UDP to use for exporting flow data and any aggregation that should be applied before export. The device driver configures NetFlow on the device according to the parameters defined in Service Activator.

For information on Service Activator support for NetFlow by device and IOS, see the Cisco IOS Device Support Guide.

This section covers the following NetFlow topics:

- **Architecture and components**
- **What is a flow?**
- **UDP formats**
- **Aggregation**

**Architecture and components**

In any NetFlow deployment, collection software gathers exported flow data from monitored devices. Service Activator currently supports the following collection software:

- Vista Plug-in for NetFlow – for use with InfoVista Server
- Cisco’s NetFlow FlowCollector
We recommend that you install the collection software on a dedicated host machine. A number of collectors may be distributed throughout the network, with each collector gathering data from a subset of devices.

**What is a flow?**

A flow is a unidirectional stream of packets between a source and a destination. Both of these parameters are defined by a network-layer IP address and a transport-layer source and destination port number. A flow is identified by the combination of the following seven field values:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Layer 3 protocol type
- ToS byte
- Input logical interface (ifIndex)

Together, these key fields define a unique flow. Additional accounting fields appear in the flow and these vary depending on which UDP format version packets have been selected for export from the device (see *UDP formats on page 34*).

Flows are processed in a NetFlow cache. NetFlow creates a NetFlow cache entry that contains information for all active flows. Within the cache, a flow record is maintained for each active flow. Each flow record contains key fields that can be used for exporting to a collection device.

The router checks the cache once per second and expires the flow if one or more of the following conditions are met:

- Transport is completed
- The cache is full
- The inactive timer has expired after traffic inactivity for the specified number of seconds
- The active timer has expired after traffic activity for the specified number of minutes

**UDP formats**

NetFlow data is exported from the router as a UDP datagram in one of four formats: Version 1, Version 5, Version 7 or Version 8. The datagram consists of a header and one or more flow records.
Service Activator supports Version 1, Version 5 and Version 8 formats.

- Version 1 was the first released version and should only be used if you need to support a legacy collection system. Typically, records are exported when the NetFlow cache is full or the flow or the timer has expired.
- Version 5 is based on version 1 with the addition of BGP AS information and flow sequence numbers.
- Version 8 supports router-based aggregation of flows in additional ‘aggregation’ caches. As flows expire from the main cache, they are added to each enabled aggregation cache. This format allows for export datagrams to contain a subset of Version 5 export data, which is valid for a particular aggregation cache scheme.

Service Activator indicates which UDP versions are supported by the device in the device capabilities. For information on retrieving and viewing device capabilities, see the Network Discovery and Basic Setup guide.

For detailed information on which UDP versions are supported by a particular device, consult the Cisco documentation.

Aggregation

Aggregating NetFlow data on the router before exporting in Version 8 format provides a number of benefits:

- It reduces the bandwidth required between the router and the machine that collects the exported NetFlow data.
- It reduces the amount of flows sent to the collector software for processing.
- It improves the scalability of high-flow-per-second routers, such as the Cisco 7500 series routers.

A range of aggregation schemes is available and you can configure each aggregation scheme with a cache size, timeout value, export destination IP address and export destination UDP port.

As flows expire in the main NetFlow cache, relevant information is extracted from the expired flow and the relevant flow entry in an aggregation cache is updated. One or more aggregation caches may be maintained, depending on the number of aggregation schemes selected for export. Each aggregation cache contains different field combinations that determine which data flows are grouped.
In Service Activator you can implement only one aggregation scheme on a device. The aggregation cache parameters – size, timeout value, and so on – are the same as those defined for the main NetFlow cache.

Data is always exported from an aggregation cache in v8 format.

The following aggregation schemes are available:

- **AS aggregation** – generates AS-to-AS traffic flow data. The scheme groups data flows by source BGP AS, destination BGP AS, input interface and output interface.

- **Destination prefix** – supports examination of flows by destination. The scheme groups data flows by destination prefix, destination prefix mask, destination BGP AS and output interface.

- **Prefix aggregation** – supports examination of flows by source and destination. The scheme groups data flows by source prefix, destination prefix, source prefix mask, destination prefix mask, source BGP AS, destination BGP AS, input interface and output interface.

- **Protocol port aggregation** – supports examination of network usage by traffic type. The scheme groups data flows by IP protocol, source port number and destination port number when applicable.

- **Source prefix aggregation** – supports examination of flows by source. The scheme groups data flows by source prefix, source prefix mask, source BGP AS and input interface.

If you are generating NetFlow reports in Service Activator, note that some aggregation schemes may not be supported by the integrated reporting software. Consult the relevant SLA monitoring guide or third-party documentation for details of supported schemes.
Configuring measurement types in Service Activator

Considerations

Depending on the type of measurement being configured and its intended use, any one of the following considerations should be taken into account when configuring a measurement parameter in Service Activator:

- Reports
- Policy target levels
- CAR MIBs
- NetFlow

Reports

If you intend to generate reports based on these measurement types, there may be additional requirements associated with the reporting tool. For more information on the reporting tool’s measurement requirements, refer to the chapter for the reporting tool in this guide.

Policy target levels

Measurement is automatically inherited to policy targets at a lower level, down to the device level. Below this level, device and interface roles specify where measurement is inherited to.

When you configure a measurement parameter, you must specify at which level measurement is activated, and this varies depending on the measurement type:

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Device level</th>
<th>Interface level</th>
<th>Sub-interface level</th>
<th>PVC level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR MIB</td>
<td>N/A</td>
<td>Select one or more options (mandatory)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIB2</td>
<td></td>
<td>Select one or more options (mandatory)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetFlow</td>
<td>Mandatory</td>
<td>Select one or more options (mandatory)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NetFlow is generally applied to Access interfaces on PE devices – that is, as close to the customer site as possible.
CAR MIBs

If you want to apply the CAR MIB measurement on a Service Activator object, the following MIBs must be present on the devices where measurement is implemented:

- CISCO-CAR-MIB
- CISCO-SMI

NetFlow

When applying NetFlow measurement, you can specify in which version of UDP packets are exported from the device to the NetFlow collector. Later versions support aggregation of flows before export, minimizing bandwidth usage. For more information on UDP and aggregation, see UDP formats on page 34 and Aggregation on page 35.

If you are monitoring NetFlow measurements using an integrated reporting tool, check which aggregation schemes are supported. A tool may not support all of the available aggregation types. For information on supported aggregation types, consult the reporting tool’s documentation.

The following limitations apply to the NetFlow measurement parameter:

- NetFlow data cannot be exported to more than one collection system.
- When configuring NetFlow measurement, you cannot configure multiple aggregation schemes on a device.
Check the device’s capabilities – device-level capabilities indicate support for NetFlow.

To do this, right-click on the device object and select Properties from the pop-up menu. Then select the Capabilities property page.

We recommend you configure a dedicated interface between the router and the collection device to prevent loss of packets.

The collection device should be a dedicated host machine.

Applying NetFlow or MIB-based measurements to a policy target

Measurement Parameters can be applied to policy targets including Networks, Devices, Interfaces, and Sub-interfaces.

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

To apply NetFlow or MIB-based measurements to a policy target

1. From the policy target’s pop-up menu, select Add Measurement Parameter.

   The Measurement Parameter dialog box opens.

2. On the Measurement property page:
In the **Name** field, enter an identifier for the measurement parameter.

Select a **Driver type** from the drop-down or select **Any** to apply measurement to all vendor devices.
As needed, specify values including **Collect class-based QoS MIB**, **Collect CAR QoS MIB statistics**, **Collect MIB2 statistics**, **Collect Juniper CoS MIB Statistics**, **Collect NetFlow statistics**, **Device level**, **Interface level**, **Sub-interface**, **PVC**, and **Netflow properties**.

If you select Netflow properties, specify values for **Netflow** fields including **Netflow**, **Cache**, **Aggregation**, **Active timeout**, **Inactive timeout**.

3. On the **Role** property page, specify the device and interface roles to which measurement applies.
Chapter 4

Setting Up Service Activator for Integration

This chapter covers the following topics:

- External systems
- Collectors
External systems

Service Activator’s SLA monitoring capability is provided through integrated third party reporting software. Before you configure Service Activator to support the third party reporting tool, it is important to understand the reporting tool’s architecture and functionality.

For more information on integrating with InfoVista, refer to Chapter 6, *InfoVista Integration*.

For more information on integrating with Micromuse, refer to Chapter 7, *Micromuse Integration*.

To enable Service Activator support for third party reporting software, you must model the reporting software’s components as ‘external systems’ through the Service Activator user interface. The type and number of components to be modelled depends on which reporting software you are using.

**Types**

Service Activator supports four types of external systems used for reporting purposes:

- InfoVista Server
- Vista Plug-in for NetFlow
- NetFlow FlowCollector
- Micromuse Netcool

As shown in the table below, you should select the external system type based on the reporting tool you will use and the statistics type you want to collect.

<table>
<thead>
<tr>
<th>External system type</th>
<th>Reporting tool</th>
<th>Statistics type</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoVista Server</td>
<td>InfoVista</td>
<td>■ SAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MIB-based data</td>
</tr>
<tr>
<td>Vista Plug-in for NetFlow</td>
<td>InfoVista</td>
<td>■ NetFlow</td>
</tr>
</tbody>
</table>
Creating an external system

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

1. On the System tab, select the External Systems folder and select Add External System from the folder’s pop-up menu.
   The External System dialog box opens.
2. Select the External System property page and specify values including Name, Remarks, Type, Primary IP, Port, Secondary IP, Port and URL.
3. Select the Security property page, and select values including Username and Login password. These apply to Infovista Server.

Collectors

One or more collectors must be added to each device that you want to monitor. The collector links the device to the external system that performs data collection and aggregation. In Service Activator, the link between the device and the external system is called a collector parameter.

A collector behaves in a similar way to Service Activator’s PHB groups in that it may be applied to any number of policy targets, and is inherited through Service Activator’s policy inheritance hierarchy. Device and interface roles specify the points from which data is collected. For more information on defining and using roles, see the Network Discovery and Basic Setup guide.
Roles and measurement parameters

The roles that you associate with a collector depend on the type of data to be collected and from which devices.

For NetFlow and MIB-based measurement, the roles should be the same as those associated with at least one measurement parameter.

SAA is configured at device level and so only the device role that you associate with a collector is significant.

Though only device role is significant, you must also specify an interface role when you apply a collector that collects SAA data. We recommend you use the system-defined Any Role.

SAA measurement can be applied to an MPLS VPN or to a measurement-only VPN.

If SAA is applied to devices that participate in an MPLS VPN, the device role you associate with a collector depends on which connection you are measuring:

- For CE to CE, the device role must be Access or Shadow
It is possible to monitor the CE to CE connection by applying SAA to the CE device, or by deploying shadow routers.

- For PE to PE, the device role is Shadow
  Shadow routers must always be deployed when monitoring the PE to PE connection.
- For PE to CE, data must be collected from devices tagged with two different roles – that is, Shadow and Access.
  It is not possible to associate two device roles with a collector. We recommend you assign an additional user-defined role to the relevant Shadow and Access devices and associate this role with the collector.

If you have applied SAA to devices that do not participate in an MPLS VPN, the device role must be Access or Shadow, depending on which role you have assigned to the monitored devices.

If you are collecting measurement data for a VPN and generating reports using InfoVista, we recommend you assign a single collector to the VPN. If you distribute collection between multiple collectors, VPN-level reports will not be available as there is currently no mechanism for aggregating VPN data from multiple collectors.

If you apply a collector at domain level, the collector is inherited through Service Activator’s logical inheritance line – that is, via customer, VPN and site. Do not apply the collector at domain level unless you have created at least one customer and VPN, populated by the devices from which you wish to collect data.

When you have associated a collector with a policy target, the measurement component exports the relevant topology details to the collector. For example, if the collector is associated with a VPN, the component exports topology details for all devices participating in the VPN. If the topology changes, the measurement component exports the updated details to the collector.
Creating a collector in Service Activator

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

**Note:** If you are creating more than one collector, and if you are creating a Vista Plug-in for Netflow, ensure that you create the plug-in collector first.

1. From a policy target’s pop-up menu, select Add Collector Parameter. The Collector Parameter dialog box opens.

2. On the Collector property page, specify values including Name, Collector, and Driver Type.

   Note that measurement data may currently be collected from Cisco devices only.

3. On the Role property page, specify the device and interface roles to which the collector applies.

   You must specify both a device and an interface role for the collector to be applied. If necessary, you can use the system-defined Any Role as the device or interface role which effectively acts as a wildcard.
Chapter 5

Generic Exporter

This chapter describes the Service Activator Generic Exporter. It covers the following topics:

- Integration with Service Activator
- Customizing integration
- Error Reporting
Integration with Service Activator

Integration between Service Activator and the Generic Exporter occurs in the following way:

- Service Activator’s Generic Exporter reads an XML export file (TopologyExporterConfig.xml) that identifies which data to collect for external reporting.
- This file is used to guide data collection from the Service Activator object model through the OSS Integration Manager (OIM) to create TopologyExporter.xml, which contains raw object model data.
- An XSL file (TopologyExporter.xsl) is then used to filter the TopologyExporter.xml file to create an export file.
**Key components**

The following table provides a brief description of the key components used with the Generic Exporter.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TopologyExporterConfig.xml</td>
<td>When the Generic Exporter is invoked, it reads this configuration file which includes variables indicating:</td>
</tr>
<tr>
<td></td>
<td>- Service Activator server to connect to</td>
</tr>
<tr>
<td></td>
<td>- name of output XML file to be used when extracting data from Service Activator</td>
</tr>
<tr>
<td></td>
<td>- transformation directives indicating the number of XSLs to be called and what sequence they are to be called in</td>
</tr>
<tr>
<td></td>
<td>- root tag to be used in XML output</td>
</tr>
<tr>
<td></td>
<td>- rules specifying which objects, attributes and children are to be extracted and which are to be filtered out from the Service Activator object model</td>
</tr>
<tr>
<td>TopologyExporter.xml</td>
<td>Object model file filtered by TopologyExporter.xsl to produce final output file TopologyExporter.txt</td>
</tr>
<tr>
<td>TopologyExporter.xsl</td>
<td>XML schema file used to transform TopologyExporter.xml. Executes filtering and formatting commands to localize data to meet output requirements.</td>
</tr>
<tr>
<td>TopologyExporter.txt</td>
<td>Filtered object model information to be transmitted via ftp to the Vista Provisioner.</td>
</tr>
<tr>
<td>Archive files</td>
<td>Whenever Service Activator creates a new TopologyExporter.txt file, the previous one is archived with the date and time of creation incorporated into the filename.</td>
</tr>
</tbody>
</table>

These files can be customized to meet the requirements of the third party components you are integrating Service Activator data with. For more information, refer to the section *Customizing integration on page 52.*
TopologyExporter.xml Filtering

The Generic Exporter filters the object model to process only the entities described in the table below. It exports an XML file containing a filtered version of the object model for both customer and network topology trees:

- Policy/Domain*/Customer*/VPN*/, Site*, ParameterSetInstance*
- Policy/Domain*/Network*/Device*, ParameterSetInstance*

Customizing integration

A service provider can customize the Generic Exporter integration between Service Activator and a third party component by modifying the following files:

- TopologyExporterConfig.xml
- TopologyExporter.xsl
- TopologyExporter.txt

TopologyExporterConfig.xml

The content of the TopologyExporter.xml file is controlled by the TopologyExporterConfig.xml file. The Generic Exporter utility uses the TopologyExporterConfig.xml file to generate the TopologyExporter.txt file.

As needed, a service provider can modify the TopologyExporterConfig.xml file to customize the topology information exported.

The TopologyExporterConfig.xml file is located in the following directory:

<ServiceActivatorHome>/modules/TopologyExporterIntegrationModule/

For a description of the fields included in the TopologyExporterConfig.xml file, refer to Appendix A, TopologyExporterConfig.xml Fields.

TopologyExporter.xsl

The TopologyExporter.xsl file applies formatting and filter rules to the TopologyExporter.xml file to generate an TopologyExporter.txt file. If the TopologyExporterConfig.xml file is modified, the filters in the TopologyExporter.xsl file should also be modified accordingly.

The TopologyExporter.xsl file is located in the following directory:

<ServiceActivatorHome>/modules/TopologyExporterIntegrationModule/
The default transform file, TopologyExporter.xsl, applies the following filters to the TopologyExporter.xml file to generate the output TopologyExporter.txt file:

**Domain(s)**
- **Network(s)**
  - When a device is nested in multiple networks, only the immediate parent of the device appears in the output network folder
- **Device(s)**
  - Must be managed
  - Must not be virtual
  - **Interface(s)/Sub-Interface(s)**
    - Must have interface and parent device roles assigned
    - Must have measurement parameter MIB2 selected
    - Must have measurement parameter Juniper CoS selected for Juniper CoS data export
  - **PVC(s)**
    - Must have measurement parameter MIB2 selected

**TopologyExporter.txt**

The TopologyExporter.txt file is derived from the TopologyExporter.xml file based on format instructions provided by the TopologyExporter.xsl file.

The TopologyExporter.txt file is deposited in the following directory:

```
<ServiceActivatorHome>/modules/TopologyExporterIntegrationModule/
```

**Invoking the Generic Exporter**

The XML data export function of the Generic Exporter is invoked using the TopologyExporter.sh script, which resides on the Service Activator server on which the Integration Manager is installed.

It can be invoked from a manually entered command, or set up to run automatically at appropriate intervals using a UNIX cron job.

The command has the following parameters:

```
<Service ActivatorHome>/modules/bin/TopologyExporter.sh <username> <password> [ftp_server_name] [ftp_user_name] [ftp_pwd] [remote_directory_name]
```

Required parameters:
- **username** user ID used to access Service Activator
password password for user ID
Optional parameters:
ftp_server_name destination ftp server to send output txt to
ftp_user_name user ID for destination ftp server
ftp_pwd password for ftp user ID
remote_directory_name destination ftp server directory in which to place TopologyExporter.txt

**Note:** To access the InfoVista Integration Module functionality in the Service Activator GUI, ensure that the **Allow concurrent logins** option is enabled for the Service Activator user. For more information on how to do this, refer to the *Service Activator Administrator’s Guide*.

### Configuring the Generic Exporter SNMP community string

The OSS Integration Manager does not pass the device SNMP write community string in open text (acting as a password) and therefore it is not available for TopologyExporter.xsl to process. A hard coded value of *private* is used for this value.

If your devices are configured to use a different value for the SNMP write community string, you can replace the string *private* by editing the TopologyExporter.xsl file.

If your devices are all configured with individual (i.e. different) write community strings, you would have to implement a look up method which integrates with the TopologyExporter.xsl file. This is beyond the scope of this document.

### Error Reporting

During the filtering of the TopologyExporter.xml file by the TopologyExporter.xsl file, if there is missing data or incorrect data, a message will be output to the Error console. The error will also be included in the XML output file as an XML comment. This allows you to confirm if any of the required data is missing or if the data is incorrectly filled.
Chapter 6

InfoVista Integration

This chapter describes how Service Activator integrates with InfoVista. It assumes a working knowledge of InfoVista concepts. For more information on InfoVista, refer to the InfoVista documentation.

This chapter covers the following topics:

- Integration with Service Activator
- Service Activator objects and InfoVista Vistas
- Customizing InfoVista integration
- Reports
- Specifying additional object information
- Deployment
- Error Reporting
Integration with Service Activator

Integration between Service Activator and InfoVista occurs in the following way:

- Service Activator’s Generic Exporter, which is automatically installed with the InfoVista Integration Module, reads an XML export (InfoVistaExportConfig.xml) file that identifies which data to collect for the InfoVista reports.
- This file is used to guide data collection from the Service Activator object model through the OSS Integration Manager (OIM) to create InfoVistaExport.xml, which contains raw object model data.
- An InfoVista.xsl is then used to filter InfoVistaExport.xml to create a comma delimited export file (InfoVista.txt) containing the object model data as required by your existing InfoVista reports.
- The Service Activator provided def.txt file acts as a data dictionary against the entries in the InfoVista.txt file.
- InfoVista.txt and def.txt are sent via FTP to the InfoVista server.
Key components

The following table provides a brief description of the key components used to enable InfoVista reporting with Service Activator.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoVista Integration Module</td>
<td>A Service Activator module that includes the Generic Exporter utility and several integration files. The Generic Exporter exports the object model and converts it to an InfoVista compliant format ready for import by the InfoVista Server.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For details on installing the InfoVista Integration module, refer to the Setup Guide.</td>
</tr>
<tr>
<td>InfoVista Server</td>
<td>Uses SNMP to poll and collect SAA and MIB-based data from the devices and NetFlow data from the Vista Plug-ins for NetFlow. Service Activator is used to allocate devices to InfoVista servers. Each InfoVista Server maintains a local object model that reflects the devices assigned to it and the measurement applied.</td>
</tr>
<tr>
<td>Vista Plug-in for Netflow</td>
<td>Collects and aggregates v1, v5 and v8 NetFlow UDP datagrams from NetFlow-enabled devices and exports the data to an InfoVista Server. The Service Activator user interface is used to allocate a device to an InfoVista server. The component is only required when the NetFlow measurement is applied. Service Activator is used to allocate devices to Vista Plug-ins for NetFlow.</td>
</tr>
<tr>
<td>Vista Provisioner</td>
<td>An InfoVista integration utility that enables users to automate a number of key tasks in the administration of an InfoVista Server.</td>
</tr>
</tbody>
</table>
## InfoVista integration files

The following table provides a brief description of the key components used to enable Service Activator integration with InfoVista.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>infoVistaExport.sh</td>
<td>Script residing on the Service Activator server on which the Integration Manager is installed, used to invoke the XML export process.</td>
</tr>
<tr>
<td>InfoVistaExportConfig.xml</td>
<td>When the Generic Exporter is invoked, it reads this configuration file which includes variables indicating:</td>
</tr>
<tr>
<td></td>
<td>- Service Activator server to connect to</td>
</tr>
<tr>
<td></td>
<td>- name of output XML file to be used when extracting data from Service Activator</td>
</tr>
<tr>
<td></td>
<td>- transformation directives indicating the number of XSLs to be called and what sequence they are to be called in</td>
</tr>
<tr>
<td></td>
<td>- root tag to be used in XML output</td>
</tr>
<tr>
<td></td>
<td>- rules specifying which objects, attributes and children are to be extracted and which are to be filtered out from the Service Activator object model</td>
</tr>
<tr>
<td>InfoVista.xml</td>
<td>Object model file filtered by InfoVista.xsl to produce final output file InfoVista.txt</td>
</tr>
<tr>
<td>InfoVista.xsl</td>
<td>XML schema file used to transform InfoVistaExport.xml into a format compliant with InfoVista. Executes filtering and formatting commands to localize specific InfoVista requirements.</td>
</tr>
<tr>
<td>InfoVista.txt</td>
<td>Filtered object model information to be transmitted via ftp to the Vista Provisioner.</td>
</tr>
</tbody>
</table>
These files can be customized as needed. For more information, refer to the section Customizing InfoVista integration on page 61.

Once the InfoVista.txt file is imported into the Vista Provisioner, the InfoVista Server polls the assigned devices for SAA and MIB-based statistics using SNMP. For NetFlow statistics, the InfoVista Server polls the Vista Plug-in for NetFlow. The Vista Plug-in for NetFlow collects the NetFlow statistics directly from the devices.

### Service Activator objects and InfoVista Vistas

Both Service Activator and InfoVista use object models:

- Service Activator models system components, the topology of the network and the policy that has been applied to the network.
  
  This information is modelled in Service Activator’s internal object model and made available to third-party applications, including InfoVista, via the OSS Integration Manager (OIM) or the Generic Exporter.

- InfoVista models the elements necessary to generate the relevant reports for the correct policy targets – for example, Frame Relay and ATM VC endpoints, WAN interfaces and NetFlow interfaces.

  A class of objects is referred to as a Vista, while an object within a class is referred to as a Vista Instance, or Instance.

### Object model synchronization

The object model between Service Activator and InfoVista is synchronized when the InfoVista.txt file generated by Service Activator’s Generic Exporter is imported by the Vista Provisioner.

Synchronization is a manual action that should occur each time monitored network or service elements are added, deleted or re-configured in Service Activator. Synchronization of the object model can also be scheduled during regular maintenance windows. For more information on how to perform this procedure, refer to Importing Service Activator integration files into InfoVista on page 68.
**InfoVista object mapping**

The following table shows how Service Activator objects are mapped to InfoVista Vistas:

<table>
<thead>
<tr>
<th>Service Activator Object</th>
<th>InfoVista Vista or Vista Instance</th>
<th>InfoVista Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Router</td>
<td>Basic</td>
<td></td>
</tr>
<tr>
<td>Device(s)</td>
<td>Router Group</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>Interface, Sub-interface</td>
<td>WAN_IF</td>
<td>Proxy</td>
<td></td>
</tr>
<tr>
<td>Interface(s), Sub-interface(s)</td>
<td>WAN_IF Group</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>Interface, Sub-interface</td>
<td>LAN_IF</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>Interface(s), Sub-interface(s)</td>
<td>LAN_IF Group</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>PVC</td>
<td>ATM_PVC</td>
<td>Proxy</td>
<td></td>
</tr>
<tr>
<td>PVC</td>
<td>FR_PVC</td>
<td>Proxy</td>
<td></td>
</tr>
<tr>
<td>External System</td>
<td>NFPLUGIN</td>
<td>Basic</td>
<td></td>
</tr>
<tr>
<td>External System, Collector IV Server, Measurement Parameter: IV Plugin</td>
<td>NFROUTER</td>
<td>Proxy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NFINTERFACE</td>
<td>Proxy</td>
<td></td>
</tr>
<tr>
<td>SAA Operation</td>
<td>SAA_JITTER</td>
<td>Proxy</td>
<td>Jitter</td>
</tr>
<tr>
<td>SAA Operation</td>
<td>SAA_RTT</td>
<td>Proxy</td>
<td>TCP, UDP, ECHO</td>
</tr>
<tr>
<td>SAA Operation(s)</td>
<td>END2END_GROUP</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>SAA Operation(s)</td>
<td>SAAGENT_GROUP</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>CAR Policy</td>
<td>VPN-IF</td>
<td>Proxy</td>
<td>CAR</td>
</tr>
</tbody>
</table>
Interface mapping depends on both the device and interface roles assigned in Service Activator.

<table>
<thead>
<tr>
<th>Service Activator</th>
<th>InfoVista Interface Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Role</td>
<td>Interface Role</td>
</tr>
<tr>
<td>Access</td>
<td>Local</td>
</tr>
<tr>
<td>Access</td>
<td>Access</td>
</tr>
<tr>
<td>Gateway</td>
<td>Access</td>
</tr>
<tr>
<td>Gateway</td>
<td>Core</td>
</tr>
<tr>
<td>Core</td>
<td>Core</td>
</tr>
<tr>
<td>All other</td>
<td>All other</td>
</tr>
</tbody>
</table>

**InfoVista.xml Filtering**

The InfoVista Generic Exporter filters the object model to process only the entities described in the table below. It exports XML file containing a filtered version of the object model for both customer and network topology trees:

- Policy/Domain*/Customer*/VPN*/, Site*, ParameterSetInstance*
- Policy/Domain*/Network*/Device*, ParameterSetInstance*

InfoVista reports must exist against entities in the topology tree for network reporting.

**Customizing InfoVista integration**

A service provider can customize the integration between Service Activator and InfoVista by modifying the following files:

- InfoVistaExportConfig.xml
- InfoVista.xsl
- infoVista.txt
- def.txt
- Report.pl
InfoVistaExportConfig.xml


The default InfoVistaExportConfig.xml file is set up to generate an InfoVista.xml file that lists all devices in Service Activator that are assigned a collector.

As needed, a service provider can modify the InfoVistaExportConfig.xml file to customize the topology information provided to InfoVista.

The InfoVistaExportConfig.xml file is located in the following directory:

<ServiceActivatorHome>/modules/Config

For a description of the fields included in the InfoVistaExportConfig.xml file, refer to Appendix A, TopologyExporterConfig.xml Fields.

InfoVista.xsl

The InfoVista.xsl file applies formatting and filter rules to the Infovista.xml file to generate an InfoVista.text file that can be imported by the Vista Provisioner. If the InfoVistaExportConfig.xml file is modified, the filters in the Infovista.xsl file should also be modified accordingly.

The InfoVista.xsl file is located in the following directory:

<ServiceActivatorHome>/modules/Config

The default InfoVista.xsl file applies the following filters to the InfoVistaExport.xml file to generate the InfoVista.text file:

**Domain(s)**

- **Network(s)**
  - When a device is nested in multiple networks, only the immediate parent of the device appears in the InfoVista network folder

- **Device(s)**
  - Must be Cisco or Juniper
  - Must be managed
  - Must not be virtual
  - Must have either InfoVista Server or Vista Plug-in for NetFlow associated

- **Interface(s)/Sub-Interface(s)**
  - Must have interface and parent device roles assigned
  - Must have measurement parameter MIB2 selected
— Must have measurement parameter Juniper CoS selected for Juniper CoS data export

— PVC(s)
— Must have measurement parameter MIB2 selected

infoVista.txt

The infoVista.txt file is derived from the InfoVista.xml file based on format instructions provided by the InfoVista.xsl file.

The infoVista.txt file is deposited in the following directory:

<ServiceActivatorHome>/modules/InfoVistaIntegrationModule

When the error message “SAA with ID=WARNING” message appears in the infoVista.txt file, this indicates that the probe ID could not be evaluated for the following reasons:

- Destination router is not monitored (a collector is not assigned to it)
- Source router is virtual and cannot be monitored with probes
- Source router concrete is not in the installed state or configuration has failed on the router

If you have duplicate routers in the same network, an "instance name duplicate" error is triggered while exporting the infoVista.txt file to the InfoVista server. Ensure there are no duplicate routers in the same network before invoking the InfoVista topology exporter.

Alternatively, the infoVista.txt file can be edited to remove duplicate entries before export to the InfoVista server.

def.txt

The def.txt file acts as a data dictionary that validates the infoVista.txt file. The default supplied def.txt file must be updated using a text editor when changes are made to either the InfoVistaExportConfig.xml or InfoVista.xsl files.

The def.txt file is located in the following directory:

<ServiceActivatorHome>/modules/InfoVistaIntegrationModule
Report.pl

Service Activator is packaged with a default report.pl file. This file determines the folder structure used by InfoVista for storing reports and can be customized as needed.

The default report.pl file specifies a folder structure that mirrors Service Activator’s policy inheritance hierarchy. At the top of the structure are domain-level report folders. Below this, report folders are held in two branches – one reflecting the physical inventory of network devices and the other reflecting customer configurations.

The Report.pl file is located in the following directory:

<ServiceActivatorHome>/ExplorerScripts/TopologyExporter/IVProvCfg

The default report.pl file specifies the following folder structure:

Domain Instance
  Network Instance
    Router Instance
      Router - individual reports
      SAA Folder
        SAA Instance Folder - individual reports
  WAN Folder
    WAN Instance Folder - individual reports
  LAN Folder
    LAN Instance Folder - individual reports
    ATM-PVC Folder - individual reports
    FR PVC Folder - individual reports
  NetFlow Folder
    Router Instance Folder
      NF Router Instance Folder - individual
    NF Interfaces Instance Folder - individual
  reports
  interfaces
    Overview LAN Folder - group Reports across all routers/
    Overview WAN Folder - group Reports across all routers/
    Overview Router Folder - group Reports across all routers/
  interfaces
  Customer Instance
    Site Instance
      Probe Summary Folder - group reports
      SLA Exec Folder - group reports
    VPN Instance
The Vista Provisioner and InfoVista Server automatically generate CAR reports when a CAR policy is detected on an interface using the VPN_IF definition in the Def.txt file. This occurs regardless of whether the Collect CAR MIB statistics checkbox is enabled or not in the Measurement Parameter dialog for a device. The automatic generation of CAR reports by InfoVista prevents the customized placement of the CAR folder. As a result, it automatically appears at the root of the report folder structure.

Group reports are generated by Vista Provisioner based on configuration information provided by the Def.txt file. Refer to the section InfoVista object mapping on page 60 for default groups.

**General tasks**

**Note:** To access the InfoVista Integration module functionality in the Service Activator GUI, ensure that the Allow concurrent logins option is enabled for the Service Activator user. For more information on how to do this, refer to the Service Activator Administrator’s Guide.

The tasks for integrating Service Activator with InfoVista include:

1. Install the InfoVista Integration Module on the Service Activator server.

   For instructions on how to install InfoVista Integration module, refer to the Service Activator Setup Guide.

2. Install the following InfoVista components in the following order:
   - InfoVista Server(s)
   - Vista Provisioner(s)
   - Plug-in(s) for NetFlow (only when the NetFlow measurement is used)
3. In the Service Activator user interface:
   - Configure the devices to generate statistics for the various measurement types.

   For more information, refer to Chapter 2, Service Assurance Agent (SAA) measurement and Chapter 3, MIB-based and NetFlow measurements.

   - Model the InfoVista Server(s) and the Plug-in(s) for NetFlow components as ‘external systems’.

   For more information, refer to the section Creating an external system on page 45 in Chapter 4, Setting Up Service Activator for Integration.

   - Add collectors to the devices that you want to monitor:
     - To collect data for SAA or MIB-based statistics, link the device to an InfoVista Server collector.
     - To collect data for NetFlow statistics, add two Collector Parameters. First link the device to the Vista Plug-in for NetFlow collector. Then link the device to an InfoVista Server collector.

   When you associate both a NetFlow and MIB-based collector with a device, the device exports NetFlow data to the NetFlow Collector (Vista Plug-in for NetFlow) which in turn is polled by the MIB-based collector (InfoVista Server).

   Service Activator calculates the relationship between the NetFlow and MIB-based collectors. You do not need to associate a NetFlow collector with a MIB-based collector in the user interface.
4. Refer to the next section for instructions on how to import Service Activator integration files into InfoVista.

Invoking the InfoVista Integration Module

The XML data export function of the InfoVista Integration Module is invoked using the infoVistaExport.sh script, which resides on the Service Activator server on which the Integration Manager is installed.

It can be invoked from a manually entered command, or set up to run automatically at appropriate intervals using a UNIX cron job.

Optionally, the infoVistaExport.sh script can instruct the InfoVista Integration Module to transmit the final output InfoVista.txt file to an external system using an ftp protocol transfer. The destination ftp server, logon credentials and a destination directory can be specified when the script is invoked.

The command has the following parameters:

\<Service ActivatorHome>/modules/bin/infoVistaExport.sh <username> 
<password> [ftp_server_name] [ftp_user_name] [ftp_pwd] [remote_directory_name]

Required parameters:
- username user ID used to access Service Activator
- password password for user ID

Optional parameters:
- ftp_server_name destination ftp server to send InfoVista.txt to
- ftp_user_name user ID for destination ftp server
- ftp_pwd password for ftp user ID
- remote_directory_name destination ftp server directory in which to place InfoVista.txt

For more information, refer to the section Creating a collector in Service Activator on page 48 in Chapter 4, Setting Up Service Activator for Integration.
Importing Service Activator integration files into InfoVista

This section describes how to import files to InfoVista once the InfoVista Integration Module has been run. See *Invoking the InfoVista Integration Module on page 67* for instructions on invoking the module to create the InfoVista.txt file.

Importing the def.txt and report.pl files into InfoVista

This procedure should be performed during the initial set up of InfoVista reporting, once installation of the InfoVista Integration Module is complete. Refer to the *Service Activator Setup Guide* for details on installation of the InfoVista Integration Module.

**To import the Service Activator Def.txt and report.pl files into InfoVista:**

1. Copy the def.txt and report.pl files from the following location to the Vista Provisioner:
   
   `<ServiceActivatorHome>/modules/InfoVistaIntegrationModule`

2. Refer to the InfoVista documentation for instructions on how to import this file into the Vista Provisioner.

Importing the Service Activator InfoVista.txt file into InfoVista

Use this procedure if the infoVistaExport.sh file is not set up to send the InfoVista.txt file via ftp directly to the Vista Provisioner. Otherwise, perform this procedure each time the network topology is updated in Service Activator.

**To import the Service Activator InfoVista.txt file into InfoVista**

1. Generate the InfoVista.txt file:
   
   See *Invoking the InfoVista Integration Module on page 67*.

2. Copy the new InfoVista.txt file from the following location to the Vista Provisioner:

   `<ServiceActivatorHome>/ServiceActivator/modules/InfoVistaIntegrationModule`

3. Refer to the InfoVista documentation for instructions on how to:
— Import the InfoVista.txt file into the Vista Provisioner.
— Update the InfoVista Server with the latest topology configuration.

Reports

This section covers the following topics:

- Report naming convention
- Interpreting reports

Report naming convention

Report names are based on the entry names included in the InfoVista.txt file. The report naming convention outlined in the following table is used.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Naming Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device/Router</td>
<td>Device Name</td>
<td>rot2611-18</td>
</tr>
<tr>
<td>WAN Interface</td>
<td>Device Name + Interface Name + IP Address</td>
<td>rot2611-18_Ethernet0/1_10.13.5.26</td>
</tr>
<tr>
<td>LAN Interface</td>
<td>Device Name + Interface Name + IP Address</td>
<td>rot2611-18_Ethernet1/1_10.13.5.29</td>
</tr>
<tr>
<td>ATM PVC</td>
<td>Device Name + PVC Name</td>
<td>rot2611-18_VPI: 1 VCI: 100</td>
</tr>
<tr>
<td>FR PVC</td>
<td>Device Name + PVC Name</td>
<td>rot2611-18_DLCI3</td>
</tr>
<tr>
<td>JITTER Probe</td>
<td>“From” Src Device Name + “To” Dest Device Name + “Name” Probe Name + “ID” + Probe ID + “DSCodePt” + DSCodePt</td>
<td>From: rot2611-18 To: rot2611-17 Name: TcpConnect ID: 1074790913 DSCodePt:0</td>
</tr>
<tr>
<td>RTT Probe</td>
<td>“From” Src Device Name + “To” Dest Device Name + “Name” Probe Name + “ID” + Probe ID + “DSCodePt” + DSCodePt</td>
<td>From: rot2611-18 To: rot2611-17 Name: TcpConnect ID: 1074790914 DSCodePt:0</td>
</tr>
<tr>
<td>NetFlow Plugin</td>
<td>Plugin Name</td>
<td>Defined by user in Service Activator</td>
</tr>
</tbody>
</table>
Interpreting reports

For detailed information on the reports generated by InfoVista, refer to the InfoVista documentation.

Configuring the InfoVista Integration module SNMP community string

The OSS Integration Manager does not pass the device SNMP write community string in open text (acting as a password) and therefore it is not available for Topology.xsl to process. A hard coded value of private is used for this value.

If your devices are configured to use a different value for the SNMP write community string, you can replace the string private by editing the InfoVista.xsl file.

If your devices are all configured with individual (i.e. different) write community strings, you would have to implement a look up method which integrates with the InfoVista.xsl file. This is beyond the scope of this document.

The entry in InfoVista.txt representing the NFPLUGIN has a hard coded community string of public as Service Activator does not manage this attribute.

Specifying additional object information

There are inventory attributes not stored in the Service Activator Object Model that are needed by the InfoVista Server for SLA collection. In order to store this information a separate GUI is available from the associated object (i.e. Device, VPN, Site) in the Service Activator GUI.

The Additional Attributes GUI is a configurable XML Java Swing GUI that allows you to enter the additional attributes. These are stored as Generic Policies in the Service Activator object model and exported along with the other Service Activator objects.
To launch the Additional Attributes GUI, right-click on the appropriate object, and select **Modules > InfoVista > Properties**.

The fields displayed depend on what object type you are setting properties for. Consult the table in the section *InfoVista.xml Filtering on page 61* to see which fields are available for which object types. The column labeled *Uses Service Activator Objects* is marked **N** for fields which are populated from the Additional Attributes GUI.

**Deployment**

For details on the versions of InfoVista components supported, refer to the *Release Notes*.

**Error Reporting**

During the filtering of the InfoVistaExport.xml file by the InfoVista.xsl file, if there is missing data or incorrect data, a message will be output to the Error console. The error will also be included in the XML output file as an XML comment. This allows you to confirm if any of the required data is missing or if the data is incorrectly filled.
Chapter 7

Micromuse Integration

This chapter describes how Service Activator integrates with the MicroMuse Netcool Impact Server through the Micromuse Integration Module. It assumes a working knowledge of MicroMuse concepts. For more information on MicroMuse, refer to the MicroMuse product documentation.

This chapter covers the following topics:

- Integration with Service Activator
- Micromuse object mapping
- Specifying additional object information
- Customizing Micromuse integration
- Deployment
- Error reporting
Integration with Service Activator

Service Activator communicates with Micromuse Netcool through Micromuse Integration Module which generates XML formatted documents. These XML files comply with the input requirements of Micromuse Netcool Impact server via the XML DSA. This allows the sharing of information between the Service Activator object model and Micromuse.

Typically the Micromuse Integration Module will be set up to generate its export XML file (i.e. Micromuse.xml) on a nightly basis. When Micromuse needs additional information on a fault it will then read the export XML file to enhance the fault information.

In more detail, integration between Service Activator and Micromuse occurs in the following way:

- Service Activator’s Generic Exporter, which is automatically installed with the Micromuse Integration Module, reads a local XML config file (MicromuseExportConfig.xml) which contains the parameters required to log in to the OIM and extract data from the object model.
- It creates MicromuseExport.xml which contains raw object model information.
- The Micromuse.xsl file is then used to filter MicromuseExport.xml to create an export file (Micromuse.xml) containing filtered object model data as required by your existing Micromuse policies.
- Micromuse then uses the Micromuse Netcool Impact server to import the export file and generate the necessary reports.
Key components

The following table provides a brief description of the key components used to enable Service Activator integration with Micromuse.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micromuse Netcool Impact Server</td>
<td>Receives information from the Service Activator object model in an XML file transmitted via ftp. The server then distributes the data to Netcool Impact and other installed Micromuse Netcool solutions.</td>
</tr>
<tr>
<td>Additional attributes GUI</td>
<td>GUI used to add additional property attributes required by Netcool to devices. This information is stored in the Service Activator object model as configuration policy information.</td>
</tr>
</tbody>
</table>
| Micromuse Integration Module     | A Service Activator module that includes the Generic Exporter and several integration files. The Generic Exporter exports data from the object model and converts it to a Micromuse compliant format ready for import.  
**Note:** For details on installing the Micromuse Integration Module, refer to the Setup Guide. |
## Micromuse integration files

As shown in the diagram on page 74, Service Activator uses the following files to perform integration with the Netcool Impact server:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>micromuseExport.sh</td>
<td>Script residing on the Service Activator server on which the Integration Manager is installed, used to invoke the XML export process. See <em>Invoking the Micromuse Integration Module</em> on page 78.</td>
</tr>
</tbody>
</table>
| MicromuseExportConfig.xml | When the Micromuse Generic Exporter is invoked, it reads this configuration file which includes variables indicating:  
  - Service Activator server to connect to  
  - name of output XML file to be used when extracting data from Service Activator  
  - transformation directives indicating the number of XSLs to be called and what sequence they are to be called in  
  - root tag to be used in XML output  
  - rules specifying which objects, attributes and children are to be extracted and which are to be filtered out from the Service Activator object model |
| MicromuseExport.xml       | Object model file filtered by Micromuse.xsl to produce final output file Micromuse.xml |
| Micromuse.xsl             | XML schema file used to transform MicromuseExport.xml into a format compliant with Micromuse. Executes filtering and formatting commands to localize specific Micromuse requirements. |
These files can be customized as needed. For more information, refer to the section [Customizing Micromuse integration](#) on page 82.

Once the Micromuse.xml file is imported into the Micromuse Netcool Impact server, it is distributed to other Micromuse solutions.

### Invoking the Micromuse Integration Module

The XML data export function of the Micromuse Integration Module is invoked using the `micromuseExport.sh` script, which resides on the Service Activator server on which the Integration Manager is installed.

It can be invoked from a manually entered command, or set up to run automatically at appropriate intervals using a UNIX cron job.

Optionally, the `micromuseExport.sh` script can instruct the Micromuse Integration Module to transmit the final output Micromuse.xml file to an external system using an ftp protocol transfer. The destination ftp server, logon credentials and a destination directory can be specified when the script is invoked.

The command has the following parameters:

```
<Service ActivatorHome>/modules/bin/micromuseExport.sh <username> <password> [ftp_server_name] [ftp_user_name] [ftp_pwd] [remote_directory_name]
```

**Required parameters:**

- `username`: user ID used to access Service Activator
- `password`: password for user ID

**Optional parameters:**

- `ftp_server_name`: destination ftp server to send Micromuse.xml to
- `ftp_user_name`: user ID for destination ftp server

---

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micromuse.xml</td>
<td>Filtered object model information to be transmitted via ftp to the Micromuse Netcool Impact server.</td>
</tr>
<tr>
<td>Archive files</td>
<td>Whenever Service Activator creates a new Micromuse.xml file the previous one is archived with the date and time of creation incorporated into the filename.</td>
</tr>
</tbody>
</table>
ftp_pwd password for ftp user ID
remote_directory_name destination ftp server directory in which to place Micromuse.xml

**Note:** To access the Micromuse Integration Module functionality in the Service Activator GUI, ensure that the **Allow concurrent logins** option is enabled for the Service Activator user. For more information on how to do this, refer to the *Service Activator Administrator’s Guide*.

---

## Micromuse object mapping

This table shows how Micromuse XML objects are mapped to Service Activator object model entities.

<table>
<thead>
<tr>
<th>Micromuse Resource</th>
<th>Service Activator Object Model Entity</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>VPN</td>
<td></td>
</tr>
<tr>
<td>VpnSite</td>
<td>Site</td>
<td></td>
</tr>
<tr>
<td>VpnAccessPoint</td>
<td>Interface</td>
<td></td>
</tr>
<tr>
<td>OspfRoutingProtocol</td>
<td>OSPF Attributes from Interface</td>
<td>This is from Service level. Service Activator stores it at the Interface level.</td>
</tr>
<tr>
<td>RipRoutingProtocol</td>
<td>RIP Attributes from Interface</td>
<td>This is from Service level. Service Activator stores it at the Interface level.</td>
</tr>
<tr>
<td>BgpRouting Protocol</td>
<td>BGB Attributes from Interface</td>
<td></td>
</tr>
<tr>
<td>Static Routing</td>
<td>Static Routing Attributes from Interface</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>Customer</td>
<td></td>
</tr>
</tbody>
</table>
There are inventory attributes not stored in the Service Activator Object Model that are needed by the Micromuse Netcool Impact Server for fault collection. In order to store this information a separate GUI is available from the associated object (i.e. Device, VPN, Site) in the Service Activator GUI.

The **Additional Attributes GUI** is a configurable XML Java Swing GUI that allows you to enter the additional attributes. These are stored as Generic Policies in the Service Activator object model and exported along with the other Service Activator objects.

To launch the Additional Attributes GUI, right-click on the appropriate object, and select **Modules > Micromuse > Properties**.

### Specifying additional object information

<table>
<thead>
<tr>
<th>Micromuse Resource</th>
<th>Service Activator Object Model Entity</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>Site properties</td>
<td></td>
</tr>
<tr>
<td>SubnetworkConnection</td>
<td>Left blank</td>
<td></td>
</tr>
<tr>
<td>ManagedElement</td>
<td>Device</td>
<td></td>
</tr>
<tr>
<td>Classifier</td>
<td>Class Of Service</td>
<td></td>
</tr>
<tr>
<td>Forwarding class</td>
<td>Classification Group</td>
<td></td>
</tr>
<tr>
<td>Loss-priority</td>
<td>Classification</td>
<td></td>
</tr>
<tr>
<td>PolicyMap</td>
<td>PHB</td>
<td></td>
</tr>
<tr>
<td>PolicyClass</td>
<td>CoS Mechanisms</td>
<td></td>
</tr>
<tr>
<td>TerminationPoint</td>
<td>Interface/ Subinterface/Segment</td>
<td></td>
</tr>
</tbody>
</table>
The fields displayed depend on what object type you are setting properties for. Consult the table in the section *MicromuseExport.xml Filtering on page 81* to see which fields are available for which object types. The column labeled **Uses Service Activator Objects** is marked N for fields which are populated from the Additional Attributes GUI.

**MicromuseExport.xml Filtering**

The Micromuse Generic Exporter filters the object model to process only the entities described in the table below. It exports an XML file containing a filtered version of the object model for both customer and network topology trees:

- Policy/Domain*/Customer*/VPN*/, Site*, ParameterSetInstance*
- Policy/Domain*/Network*/Device*, ParameterSetInstance*

Micromuse collector parameters must exist against entities in the topology tree for network reporting.

**NOTE:** Classifier, forwarding-class and loss-priority are Juniper-only functionality and are not currently supported.
Customizing Micromuse integration

A service provider can customize the integration between Service Activator and Micromuse by modifying the following files:

- MicromuseExportConfig.xml
- Micromuse.xsl

MicromuseExportConfig.xml

The Generic Exporter generates the MicromuseExport.xml file following content guidelines stored in the MicromuseExportConfig.xml file.

The default MicromuseExportConfig.xml file is set up to generate a MicromuseExport.xml file that lists all devices in Service Activator that are assigned a Micromuse collector.

A service provider can modify the MicromuseExportConfig.xml file appropriately to customize the topology information provided to Micromuse.

The MicromuseExportConfig.xml file is located in the following directory:

<ServiceActivatorHome>/modules/Config

For a description of the fields included in the default MicromuseExportConfig.xml file, refer to Appendix A, TopologyExporterConfig.xml Fields.
Micromuse.xsl

The Micromuse.xsl file applies formatting and filter rules to the MicromuseExport.xml file to generate an xml file that can be imported by the Micromuse Netcool Impact server. If the MicromuseExportConfig.xml file is modified, the filters in the Micromuse.xsl file should also be modified accordingly.

The Micromuse.xsl file is located in the following directory:

```
<ServiceActivatorHome>/modules/Config
```

It applies the following filters:

**Domain(s)**

**Network(s)**

**Device(s)**
- Must be from Cisco, or Juniper
- Must be managed
- Must not be virtual
- Must have Micromuse Server associated

**Interface(s)**
- Must have interface and parent device roles assigned
- Must have measurement parameter MIB2 selected

**Sub-interface(s)**
- Must have interface and parent device roles assigned
- Must have measurement parameter MIB2 selected

**PHB Group(s) for Interface/Sub-interface**
- Must have concrete object installed

**Policing Rule(s) for Interface/Sub-interface**
- Must have concrete object installed

**Customer(s)**
- No filters are applied directly to the Customers

**VPN(s)**
- No filters are applied directly to the VPNs

**Site(s)**
- No filters are applied directly to the sites

**Device(s)**
Deployment

For details on the versions of MicroMuse components supported, refer to the Release Notes.

Error reporting

During the filtering of the MicromuseExport.xml file by the Micromuse.xsl file, if there is missing data or incorrect data, a message will be output to the Error console. The error will also be included in the XML output file as an XML comment. This allows you to confirm if any of the required data is missing or if the data is incorrectly filled.
Appendix A

TopologyExporterConfig.xml Fields

This appendix describes fields included in the default TopologyExporterConfig.xml files.
Directory locations

The TopologyExporterConfig.xml files are located in the following directory:

<ServiceActivatorHome>/modules/Config/

Configuration fields

<table>
<thead>
<tr>
<th>Configuration field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipsaServer</td>
<td>The IP address where the Service Activator Integration Manager is running.</td>
</tr>
<tr>
<td>ipAddress</td>
<td>The port where the Service Activator Integration Manager is running.</td>
</tr>
<tr>
<td>port</td>
<td>The filename of the xml export file.</td>
</tr>
<tr>
<td>primaryEntry</td>
<td>Multiple transformations can be performed by surrounding each transformation with the &lt;item&gt; tag.</td>
</tr>
<tr>
<td>fileNameToWrite</td>
<td>The filename that acts as the source document for the transformation. This entry is usually the filenameToWrite from the primaryEntry.</td>
</tr>
<tr>
<td>transformations</td>
<td>The filename of the xsl document that is used for the transformation.</td>
</tr>
<tr>
<td>fileNameToWrite</td>
<td>The filename where the transformation is written to.</td>
</tr>
<tr>
<td>rootTagName</td>
<td>The root xml tag (top-level) that is inserted into the output xml document. The default is 'root'.</td>
</tr>
</tbody>
</table>
## Output rules

<table>
<thead>
<tr>
<th>Output rules field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startFromObject</td>
<td>The object type in Service Activator that the export starts the cascade from.</td>
</tr>
<tr>
<td>objectSubscription</td>
<td>Filters which objects are exported.</td>
</tr>
<tr>
<td>subscribe</td>
<td>Identifies which object types to export. If all objects should be exported, insert object type 'all'.</td>
</tr>
<tr>
<td>unsubscribe</td>
<td>Identifies which object types not to export.</td>
</tr>
<tr>
<td>attributeSubscription</td>
<td>Filters which attributes to export.</td>
</tr>
<tr>
<td>subscribe</td>
<td>Identifies what attributes to export. If all attributes should be exported, then enter the single subscribe 'all'.</td>
</tr>
<tr>
<td>requiresAttributeValue</td>
<td>Filters which objects to export based on an attribute value.</td>
</tr>
<tr>
<td>objectType</td>
<td>Type of object the attribute is on.</td>
</tr>
<tr>
<td>attribute</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>The name of the attribute.</td>
</tr>
<tr>
<td>value</td>
<td>The value the attribute must have for the object to be exported.</td>
</tr>
<tr>
<td>requiresChild</td>
<td>Filters which objects to export based on associated child type.</td>
</tr>
<tr>
<td>objectType</td>
<td>Identifies type of object to export based on child type.</td>
</tr>
<tr>
<td>requires</td>
<td>The type of child object that is required for the above object type to be exported. For example, if the objectType is Customer and requires is set to VPN, only Customers that have VPNs will be exported.</td>
</tr>
</tbody>
</table>
### TopologyExporterConfig.xml Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ignoreChild</td>
<td>Filters which objects not to export based on whether they are the child of a given parent.</td>
</tr>
<tr>
<td>objectType</td>
<td>Identifies the object type of the parent.</td>
</tr>
<tr>
<td>ignore</td>
<td>Type of child object type that will not be exported based on the above parent type. For example, if the parent objectType is Customer and ignore is set to VPN, any VNP object with a parent of Customer will not be exported.</td>
</tr>
<tr>
<td>stopChildCascade</td>
<td>Stops cascading one level below a certain object type. For example, if stopChildCascade is given for object type Site, only direct children of the Site object are exported.</td>
</tr>
<tr>
<td>item</td>
<td>The object type to stop cascading on (note: you will still get one level of children below this object type)</td>
</tr>
<tr>
<td>requiresCollectorParameter</td>
<td>Filters which objects to export based on whether they are associated with a collector of a certain type.</td>
</tr>
<tr>
<td>objectType</td>
<td>The object type, typically Device, that is associated with a collector type.</td>
</tr>
<tr>
<td>collectorType</td>
<td>The collector type that the above object type is associated with. For example, if the objectType is Device and the collectorType is InfoVista Server, only devices that are associated with an InfoVista Server collector are exported.</td>
</tr>
<tr>
<td>Output rules field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cascadeMeasurementParameter</td>
<td>This is not really a filter. Measurement parameters are not children of objects, but instead have targets to which they apply. To make Measurement Parameters appear as children, the exporter can cascade the xml definition of the MeasurementParameter down to the object which it targets. For example, if cascadeMeasurementParameter is applied to the object type Interface, the measurement parameter will appear under the interface object when it is exported.</td>
</tr>
<tr>
<td>item</td>
<td>The object type that measurement parameters are cascaded on.</td>
</tr>
</tbody>
</table>
Index

A
Additional Attributes
   specifying 80
Additional Attributes GUI 70, 80
Aggregation, in NetFlow 35
Archive files
   Micromuse 78

C
CB QoS MIB measurement
   guidelines for collector roles 46
   setting up 32
CE to CE measurement 22
Collectors
   and inheritance 47
   and roles 45
   and VPNs 47
   roles, guidelines for measurement types 46
Configurable XML Exporter 74
customer support vii

D
Data gathering, overview of 50, 56
def.txt 63, 68
Deploying
   SAA 20
documentation
   downloading viii
   Service Activator ix

E
Export.xml 51
ExportConfig.xml 50
External systems
   description of 3

F
Flow, description of in NetFlow 34
ftp parameters
   Micromuse 78
Full duplex tests in SAA 10

G
Generic Exporter 50, 56

H
Half duplex tests in SAA 10

I
ICMP Echo, SAA operation 15
InfoVista Integration
   configuring SNMP 54, 70
   customizing 52, 61
   error reporting 54, 71
   ftp configuration 54, 67
   invoking 53, 67
   key components 51, 57
   object mapping 60
   Overview 49, 55
   reports 69
   specifying additional objects 70
InfoVista Integration Module 57
   installing 65
InfoVista Server 57
InfoVista.txt 56, 58, 63
InfoVista.xsl 56, 58
Infovista.xsl 62
infoVistaExport.sh 58
infoVistaExport.xml 52, 56, 58, 61
InfoVistaExportConfig.xml 52, 56, 58, 62
Inheritance and collectors 47
Instance
   definition of 59
J
Jitter, SAA operation 15

L
Limits, when applying SAA 26

M
Measurement
  overview of types 2
Measurement and reporting, overview of setup tasks 65
MIB2 measurement
  guidelines for collector roles 46
  overview of 32
Micromuse Integration
  components 76
  customizing 82
  error reporting 84
  files 77
  Invoking 78
  object mapping 79
  Overview 73
Micromuse.xml 78
Micromuse.xsl 74, 77
micromuseExport.sh 77
MicromuseExport.xml 74, 77
  filtering 81
MicromuseExportConfig.xml 82
Modelling third-party components 3

N
Netcool Impact server 74
NetFlow 31
  aggregation 35
  aggregation schemes 36
  architecture and distribution 33
  export formats 34
  flow description 34
  guidelines for collector roles 46
  overview of 33
  UDP formats 34
  where you can apply 37
Service Assurance Agent See SAA 8

O
Operations
  and VPN topology 9
  SAA, overview of 8

P
PE to CE measurement 25
PE to PE measurement 23
Pre-requisites
  SAA 19
  products
    downloading viii

R
report.pl 64, 68
Reporting, overview of 50, 56
Roles
  associating with collectors 45

S
SAA 7–26
  CE to CE measurement, description of 22
  deployment guidelines 20
  guidelines for collector roles 46
  limits 26
  number of operations configured by 9
  one-way test 10
  operation types 8
  overview of 8
  PE to CE measurement, description of 25
  PE to PE measurement, description of 23
  pre-requisites 19
  shadow routers, description of 21
  test connection to an unmanaged device 14
  threshold violations, configuring SNMP trap/SNA NMVT alert 16
  two-way test 10
  using virtual devices 14
SAA Responder, description of 16
SAA template
  overview of 16
Setting up
  CB QoS MIB measurement 32
Setting up measurement and reporting,
  overview 65
Shadow routers, description of 21
  support
    customer vii

T
TCP Connect, SAA operation 15
Threshold violations, configuring SNMP trap/
  SNA NMVT alert 16
TopologyExporter.txt 51, 53
TopologyExporter.xml 50
TopologyExporter.xsl 51, 52
TopologyExporterConfig.xml 51

U
UDP Echo, SAA operation 15
UDP formats and NetFlow 34
Unmanaged devices, testing connection to
  with SAA 14

V
Virtual devices, testing connection to with
  SAA 14
Vista Instance, definition of 59
Vista Plug-in for Netflow 57
Vista Provisioner 57
Vista, definition of 59
VPNs
  collecting measurement data for 47
  fully-meshed and SAA 9
  hub and spoke and SAA 9