EMC® Smarts®
IP Management Suite

Version 7.0

Configuration Guide

P/N 300-004-086
REV A03

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Copyright © 1996-2007 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Contents

Preface v

Chapter 1 Introduction

Important terminology 1-2
Architectural and functional overview 1-3
Configuration overview 1-6
What's next after configuration? 1-8

Chapter 2 Setting Configuration Parameters

User configuration parameters 2-2
Methods of modifying user configuration parameters 2-3
Description of discovery.conf 2-6
Description of tpmgr-param.conf 2-11
Description of name-resolver.conf 2-15
Description of partition.conf 2-16
Description of user-defined-connections.conf 2-17
Fine tuning ICMP and SNMP polling related parameters 2-19
Configuring IP tagging for backward compatibility 2-21
Enabling light discovery for satellite Domain Managers 2-22
Enabling the concatenation of InterfaceAlias fields in ERX devices 2-24
Enabling the logging of failed DNS lookups during discovery 2-26
Enabling the SystemRestarted and ConfigChange events 2-28
Changing the IPSec short pending list interval 2-37

Chapter 3 Setting Up Communication Links

Configuring the Global Manager to communicate with the IP Manager 3-2
Configuring SNMP trap receiving and processing 3-5
Configuring SNMP trap notifying 3-16

Chapter 4 Excluding Device-Access Interfaces

Understanding interface-matching filters 4-2
Example of configuring interface-matching filters 4-10
Optimizing the discovery of ERX devices 4-13
Enabling the creation of a physical connection for a shared MAC address 4-16

Chapter 5 Using the Polling and Thresholds Console

Opening the Polling and Thresholds Console 5-2
Understanding groups and settings 5-5
Working with groups and settings 5-7

Chapter 6 Customizing Polling

Viewing the Polling tab 6-2
Understanding polling groups and settings 6-4
Default polling groups and settings for IP Availability Manager 6-5
Default polling groups and settings for IP Performance Manager 6-8
Enabling IPv6 address polling 6-12

Chapter 7 Customizing Polled Data Thresholds

Viewing the Thresholds tab 7-2
Understanding threshold groups and settings 7-4
Default threshold groups and settings for IP Availability Manager 7-5
Default threshold groups and settings for IP Performance Manager 7-12
Using a policy setting to control the managed state of interfaces 7-22
Using a policy setting to ensure appropriate VLAN assignments 7-25
Using a policy setting to prevent VLAN impacts 7-28

Chapter 8 Creating Credential Sets

Viewing the Device Access tab 8-2
Understanding device access groups and settings 8-3
Default device access groups and settings 8-4
Overview of creating a device access group 8-5

Chapter 9 Creating IP Tagging Filters

Viewing the Tagging tab 9-2
Understanding the IP tagging feature 9-3
Understanding tagging filter groups and settings 9-10
Default tagging filter groups and settings 9-11
Overview of creating an IP tag filter group 9-15
Example of creating an IP tag filter group 9-17

Appendix A Wildcard Patterns

Types of wildcard patterns A-2

Appendix B Polling for Analysis

Polling overview B-2
ICMP poller B-3
SNMP poller B-4
ICMP and SNMP polling coordination B-5

Index i-1

Preface

The EMC Smarts IP Management Suite is a family of products intended to diagnose availability and performance problems for IP networks.

This document provides an architectural overview of the EMC Smarts IP discovery products and presents procedures for configuring them. The IP products that provide discovery are:

- EMC Smarts IP Availability Manager (IP Availability Manager)
- EMC Smarts IP Performance Manager (IP Performance Manager)
- EMC Smarts IP Server Performance Manager (IP Server Performance Manager)
- EMC Smarts IP Discovery Manager (IP Discovery Manager)

IP Availability Manager discovers and monitors network data for the purpose of performing availability analysis, and IP Performance Manager and IP Server Performance Manager discover and monitor network data for the purpose of performing performance analysis. IP Discovery Manager has the same discovery capabilities as IP Performance Manager except that it does not perform any availability or performance analysis.

Note: In this document, "IP Manager" is used to collectively represent IP Availability Manager, IP Performance Manager, IP Server Performance Manager, and IP Discovery Manager.

Audience for this guide

This document is intended for administrators and integrators who need to configure and maintain the IP Manager.

Prerequisites

It is assumed that readers of this document have a general understanding of operating systems. Readers must have administrative privileges on the local host to perform the configurations.

In addition, the following software must be installed:

- EMC Smarts IP Availability Manager, EMC Smarts IP Performance Manager, EMC Smarts IP Server Performance Manager, or EMC Smarts IP Discovery Manager
- EMC Smarts Service Assurance Manager (Global Manager)
- EMC Smarts Global Console

The *EMC Smarts IP Management Suite Installation Guide* and the *EMC Smarts Service Assurance Management Suite Installation Guide* provide information about installing these products.

EMC Smarts IP Management Suite installation directory

In this document, the term BASEDIR represents the location where EMC Smarts software is installed.

- For UNIX, this location is: /opt/InCharge*<n>*/*<productsuite>*.
- For Windows, this location is: C:\InCharge*<n>**<productsuite>*.

The <n> represents the EMC Smarts software platform version number. The <productsuite> represents the EMC Smarts product suite to which the product belongs. For example, on UNIX operating systems, EMC Smarts IP Availability Manager is, by default, installed to: /opt/InCharge7/IP/smarts. On Windows operating systems, this product is, by default, installed to: C:\InCharge7\IP\smarts. This location is referred to as BASEDIR/smarts.

Optionally, you can specify the root of BASEDIR to be something different, but you cannot change the <productsuite> location under the root directory.

For more information about the directory structure of EMC Smarts software, refer to the *EMC Smarts System Administration Guide*.

EMC Smarts IP Management Suite products

The EMC Smarts IP Management Suite includes the following products:

- EMC Smarts IP Availability Manager
- EMC Smarts IP Discovery Manager
- EMC Smarts IP Performance Manager
- EMC Smarts IP Server Performance Manager
- EMC Smarts Adapter for HP OpenView NNM
- EMC Smarts Adapter for IBM/Tivoli NetView

Related documentation

In addition to this document, EMC Corporation provides a Help system for command line programs as well as product documentation.

Help for command line programs

Descriptions of command line programs are available as HTML pages. The index.html file, which provides an index to the various commands, is located in the BASEDIR/smarts/doc/html/usage directory.

EMC Smarts documentation

Readers of this guide may find the following related documentation helpful. It can be found in the BASEDIR/smarts/doc/pdf directory.

Note: These documents are updated periodically. Electronic versions of the updated manuals are available on the Powerlink website: **http://Powerlink.EMC.com**.

- *EMC Smarts Documentation Catalog*
- *EMC Smarts System Administration Guide*

- *EMC Smarts ICIM Reference*
- *EMC Smarts Common Information Model Infrastructure Models Chart*
- *EMC Smarts Common Information Model Application/ Business Models Chart*
- *EMC Smarts ASL Reference Guide*
- *EMC Smarts Perl Reference Guide*
- *EMC Smarts Dynamic Modeling Tutorial*
- *EMC Smarts MODEL Reference Guide*

EMC Smarts IP Management Suite documentation

The following EMC Smarts documents are relevant to users of the IP Management product suite:

- *EMC Smarts IP Management Suite Installation Guide*
- *EMC Smarts IP Management Suite Deployment Guide*
- *EMC Smarts IP Management Suite Configuration Guide*
- *EMC Smarts IP Management Suite Discovery Guide*
- *EMC Smarts IP Availability Manager User's Guide*
- *EMC Smarts IP Performance Manager and Server Performance Manager User's Guide*
- *EMC Smarts IP Adapters User's Guide*
- *EMC Smarts IP Availability Manager Extension for IPv6 User's Guide*
- *EMC Smarts IP Availability Manager Extension for NAS User's Guide*
- *EMC Smarts Quick Start Guide*

Refer to the *EMC Smarts Documentation Catalog* for documentation resources provided with other EMC Smarts product suites.

Suggestions for searching PDF files

You may search across multiple PDF files using the Adobe Acrobat Reader software.

1. If the documentation is not accessible to all users of the EMC Smarts product suite, copy the contents of the *BASEDIR/smarts/doc/pdf d*irectory to a central location, such as a shared drive on your LAN, that operators and others may view.
2. To search throughout the documentation library, open the Acrobat Reader software.
   1. Choose **Edit** > **Search**, and enter a word or phrase.
   2. In the **Where would you like to search** option, select **All PDF Documents in** and type the pathname of the location where the PDF documents reside.

If you have more than one EMC Smarts product suite installed, you can set up cross-product document searches by copying files from the BASEDIR/smarts/doc/pdf directory for each product suite into this common documentation directory path.

Conventions used in this guide

EMC uses the following conventions for notes and cautions.

Note: A note presents information that is important, but not hazard-related.

IMPORTANT

An important notice contains information essential to the operation of the software.

CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment. The caution may apply to hardware or software.

Typographical conventions

EMC uses the following conventions in this guide:

Normal font	In running text: • Interface elements (for example, button names, dialog box names) outside of procedures • Items that the user selects outside of procedures • Java classes and interface names • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, filenames, functions, menu names, utilities • Pathnames, URLs, filenames, directory names, computer names, links, groups, service keys, file systems, environment variables (for example, command line and text), notifications
Bold	In procedures: • Names of dialog boxes, buttons, icons, menus, fields, and maps • Selections from the user interface, including menu items, and field entries • Key names • Window names • Code examples • Commands • User interface tabs
Italic	Used for: • Full publications titles referenced in text In procedures: • Directory paths • Filenames • Scripts • Important/unique words • URLs • Utilities • Variables
`Courier`	Used for: • System output • User entry
< >	Angle brackets used to enclose parameter or variable values supplied by the user.
[ ]	Square brackets used to indicate optional values.
|	Vertical bar used to indicate alternate selections. The bar means “or”.
{ }	Braces used to indicate content that you must specify (that is, x or y or z).
...	Ellipses used to indicate nonessential information omitted from the example.

%	Used to indicate a C shell prompt.
#	Used to indicate a C shell superuser prompt.
▼▲	Used to indicate a command is wrapped over one or more lines. The command must be typed as one line.

Pathname conventions

Directory pathnames are shown with forward slashes (/). Users of the Windows operating systems should substitute back slashes (\) for forward slashes.

Graphical conventions

If there are figures illustrating consoles in this document, they represent the consoles as they appear in Windows. Under UNIX, the consoles appear with slight differences. For example, in views that display items in a tree hierarchy such as the Topology Browser, a plus sign displays for Windows and an open circle displays for UNIX.

Smarts Manager

Unless otherwise specified, the term Smarts Manager is used to refer to EMC Smarts programs such as Domain Managers, Global Managers, and adapters.

Where to get help

EMC support, product, and licensing information can be obtained as follows.

Product information — For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Powerlink™ website (registration required) at:

http://Powerlink.EMC.com

Technical support — For technical support, go to EMC Customer Service on Powerlink. To open a service request through Powerlink, you must have a valid support agreement. Please contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Sales and customer service contacts — For the list of EMC sales locations, please access the EMC home page at:

http://EMC.com/contact

1

Introduction

This chapter provides a brief architectural and functional overview of the IP Manager and highlights the configuration tasks needed to set up and maintain the IP Manager. It consists of the following sections:

- Important terminology 1-2
- Architectural and functional overview 1-3
- Configuration overview 1-6
- What's next after configuration? 1-8

Important terminology

In this document, “IP Manager” collectively represents EMC® Smarts® IP Availability Manager (IP Availability Manager), EMC Smarts IP Performance Manager (IP Performance Manager), EMC Smarts IP Server Performance Manager (IP Server Performance Manager), and EMC Smarts IP Discovery Manager (IP Discovery Manager).

Note: IP Server Performance Manager is an enhanced version of IP Performance Manager and is available through a feature license.

What do system and device mean?

The term "system," as used in this document, is a generic term representing a computer-based network entity, such as a switch, router, hub, bridge, or host. The term "device," as used in this document, has essentially the same meaning as system except that "device" also conveys the sense of a specific model, such as a specific model of switch, router, hub, bridge, or host.

What is a Domain Manager?

IP Availability Manager, IP Performance Manager, and IP Server Performance Manager are examples of EMC Smarts Domain Managers, where a Domain Manager is a service-assurance application associated with a particular type of information-technology domain (networks, systems, applications, application services). For IP Availability Manager, IP Performance Manager, and IP Server Performance Manager, the domain is IP network transport. Each Domain Manager is autonomous in the sense that it maintains its own data models, repository, and problem signatures, and monitors and analyzes the discovered objects in its own domain.

IP Discovery Manager performs the same discovery as IP Performance Manager, but it does not perform any analysis on the discovered objects. Its purpose is to provide a database of discovered network objects and their connectivity to any application requiring such information.

What does modeled topology mean?

The IP Manager uses EMC Smarts ICIM class models to create (within its repository) instances of the managed network systems (switches, routers, hubs, bridges, hosts), their dependencies, their internal components, and their connections. The "modeled topology" mirrors the real topology in the managed network.

Architectural and functional overview

Figure 1-1 identifies the components of a typical IP Manager deployment and shows the flow of information among the components.

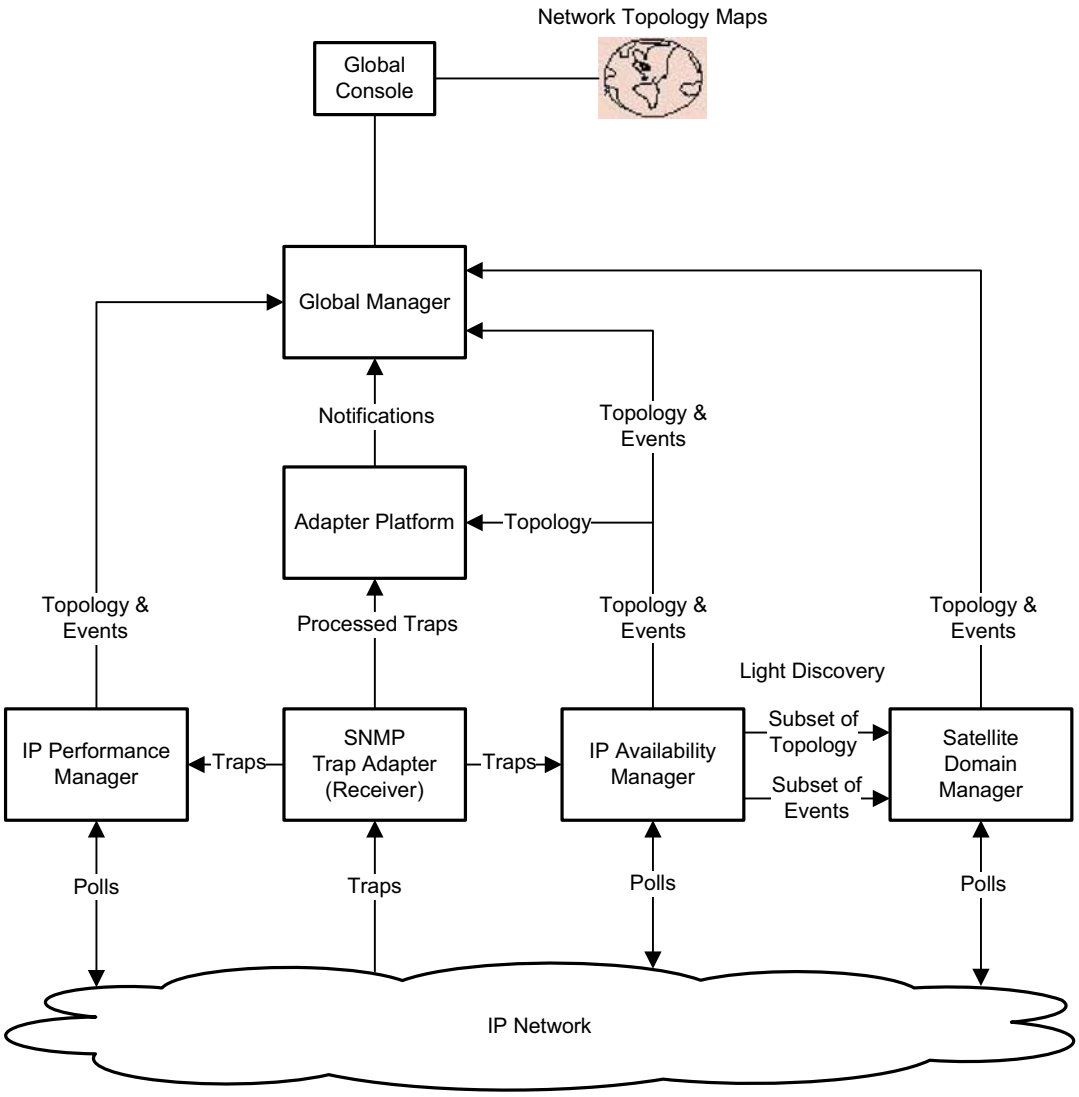


Figure 1-1 **IP Manager Deployment**

Note: In Figure 1-1, running separate instances of IP Availability Manager and IP Performance Manager is necessary if the two applications are running on different hosts. When running on the same host, a more common configuration would be IP Availability Manager and IP Performance Manager running as a single process.

IP Manager

IP Availability Manager and IP Performance Manager discover physical and logical Layer 2 and Layer 3 network objects in multi-vendor, switched, and routed networks. Together, they monitor and analyze network connectivity and performance and send network topology and event information to the EMC Smarts Service Assurance Manager (Global Manager).

Note: IP Discovery Manager can also interface with the Global Manager.

In addition, depending on the configuration, IP Availability Manager also sends network topology to the EMC Smarts Service Assurance Manager Adapter Platform (Adapter Platform) and to satellite Domain Managers.

Satellite Domain Manager (optional)

A satellite Domain Manager refers to any of the EMC Smarts Domain Managers that import topology, topology updates, and events from IP Availability Manager. Satellite Domain Managers include:

- EMC Smarts Application Connectivity Monitor
- EMC Smarts MPLS Manager
- EMC Smarts Multicast Manager
- EMC Smarts Network Protocol Manager for BGP
- EMC Smarts Network Protocol Manager for EIGRP
- EMC Smarts Network Protocol Manager for IS-IS
- EMC Smarts Network Protocol Manager for OSPF
- EMC Smarts VoIP Manager

IP Availability Manager Extension for NAS supplies additional event information to EMC Smarts Storage Insight for Availability.

Note: A feature license controls whether IP Availability Manager Extension for NAS software is available for configuration. With this extension, IP Availability Manager can discover and monitor EMC Celerra® NAS gateways and diagnose network-attached storage (NAS) connectivity problems.

Note: Similarly, a feature license controls whether IP Availability Manager Extension for IPv6 software is available for configuration. With this extension, IP Availability Manager can discover and monitor IPv6 addresses and diagnose IPv6 connectivity problems.

Global Manager (mandatory)

The EMC Smarts Service Assurance Manager (Global Manager) integrates the topology and event information imported from IP Availability Manager, IP Performance Manager, and the satellite Domain Manager and relates the information to services and customers. It also provides cross-domain and end-to-end impact analysis.

The Global Manager displays the topology, event, and impact information through the Global Console.

Adapter Platform (optional)

The EMC Smarts Service Assurance Manager Adapter Platform (Adapter Platform) imports and normalizes topology from IP Availability Manager, and imports and normalizes events of interest from EMC Smarts Adapters such as the SNMP Trap Adapter. Normalize means to convert topology or event information into a common form understood by the Global Manager. The normalized event information is transferred as EMC Smarts notifications to the Global Manager.

An Adapter Platform feature, called the Topology Importer, collects IP addresses and hostnames from IP Availability Manager so that the Adapter Platform can accurately place events in their topological context. The Adapter Platform updates this list as the information changes in IP Availability Manager.

Global Console (mandatory)

The Global Console provides a graphical user interface for configuration and administration of Global Managers, Adapter Platforms, Domain Managers, and externally running Adapters such as the SNMP Trap Adapter.

When attached to the Global Manager, a user can browse the network protocol topology in various forms, including maps, and to view notifications about events that impact availability and performance.

EMC Smarts Broker (mandatory)

The EMC Smarts Broker facilitates Global Console connections to the component applications in an EMC Smarts system deployment.

When a user starts a Global Console process, the process connects to the Broker, and the Broker displays a window through which the user views and selects any component application registered with the Broker. After the user selects an application, the Global Console connects to the application and disconnects from the Broker.

Configuration overview

The following sections highlight the configuration tasks associated with the setup of an IP Manager deployment. These tasks involve configuring the IP Manager, Global Manager, Adapter Platform, and satellite Domain Manager applications that are part of the deployment.

Some of the configuration tasks are performed before the applications are started, such as the editing of configuration files, while others are performed through the Global Console when the applications are up and running.

IP Manager configuration tasks

Configuring the IP Manager consists of the following tasks:

- Configure the IP Manager to discover the network systems
- Configure the IP Manager to monitor the discovered network systems
- Enable light discovery for the satellite Domain Managers
- Create device access credential sets for the satellite Domain Managers
- Create IP tag filters to enable the IP Manager to model overlapping IP addresses in its repository

This guide provides the procedures for performing these tasks.

Global Manager configuration tasks

In addition to performing the configuration and administration tasks common to all Global Managers, you perform the following additional task to set up the Global Manager in an IP Manager deployment:

- Configure the Global Manager to communicate with the IP Manager and the satellite Domain Managers

Chapter 3, "Setting Up Communication Links," provides the procedure for performing this task. Procedures for configuring all aspects of the Global Manager are given in the *EMC Smarts Service Assurance Manager Configuration Guide*.

Adapter Platform configuration tasks

In addition to performing the configuration and administration tasks common to all Adapter Platforms, you perform the following additional tasks to set up the Adapter Platform in an IP Manager deployment:

- Configure the Adapter Platform Topology Importer feature to import topology from the IP Manager (specifically, from IP Availability Manager)
- Configure the SNMP Trap Adapter to forward trap messages to the IP Manager

Procedures for configuring all aspects of the Adapter Platform and the SNMP Trap Adapter are given in the *EMC Smarts Service Assurance Manager Adapter Platform User's Guide*.

Satellite Domain Manager configuration tasks

In addition to performing the configuration and administration tasks common to all Domain Managers, you perform the following additional task to set up a satellite Domain Manager in an IP Manager deployment:

- Add the IP Manager (specifically, IP Availability Manager) as a source, to import initial topology and device access credential sets (if required) from the IP Manager.

Procedures for configuring all aspects of a satellite Domain Manager are given in the user documentation set for the particular Domain Manager.

What's next after configuration?

Upon configuring your IP Manager deployment, you are ready to initiate discovery. To understand, prepare, and initiate discovery, see the *EMC Smarts IP Management Suite Discovery Guide*.

2

Setting Configuration Parameters

This chapter identifies the IP Manager configuration parameters relevant to discovery, describes their settings, and provides instructions for changing their settings. It consists of the following sections:

- User configuration parameters 2-2
- Methods of modifying user configuration parameters 2-3
- Description of discovery.conf 2-6
- Description of tpmgr-param.conf 2-11
- Description of name-resolver.conf 2-15
- Description of partition.conf 2-16
- Description of user-defined-connections.conf 2-17
- Fine tuning ICMP and SNMP polling related parameters 2-19
- Configuring IP tagging for backward compatibility 2-21
- Enabling light discovery for satellite Domain Managers 2-22
- Enabling the concatenation of InterfaceAlias fields in ERX devices 2-24
- Enabling the logging of failed DNS lookups during discovery 2-26
- Enabling the SystemRestarted and ConfigChange events 2-28
- Changing the IPSec short pending list interval 2-37

User configuration parameters

A major part of configuring the IP Manager involves setting user configuration parameters in the files identified in Table 2-1.

Table 2-1 User configuration files for the IP Manager

Directory under BASEDIR	Filename	Description
smarts/conf/discovery	discovery.conf	File in which you fine tune configurations to improve the performance of IP Manager discovery.
	tpmgr-param.conf	File in which you customize IP Manager discovery.
	name-resolver.conf	File in which you control how the IP Manager names a discovered system.
	partition.conf	File in which you specify display names for partitions.
	user-defined-connections.conf	File in which you specify connections between discovered network objects.
smarts/conf/trapd	trapd.conf (See "IP trap adapter's configuration file: trapd.conf" on page 3-8.)	File in which you configure the IP trap adapter.
smarts/conf/notifier	trap-notify.conf (See "IP trap notification adapter's configuration file: trap-notify.conf" on page 3-17.)	File in which you configure the IP trap notification adapter.

Methods of modifying user configuration parameters

There are two methods to modify user configuration parameters:

1. Edit the user configuration files and modify configuration parameters.
2. Issue commands to temporarily modify configuration parameters when the IP Manager is running.

The second method can be used to temporarily modify most but not all user configuration parameters. Some configuration changes can only be made by modifying the parameter settings in the configuration files and then restarting the IP Manager. Every time that the IP Manager starts, it sets its configurations in accordance to the settings in the user (and other) configuration files.

Editing configuration files to modify configuration parameters

EMC Smarts provides the sm_edit utility to ensure that modified files are always saved to the appropriate local area and that original copies of the files remain unchanged. Figure 2-1 previews the operation of the sm_edit utility.

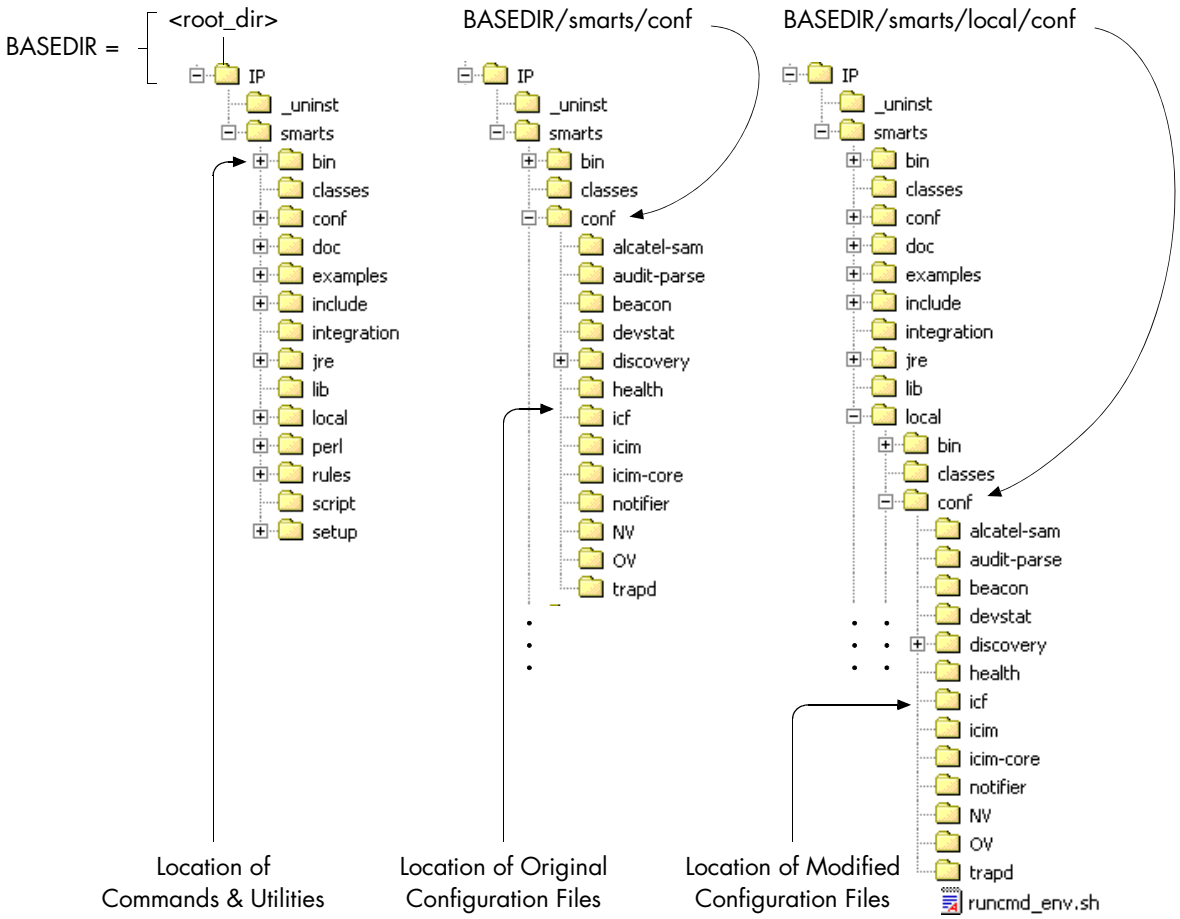


Figure 2-1 **Preview of how the sm_edit utility works**

Note: All EMC Smarts product suites use the same basic installation directory structure shown in Figure 2-1.

To invoke the sm_edit utility from the command line, go to the BASEDIR/smarts/bin directory and specify the path and the name of the file (relative to the BASEDIR/smarts directory) that you want to edit. For example,

```
sm_edit conf/discovery/discovery.conf
```

opens a local copy of the discovery.conf file in BASEDIR/smarts/local/conf/discovery (or an original copy of the discovery.conf file in BASEDIR/smarts/conf/discovery if no local copy exists) in a text editor and saves the modified version of the file to the BASEDIR/smarts/local/conf/discovery directory.

You can use the sm_edit utility to edit any text file (configuration file, rule set file, tool script, template) in the BASEDIR/smarts (or BASEDIR/smarts/local) directory. Because sm_edit assumes a starting point of BASEDIR/smarts (or BASEDIR/smarts/local), you specify the path and the name of the text file beginning with the directory name (conf, rules, script) under the BASEDIR/smarts directory.

The *EMC Smarts System Administration Guide* provides additional information about the sm_edit utility.

Note: Original versions of files may be changed or updated as part of an EMC Smarts software upgrade. However, files located in the BASEDIR/smarts/local directory are retained during an upgrade.

Issuing commands to modify configuration parameters

You can issue any of the following commands from the BASEDIR/smarts/bin directory to temporarily set user configuration parameters for the IP Manager:

- dmctl -s <IP Manager instance name> [-b <broker>] put ICF_TopologyManager::ICF-TopologyManager::<configuration parameter name> <value>

  For example:

  ```
  dmctl -s INCHARGE-AM put
     ICF_TopologyManager::ICF-TopologyManager::numberProbeThreads 12
  ```

  Note: The option -b <broker> is used to specify the host and port (<host:port>) location of the Broker if the IP Manager is attached to a Broker other than "localhost:426."

  You use this command to temporarily overwrite parameter settings specified in the discovery.conf file and the name-resolver.conf file. These parameters are defined as attributes of the ICF-TopologyManager.

- dmctl -s <IP Manager instance name> [-b <broker>] invoke ICF_TopologyManager::ICF-TopologyManager insertParameter <configuration parameter name> <value>

  For example:

  ```
  dmctl -s INCHARGE-AM invoke
     ICF_TopologyManager::ICF-TopologyManager insertParameter
     DisableMPLSDiscovery false
  ```

  You use this command to temporarily overwrite parameter settings in the tpmgr-param.conf file. These parameters are entries for a table defined inside the ICF-TopologyManager.

- sm_tpmgr -s <IP Manager instance name> [-b <broker>]
  --load-conf=<configuration file name>

  For example:

  ```
  sm_tpmgr -s INCHARGE-AM --load-conf=user-defined-connections.conf
  ```

  You use this command to reload the parameters in any of the IP Manager user configuration files.

Using these commands will overwrite most parameter settings in the user configuration files but not all; for example, you cannot use these commands to overwrite the DisableOverlappingIP setting in the tpmgr-param.conf file. Even after manually modifying the DisableOverlappingIP setting in the tpmgr-param.conf file and reloading the file using sm_tpmgr, your change will not take effect.

For parameter settings that cannot be overwritten using the dmctl or the sm_tpmgr commands, use the sm_edit utility to modify the settings in the appropriate configuration files and then restart the IP Manager. Table 2-2 identifies the parameters for which a restart is required.

Table 2-2 Parameter setting changes requiring a restart of the IP Manager

Configuration file	Parameter
discovery.conf	HSRPEnabled
	HostResourcesEnabled
	VirtualRouterEnabled
tpmgr-param.conf	DisableOverlappingIP
	OITrapAdapterPort
	UserDefinedConnectionsOverwrite
name-resolver.conf	none
partition.conf	none
user-defined-connections.conf	none
trapd.conf	PORT
trap-notify.conf	none

Description of discovery.conf

The discovery.conf file contains configuration parameters that you edit to improve the performance of IP Manager discovery. The parameters are a subset of attributes defined for the ICF-TopologyManager.

For parameters such as SNMPPort, you can override the value specified in the discovery.conf file by specifying an alternate value in a discovery filter, a seed file, or an Add Agent command.

Table 2-3 describes the parameters in the discovery.conf file. The parameters and their values are case-sensitive.

Table 2-3 Parameters in the discovery.conf file

Parameter	Value	Description
ipExcludeList	Not specified	Use ipExcludeList to prevent certain device (loopback) IP addresses from being added to the topology. By not adding the IP addresses to the topology, autodiscovery will not discover the systems associated with the IP addresses. You can use wildcards, described in Appendix A, "Wildcard Patterns," to specify matching patterns in the IP exclude patterns. Any IP address that matches an exclude pattern will NOT be added to the topology. The following example excludes IP addresses from the specified IP network: • ipExcludeList += "10.10.9.*"
ShowPendingNONSNMP	TRUE, FALSE Default: FALSE	Controls whether non-SNMP systems found by autodiscovery or topology import are added to the IP Manager's Pending Devices list or are discarded. A non-SNMP system is one that responds to ICMP polls but does not respond to SNMP polls. • A value of TRUE indicates that non-SNMP systems are added to the Pending Devices list. • A value of FALSE indicates that non-SNMP systems are discarded. **Note:** Setting the value to TRUE and having autodiscovery enabled can result in adding many hosts, workstations, and other non-SNMP systems to the Pending Devices list.
LogDiscoveryProgress	TRUE, FALSE Default: FALSE	Controls whether verbose logging is used to log discovery-related messages to the IP Manager's log file. • A value of TRUE enables verbose logging. • A value of FALSE disables verbose logging. **Note:** This parameter is used mainly for debugging purposes.
defaultSNMPPort	0 to 65535 Default: 161	Specifies the default port used by the IP Manager for SNMP discovery and monitoring. You can override this value in a discovery filter, a seed file, or an Add Agent command. **Note:** If 0 is specified, port 161 is used.
defaultRetries	Default: 5 retries	Number of SNMP retry polls to use during Phase 3 of discovery. Phase 3 of discovery is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide*. "SNMP polling discovery settings" on page 2-19 provides recommendations about changing this value.

Table 2-3 Parameters in the discovery.conf file (continued)

Parameter	Value	Description
defaultTimeout	Default: 1000 milliseconds	Timeout for an SNMP poll during Phase 3 of discovery. Phase 3 of discovery is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide*. "SNMP polling discovery settings" on page 2-19 provides recommendations about changing this value.
defaultICMPAutoRetries	Default: 3 retries	Number of ICMP polls to try during Phase 1 of discovery. Phase 1 of discovery is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide*. "ICMP polling discovery settings" on page 2-19 provides recommendations about changing this value.
defaultICMPAutoTimeout	Default: 500 milliseconds	Timeout for an ICMP poll during Phase 1 of discovery. Phase 1 of discovery is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide*. "ICMP polling discovery settings" on page 2-19 provides recommendations about changing this value.
defaultSNMPAutoRetries	Default: 3 retries	Number of SNMP polls to try during Phase 1 of discovery. Phase 1 of discovery is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide*. "SNMP polling discovery settings" on page 2-19 provides recommendations about changing this value.
defaultSNMPAutoTimeout	Default: 500 milliseconds	Timeout for an SNMP poll during Phase 1 of discovery. Phase 1 of discovery is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide*. "SNMP polling discovery settings" on page 2-19 provides recommendations about changing this value.
numberProbeThreads	Default: 10 threads	Number of discovery threads used by the discovery process. If you are experiencing slow response SNMP discovery polling and your current CPU usage is relatively low (say 5 to 10 percent) consider increasing the number of discovery threads to improve your discovery performance. Increasing this number is one of the best and simplest methods for improving discovery. Increasing threads to 20, 30 or even 50 discovery threads is acceptable, but keep in mind that more threads may require additional or more capable CPUs and that the requirements during discovery will increase. Increase threads conservatively while monitoring SNMP Processing Statistics changes, as described in the *EMC Smarts IP Management Suite Deployment Guide*.
ICMPSleepTime	0 to 100 milliseconds Default: 0	Interval between successive ICMP poll requests. During the initial discovery phase, the IP Manager sends large bursts of ICMP polls. If ICMP has a low priority, the network is congested, or the routers on the path are experiencing high processor utilization, a number of these ICMP packets may be dropped. As a result, the IP Manager receives a limited number of ICMP responses, which causes a number of systems that might otherwise be successfully discovered to be placed on the Pending Devices list as UNDISCOVERED. Other symptoms of packet loss include DiscoveryError notifications for systems that were previously discovered successfully and the full topology is not discovered after two full discoveries. Increasing the sleep time increases the amount of time between ICMP polls. You can specify a value between 0 and 100 milliseconds. A value of 100 milliseconds translates to 10 ICMP polls per second. Specifying a long sleep time can significantly prolong the discovery process. Estimate the total additional time by multiplying the sleep time by the number of IP addresses to be processed.

Table 2-3 Parameters in the discovery.conf file (continued)

Parameter	Value	Description
defaultReadCommunity	Default: public	Read-only community string used by the IP Manager when no community string is specified in the Community Strings field in the Topology tab of the Domain Manager Administration Console. The Community Strings field is described in Chapter 11, "Working with the Topology," of the *EMC Smarts IP Management Suite Discovery Guide*. If you do not want the IP Manager to use a default community string during discovery, you must remove "public" from the Community Strings field and from this parameter.
MaximumCommunities	Default: 4 strings	Number of read-only community strings that can be specified in the Community Strings field in the Topology tab of the Domain Manager Administration Console. The Community Strings field is described in Chapter 11, "Working with the Topology," of the *EMC Smarts IP Management Suite Discovery Guide*. You can change this value to address networks that have more that four different read-only community strings.
MaximumHostBits	Default: 8 bits	Number of host bits in the netmask used by autodiscovery to find additional discovery candidate systems. The default value of 8 corresponds to a netmask of 255.255.255.0. If, for example, your network includes systems with a netmask of 255.255.254.0, these systems will not be found using autodiscovery. Modify the default value of the MaximumHostBits parameter with caution because as you increase the value, so do you increase the number of ICMP polls generated during every discovery. It is recommended that you set the value only as high as needed. If necessary, modify the value during the initial topology discovery and once the devices are discovered lower the value so that subsequent pending and full discoveries do not attempt to look beyond the default netmask (this will not affect systems already discovered). For example, the default value of 8 provides a maximum of 254 host IP addresses. A value of 16 provides a maximum of 65,534 host IP addresses. 192.168.1.1/24 = 8 host bits 192.168.1.1/18 = 14 host bits 192.168.1.1/16 = 16 host bits 192.168.1.1/8 = 24 host bits Valid values range from 2 - 24. If a value greater than 24 is specified, a value of 8 is used.

Table 2-3 Parameters in the discovery.conf file (continued)

Parameter	Value	Description
AccessAddressFormat	AUTOADDRESS, LOOPBACKADDRESS Default: AUTOADDRESS	Determines how the discovery process obtains an IP address to communicate with discovered systems. This address is used for ICMP and SNMP polls during discovery and for monitoring and analysis. Valid values are AUTOADDRESS and LOOPBACKADDRESS. AUTOADDRESS means that all the IP addresses of a device are polled using ICMP. If a device has more than ten IP addresses, only ten of the IP addresses are polled using SNMP. The addresses to be polled by SNMP are chosen as follows: • IP address is managed and is not a duplicate • InterfaceMode of the IP, if any, is NORMAL • If the system has more than ten IP addresses, the first ten IP addresses are chosen in the following manner: IP addresses of type SOFTWARELOOPBACK, ETHERNET, TOKENRING, and GENERIC are chosen first in the order that the IP addresses appear in the IP address table (ipAddrTable). IP addresses of type GENERIC must not have a netmask of 255.255.255.252. The remaining IP addresses are chosen in the order that they appear in the IP address table. LOOPBACKADDRESS means that the discovery process uses the loopback address for both ICMP and SNMP for monitoring and discovery. (MIB-II ifType 24). Note that discovery cannot determine the loopback address until after the system has been discovered. If a qualified IP address is not found, the last known SNMP address is used and a DiscoveryError is notified. The DiscoveryError contains the message: Qualified access address not found.
DisplayNameFormat	AUTOASSIGNED, MIBIISYSNAME Default: AUTOASSIGNED	Determines how the discovery process populates the DisplayName attribute of a system. Valid values are AUTOASSIGNED and MIBIISYSNAME. AUTOASSIGNED means that the DisplayName of a system is set to the value of the Name attribute of the system. (Using the NameFormat parameter, described in the table "Parameters in the name-resolver.conf file" on page 2-15, you can control how the value of the Name attribute is determined.) Also, for AUTOASSIGNED, you can change the value of DisplayName using dmctl or an Adapter Scripting Language (ASL) script; the value of DisplayName is not retained across subsequent discoveries. MIBIISYSNAME means that the DisplayName of a system is set to the value of the MIB-II sysName variable. Any change to the DisplayName using dmctl or an ASL script is not retained across subsequent discoveries.
HSRPEnabled	TRUE, FALSE Default: TRUE	Enables or disables Hot Standby Router Protocol (HSRP) analysis. • A value of TRUE enables HSRP analysis. • A value of FALSE disables HSRP analysis.
HostResourcesEnabled	TRUE, FALSE Default: TRUE	Enables or disables host resources discovery and monitoring. • A value of TRUE enables host resources discovery and monitoring. • A value of FALSE disables host resources discovery and monitoring.
VirtualRouterEnabled	TRUE, FALSE Default: TRUE	Enables or disables virtual router discovery and monitoring. • A value of TRUE enables virtual router discovery and monitoring. • A value of FALSE disables virtual router discovery and monitoring.
MinimumDemandGet	TRUE, FALSE Default: FALSE	Controls whether to allow only minimum Demand Get requests or to allow all Demand Get requests. • A value of TRUE allows only minimum Demand Get requests. • A value of FALSE allows all Demand Get requests. Setting this parameter to TRUE optimizes discovery performance by avoiding unnecessary Demand Get requests.

Table 2-3 Parameters in the discovery.conf file (continued)

Parameter	Value	Description
DuplexAssumed	TRUE, FALSE Default: FALSE	Controls how to set the duplex mode for network adapters (ports or interfaces) in the topology. Valid values are TRUE and FALSE. • A value of TRUE indicates that duplex mode is assumed for all network adapters regardless of whether duplex mode can be determined by checking the MIBs. • A value of FALSE indicates that duplex mode is set only for the network adapters for which duplex mode can be determined by checking the MIBs. As further clarification, a value of TRUE means that the IP Manager will calculate current utilization for all ports and interfaces in the topology regardless of whether their duplex mode can be determined by checking the MIBs. For any port or interface whose duplex mode cannot be determined by checking the MIBs, the IP Manager sets the port's/interface's DuplexMode attribute to FULLDUPLEX and its DuplexSource attribute to ASSUMED. The one exception is the 10 megabit (Mb) Ethernet adapter, for which the IP Manager sets the adapter's DuplexMode attribute to HALFDUPLEX. A value of FALSE means that the IP Manager will calculate only current utilization for those ports or interfaces that can be determined as full-duplex or half-duplex by reading the Enterprise MIB, ETHERLIKE-MIB, and Neighbor MIB. Any port or interface having an unspecified duplex mode setting will be ignored. Be aware that calculating current utilization for all ports and interfaces (DuplexAssumed = TRUE) requires considerable system resources. Ensure that your server is sized properly to accommodate the size of your environment, as described in the *EMC Smarts IP Management Suite Deployment Guide*. Chapter 3, "Network Objects and Their Failures," in the *EMC Smarts IP Performance Manager and Server Performance Manager User's Guide* provides more information about duplex mode and current utilization calculations.
autoReprobe_short	TRUE, FALSE Default: FALSE	Enables or disables short discovery for IPSec tunnels. • A value of TRUE enables short discovery. • A value of FALSE disables short discovery. Short discovery for IPSec tunnels is described in Chapter 2, "Discovery Process," in the *EMC Smarts IP Management Suite Discovery Guide*.
importExternalTagsConnections	TRUE, FALSE Default: FALSE	Controls whether MPLS tags and connections are imported by the IP Manager from MPLS Manager. • A value of TRUE indicates that the MPLS tags and connections are imported by the IP Manager. • A value of FALSE indicates that the MPLS tags and connections are not imported by the IP Manager.
IPv6DiscoveryEnabled	TRUE, FALSE Default: TRUE	Enables or disables the discovery of IPv6 addresses. • A value of TRUE enables the discovery of IPv6 addresses. • A value of FALSE disables the discovery of IPv6 addresses.

Description of tpmgr-param.conf

The tpmgr-param.conf file contains configuration parameters that you edit to customize IP Manager discovery. The parameters are a subset of table entries defined for the ICF-TopologyManager.

You use the parameters in this file to accomplish such tasks as disabling the discovery of certain interfaces or ports, enabling the discovery of topology collection sets for satellite Domain Managers, and changing the operation of the IP Manager to be backward compatible with earlier versions of satellite Domain Managers.

Table 2-4 describes the parameters in the tpmgr-param.conf file. The parameters and their values are case-sensitive.

Table 2-4 Parameters in the tpmgr-param.conf file

Parameter	Value	Description
EnableShadowIP	TRUE, FALSE Default: TRUE	Controls whether shadow IP objects and shadow IPNetwork objects are created. • A value of TRUE indicates that shadow objects are created. • A value of FALSE indicates that shadow objects are not created. This parameter is used for backward compatibility so that the current release of the IP Management Suite is able to work with earlier versions of satellite Domain Managers.
Interface pattern filters	Not applicable	Filters that disallow the creation of certain interfaces for certain discovered devices. Interface pattern filters are described in Chapter 4, "Excluding Device-Access Interfaces."
EnableTunnelHeuristic	TRUE, FALSE Default: FALSE	Controls whether tunnel connections are processed. • A value of TRUE indicates that tunnel connections are processed. • A value of FALSE indicates that tunnel connections are not processed.
EnableClearByLinkUp	TRUE, FALSE Default: FALSE	Controls whether LinkUp traps set the OperStatus to UP. • A value of TRUE indicates that LinkUp traps set the OperStatus to UP. • A value of FALSE indicates that LinkUp traps do not set the OperStatus to UP.
DisableProtocolDiscovery	TRUE, FALSE Default: TRUE	Controls whether the IP Manager discovers topology for Network Protocol Manager. • A value of TRUE indicates that the IP Manager does not discover topology for Network Protocol Manager. • A value of FALSE indicates that the IP Manager does discover topology for Network Protocol Manager.
DisableMPLSDiscovery	TRUE, FALSE Default: TRUE	Controls whether the IP Manager discovers topology for MPLS Manager. • A value of TRUE indicates that the IP Manager does not discover topology for MPLS Manager. • A value of FALSE indicates that the IP Manager does discover topology for MPLS Manager.
DisableMulticastDiscovery	TRUE, FALSE Default: TRUE	Controls whether the IP Manager discovers topology for Multicast Manager. • A value of TRUE indicates that the IP Manager does not discover topology for Multicast Manager. • A value of FALSE indicates that the IP Manager does discover topology for Multicast Manager.

Table 2-4 Parameters in the tpmgr-param.conf file (continued)

Parameter	Value	Description
DisableOverlappingIP	TRUE, FALSE Default: TRUE	Controls whether the IP Manager discovers MPLS-related VRF IP addresses. • A value of TRUE indicates that the IP Manager does not discover VRF IP addresses. • A value of FALSE indicates that the IP Manager does discover VRF IP addresses.
UseCDPEthernetConnection	TRUE, FALSE Default: TRUE	Controls whether Cisco Discovery Protocol (CDP) connections are created for Ethernet interfaces. • A value of TRUE indicates that CDP connections are created for Ethernet interfaces. • A value of FALSE indicates that CDP connections are not created for Ethernet interfaces.
UserDefinedConnectionsOverwrite	TRUE, FALSE Default: FALSE	Controls whether user-defined connections overwrite discovered connections. • A value of TRUE indicates that user-defined connections overwrite discovered connections. • A value of FALSE indicates that user-defined connections do not overwrite discovered connections.
masklistPattern	Default: {32,30,29,28,27,26,25,24,16,8}	Range of subnetworks to be tried to find the best routing match.
EnableARPHeuristicForVPNIP	TRUE, FALSE Default: FALSE	Controls whether the ARP heuristic is used to find VPN-IPs (VRF IPs) not available in the IP address table (ipAddrTable). • A value of TRUE indicates that the ARP heuristic is used to find VPN-IPs not available in the IP address table. • A value of FALSE indicates that the ARP heuristic is not used to find VPN-IPs not available in the IP address table.
IgnoreIPForPersistentMgmtState	TRUE, FALSE Default: FALSE	Enables or disables the IP set comparison among the new and old candidates. • A value of TRUE enables the IP set comparison among the new and old candidates. • A value of FALSE disables the IP set comparison among the new and old candidates.
AllowNonPrivateIPAsName	TRUE, FALSE Default: TRUE	Controls whether a non-private IP is to be used as the name for the NameFormat TM_USENONPRIVATEIP parameter (an AutoNameOrder parameter described in the table "Parameters in the name-resolver.conf file" on page 2-15) in case non-private IPs cannot be resolved. • A value of TRUE indicates that a non-private IP is to be used as the name for the NameFormat TM_USENONPRIVATEIP parameter in case non-private IPs cannot be resolved. • A value of FALSE indicates that a non-private IP is not to be used as the name for the NameFormat TM_USENONPRIVATEIP parameter in case non-private IPs cannot be resolved.
AllowPrivateIPAsName	TRUE, FALSE Default: FALSE	Controls whether a private IP is to be used as the name for the NameFormat TM_USEPRIVATEIP parameter (an AutoNameOrder parameter described in the table "Parameters in the name-resolver.conf file" on page 2-15) in case private IPs cannot be resolved. • A value of TRUE indicates that a private IP is to be used as the name for the NameFormat TM_USEPRIVATEIP parameter in case private IPs cannot be resolved. • A value of FALSE indicates that a private IP is not to be used as the name for the NameFormat TM_USEPRIVATEIP parameter in case private IPs cannot be resolved.

Table 2-4 Parameters in the tpmgr-param.conf file (continued)

Parameter	Value	Description
AllowAllLocalInterfaceIPs	TRUE, FALSE Default: FALSE	Controls whether a local IP is to be discovered from the ipNetToMediaPhysAddress MIB. • A value of TRUE indicates that a local IP is to be discovered from the ipNetToMediaPhysAddress MIB. • A value of FALSE indicates that a local IP is not to be discovered from the ipNetToMediaPhysAddress MIB.
DiscoverCiscoTrunkPortVLANs	TRUE, FALSE Default: TRUE	Enables or disables the discovery of VLANs for which trunk ports are enabled. • A value of TRUE enables the discovery of VLANs for which trunk ports are enabled. • A value of FALSE disables the discovery of VLANs for which trunk ports are enabled.
PropagateVRIfAlias	TRUE, FALSE Default: FALSE	Enables or disables the concatenation of the InterfaceAlias fields of the virtual routers on a Juniper ERX physical router into the InterfaceAlias field of the main router. • A value of TRUE enables the concatenation of the InterfaceAlias fields. • A value of FALSE disables the concatenation of the InterfaceAlias fields. "Enabling the concatenation of InterfaceAlias fields in ERX devices" on page 2-24 clarifies the use of this parameter.
LogDNSFailure	TRUE, FALSE Default: FALSE	Enables or disables the logging of error messages for a device (loopback) IP address that cannot be resolved to a name by the Domain Name System (DNS) during autodiscovery. • A value of TRUE enables the logging of error messages for a device IP address that cannot be resolved to a name through DNS during autodiscovery. • A value of FALSE disables the logging of error messages for a device IP address that cannot be resolved to a name through DNS during autodiscovery. "Enabling the logging of failed DNS lookups during discovery" on page 2-26 clarifies the use of this parameter.
ERXRtrUseDefaultComm	TRUE, FALSE Default: FALSE	Enables or disables the appending of '@default' to the community string used to query ERX physical routers. • A value of TRUE enables the appending of '@default' to the community string used to query ERX physical routers. • A value of FALSE disables the appending of '@default' to the community string used to query ERX physical routers.
EnableERXMainRouterIFDiscovery	TRUE, FALSE Default: TRUE	Enables or disables interface discovery for the main router of Juniper ERX physical routers. • A value of TRUE enables interface discovery for the ERX main router. • A value of FALSE disables interface discovery for the ERX main router.
OITrapAdapterHost	localhost	The name or IP address of the host system where the SNMP Trap Adapter configured as a trap receiver is running. "Enabling the SystemRestarted event" on page 2-29 and "Enabling the ConfigChange event" on page 2-33 clarify the use of this parameter.
OITrapAdapterPort	162	The port on the host system where the SNMP Trap Adapter configured as a trap receiver is running. "Enabling the SystemRestarted event" on page 2-29 and "Enabling the ConfigChange event" on page 2-33 clarify the use of this parameter.

Table 2-4 Parameters in the tpmgr-param.conf file (continued)

Parameter	Value	Description
EnableConfigChange	TRUE, FALSE Default: FALSE	Controls whether the ConfigChange event results in the adding of the device that caused the event to the Pending Devices list. • A value of TRUE indicates that the ConfigChange event results in the adding of the device that caused the event to the Pending Devices list. • A value of FALSE indicates that the ConfigChange event does not result in adding the device that caused the event to the Pending Devices list. "Enabling the ConfigChange event" on page 2-33 clarifies the use of this parameter.
EnableSystemRestart	TRUE, FALSE Default: FALSE	Controls whether the SystemRestarted event results in the adding of the device that caused the event to the Pending Devices list. • A value of TRUE indicates that the SystemRestarted event results in the adding of the device that caused the event to the Pending Devices list. • A value of FALSE indicates that the SystemRestarted event does not result in the adding of the device that caused the event to the Pending Devices list. "Enabling the SystemRestarted event" on page 2-29 clarifies the use of this parameter.
EnableNonSlash30IPs	TRUE, FALSE Default: FALSE	Controls whether a Layer 2 (L2) network connection is created when CidrRoute is used as the neighbor probe. • A value of TRUE indicates that an L2 network connection is created when CidrRoute is used as the neighbor probe. • A value of FALSE indicates that an L2 network connection is not created when CidrRoute is used as the neighbor probe.
UseVPNIP	TRUE, FALSE Default: TRUE	Controls whether VPNIP and VPNIPNetwork objects or IP and IPNetwork objects are created for discovered MPLS-related VRF IP addresses. • A value of TRUE indicates that VPNIP and VPNIPNetwork objects are created for discovered MPLS-related VRF IP addresses. • A value of FALSE indicates that IP and IPNetwork objects are created for discovered MPLS-related VRF IP addresses. **Note:** The DisableOverlappingIP parameter must be set to FALSE for the UseVPNIP parameter to have any effect.

Description of name-resolver.conf

The name-resolver.conf file contains configuration parameters that you edit to control how the IP Manager names discovered systems. The parameters are a subset of attributes defined for the ICF-TopologyManager.

Table 2-5 describes the parameters in the name-resolver.conf file. The parameters and their values are case-sensitive.

Table 2-5 Parameters in the name-resolver.conf file

Parameter	Value	Description
NameFormat	TM_USEAUTONAME, TM_USESEEDNAME Default: TM_USEAUTONAME	Determines how discovered systems are named. Valid values are: • TM_USEAUTONAME The discovery process uses the parameter values in the AutoNameOrder section of this file as the source for system names. • TM_USESEEDNAME The discovery process uses the seed name as the system name. How the discovery process names a system is described in Chapter 3, "Name Resolution," of the *EMC Smarts IP Management Suite Discovery Guide*.
AutoNameOrder section		
TM_USELOOPBACK	0, 1, 2, 3, or 4 order priority Default: 0	Resolve loopback IP address: If a loopback interface IP address can be resolved to a name, that name will be used to name the system. By default, the default loopback interface type number is 24 (represents the “software loopback” interface type).
TM_USESYSNAME	0, 1, 2, 3, or 4 order priority Default: 1	Resolve MIB-II sysName: If the sysName can be resolved to at least one IP address and at least one of the resolved IP addresses resolves back to the sysName, the sysName will be used to name the system.
TM_USENONPRIVATEIP	0, 1, 2, 3, or 4 order priority Default: 2	Resolve non-private IP address: If at least one of the IP addresses is not a private IP address, the non-private IP address will be used to resolve the name. Once a name is resolved, the name will be used to name the system.
TM_USEPRIVATEIP	0, 1, 2, 3, or 4 order priority Default: 3	Resolve private IP address: If at least one of the IP addresses is a private IP address, the private IP address will be used to resolve the name. Once a name is resolved, the name will be used to name the system. By default, the private IP address pattern is 10.*|192.168.*|172.<16-31>.*, which is defined as: • Any IP address with the value of 10 in the first octet (10.*.*.*). • Any IP address with a value of 172 in the first octet and a value between 16 and 31 in the second octet (172.<16-31>.*.*). • Any IP address with a value of 192 in the first octet and a value of 168 in the second octet (192.168.*.*).
TM_USEAGENTADDRESS	0, 1, 2, 3, or 4 order priority Default: 4	Resolve SNMP agent address: If an SNMP agent address can be resolved to a name, that name will be used to name the system.

Description of partition.conf

You use the partition.conf file to assign display names to partitions formed by the IP Manager. When the topology is incomplete, the IP Manager creates partitions to allow analysis to proceed.

The partition.conf file is shown in Figure 2-2.

```
#
# partition.conf
#
# Users can specify DisplayName for partitions.
# The first column is a hostname or IP Address. The rest is treated
# as the DisplayName for the partition.
# You could have multiple entries for one partition. The last one
   wins.
#
# The following line set the DisplayName of the partition of which
   router1 is
# a member to "Routers in NYC"
# router1 Routers in NYC
#
# The following example shows how to set DisplayName of the partition
   that
# consists of a network device that hosts IP address 10.64.1.1
#
# 10.64.1.1 Servers in LA
```

Figure 2-2 User configuration file partition.conf

To name a Partition object, the IP Manager uses the value of the Name attribute for the object to set the value of the DisplayName attribute for the object. The IP Manager names a partition by assigning it the name Partition-<number>, starting with 0, and incrementing this value by one for each new partition.

The partition number is followed by the name of the Domain Manager that created the partition. For example, the name of the first partition created by and IP Availability Manager named INCHARGE-AM is Partition-0/INCHARGE-AM, the second is Partition-1/INCHARGE-AM, and so on.

To set the DisplayName of a partition, specify the hostname or IP address of a partition member followed by its name. You should specify a name or IP address that is already known to the IP Manager. Specify one hostname or IP address and name combination on each line.

One of the examples included in the partitions.conf file (Figure 2-2) sets the DisplayName of the partition of which a network system named “10.64.1.1” is a member to “Servers in LA”:

```
10.64.1.1 Servers in LA
```

If you specify multiple names for a partition, which is done by providing separate entries, the last name listed is used.

You do not need to restart the IP Manager after making a change to the partition.conf file. The IP Manager automatically loads the partition.conf file each time that it performs post-processing.

Description of user-defined-connections.conf

You use the user-defined-connections.conf file to add your own Layer 2 and Layer 3 connections to the managed topology. Doing so may be useful in cases where these connections are not discovered because of incomplete information in the MIB. If you know the necessary information, you can create these connections by specifying them in this file. The types of connections that are created include Cable, TrunkCable, and NetworkConnection.

The user-defined-connections.conf file is shown in Figure 2-3.

```
# This is a sample user-defined connections for the InCharge Server.
#
# A connection must contain FOUR fields separated by |
# - The first field is the local network device.
#   You can specify either a name or an IP address.
# - The second field is the network adapters on the local device.
#   It can be ifIndex or ifDescr or ifName as defined in MIB2.
#   It can also be n/m or n.m where n is the module number and m is
#   port number.
# - The third field is the remote network device.
#   You can specify either a name or an IP address.
# - The fourth field is the network adapters on the remote device.
#   It can be ifName or ifDescr or ifIndex as defined in MIB2.
#   It can also be n/m or n.m where n is the module number and m is
#   port number.
#
# The search for network adapter is performed in the order of
# ifName, ifDescr, ifIndex. If it fails to return a network adapter
# object, the specified string is used as interfaceKey/PortKey to
# find interface/port objects.
#
# To make your file more readable, you can
# include blank and comment lines.
# A comment line is one whose first character is "#".
#
# Following are a few examples. NOTE - the lines are commented, so the
# discovery won't really try to make these connections.
#
# 192.168.1.200|Serial0|router2.smarts.com|5|
# 10.64.1.1|2.4|Core-switch.smarts.com|2/4|
#
```

Figure 2-3 User configuration file user-defined-connections.conf

As indicated in Figure 2-3, a user-defined connection contains the following four fields, each separated by a pipe character (|):

- The first field is the local network. You can specify the local network by name or IP address.
- The second field is the network adapter on the local device. You can specify the network adapter by its ifIndex, ifDescr, or ifName, as defined in RFC 1213. You can also use the <n>/<m> or <n>.<m> syntax, where <n> is the module number, and <m> is the port number.
- The third field is the remote network device. You can specify the remote network device by name or IP address.

- The fourth field is the network adapter on the remote device. You can specify the network adapter by its ifIndex, ifDescr, or ifName, as defined in RFC 1213. You can also use the <n>/<m> or <n>.<m> syntax, where <n> is the module number, and <m> is the port number.

Note: Any objects specified in these four fields must be in the managed topology.

The IP Manager searches for the network adapter in the order of ifName, ifDescr, and ifIndex. If a network adapter is not found, the specified string is used as InterfaceKey/PortKey to find the network adapter.

The following are two examples of user-defined connections:

```
192.168.1.200|Serial0|router2.smarts.com|5|
10.64.1.1|2.4|Core-switch.smarts.com|2/4|
```

Note that the network adapter field for the remote device must end with a pipe character (|).

As with changes to the partition.conf file, you do not need to restart the IP Manager after making a change to the user-defined-connections.conf file. The IP Manager automatically loads the user-defined-connections.conf file each time that it performs post-processing.

Fine tuning ICMP and SNMP polling related parameters

The discovery.conf file contains several settings that you can edit to improve performance during topology discovery. However, changing these values beyond the recommended values can adversely affect the discovery process. These settings include:

- defaultICMPAutoRetries
- defaultICMPAutoTimeout
- defaultSNMPAutoRetries
- defaultSNMPAutoTimeout
- defaultRetries
- defaultTimeout

When a timeout occurs because of packet loss in the network, you should increase the number of retries. If the timeout occurs because the network is slow or the system is busy, you should increase the value of the timeout.

Note: The defaultICMPAutoRetries, defaultICMPAutoTimeout, defaultSNMPAutoRetries, and defaultSNMPAutoTimeout settings apply to discovery. These settings do not affect the operation of the ICMP poller or the SNMP poller that polls objects to perform analysis.

ICMP polling discovery settings

Before sending an SNMP request, the discovery process sends an ICMP poll to the system to make sure it is reachable. The defaultICMPAutoRetries and defaultICMPAutoTimeout control the ICMP poll used during the initial stage of discovery.

Retries is the number of times an ICMP poll is sent to a system after the initial attempt fails. The default value is three. Timeout is the amount of time that should elapse before considering that the previous ICMP poll has failed or timed out. The default value is 500 milliseconds.

The value of the timeout is constant, meaning that the timeout value stays the same between retries. In contrast, the value of the timeout for the SNMP poller doubles after each retry.

SNMP polling discovery settings

The defaultSNMPAutoRetries and defaultSNMPAutoTimeout settings control SNMP polling during the first phase of the discovery process.

The defaultSNMPAutoRetries is the number of times that the SNMP poller should try to reach a system after the initial attempt fails. The default value is three. The defaultSNMPAutoTimeout is the amount of time that the SNMP poller should wait before it considers an SNMP request to have failed or timed out. The default value is 500 milliseconds.

The defaultRetries and defaultTimeout settings control the SNMP poller during the third phase of the discovery process, when the systems are fully probed and the topology is created.

The defaultRetries is the number of times that the SNMP poller should try to reach a system after the initial attempt fails. The default value is five. The defaultTimeout is the amount of time that the SNMP poller should wait before it considers an SNMP request to have failed or timed out. The default value is 1000 milliseconds.

It is recommended that you adjust the timeout and retries settings as follows:

- If random systems are experiencing excessive timeouts, a slow system response time is the most probable cause. In this case, you should increase the timeout value by 50% and decrease the number of retries by 1.
- If several systems within the same subnet are experiencing excessive timeouts, a high packet loss rate is the most probable cause. In this case, you should increase the number of retries by 1 and decrease the timeout value by 50%.

You should not increase the value of the timeout beyond 8000 milliseconds. The timeout doubles with each retry. Therefore, an initial value of 8000 milliseconds with the default number of retries (five) results in a cumulative timeout interval of 504000 milliseconds (more than eight minutes) during the discovery process. In addition, because increasing the number of retries by one effectively doubles the total timeout, you should not increase both the timeout and the number of retries.

Configuring IP tagging for backward compatibility

To support earlier versions of satellite Domain Managers, ensure that the following configuration parameters are set as shown:

- DisableOverlappingIP = TRUE (default)
- UseVPNIP = TRUE (default)
- EnableShadowIP = TRUE (default)

To support earlier versions of satellite MPLS Managers and Network Protocol Managers that want to import VPNIP and VPNIPNetwork objects from IP Availability Manager, ensure that the following configuration parameters are set as shown:

- DisableOverlappingIP = FALSE
- UseVPNIP = TRUE
- EnableShadowIP = TRUE

All three parameters are in the tpmgr-param.conf file, which is described in "Description of tpmgr-param.conf" on page 2-11.

Procedure for configuring IP tagging for backward compatibility

To configure IP Availability Manager to support earlier versions of MPLS Manager and Network Protocol Manager that want to import VPNIP and VPNIPNetwork objects from IP Availability Manager:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Set the following three parameters to the values shown:

   ```
   DisableOverlappingIP FALSE
   UseVPNIP TRUE
   EnableShadowIP TRUE
   ```

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.
4. If IP Availability Manager was running before you edited the *tpmgr-param.conf* file, restart IP Availability Manager.

Enabling light discovery for satellite Domain Managers

By default, IP Availability Manager is not configured to discover the topology collection sets for the following satellite Domain Managers (Figure 2-4):

- MPLS Manager
- Multicast Manager
- Network Protocol Manager for BGP
- Network Protocol Manager for OSPF
- Network Protocol Manager for EIGRP
- Network Protocol Manager for IS-IS

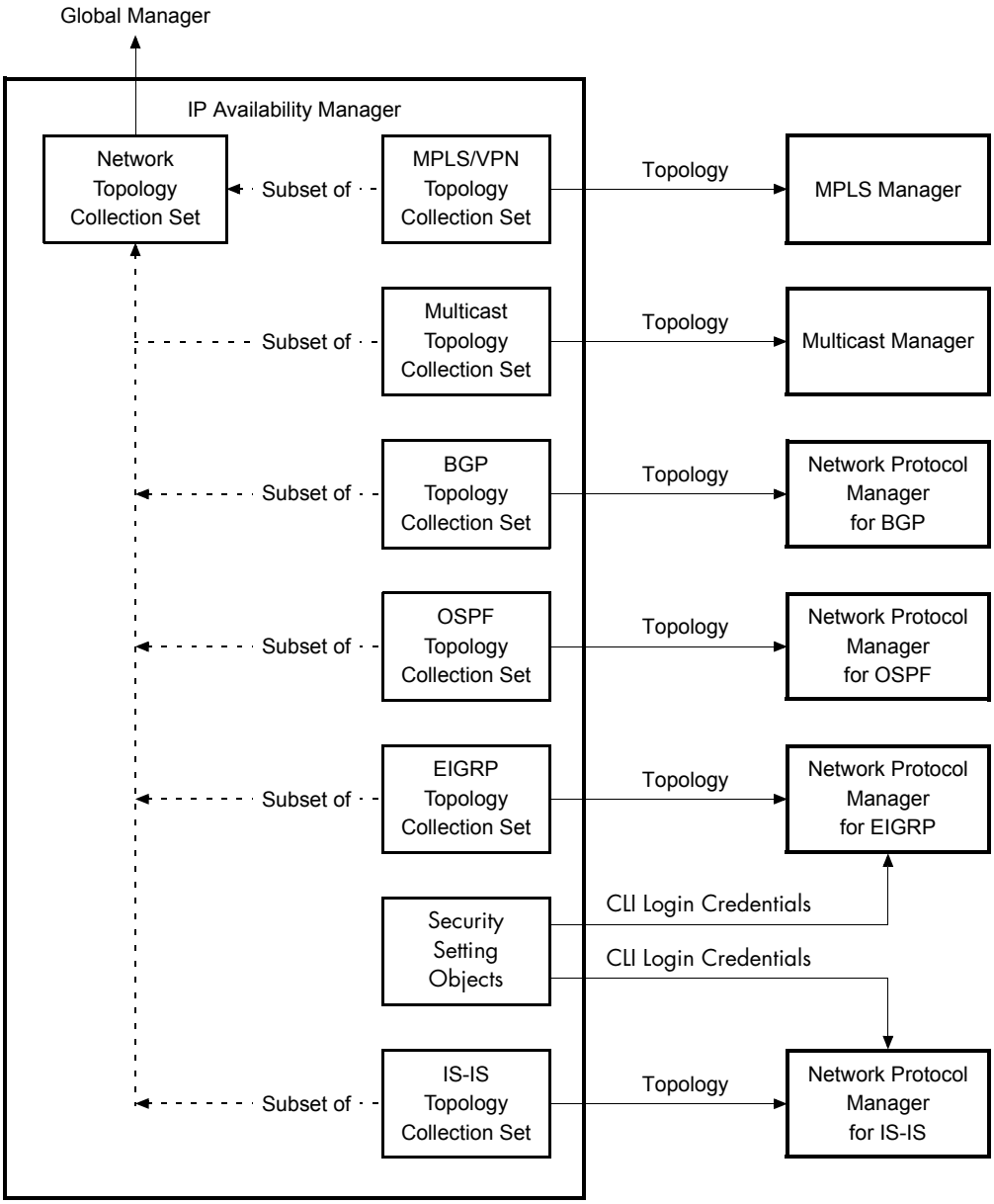


Figure 2-4 Exporting customized topology collection sets to satellite Domain Managers

To configure IP Availability Manager to discover the topology collection set for MPLS Manager, the following parameter must be set to FALSE:

```
DisableMPLSDiscovery FALSE
```

To configure IP Availability Manager to discover the topology collection set for Multicast Manager, the following parameter must be set to FALSE:

```
DisableMulticastDiscovery FALSE
```

To configure IP Availability Manager to discover the topology collection set for Network Protocol Manager, the following parameter must be set to FALSE:

```
DisableProtocolDiscovery FALSE
```

All three parameters are in the tpmgr-param.conf file, which is described in "Description of tpmgr-param.conf" on page 2-11.

Procedure for enabling light discovery for satellite Domain Managers

To configure IP Availability Manager to discover the topology collection sets for MPLS Manager, Multicast Manager, and Network Protocol Manager:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Set the following three parameters to FALSE:

   ```
   DisableMPLSDiscovery FALSE
   DisableMulticastDiscovery FALSE
   DisableProtocolDiscovery FALSE
   ```

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=tpmgr-param.conf
   ```

Or, as an alternative to the preceding procedure, use the dmctl utility to temporarily set these parameters to FALSE:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following commands:

   ```
   dmctl -s <IP Manager instance name> [-b <broker>] invoke
      ICF_TopologyManager::ICF-TopologyManager insertParameter
      DisableMPLSDiscovery FALSE

   dmctl -s <IP Manager instance name> [-b <broker>] invoke
      ICF_TopologyManager::ICF-TopologyManager insertParameter
      DisableMulticastDiscovery FALSE

   dmctl -s <IP Manager instance name> [-b <broker>] invoke
      ICF_TopologyManager::ICF-TopologyManager insertParameter
      DisableProtocolDiscovery FALSE
   ```

Enabling the concatenation of InterfaceAlias fields in ERX devices

The IP Manager can be configured to gather the InterfaceAlias-field descriptions from the interface objects of the virtual routers on a Juniper ERX physical router and append the descriptions to the InterfaceAlias field of the corresponding physical interface object of the main router (default router). (After the concatenation, the InterfaceAlias fields of the interface objects of the virtual routers still hold their original descriptions.) Table 2-6 helps clarify some of the terminology used here.

Table 2-6 Juniper ERX routing platform terminology

Term	Meaning
Juniper ERX physical router	Synonymous with the Juniper ERX routing platform, which consists physically of a shelf of modules, and functionally of a main router and one or more user-created virtual routers.
Juniper ERX main router	Also known as the "default router." Provides a command line interface through which users create and configure virtual routers. The main router implements the traditional concept of a router and supports routing across all physical and virtual interfaces that have not been assigned to a virtual router.
Juniper ERX virtual router	A user-created logical router in a Juniper ERX physical router. Each virtual router has its own forwarding table, IP interfaces, and routing-protocol instances. **Note:** The discovered main router and virtual routers are represented as Router objects in the IP Manager's modeled topology.
Juniper ERX device (or just ERX device)	Used throughout this document to mean any instance of a Juniper ERX main or virtual router.
ifAlias description	A description configured on an individual interface by a network manager. All interfaces on a Juniper ERX physical router have the same DeviceID—the device identification of the physical router. **Note:** For any interface object discovered by the IP Manager, the ifAlias description is stored in the object's InterfaceAlias attribute.

This concatenation configuration is controlled by the PropagateVRIfAlias parameter, which is described in "Description of tpmgr-param.conf" on page 2-11.

Procedure for enabling the concatenation of InterfaceAlias fields

To enable the concatenation of InterfaceAlias fields in Juniper ERX devices:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Set the following parameter to the value shown:

   ```
   PropagateVRIfAlias TRUE
   ```

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=tpmgr-param.conf
   ```

Enabling the logging of failed DNS lookups during discovery

Autodiscovery of a new device added to the managed network will fail if the Domain Name System (DNS) on the host system where the IP Manager is running does not have an entry for the device. The chain of events for this scenario is as follows:

1. A new device has been added to the managed network.
2. An entry for the device has yet to be added to the DNS database.
3. During the next scheduled full discovery, the IP Manager sends an ICMP poll and then an SNMP poll to the IP address of the new device. The SNMP poll returns system-related data for the device.
4. The IP Manger initiates a DNS lookup of the new device to obtain its hostname; the lookup fails, forcing the IP Manager to use the device's IP address as the device's "name."
5. The IP Manager compares the IP address "name" and the system-related data returned from the SNMP poll against the discovery filters configured for the IP Manager.

   Note: Discovery filters are described in Chapter 7, "Creating Discovery Filters," in the *EMC Smarts IP Management Suite Discovery Guide*.

6. The IP address "name" does not match any of the "System Name" fields of the discovery filters because the "System Name" fields all contain hostname patterns. The new device is not discovered.

   Note: If the "System Name" fields contained a lone asterisk (*), the new device would match a discovery filter and be discovered.

7. Because (by default) DNS lookup failures are not logged, users will not be aware that the newly added device failed discovery.

By setting the LogDNSFailure parameter to TRUE, users will become aware in a timely manner when autodiscovery fails due to a failed DNS lookup: All failed DNS lookups occurring during autodiscovery will be logged to the IP Manager's main log file (for example, INCHARGE-AM) in the BASEDIR/smarts/local/logs directory. Here's an example error message for this type of failure:

```
DNSLookupFailure: 22-Feb-2007 10:04:03 EST, 10.10.10.10 failed DNS
   lookup during Auto Discovery. IP address: 10.10.10.10
```

Through these logged error messages, users will be able to react quickly by adding new entries to the DNS and initiating new discoveries.

The LogDNSFailure parameter is described in "Description of tpmgr-param.conf" on page 2-11.

Procedure for enabling the logging of failed DNS lookups

To enable the logging of failed DNS lookups during discovery:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Set the following parameter to the value shown:

   ```
   LogDNSFailure TRUE
   ```

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=tpmgr-param.conf
   ```

Enabling the SystemRestarted and ConfigChange events

Figure 2-5 shows the events detected by IP Availability Manager for a managed system (device) that has been restarted or reconfigured. When IP Availability Manager detects a "system restarted" or "configuration change" condition, it creates and sends a SystemRestarted or ConfigChange trap to the Adapter Platform. The Adapter Platform, in turn, transforms the SystemRestarted or ConfigChange trap to a SystemRestarted or ConfigChange event, for import as a notification by the Global Manager.

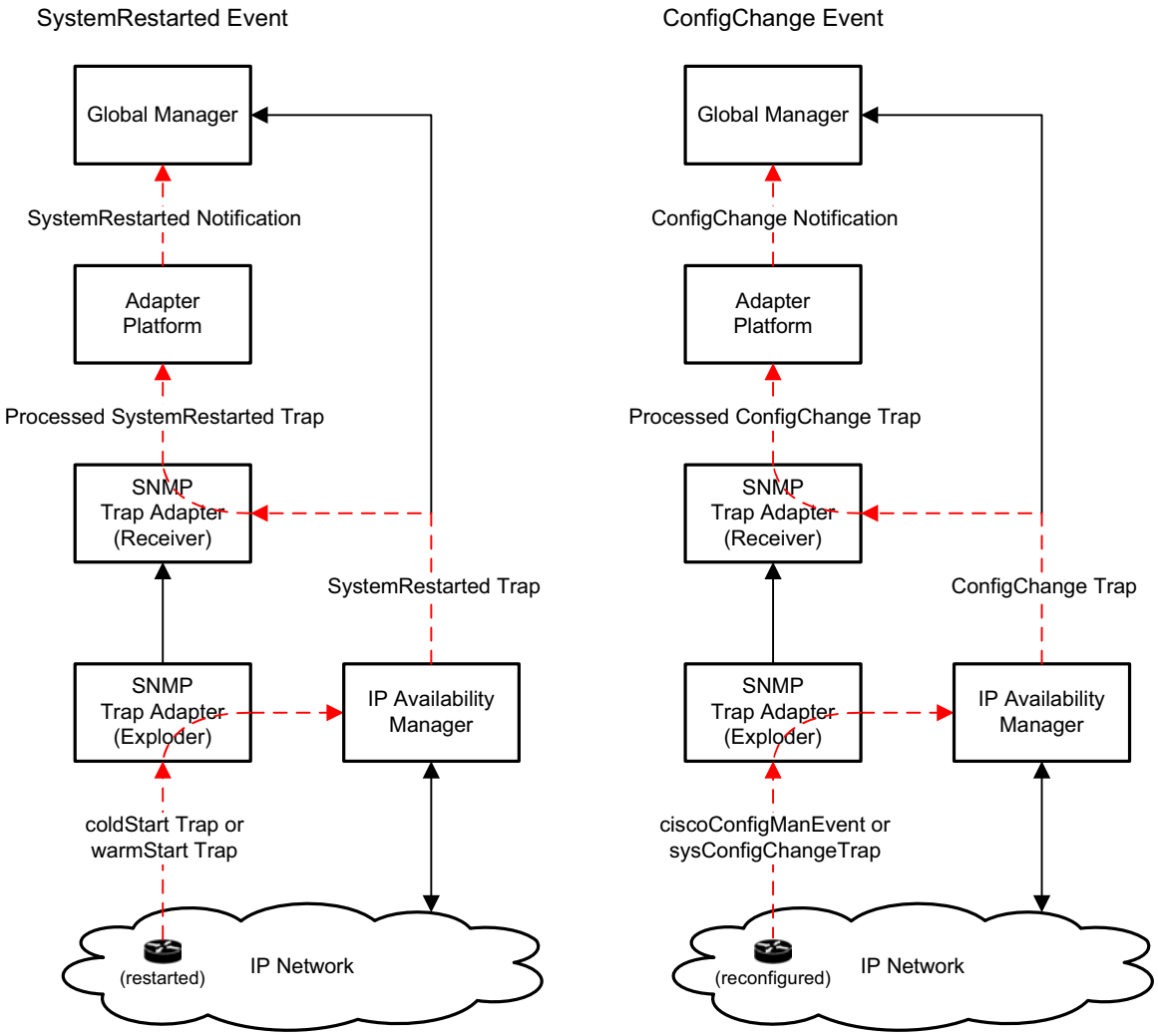


Figure 2-5 Generation of SystemRestarted and ConfigChange events

Enabling the SystemRestarted event

Figure 2-6 shows the enabling and processing of the SystemRestarted event.

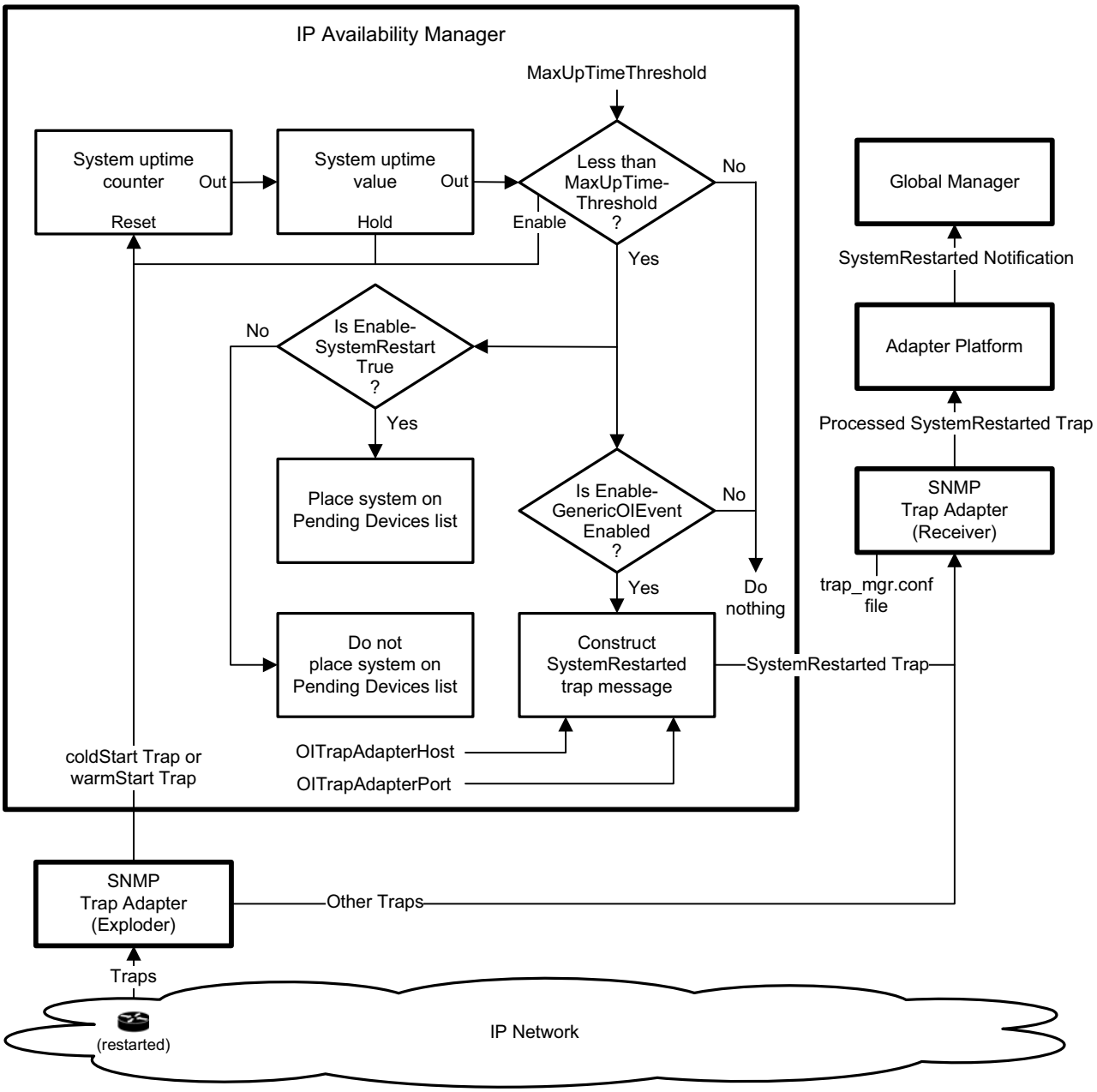


Figure 2-6 How the SystemRestarted event is enabled and processed

Upon receiving a coldStart or warmStart trap for a system already in the modeled topology, IP Availability Manager (if so configured) places the system on the Pending Devices list and generates a SystemRestarted trap for input to the Adapter Platform. During the next pending discovery, IP Availability Manager will send SNMP polls to the system to rediscover the system and its components.

Parameters for the SystemRestarted event

The parameters identified in Table 2-7 (and shown in Figure 2-6) enable and control the SystemRestarted event.

Table 2-7 Parameters for the SystemRestarted event

Parameter name	Allowed values	Description
EnableGenericOIEvent	ENABLED, DISABLED Default: DISABLED	Belongs to Connectivity setting and is described in "Connectivity threshold setting" on page 7-9.
MaxUpTimeThreshold	1 to 1436 minutes Default: 60 minutes	
OITrapAdapterHost	String Default: localhost (Name or IP address of the host system where the SNMP Trap Adapter configured as a trap receiver is running)	Belongs to tpmgr-param.conf file and is described in "Description of tpmgr-param.conf" on page 2-11.
OITrapAdapterPort	Integer Default: 162 (Port on the host system where the SNMP Trap Adapter configured as a trap receiver is running)	
EnableSystemRestart	TRUE, FALSE Default: FALSE	

As shown in Figure 2-6, a coldStart or warmStart trap for a system resets the uptime counter for the system. Whenever a coldStart or warmStart trap for a system is received before the uptime counter reaches the value specified for the MaxUpTimeThreshold parameter, IP Availability Manager considers the system to have been restarted.

Trap parsing rules for the SystemRestarted event

The Adapter Platform transforms SystemRestarted traps into SystemRestarted notifications based on rules specified in the trap_mgr.conf file, which is located in the BASEDIR/smarts/conf/icoi directory of the Service Assurance Management Suite installation area. The parsing rules defined for the SystemRestarted trap are:

```
#System restart traps

#BEGIN_TRAP .1.3.6.1.4.1.733 6 13
#  ClassName:              SNMPTrap
#  InstanceName:           $V1$
#  ElementClassName:       $V2$
#        ElementName:            $V1$
#  EventName:              SystemRestarted
#  Severity:               3
#  EventText:              System Restarted
#  Expiration:             600
#  State:                  NOTIFY
#  UnknownAgent:           CREATE
#END_TRAP

#BEGIN_TRAP * 0 0
#         ClassName:         SNMPTrap
#         InstanceName:      $SYS$
#         EventName:         SystemRestarted
```

```
#        Severity:         5
#        Expiration:       600
#        State:            NOTIFY
#        EventType:        MOMENTARY
#END_TRAP

#BEGIN_TRAP * 1 0
#        ClassName:        SNMPTrap
#        InstanceName:     $SYS$
#        EventName:        SystemRestarted
#        Severity:         5
#        Expiration:       600
#        State:            NOTIFY
#        EventType:        MOMENTARY
#END_TRAP
```

The class name and instance name of the system for which the SystemRestarted notification applies are listed in the ElementClassName and ElementName attributes of the notification definition.

Note: To be more precise, the trap_mgr.conf file defines how the SNMP Trap Adapter configured as a trap receiver maps a trap into the data fields of an EMC Smarts notification. The Adapter Platform, in turn, creates the EMC Smarts notification (object) from the data fields.

Procedure for enabling the SystemRestarted event

To enable the SystemRestarted event:

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager, click the **Thresholds** tab to display the thresholds group tree. (Accessing the Polling and Thresholds Console is described in "Opening the Polling and Thresholds Console" on page 5-2.)
2. In the thresholds group tree, click the **+** signs to expand the thresholds group tree to display the system resource groups: Switches, Routers, Hubs and Bridges, and Other Systems.
3. To enable the SystemRestarted event for a particular system resource group, click its **Connectivity** setting, set the **EnableGenericOIEvent** parameter to **ENABLED**, accept or modify the value of the **MaxUpTimeThreshold** parameter, and click the **Apply** button to save your changes. (Procedures for modifying threshold groups are given in "Working with groups and settings" on page 5-7.)
4. Repeat Step 3 for each system resource group for which you want to enable the SystemRestarted event.
5. Click the **Reconfigure** toolbar button to make your changes take effect.
6. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

7. Set the **EnableSystemRestart** parameter to **TRUE** and accept or modify the values of the **OITrapAdapterHost** and the **OITrapAdapterPort** parameters.
8. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

9. Issue the following command from *BASEDIR/smarts/bin* to make your changes to the *tpmgr-param.conf* file take effect:

```
sm_tpmgr -s <IP Manager instance name> [-b <broker>]
   --load-conf=tpmgr-param.conf
```

10. Go to the *BASEDIR/smarts/bin* directory in the Service Assurance Management Suite installation area and issue the following command to open the *trap_mgr.conf* file:

```
sm_edit conf/icoi/trap_mgr.conf
```

11. Uncomment the following notification definition (shown uncommented) to enable the Adapter Platform to transform SystemRestarted traps into notifications:

```
#System restart traps

BEGIN_TRAP .1.3.6.1.4.1.733 6 13
   ClassName:              SNMPTrap
   InstanceName:           $V1$
   ElementClassName:       $V2$
       ElementName:            $V1$
   EventName:              SystemRestarted
   Severity:               3
   EventText:              System Restarted
   Expiration:             600
   State:                  NOTIFY
   UnknownAgent:           CREATE
END_TRAP

BEGIN_TRAP * 0 0
        ClassName:         SNMPTrap
        InstanceName:      $SYS$
        EventName:         SystemRestarted
        Severity:          5
        Expiration:        600
        State:             NOTIFY
        EventType:         MOMENTARY
END_TRAP

BEGIN_TRAP * 1 0
        ClassName:         SNMPTrap
        InstanceName:      $SYS$
        EventName:         SystemRestarted
        Severity:          5
        Expiration:        600
        State:             NOTIFY
        EventType:         MOMENTARY
END_TRAP
```

12. Save the *trap_mgr.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/icoi* directory.

13. If the Adapter Platform was running before you edited the *trap_mgr.conf* file, restart the Adapter Platform.

Note: The *EMC Smarts Service Assurance Manager Adapter Platform User's Guide* provides detailed procedures for configuring the Adapter Platform and the SNMP Trap Adapter.

Enabling the ConfigChange event

Figure 2-7 shows the enabling and processing of the ConfigChange event.

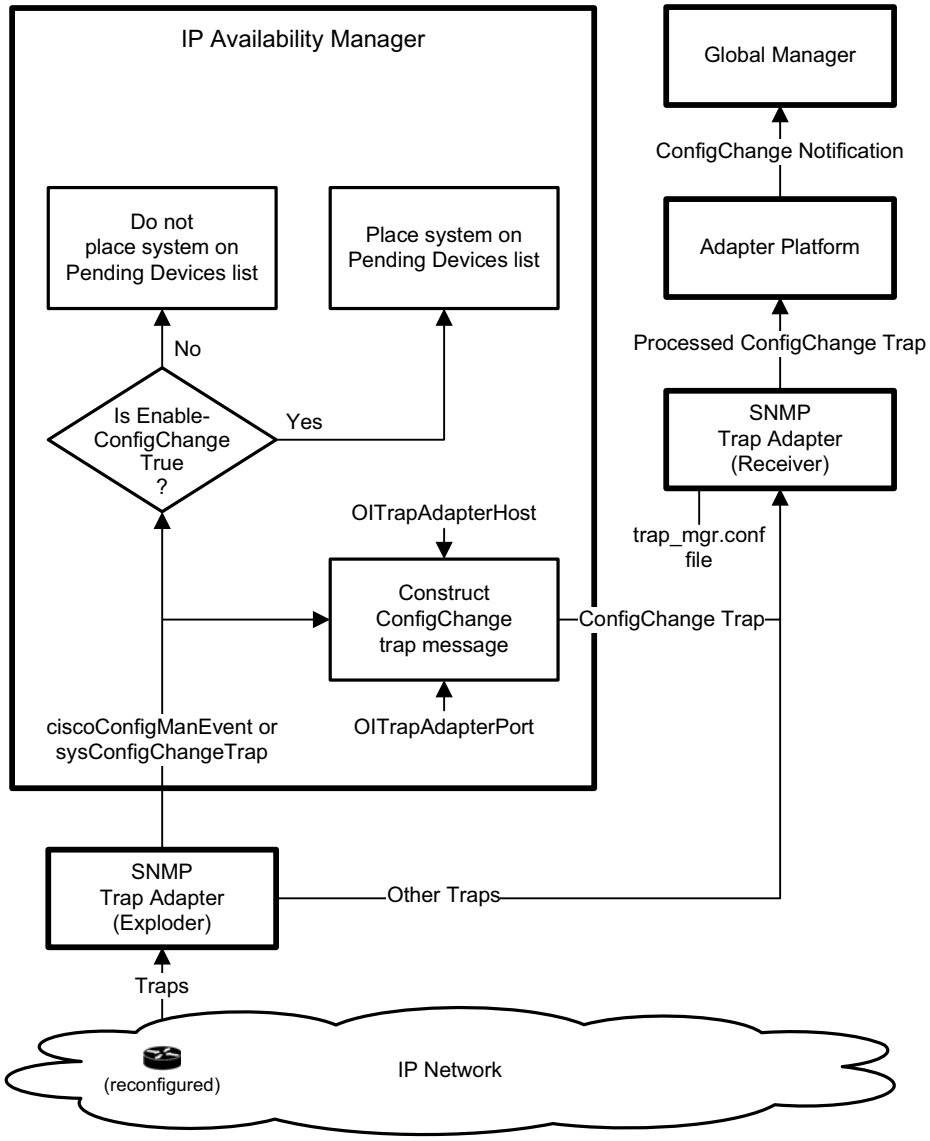


Figure 2-7 How the ConfigChange event is enabled and processed

Upon receiving a ciscoConfigManEvent or sysConfigChangeTrap trap for a system already in the modeled topology, IP Availability Manager (if so configured) places the system on the Pending Devices list and generates a ConfigChange trap for input to the Adapter Platform. During the next pending discovery, IP Availability Manager will send SNMP polls to the system to rediscover the system and its components.

Parameters for the ConfigChange event

The parameters identified in Table 2-8 (and shown in Figure 2-7) enable and control the ConfigChange event.

Table 2-8 Parameters for the ConfigChange event

Parameter name	Allowed values	Description
OITrapAdapterHost	String Default: localhost (Name or IP address of the host system where the SNMP Trap Adapter configured as a trap receiver is running)	Belongs to tpmgr-param.conf file and is described in "Description of tpmgr-param.conf" on page 2-11.
OITrapAdapterPort	Integer Default: 162 (Port on the host system where the SNMP Trap Adapter configured as a trap receiver is running)	
EnableConfigChange	TRUE, FALSE Default: FALSE	

Trap parsing rules for the ConfigChange event

The Adapter Platform transforms ConfigChange traps into ConfigChange notifications based on rules specified in the trap_mgr.conf file, which is located in the BASEDIR/smarts/conf/icoi directory of the Service Assurance Management Suite installation area. The parsing rules defined for the ConfigChange trap are:

```
# Cisco Configuration Management Traps
# ------------------------------------
BEGIN_TRAP .1.3.6.1.4.1.9.9.43.2 6 1
        ClassName:        SNMPTrap
        InstanceName:     $SYS$
        EventName:        ConfigChange
        Severity:         3
        EventText:        Cisco Command Monitoring -> Event Command
   Source: $V1$ | Event Config Source: $V2$ | Event Config Destination:
   $V3$
        Map:            {
                          V1
                          1 = Command Line
                          2 = SNMP
                        }
                        {
                          V2
                          1 = Erase
                          2 = Command Source
                          3 = Running
                         4 = Startup
                          5 = Local
                          6 = Network TFTP
                          7 = Network RCP
                        }
                        {
                         V3
                          1 = Erase
                          2 = Command Source
                          3 = Running
                          4 = Startup
                          5 = Local
```

```
                                6 = Network TFTP
                                7 = Network RCP
                              }
        Expiration:             600
        State:                  NOTIFY
        EventType:              MOMENTARY
END_TRAP
.
.
.
# Cisco Configuration change Traps
# ------------------------------------

#BEGIN_TRAP .1.3.6.1.4.1.9.5 6 9
#        ClassName:        SNMPTrap
#        InstanceName:     $SYS$
#        EventName:        ConfigChange
#        Severity:         3
#        EventText:        sysConfigChangeInfo = $V2$ |
   sysConfigChangeTime = $V1$
#        Expiration:       600
#        State:            NOTIFY
#        EventType:        DURABLE
#END_TRAP
```

The instance name of the system for which the ConfigChange notification applies is listed in the InstanceName attribute of the notification definition.

Note: To be more precise, the trap_mgr.conf file defines how the SNMP Trap Adapter configured as a trap receiver maps a trap into the data fields of an EMC Smarts notification. The Adapter Platform, in turn, creates the EMC Smarts notification (object) from the data fields.

Procedure for enabling the ConfigChange event

To enable the ConfigChange event:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Set the **EnableConfigChange** parameter to **TRUE** and accept or modify the values of the **OITrapAdapterHost** and the **OITrapAdapterPort** parameters.

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes to the *tpmgr-param.conf* file take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=tpmgr-param.conf
   ```

5. Go to the *BASEDIR/smarts/bin* directory in the Service Assurance Management Suite installation area and issue the following command to open the *trap_mgr.conf* file:

   ```
   sm_edit conf/icoi/trap_mgr.conf
   ```

6. Uncomment the following notification definition (shown uncommented) to enable the Adapter Platform to transform ConfigChange traps into notifications:

```
# Cisco Configuration change Traps
# ------------------------------------

BEGIN_TRAP .1.3.6.1.4.1.9.5 6 9
        ClassName:        SNMPTrap
        InstanceName:     $SYS$
        EventName:        ConfigChange
        Severity:         3
        EventText:        sysConfigChangeInfo = $V2$ |
   sysConfigChangeTime = $V1$
        Expiration:       600
        State:            NOTIFY
        EventType:        DURABLE
END_TRAP
```

7. Save the *trap_mgr.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/icoi* directory.

8. If the Adapter Platform was running before you edited the *trap_mgr.conf* file, restart the Adapter Platform.

Note: The *EMC Smarts Service Assurance Manager Adapter Platform User's Guide* provides detailed procedures for configuring the Adapter Platform and the SNMP Trap Adapter.

Changing the IPSec short pending list interval

When detecting an underlying interface event while monitoring IPSec tunnels, the IP Manager places the IPSec-capable host router on the short pending list for a short discovery (assuming that the autoReprobe_short parameter in the discovery.conf file is set to TRUE) to rediscover the host and to recreate the IPSec tunnels.

By default, as shown in Table 2-9, the interval for the short pending list is set to 900 seconds (15 minutes). If you wish, you can change this interval.

Table 2-9 reprobePeriod_short parameter

Parameter	Value	Description
reprobePeriod_short	60 or more seconds Default: 900 seconds	Interval between successive short pending list discoveries of IPSec tunnels. Short discovery for IPSec tunnels is described in Chapter 2, "Discovery Process," of the *EMC Smarts IP Management Suite Discovery Guide.*

Procedure for changing the IPSec short pending list interval

To change the short pending list interval:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *discovery.conf* file:

   ```
   sm_edit conf/discovery/discovery.conf
   ```

2. Add the following line to the file:

   ```
   reprobePeriod_short = <value in seconds>
   ```

   Note: Ensure that the specified value is equal to or greater than 60 seconds.

3. Save the *discovery.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=discovery.conf
   ```

Or, as an alternative to the preceding procedure, use the dmctl utility to temporarily change the short pending list interval:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command:

   ```
   dmctl -s <IP Manager instance name> [-b <broker>] put
      ICF_TopologyManager::ICF-TopologyManager::reprobePeriod_short
      <value in seconds>
   ```

3

Setting Up Communication Links

This chapter describes setting up communication links for communication with the Global Manager and for receiving and sending SNMP traps. It consists of the following sections:

- Configuring the Global Manager to communicate with the IP Manager 3-2
- Configuring SNMP trap receiving and processing 3-5
- Configuring SNMP trap notifying 3-16

Configuring the Global Manager to communicate with the IP Manager

For the Global Manager to import topology and events from the IP Manager, you edit and uncomment the entries for the IP Manager in the Global Manager ics.conf file. This file, located in the BASEDIR/smarts/conf/ics directory in the Service Assurance Management Suite installation area, contains parameters for domain naming, tagging, system default settings, and business topology. The portions of the ics.conf file relevant to this discussion are contained in the DomainSection of the file.

```
DomainSection
{
    DomainType
    {
        ConfFile          = "dxa-conn.conf";
        MinimumCertainty  = 0.01;
        SmoothingInterval = 65;
#        HookScript        = "ics/dxa-sample-hook.asl";
        Name              = "INCHARGE-AM";
#        Name              = "NCM";
#        Tagging
#        {
#            Name    = "INCHARGE-AM2";
#            TagType = "Private IP";
#            Tag     = "abc";
#        }
    }

#   DomainType
#   {
#       ConfFile          = "dxa-conn-perf.conf";
#       MinimumCertainty  = 0.01;
#       SmoothingInterval = 65;
##       HookScript        = "ics/dxa-sample-hook.asl";
#       Name              = "INCHARGE-AM-PM";
#   }

#   DomainType
#   {
#       ConfFile          = "dxa-perf.conf";
#       MinimumCertainty  = 0.01;
#       SmoothingInterval = 65;
##       HookScript        = "ics/dxa-sample-hook.asl";
#       Name              = "INCHARGE-PM";
#   }

.
.
.

#   DomainType definition for INCHARGE-DISCMGR.
#   DomainType
#   {
#       ConfFile          = "dxa-dm.conf";
#       MinimumCertainty  = 0.24;
#       SmoothingInterval = 65;
##       HookScript        = "ics/dxa-sample-hook.asl";
#       Name              = "INCHARGE-DISCMGR";
#   }

}
.
.
.
```

DomainType entries:

IP Availability Manager

IP Availability Manager and IP Performance Manager (running as a single process)

IP Performance Manager

IP Discovery Manager

The DomainSection of the ics.conf file specifies the underlying Domain Managers from which the Global Manager is to import topology and event information. It contains a DomainType entry for every type of Domain Manager product. By default, all configuration fields of a DomainType entry are commented out (preceded with a pound sign—#) except for the configuration fields of the DomainType entry for IP Availability Manager.

Note: Field names and their values are case-sensitive.

Edit the ics.conf file

Use the sm_edit utility to edit the ics.conf file and specify a DomainType entry for each Domain Manager involved in your deployment. At a bare minimum, for each entry, uncomment the same lines that are uncommented for the IP Availability Manager entry and change the value of the Name field to the name of the Domain Manager.

For example, to specify a DomainType entry for IP Availability Manager and IP Performance Manager running as a single process named INCHARGE-MY-AM-PM:

1. Go to the *BASEDIR/smarts/bin* directory in the Service Assurance Management Suite installation area and issue the following command to open the *ics.conf* file:

```
sm_edit conf/ics/ics.conf
```

2. Find the DomainType entry for IP Availability Manager and IP Performance Manager running as a single process:

```
#   DomainType
#   {
#       ConfFile          = "dxa-conn-perf.conf";
#       MinimumCertainty  = 0.01;
#       SmoothingInterval = 65;
##       HookScript        = "ics/dxa-sample-hook.asl";
#       Name              = "INCHARGE-AM-PM";
#   }
```

3. Uncomment the DomainType entry, change the value of the Name parameter to “INCHARGE-MY-AM-PM”, and add a leading comment line:

```
#   DomainType definition for EMC Smarts AM-PM.
    DomainType
    {
        ConfFile          = "dxa-conn-perf.conf";
        MinimumCertainty  = 0.01;
        SmoothingInterval = 65;
#        HookScript        = "ics/dxa-sample-hook.asl";
        Name              = "INCHARGE-MY-AM-PM";
    }
```

4. Save the *ics.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/ics* directory.

The ConfFile field associates an underlying Domain Manager with the data exchange file in which the topology and events to be imported by the Global Manager from that Domain Manager are specified. DXA files are located in the BASEDIR/smarts/conf/ics directory.

The other DomainType entry fields, as well as other aspects of the ics.conf file, are described in the *EMC Smarts Service Assurance Manager Configuration Guide*.

Force the Global Manager to read the edited ics.conf file

At startup, the Global Manager reads the ics.conf file and subscribes to the topology and events specified in the file. If you edit this file while the Global Manager is running, you must reconfigure the Global Manager.

To make any change in the ics.conf file known to a running Global Manager:

1. Go to the *BASEDIR/smarts/bin* directory in the Service Assurance Management Suite installation area and issue the following command on one line:

   ```
   sm_adapter -s <Global Manager instance name> [-b <broker>]
      BASEDIR/SAM/smarts/rules/ics/ICS_RemoteConfig.asl
   ```

   For example:

   ```
   sm_adapter -s INCHARGE-SA
      c:\InCharge7\SAM\smarts\rules\ics\ICS_RemoteConfig.asl
   ```

   Note: The option -b <broker> is used to specify the host and port (<host:port>) location of the Broker if the Global Manager is attached to a Broker other than "localhost:426."

2. When prompted for a username and password, specify an administration account; for example, *admin* and *changeme*.

DXA version files for the IP Manager

For the current release of the IP Management Suite, the following DXA files (located in the BASEDIR/smarts/conf/ics directory in the Service Assurance Management Suite installation area) are applicable:

- dxa.conn.conf
- dxa.conn-perf.conf
- dxa.dm.conf
- dxa.perf.conf

For 6.x releases of the IP Management Suite, the following DXA files are applicable:

- dxa6X.conn.conf
- dxa6X.conn-perf.conf
- dxa6X.dm.conf
- dxa6X.perf.conf

For 4.x releases of the IP Management Suite, the following DXA files are applicable:

- dxa4X.conn.conf
- dxa4X.conn-perf.conf
- dxa4X.perf.conf

Configuring SNMP trap receiving and processing

The IP Manager's trap adapter obtains system status information from the SNMP trap messages received either directly from the managed network systems or from an SNMP Trap Adapter. The IP trap adapter parses and delivers the trap messages in the form of status updates to the IP Manager's correlation analysis software, which in turn applies the status updates to the appropriate system objects in the modeled topology.

Note: Trap messages make the immediate asynchronous notification of status change available. Without trap messages, the response time is determined by the length of the SNMP polling interval (240 seconds by default), described in Chapter 6, "Customizing Polling."

IP trap adapter

The IP trap adapter is designed to receive and process SNMPv1, v2c, and v3 traps. It reads the following files in the IP Management Suite installation area whenever the IP Manager starts up:

- BASEDIR/smarts/local/conf/trapd/trapd.conf
  (or BASEDIR/smarts/conf/trapd/trapd.conf if no local trapd.conf file exists)
- BASEDIR/smarts/local/conf/seedfile

The IP trap adapter sets its configurations to the configuration settings in the trapd.conf file. By default, the trap adapter listens on port 9000 for incoming traps.

The IP trap adapter collect SNMPv3 authentication and privacy credentials from the seedfile and passes the credentials to the SNMP library. It uses the credentials to authenticate and decrypt SNMPv3 traps received from SNMPv3-capable seed systems once they are discovered. IP seed files are described in Chapter 6, "Creating a Seed file," in the *EMC Smarts IP Management Suite Discovery Guide*.

Traps received directly from the managed network systems

For this trap processing design (Figure 3-1), the managed network systems send SNMPv1, v2c, or v3 traps directly to the IP trap adapter. The trap adapter uses the authentication and privacy credentials obtained from the BASEDIR/smarts/local/conf/seedfile to authenticate and decrypt SNMPv3 traps.

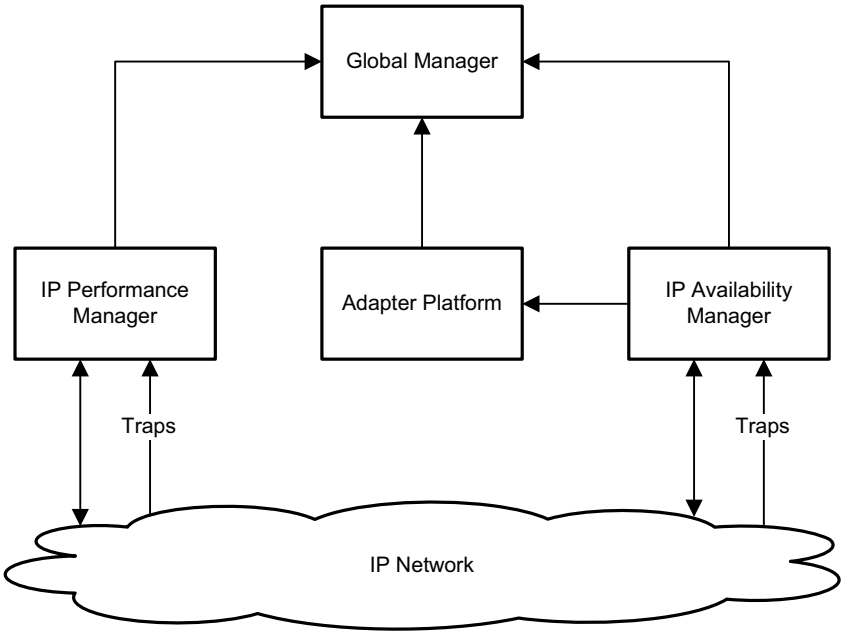


Figure 3-1 Traps received directly from the managed network systems

Traps received from the SNMP Trap Adapter

For the trap processing design shown in Figure 3-2, the SNMP Trap Adapter configured as a trap receiver forwards SNMPv1 and v2c traps to the IP trap adapter. For the trap processing design shown in Figure 3-3 (recommended), the SNMP Trap Adapter configured as a trap exploder forwards SNMPv1 and v2c trap messages to the IP trap adapter. The SNMP Trap Adapter is part of the Service Assurance Management Suite.

In either design, the SNMP Trap Adapter authenticates and decrypts received SNMPv3 traps, converts them to SNMPv2c traps, and forwards the SNMPv2c traps to the configured destinations, including IP Availability Manager and IP Performance Manager. The SNMP Trap Adapter uses the authentication and privacy credentials obtained from a seed file (similar to the one maintained by the IP Manager) to authenticate and decrypt SNMPv3 traps.

The *EMC Smarts Service Assurance Manager Adapter Platform User's Guide* provides procedures for configuring the SNMP Trap Adapter to create the trap processing designs shown in Figure 3-2 and Figure 3-3.

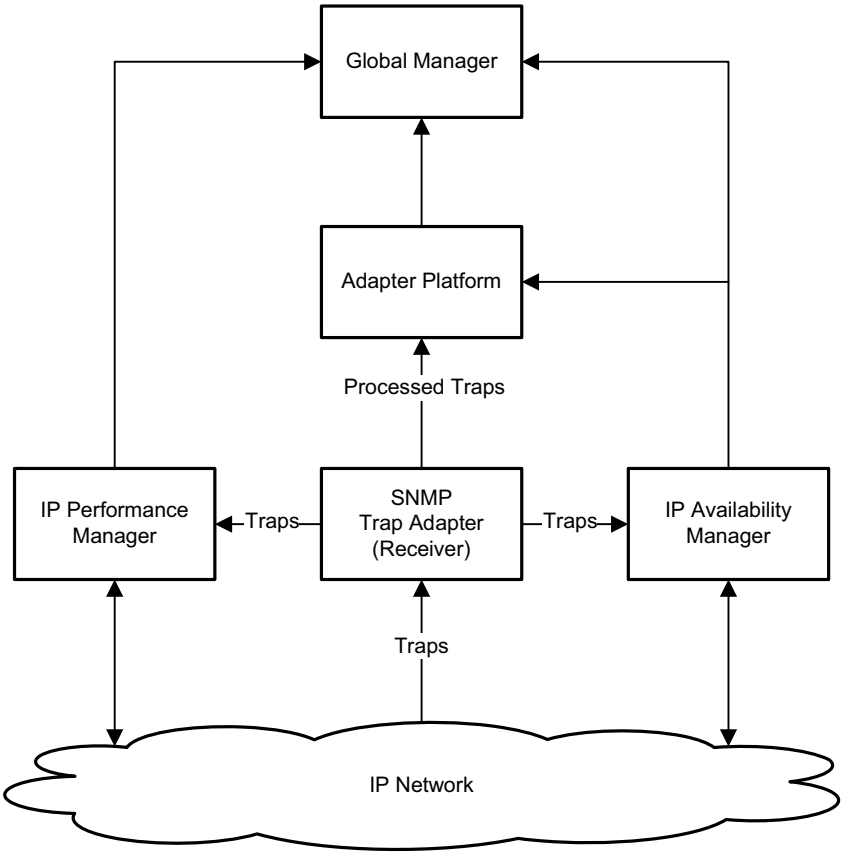


Figure 3-2 A trap processing design involving one instance of the SNMP Trap Adapter

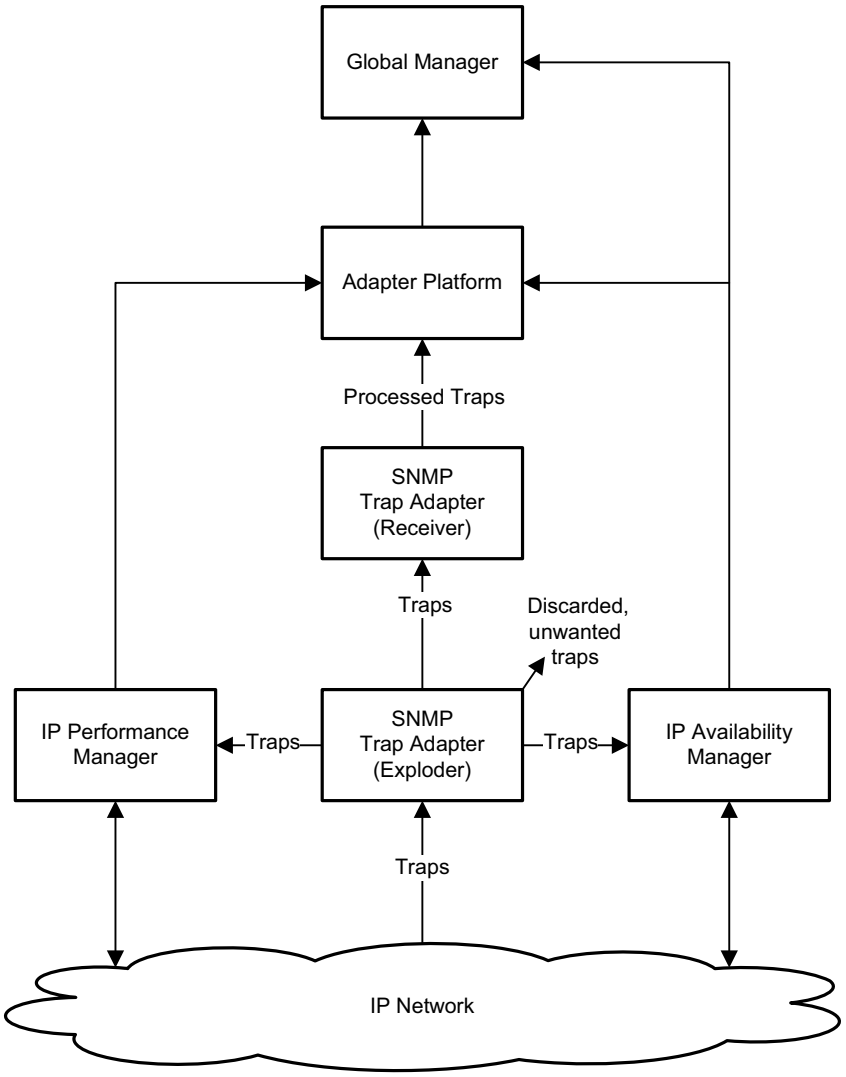


Figure 3-3 A trap processing design involving two instances of the SNMP Trap Adapter

IP trap adapter's configuration file: trapd.conf

The IP trap adapter's configuration file trapd.conf, located in the BASEDIR/smarts/conf/trapd directory of the IP Management Suite installation area, is a global configuration file used to configure many of the internal trap adapters included in the EMC Smarts Manager products. (It is also used to configure the SNMP Trap Adapter as a trap exploder.) Only a few of the configuration parameters in this file are applicable to the IP trap adapter because the IP trap adapter is configured as a trap receiver: It collects, parses, and delivers trap messages in the form of status updates to the IP Manager correlation analysis software.

Figure 3-4 shows the portions of the trapd.conf file that are of interest to this discussion and the discussions that follow. The PORT configuration parameter is of primary interest, and is set to 9000 by default. Table 3-1 presents detailed descriptions of the configuration parameters in the trapd.conf file.

Note: Any line in the trapd.conf file preceded with a pound sign (#) is not read by the IP trap adapter or any other software program.

```
# trapd.conf - Configuration file for SNMP Trap Adapter
.
.
.
# Following is a list of parameters with a brief description.
# More information is available at the end of this file.
#
# PORT      UDP port number the trap adapter listens to.
#
# WINDOW    De-duplication window, in seconds.
#
# ASCII     Controls formatting of non-printable characters.
#
# TAG       Enables tagging of varbind values.
#
# ENABLE_FWD   Enables trap forwarding.
#
# MATCH     Determines whether traps are tested against all forwarding
#           criteria or up to the first criterion that matches.
#
# FORWARD   Specifies matching criteria for traps and the forwarding
#           destinations for matched traps.

##########
#
# Set the parameters here.

PORT: 9000

#WINDOW: 10

#ASCII: FALSE

#TAG: FALSE

ENABLE_FWD: TRUE

#MATCH: all

#QUEUE_LIMIT_MEGS: 0

#QUEUE_LIMIT_SECONDS: 0
.
.
.
# Traps required by InCharge IP Availability Manager (AM)
#
# Generic: coldStart, warmStart, LinkUp, LinkDown
#FORWARD: *.*.*.* .* <0-3> * host:port
.
.
.
# Traps required by InCharge IP Performance Manager (PM)
#
# Cisco: EnvMon Voltage, Temperature, Fan, RedundantSupply
#FORWARD: *.*.*.* .1.3.6.1.4.1.9.9.13.3 6 <2-5> host:port
.
.
.
```

Figure 3-4 **Parts of the trapd.conf file that are of interest to this discussion**

Table 3-1 Detailed descriptions of the configuration parameters in the trapd.conf file

Parameter	Description
PORT	UDP port number on which the trap adapter listens for traps. Valid values are 0-65535. The default port is 9000 or 162. If NetView or OpenView is running on this host system, the default port is set to 9000 during the installation of the IP Management Suite software; otherwise, the default port is set to 162.
WINDOW	De-duplication window, in seconds. The maximum amount of time between receiving similar traps before the second trap is considered unique. Valid values are non-negative integers, including 0. The default is 0. If not set or set to 0, the de-duplication feature is disabled, meaning that all traps are considered unique.
ASCII	Controls formatting of non-printable characters. Valid values are TRUE and FALSE. The default is FALSE. • When TRUE, non-printable characters are represented as dots (.). • When FALSE, non-printable characters are represented by their hex values.
TAG	Enables tagging of varbind values. Valid values are TRUE and FALSE. The default is FALSE. • When TRUE, the type of the varbind value appears before each value; for example, INTEGER-32 3. • When FALSE, the type of the varbind value does not appears before each value.
QUEUE_LIMIT_MEGS	Limits the size of internal trap queue to the stated size, in megabytes. Valid values are non-negative integers, including 0. The default is 0, which means that there is no limit on the size of the internal trap queue. **Note:** The limit is not exact: The queue may grow slightly larger than the value specified. When the limit is reached, some traps will be discarded.
QUEUE_LIMIT_SECONDS	Limits the time that a trap can spend in the internal trap queue, in seconds. Valid values are non-negative integers, including 0. The default is 0, which means that there is no limit on the time that a trap can spend in the internal trap queue. **Note:** This limit is even less exact than the limit set for QUEUE_LIMIT_MEGS. In general, you should specify values for both QUEUE_LIMIT_MEGS and QUEUE_LIMIT_SECONDS. When the limit is reached, some traps will be discarded.

Table 3-1 Detailed descriptions of the configuration parameters in the trapd.conf file (continued)

Parameter	Description
ENABLE_FWD	Determines whether uncommented FORWARD parameters are enabled or disabled. Valid values are TRUE and FALSE. The default is TRUE. • When TRUE, uncommented FORWARD parameters are enabled: Forwarding criteria specified in uncommented FORWARD parameters are read. • When FALSE, uncommented FORWARD parameters are disabled: Forwarding criteria specified in uncommented FORWARD parameters are not read.
MATCH	Determines whether traps are tested against all forwarding criteria specified in an uncommented FORWARD parameter, or tested up to the first criterion that matches. Valid values are “all” or “first.” The default is “first.” If no forwarding criteria is specified, the MATCH parameter is ignored.
FORWARD	Specifies matching criteria for traps and the forwarding destinations for matched traps. Valid syntax is: <address> <OID> <generic type> <specific type> \ <host <[:port] | [:port:community]>> [<host<[:port] | [:port:community]>> ...] Example: FORWARD: *.*.*.* .* * * snake:9099:public1 All traps received from all network systems will be sent to port 9099 on a host named “snake”; the community string “public1” will be assigned to the forwarded traps. Other trap forwarding examples are presented at the end of the trapd.conf file.

Configuring a single instance of the IP trap adapter

When configuring a single instance of the IP trap adapter, perform the procedure in this section. When configuring multiple instances of the IP trap adapter, perform the procedure in“Configuring multiple instances of the IP trap adapter” on page 3-12 .

At startup and by default, the IP trap adapter for an IP Manager instance (IP Availability Manager, IP Performance Manager, IP Server Performance Manager, or IP Availability Manager and IP Performance Manager running as a single process) attempts to listen for SNMP traps on port 9000. Port 9000 is the standard port destination that network systems use when sending traps.

For the trap processing design shown in Figure 3-1 on page 3-6, the only configuration requirement is to configure the managed network systems to send their traps to the trap listening port of the host on which the IP Manager is running. For the trap processing designs shown in Figure 3-2 on page 3-7 and Figure 3-2 on page 3-7, the only configuration requirements are:

- To configure which traps received by the SNMP Trap Adapter are to be forwarded to the IP Manager and
- To configure the SNMP Trap Adapter to forward the IP Manager traps to the trap listening port of the host on which the IP Manager is running.

Note: The *EMC Smarts Service Assurance Manager Adapter Platform User’s Guide* provides procedures for configuring the SNMP Trap Adapter.

Of course, if you choose a trap listening port other than port 9000, you must edit the trapd.conf file and change the default port number to the new port number.

To change the trap listening port number for the IP Manager:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *trapd.conf* file:

   ```
   sm_edit conf/trapd/trapd.conf
   ```

2. In the *trapd.conf* file, set the port number (0 - 65535) of the following PORT entry to the new trap listening port:

   ```
   PORT: <port number>
   ```

3. Save and close the file. The modified version of the *trapd.conf* file is saved to the *BASEDIR/smarts/local/conf/trapd* directory.
4. If the IP Manager was running before you edited the *trapd.conf* file, restart the IP Manager.

Configuring multiple instances of the IP trap adapter

Each IP Manager instance (IP Availability Manager, IP Performance Manager, IP Server Performance Manager, or IP Availability Manager and IP Performance Manager running as a single process) has its own IP trap adapter, and each IP trap adapter requires its own trap listening port. So, for example, if you start two IP Manager instances from the same installation area, or from two different installation areas on the same host, each of those instances will need its own trap listening port. Put another way, multiple IP Manager instances running on the same host cannot share a trap listening port.

The technical issue here is that by default any combination of IP Manager instances running on the same host will compete for the same listening port: port 9000, the default SNMP trap receive port for an IP Management Suite installation. So, for example, if an IP Availability Manager, an IP Performance Manager, and an IP Availability Manager and IP Performance Manager running as a single process are started on a host (not practical but possible), the second and third to start will log an error stating that they cannot open port 9000 and therefore are disabling their SNMP trap receivers. The second and third applications to start will still run, however.

Of course, you can solve this port-9000 conflict problem by creating three different IP Management Suite installation areas, editing the trapd.conf files in two of the installation areas and changing their trap port numbers to values other than 9000, and then starting an IP Manager instance from each installation area.

A more practical solution to the port-9000 conflict problem is the one presented in this section. The solution involves starting two instances of IP Availability Manager from the same IP Management Suite installation area. One instance will listen for traps on default port 9000, and the other will listen for traps on port 7000.

Figure 3-5 previews the directory and file duplication that makes this solution possible. Table 3-2 presents a brief description of the icf directory and the TRAP.import file.

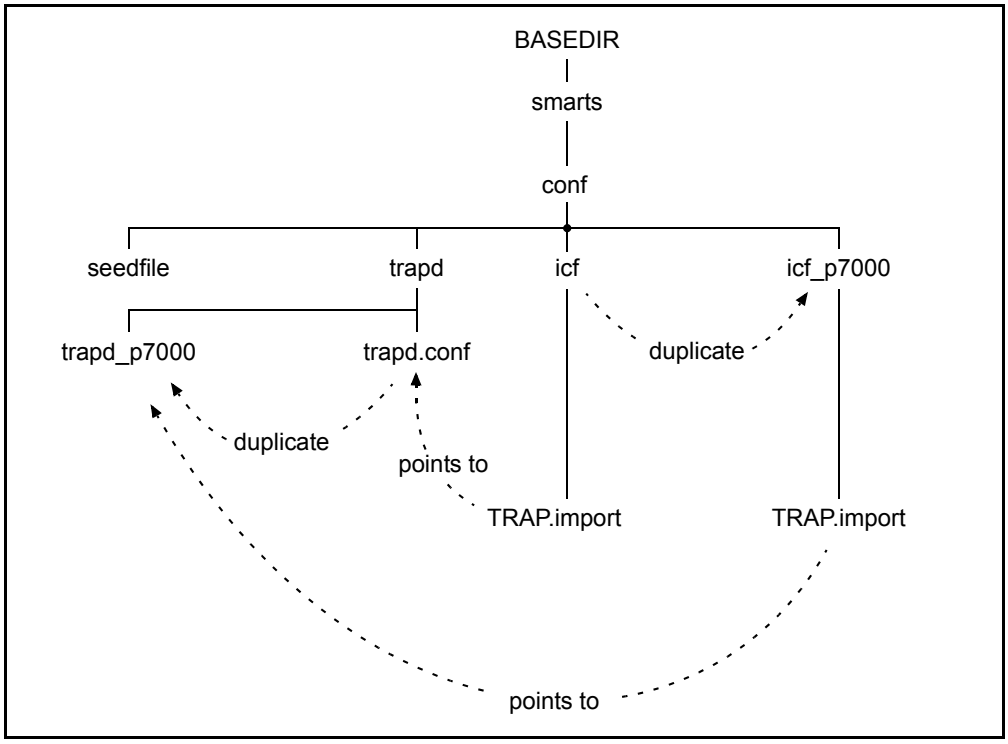


Figure 3-5 **Directory and file duplication for running two IP Manager instances**

Table 3-2 **Startup directory and file for the IP trap adapter**

Directory under BASEDIR	Directory name or filename	Description
smarts/conf	icf (directory)	Contains bootstrap and other configuration files for the IP Manager.
smarts/conf/icf	TRAP.import (file)	Calls a driver that invokes the IP trap adapter. Included in the driver specification is the path to the trapd.conf file.

Here are the assumptions and particulars for the proposed solution:

- The two IP Availability Manager instances are to be started from the same IP Management Suite installation area and are named INCHARGE-AM and INCHARGE-AM-P7000.
- The name of the host on which the IP Management Suite installation resides is Host1.
- An SNMP Trap Adapter has been configured on a remote host to forward the appropriate traps to Host1:9000 for INCHARGE-AM, and to Host1:7000 for INCHARGE-AM-P7000.

Create a second trapd.conf file

To create a trapd.conf file named trapd_p7000.conf:

1. In the IP Management Suite installation area on Host1, copy the *BASEDIR/smarts/conf/trapd/trapd.conf* file to a sibling file named *trapd_p7000.conf*.
2. On Host1, go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and enter the following command to open the *trapd_p7000.conf* file:

   ```
   sm_edit conf/trapd/trapd_p7000.conf
   ```

3. In the *trapd_p7000.conf* file, set the port number to the destination port (Host1:7000) configured for INCHARGE-AM-P7000 in the SNMP Trap Adapter's configuration file:

   ```
   PORT: 7000
   ```

4. Save and close the file. The modified version of the *trapd_p7000.conf* file is saved to the *BASEDIR/smarts/local/conf/trapd* directory in the IP Management Suite installation area.

Create a second icf directory

To create an icf directory named icf_p7000:

1. In the IP Management Suite installation area on Host1, copy the *BASEDIR/smarts/conf/icf* directory to a sibling directory named *icf_p7000.conf*.
2. On Host1, go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and enter the following command to open the *TRAP.import* file in the *icf_p7000* directory:

   ```
   sm_edit conf/icf_p7000/TRAP.import
   ```

3. In the *TRAP.import* file, set the following "filename" line to point to the *trapd_p7000.conf* file:

   ```
   filename = "$SM_SITEMOD/conf/trapd/trapd_p7000.conf"
   ```

4. Save and close the file. The modified version of the *TRAP.import* file is saved to the *BASEDIR/smarts/local/conf/icf_p7000* directory in the IP Management Suite installation area.
5. Issue the following command from *BASEDIR/smarts/bin* to start an IP Availability Manager instance named INCHARGE-AM:

   ```
   sm_server --name=INCHARGE-AM --config=icf
      --bootstrap=bootstrap-am.conf --port=0 --subscribe=default
      --ignore-restore-errors --output
   ```

   The IP Availability Manager instance named INCHARGE-AM will register with the Broker for the deployment, read the *trapd.conf* file in the *BASEDIR/smarts/local/conf/trapd* directory, and begin listening for traps on the trap port (9000) specified in the *trapd.conf* file.
6. Issue the following command from *BASEDIR/smarts/bin* to start an IP Availability Manager instance named INCHARGE-AM-P7000:

   ```
   sm_server --name=INCHARGE-AM-P7000 --config=icf_p7000
      --bootstrap=bootstrap-am.conf --port=0 --subscribe=default
      --ignore-restore-errors --output
   ```

The IP Availability Manager instance named INCHARGE-AM-P7000 will register with the Broker for the deployment, read the *trapd_p7000.conf* file in the *BASEDIR/smarts/local/conf/trapd* directory, and begin listening for traps on the trap port (7000) specified in the *trapd_p7000.conf* file.

Configuring SNMP trap notifying

The IP trap notification adapter subscribes to either IP notifications or Service Assurance notifications and logs them by sending them as trap messages. The configuration file for the IP trap notification adapter, named trap-notify.conf, is identified in Figure 3-6.

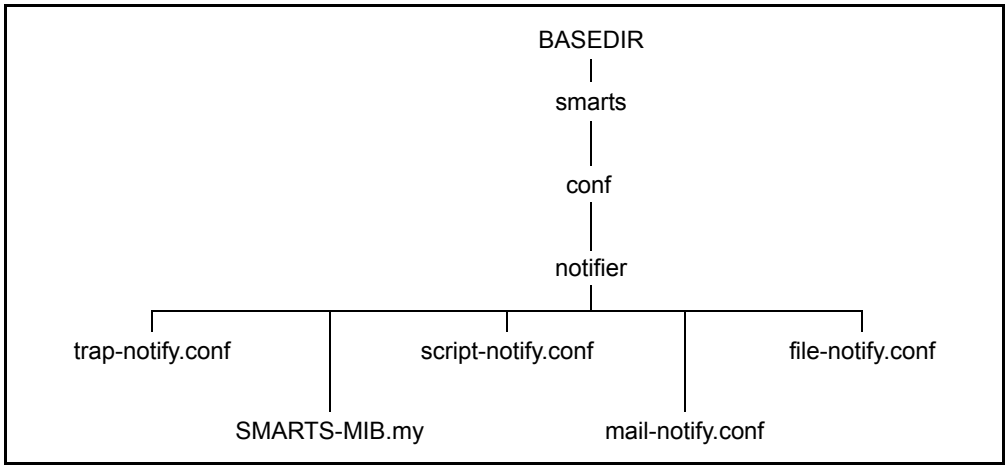


Figure 3-6 The trap-notify.conf file and its location in the IP Management Suite installation area

The main purpose of the IP trap notification adapter is to accommodate third-party management systems that do not have the software to read EMC Smarts notifications but do have the software to listen to SNMP trap messages. The IP trap notification adapter converts and delivers the EMC Smarts notifications as traps to these management systems.

Originally, the IP trap notification adapter converted only IP Manager notifications to traps. Later, the IP trap notification adapter was upgraded to also convert Service Assurance notifications to traps.

The trap notification adapter and the following other notification adapters are part of the EMC Smarts base code and therefore shipped with every EMC Smarts management suite, not just the IP Management Suite:

- Script notification adapter

  Runs a user-defined script that extracts information from a notification and forwards the information.

- Mail notification adapter

  Sends an email message when it receives a notification.

- File notification adapter

  Writes information about a notification to a file.

The configuration files for these three notification adapters are also identified in Figure 3-6.

IP trap notification adapter's configuration file: trap-notify.conf

Figure 3-7 shows the trap-notify.conf file in its entirety.

```
# trap-notify.conf - Configuration for the trap notification adapter
#
# Copyright (C) 1999-2000 System Management ARTS (SMARTS)
# All Rights Reserved
#
# $Id: trap-notify.conf,v 1.8 2004/03/11 01:25:37 guw Exp $
#
GNA_Notifier::trap-Notifier
{
  #
  # Name of the SAM Server from which to receive notifications.
  #
  serverName = "INCHARGE-SA"

  ConfiguredBy = TrapConfiguration::Trap-Configuration
  {
     #
     # Comma-separated list of destinations to which the traps are sent.
     # Each entry in the list has three fields:
     # o address -- host-name or IP-address
     # o port    -- UDP port number
     # o version -- SNMP version; either "V1" or "V2C"
     # o community -- community string
     #
#     Destinations = { {"localhost", 162, "V1"},
#                      {"other-host", 30162, "V2C"}
#                    }
#     DestinationsWithCommunity = { {"localhost", 162, "V1", "public1"},
#                      {"other-host", 30162, "V2C", ""}
#                    }
  }

  ReadsInputFrom = GA_SubscriberFE::trap-Subscriber-FrontEnd
  {
     #
     # How long, in seconds, an event must remain active before the
     # adapter sends a notification.
     #
     eventSmoothingInterval = 0

     #
     # Notification threshold; discard notifications with a certainty
     # below this value, which should be between 0.0 and 1.0.
     #
     minimumCertainty = 0.01

     #
     # Subscriptions.
     #
     SubscribesTo =
     {
#          #
```

Figure 3-7 The trap-notify.conf file (sheet 1 of 2)

```
#             # Choice subscription.
#             #
#             GA_ChoiceSubscription::trap-All-Problems-Subscriptions
#             {
#                      className = ".*"
#                   instanceName = ".*"
#                      eventName = ".*"
#                       problems = TRUE
#                     aggregates = FALSE
#                       symptoms = FALSE
#             }

#             #
#             # NL subscription.
#             #
#             GA_NLSubscription::trap-Default-NL-Subscriptions
#             {
#                      NLName = "Default"
#             }

             #
             # Profile subscription.
             #
             GA_ProfileSubscription::trap-Default-Profile-Subscriptions
             {
                  profileName = "default"
             }
     }
  }
  #
  # How long, in seconds, to ignore notifications from the server after it
  # connects. Use this parameter to avoid receiving the initial flood of
  # notifications when the server (re)connects.
  #
  initialEventDelay = 0
}
#
# Local Variables:
# mode: Makefile
# comment-column: 32
# fill-column: 79
# End:
```

Figure 3-7 The trap-notify.conf file (sheet 2 of 2)

Service Assurance notification subscription

Figure 3-8 shows the content of the trap-notify.conf file with all commented lines removed. By default, the trap-notify.conf is set up for Service Assurance notification subscription, but because no destination (Destinations, DestinationsWithCommunity) is specified, the trap notification adapter does nothing.

```
GNA_Notifier::trap-Notifier
{
  serverName = "INCHARGE-SA"

  ConfiguredBy = TrapConfiguration::Trap-Configuration
  {
#  (No destination specified here.)
  }

  ReadsInputFrom = GA_SubscriberFE::trap-Subscriber-FrontEnd
  {
     eventSmoothingInterval = 0
     minimumCertainty = 0.01
     SubscribesTo =
     {
            GA_ProfileSubscription::trap-Default-Profile-Subscriptions
             {
                  profileName = "default"
             }
     }
  }
  initialEventDelay = 0
}
```

Figure 3-8 The trap-notify.conf file with all commented lines removed

Destinations setting versus DestinationsWithCommunity setting

The two destination settings are defined in the following lines of the trap-notify.conf file:

```
ConfiguredBy = TrapConfiguration::Trap-Configuration
{
   #
   # Comma-separated list of destinations to which the traps are sent.
   # Each entry in the list has three fields:
   # o address -- host-name or IP-address
   # o port    -- UDP port number
   # o version -- SNMP version; either "V1" or "V2C"
   # o community -- community string
   #
#   Destinations = { {"localhost", 162, "V1"},
#                    {"other-host", 30162, "V2C"}
#                  }
#   DestinationsWithCommunity = { {"localhost", 162, "V1", "public1"},
#                    {"other-host", 30162, "V2C", ""}
#                  }
}
```

Note: “other-host” is replaced with the name of the host that is to receive the sent traps.

The difference between the Destinations and DestinationsWithCommunity settings is that the latter lets you specify the community string that you want to assign to the sent traps, while the former assigns the sent traps a default community string, most likely "public.”

Currently, however, the community string that you specify in the DestinationsWithCommunity setting is ignored. As explanation, the code passes the specified community string to the underlying code that eventually sends the trap, but the Model implementation code does not do anything with the community string. EMC Corporation is investigating this behavior.

The “other-host” that you specify may be a host on which the trap notification adapter resides, or it may be a remote host. Because the trap notification adapter and all other notification adapters use the EMC Smarts Remote API to connect to a host, they can run anywhere.

IP Manager notification subscription

Figure 3-9 is an example of a modified trap-notify.conf file that directs the trap notification adapter to subscribe to IP notifications.

Note: Use the sm_edit utility to modify the trap-notify.conf file.

```
# trap-notify.conf - Configuration for the trap notification adapter
#
# Copyright (C) 1999-2000 System Management ARTS (SMARTS)
# All Rights Reserved
#
# $Id: trap-notify.conf,v 1.8 2004/03/11 01:25:37 guw Exp $
#
GNA_Notifier::trap-Notifier
{
  #
  # Name of the SAM Server from which to receive notifications.
  #
  serverName = "INCHARGE-AM"

  ConfiguredBy = TrapConfiguration::Trap-Configuration
  {
     #
     # Comma-separated list of destinations to which the traps are sent.
     # Each entry in the list has three fields:
     # o address -- host-name or IP-address
     # o port    -- UDP port number
     # o version -- SNMP version; either "V1" or "V2C"
     # o community -- community string
     #
      Destinations = { {"localhost", 162, "V1"},
                       {"qa-fancy", 30162, "V2C"}
                     }
#     DestinationsWithCommunity = { {"localhost", 162, "V1", "public1"},
#                       {"other-host", 30162, "V2C", ""}
#                     }
  }

  ReadsInputFrom = GA_SubscriberFE::trap-Subscriber-FrontEnd
  {
     #
     # How long, in seconds, an event must remain active before the
     # adapter sends a notification.
     #
     eventSmoothingInterval = 0

     #
     # Notification threshold; discard notifications with a certainty
     # below this value, which should be between 0.0 and 1.0.
     #
     minimumCertainty = 0.01

     #
     # Subscriptions.
     #
     SubscribesTo =
     {
#          #
```

Figure 3-9 The trap-notify.conf file modified for IP Manager notification subscription (sheet 1 of 2)

```
#             # Choice subscription.
#             #
              GA_ChoiceSubscription::trap-All-Problems-Subscriptions
              {
                       className = ".*"
                     instanceName = ".*"
                        eventName = ".*"
                         problems = TRUE
                       aggregates = FALSE
                         symptoms = FALSE
              }

              #
              # NL subscription.
              #
              GA_NLSubscription::trap-Default-NL-Subscriptions
              {
                       NLName = "Default"
              }

             #
             # Profile subscription.
             #
#           GA_ProfileSubscription::trap-Default-Profile-Subscriptions
#           {
#                profileName = "default"
#           }
     }
  }
  #
  # How long, in seconds, to ignore notifications from the server after it
  # connects. Use this parameter to avoid receiving the initial flood of
  # notifications when the server (re)connects.
  #
  initialEventDelay = 0
}
#
# Local Variables:
# mode: Makefile
# comment-column: 32
# fill-column: 79
# End:
```

Figure 3-9 The trap-notify.conf file modified for IP Manager notification subscription (sheet 2 of 2)

4

Excluding Device-Access Interfaces

This chapter describes how to use interface-matching filters to exclude device-access interfaces from the discovered topology. It consists of the following sections:

- Understanding interface-matching filters 4-2
- Example of configuring interface-matching filters 4-10
- Optimizing the discovery of ERX devices 4-13
- Enabling the creation of a physical connection for a shared MAC address 4-16

Understanding interface-matching filters

There are several reasons for wanting to exclude device-access interfaces from the discovered topology, foremost of which is to limit the interfaces discovered for access devices. An access device is a network device that contains interfaces (also known as access ports) that connect to hosts. Access ports are only managed if the connected host is also a member of the managed topology or if the interfaces, themselves, are explicitly managed.

Note: Devices that are certified as access devices are identified in the *EMC Smarts Certified Device List*.

To exclude interfaces for a discovered device, a user creates interface-matching filters for the device and adds the filters to the tpmgr-param.conf file. The filters are based on the following interface table (ifTable) objects of the device's MIB:

- ifIndex
- ifDescr
- ifType

The object identifier (OID) paths to these objects in the Internet-standard MIB-II are shown in Figure 4-1.

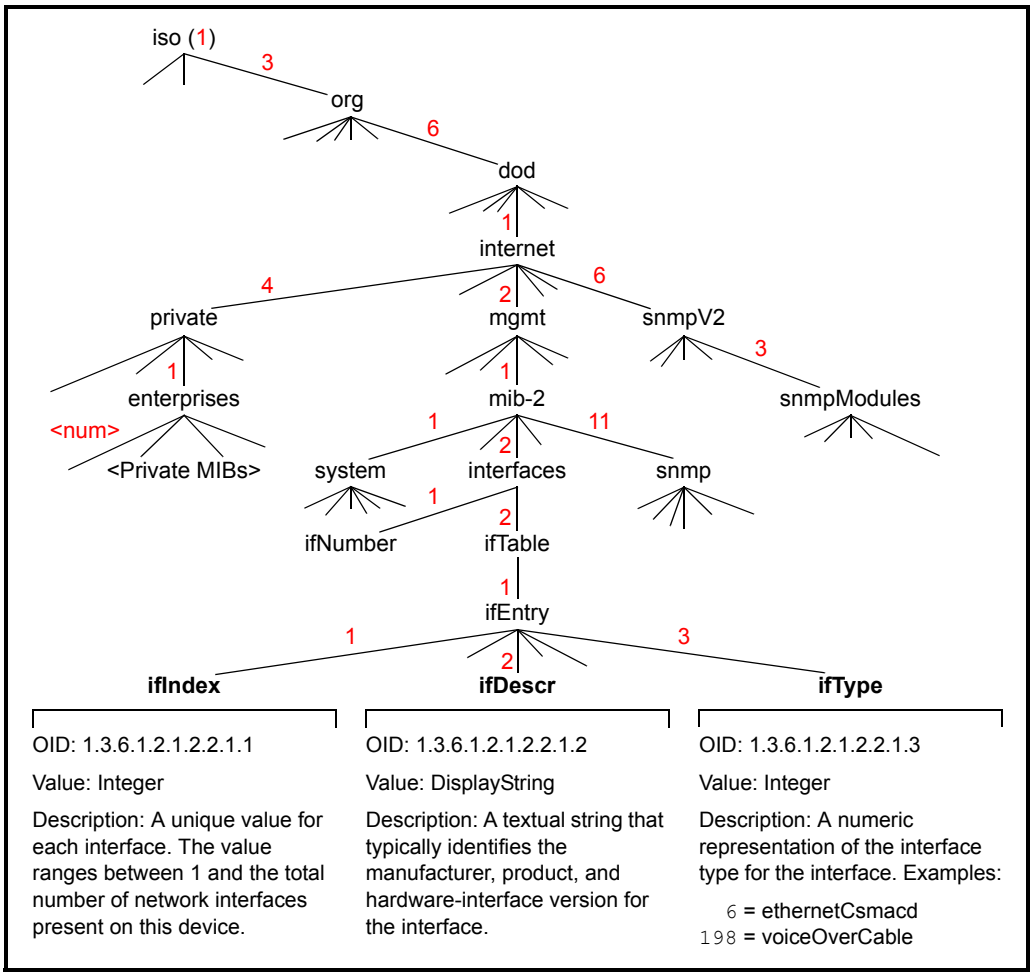


Figure 4-1 **OID-tree view of the MIB-II ifTable's ifIndex, ifDescr, and IfType objects**

For a device using a private MIB, the OID paths to the ifIndex, ifDescr, and ifType objects all begin with the prefix 1.3.6.1.4.1. As shown in Figure 4-1, a prefix of 1.3.6.1.4.1.<number> points to the objects in an enterprise's private MIBs.

The following display is an example of a set of interface-matching filters created for a Redback SMS 1800 Router device.

Key (interface pattern type for this device)	Value (Interface-matching patterns)
IFIndexPattern-.1.3.6.1.4.1.2352.1.3	~*
IFDescrPattern-.1.3.6.1.4.1.2352.1.3	~LOGICAL INTERFACE~PPP LAYER~0
IFTypePattern-.1.3.6.1.4.1.2352.1.3	~23~53~0

The "key" part of an interface-matching filter specifies the interface pattern type (IFIndex, IFDescr, or IFType) for the device, where the device is identified by its system object identifier (sysObjectID). (The sysObjectID for a Redback SMS 1800 Router is 1.3.6.1.4.1.2352.1.3.) The "value" part of an interface-matching filter specifies the interface-matching pattern for the filter's interface pattern type.

Any interface retrieved from the target device that matches any interface-matching pattern for any interface pattern type for the device is included in the discovered topology. Conversely, any interface retrieved from the target device that does not match any of the interface-matching patterns is excluded from the discovered topology.

As indicated in the example, a set of interface-matching filters for a device consists of three different interface pattern types:

- IFIndexPattern pertains to the ifIndex MIB object type.
- IFDescrPattern pertains to the ifDescr MIB object type.
- IFTypePattern pertains to the ifType MIB object type.

The default value (interface-matching pattern) for IFIndexPattern is "~*", the default value for IFDescrPattern is "~*", and the default value for IFTypePattern is "*". As examples:

- IFIndexPattern-.1.3.6.1.4.1.164.6.1.3.183 ~*
- IFDescrPattern-.1.3.6.1.4.1.164.6.1.3.183 ~*
- IFTypePattern-.1.3.6.1.4.1.164.6.1.3.183 *

Note: "*" matches anything, and ~*" matches nothing because of the leading tilde. Wildcard syntax is discussed in Appendix A, "Wildcard Patterns."

Each pattern type and its value is considered an interface-matching filter.

The interface-matching filters for a device are inclusive, meaning that if an interface retrieved from the target device matches just one of the matching patterns specified for IFIndexPattern, IFDescrPattern, or IFTypePattern, the interface will be included in the discovered topology. So, by default (because of the "*" in the IFTypePattern), all of the interfaces retrieved from the target device will be included in the topology.

Interface-matching filters in the tpmgr-param.conf file

You can use the interface-matching filters in the tpmgr-param.conf configuration file as a starting point to configure interface-matching filters for access devices.

The tpmgr-param.conf file contains the IFTypePattern and IFDescrPattern interface-matching filters listed in Table 4-1. You can modify any of these filters or add your own filters to the file.

Note: Currently, the tpmgr-param.conf file does not contain any IFIndexPattern interface-matching filters; thus, by default, these filters are set to ~*.

Table 4-1 IFTypePattern and IFDescrPattern interface-matching filters in the tpmgr-param.conf file

Device	Interface pattern type and device	Interface-matching pattern
Ascend Max-1800 Switch	IFTypePattern-.1.3.6.1.4.1.529.1.2.2	~1 Interface type definitions: 1 = other (generic) **Note:** For any discovered device matching the sysObjectID of this filter, any interface having an ifType value other than 1 will match this filter and therefore be created.
Ascend Max-2000 Switch	IFTypePattern-.1.3.6.1.4.1.529.1.2.3	
Ascend Max-3000 Switch	IFTypePattern-.1.3.6.1.4.1.529.1.2.9	
Cisco IOS 6509 NEB Router	IFTypePattern-.1.3.6.1.4.1.9.1.310	~51 Interface type definitions: 51 => sonetVT
Paradyne 3160 Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.2.1	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.2.1	~COM*|~Modem*|~DTE*|~AUX* **Note:** For any discovered device matching the sysObjectID of this filter, any interface having an ifDescr description not containing COM, Modem, DTE, or AUX will match this filter and therefore be created.
Paradyne T1 SLV Model 9124 Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.5	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.5	~RMON*|~DSX-1*
Paradyne DDS SLV Model 9626 Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.1.6	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.1.6	~RMON*|~COM*|~Modem*
Paradyne DDS SLV Model 9624-OS Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.1.8	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.1.8	~RMON*|~COM*
Paradyne T1 SLV Model 9126-IISLV Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.7	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.7	~RMON*|~Modem*|~DSX-1*|~Ethernet*
Paradyne T1 SLV Model 9128-II Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.8	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.8	~RMON*|~DSX-1*|~COM*|~Modem*

Table 4-1 IFTypePattern and IFDescrPattern interface-matching filters in the tpmgr-param.conf file (continued)

Device	Interface pattern type and device	Interface-matching pattern
Paradyne T1 SLV Model 9124-II Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.9	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.9	~RMON*|~DSX-1*
Paradyne T1 FLEX Model 9123-C Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.11.1	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.11.1	~COM*
Paradyne T1 FLEX Model 9123-SLV Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.11.2	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.11.2	~RMON*|~COM*
Paradyne T1 SLV Model 9124-OS Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.4.12	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.4.12	~RMON*|~DSX-1*
Paradyne T3 DSU/CSU Model 9520-II Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.5.3	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.5.3	~RMON*|~COM*|~Modem*
Paradyne T1 DSU/CSU Model 7123 Probe	IFTypePattern-.1.3.6.1.4.1.1795.1.14.2.4.12.1.1	~*
	IFDescrPattern-.1.3.6.1.4.1.1795.1.14.2.4.12.1.1	~COM*
Radware IP-mux1 Router	IFTypePattern-.1.3.6.1.4.1.164.6.1.3.82	~80~81~82 Interface type definitions: 80 => atmLogical 81 => ds0 82 => ds0Bundle **Note:** For any discovered device matching the sysObjectID of this filter, any interface having an ifType value other than 80, 81, or 82 will match this filter and therefore be created.
Radware IP-mux16 Router	IFTypePattern-.1.3.6.1.4.1.164.6.1.3.83	
Redback SMS 1800 Router	IFTypePattern-.1.3.6.1.4.1.2352.1.3	~23~53~0 Interface type definitions: 23 => ppp 53 => propVirtual 0 => invalid value (ignored) **Note:** For any discovered device matching the sysObjectID of this filter, any interface having an ifType value other than 23 or 53 will match this filter and therefore be created.
	IFDescrPattern-.1.3.6.1.4.1.2352.1.3	~LOGICAL INTERFACE~PPP LAYER~0 **Note:** For any discovered device matching the sysObjectID of this filter, any interface having an ifDescr description not containing LOGICAL INTERFACE, PPP LAYER, or 0 will match this filter and therefore be created.

Table 4-1 IFTypePattern and IFDescrPattern interface-matching filters in the tpmgr-param.conf file (continued)

Device	Interface pattern type and device	Interface-matching pattern
Redback SMS 10000 Router	IFTypePattern-.1.3.6.1.4.1.2352.1.4	~23~53~0 Interface type definitions: 23 => ppp 53 => propVirtual 0 => invalid value (ignored)
	IFDescrPattern-.1.3.6.1.4.1.2352.1.4	~LOGICAL INTERFACE~PPP LAYER~0
Unisphere ERX-1440 Router	#IFTypePattern-.1.3.6.1.4.1.4874.1.1.1.1.1 (Commented out)	* (Described in "Optimizing the discovery of ERX devices" on page 4-13.)
	#IFDescrPattern-.1.3.6.1.4.1.4874.1.1.1.1.1 (Commented out)	~* (Described in "Optimizing the discovery of ERX devices" on page 4-13.)

Other interface-limiting parameter types in the tpmgr-param.conf file

The tpmgr-param.conf file also contains other interface-limiting parameter types, all of which are listed in Table 4-2. The table includes an example, the default value, and the definition of each parameter type.

Several of the parameter types in Table 4-2 are interface-matching filters that are also based on the ifIndex, ifDescr, and ifType objects. However, they have slightly different interface-matching pattern names and employ slightly different syntax than the standard interface-matching filters previously discussed.

Table 4-2 Other interface-limiting parameter types in the tpmgr-param.conf file

Parameter type	Default value	Definition
IFTypePatternIFExt<sysObjectID> <interface type pattern> Example (in tpmgr-param.conf file): • IFTypePatternIFExt.1.3.6.1.4.1.119.1.3.13.4 39|53|1 Where: sysObjectID is the NEC Bluefire 5003 Switch; and Interface types are 39 => sonet, 53 => ds3, and 1 => other (generic). For any discovered device matching the sysObjectID, the filter will override the 30|37|134 default interface type pattern with 39|53|1. Only the interfaces having an ifType value of 39, 53, or 1 will be created.	30|37|134	A filter that restricts the creation of interfaces (ports) for a specific device based on a specific interface type pattern. The IFTypePatternIFExt filter, as well as the IFDescrPatternIFExt and IFIndexPatternIFExt filters, apply to some of the first Cisco, Fore, Huawei, NEC, and Riverstone devices certified for the IP Management Suite. The default values for these filters are: • IFIndexPatternIFExt defaults to ~* • IFDescrPatternIFExt defaults to ~* • IFTypePatternIFExt defaults to 30|37|134
IFTypePattern-SwitchPort[sysObjectID] <interface type pattern> Example (in tpmgr-param.conf file): • IFTypePattern-SwitchPort.1.3.6.1.4.1.119.1.14.8 37 Where: sysObjectID is the NEC Bluefire 8000 series 2.5G ATM Switch; and Interface type is 37 => atm. For any discovered device matching the sysObjectID, the filter will override the 6|26|62|69|117 default interface type pattern: Only the interfaces having an ifType value of 37 will be created.	6|26|62|69|117	A filter that restricts the creation of interfaces (ports) for a specific device based on a specific interface type pattern. The IFTypePattern-SwitchPort filter, as well as the IFDescrPattern-SwitchPort and the IFIndexPattern-SwitchPort filters, apply to some of the first Cisco, NEC, and Lucent devices certified for the IP Management Suite. The default values for these filters are: • IFIndexPattern-SwitchPort defaults to ~* • IFDescrPattern-SwitchPort defaults to ~* • IFTypePattern-SwitchPort defaults to 6|26|62|69|117

Table 4-2 Other interface-limiting parameter types in the tpmgr-param.conf file (continued)

Parameter type	Default value	Definition
IFDescrPattern-SwitchPort[sysObjectID] <interface description pattern> Example (commented out in tpmgr-param.conf file): • #IFDescrPattern-SwitchPort.1.3.6.1.4.1.9.1.501 Vlan888 Where: sysObjectID is the Cisco C4507-with-Sup Switch. For any discovered device matching the sysObjectID, the filter will override the ~* default interface description pattern: Only the interfaces containing Vlan888 in their ifDescr descriptions will be created.	~*	A filter that restricts the creation of interfaces (ports) for a specific device based on a specific interface description pattern. When this filter is used, the associated IFTypePattern-SwitchPort filter is typically set to "~*".
CiscoStackWithATM-Pattern <sysObjectID1>|<sysObjectID2>| <sysObjectIDn> Example (in tpmgr-param.conf file): • CiscoStackWithATM-Pattern .1.3.6.1.4.1.9.5.7|.1.3.6.1.4.1.9.5.17|.1.3.6.1.4.1.9.5.34 Where: sysObjectID1 is the Cisco Catalyst WS-C5000 Switch; sysObjectID2 is the Cisco 5500 Switch; and sysObjectID3 is the Cisco Catalyst 5505 Switch. For any discovered ATM switch matching any of these sysObjectIDs, ATM peer connections will be created between the ATM switch ports and the ports on the neighboring ATM switches.	.1.3.6.1.4.1.9.5.7, .1.3.6.1.4.1.9.5.17, .1.3.6.1.4.1.9.5.34	Identifies which devices are ATM switches so that ATM peer connections are created between the ATM switch ports and the ports on the neighboring ATM switches.
Layer3SwitchPattern <sysObjectID1>|<sysObjectID2>| <sysObjectID3>|<sysObjectIDn> Example (in tpmgr-param.conf file): • Layer3SwitchPattern .1.3.6.1.4.1.9.1.400|.1.3.6.1.4.1.9.5.44| .1.3.6.1.4.1.4641.1.*|.1.3.6.1.4.1.5567.1.1.* Where: sysObjectID1 is the Cisco 6513 IOS Switch; sysObjectID2 is the Cisco 6509 Switch; sysObjectID3 is the Tellabs 8820 or 8860 Switch; and sysObjectID4 is the Riverstone Networks Procut Line of Multi Layer Switch Router. For any discovered Layer 3 device matching any of these sysObjectIDs, WAN network connections will be created between the Layer 3 switch ports and the ports on the neighboring routers.	.1.3.6.1.4.1.9.1.400, .1.3.6.1.4.1.9.5.44	Identifies which devices are Layer 3 switches so that WAN network connections are created between the Layer 3 switch ports and the ports on the neighboring routers. If a user comments out the default line for this parameter in the tpmgr-param.conf file, the IP Manager consults the rule set in the following file to determine how to create WAN network connections during the post-processing phase of the discovery process: BASEDIR/smarts/rules/discovery/ic-post-wan-link.asl
DisplayNameByIfDescr-<sysObjectID> TRUE|FALSE Example (in tpmgr-param.conf file): • DisplayNameByIfDescr-.1.3.6.1.4.1.800.3.1.1.9 TRUE Where: sysObjectID is the Alcatel OmniStack 6024 Switch. For any discovered device matching the sysObjectID, any interface discovered without an ifName string will have its ifDescr description included in its display name.	FALSE	Controls how to name the interface of a specific device when no ifName string is discovered for the interface. • A value of TRUE indicates that the ifDescr description for the interface will be included in the display name for the interface. • A value of FALSE indicates that the DeviceID for the interface will be included in the display name for the interface. If the DeviceID is empty, the ifDescr description will be included in the display name. **Note:** ifName is a text string containing the name used by the device to represent the interface. **Note:** DeviceID refers to the device to which this interface belongs.

Table 4-2 Other interface-limiting parameter types in the tpmgr-param.conf file (continued)

Parameter type	Default value	Definition
ERXIfExcludeSysPattern <name pattern> Example (in tpmgr-param.conf file): • ERXIfExcludeSysPattern * Any discovered Juniper ERX device whose Name attribute string matches * (all discovered ERX devices will match this filter) will be subjected to the following two filters: ERXIfExcludeTypePattern ERXIfExcludeDescrPattern	*	Identifies via a name pattern which Juniper ERX devices are to be subjected to the following two filters: • ERXIfExcludeTypePattern • ERXIfExcludeDescrPattern "Optimizing the discovery of ERX devices" on page 4-13 clarifies the use of this parameter. **Note:** The term "Juniper ERX device" or just "ERX device," as used here, represents an instance of any discovered Juniper ERX main or virtual router.
ERXIfExcludeTypePattern <interface type pattern> Example (in tpmgr-param.conf file): • ERXIfExcludeTypePattern 1|23|126|134 Where: Interface types are 1 => other (generic), 23 => ppp, 126 => ip, and 134 => atmSubInterface. For any discovered Juniper ERX device matching the ERXIfExcludeSysPattern filter, any interface having an ifType value other than 1, 23, 126, or 134 will be created.	1|23|126|134	A filter that restricts the creation of interfaces (ports) for the Juniper ERX devices matching the ERXIfExcludeSysPattern filter. The filtering is based on a specific interface type pattern. Any interface type matching the ERXIfExcludeTypePattern filter is excluded from the discovered topology, and any interface type not matching the filter is included in the discovered topology. "Optimizing the discovery of ERX devices" on page 4-13 clarifies the use of this parameter.
ERXIfExcludeDescrPattern <interface description pattern> Example (commented out in tpmgr-param.conf file): • #ERXIfExcludeDescrPattern *ATM*<0-20>/<0-20>.<1800-900000>| *LOOPBACK<200-10000>|*PPPOE* For any discovered Juniper ERX device matching the ERXIfExcludeSysPattern filter, any interface having an ifDescr description matching any of these patterns will NOT be created, and any interface having an ifDescr description NOT matching any of these patterns will be created.	~*	A filter that restricts the creation of interfaces (ports) for the Juniper ERX devices matching the ERXIfExcludeSysPattern filter. The filtering is based on a specific interface description pattern. Any interface description matching the ERXIfExcludeDescrPattern filter is excluded from the discovered topology, and any interface description not matching the filter is included in the discovered topology. "Optimizing the discovery of ERX devices" on page 4-13 clarifies the use of this parameter.
ERXVRNonIfIPExcludeSysPattern <name pattern> Example (commented out in tpmgr-param.conf file): • #ERXVRNonIfIPExcludeSysPattern * For any discovered Juniper ERX device whose Name attribute string matches * (all discovered ERX devices will match this filter), any IP address (discovered on the device) that is not related to interfaces will be excluded from the discovered topology.	~*	Identifies via a name pattern which Juniper ERX devices are to have all discovered IP addresses not related to interfaces excluded from the discovered topology.

Table 4-2 Other interface-limiting parameter types in the tpmgr-param.conf file (continued)

Parameter type	Default value	Definition
IfVirtualTypePattern-<sysObjectID> <interface type pattern> Example (commented out in tpmgr-param.conf file): • #ifVirtualTypePattern-.1.3.6.1.4.1.9.1.452 135|136 Where: sysObjectID is the Cisco WS-C3550-24Dc Switch; and Interface types are 135 => l2vlan and 136 => l3ipvlan. For any discovered device matching the sysObjectID, the IP Manager will NOT create a "LayeredOver/Underlying" relationship *between* a virtual interface associated with ifType 135 or 136 *and* the underlying MAC address. Thus, only the "LayeredOver/Underlying" relationship between the underlying MAC address and the layered-over physical interface will exist for the MAC address. Accordingly, the IP Manager will create the network connection for the physical interface during the post-processing phase of the discovery process.	~*	For a specific device, enables the creation of a network connection for a physical interface that shares a physical MAC address with one or more virtual interfaces. This parameter can be used for any device that has multiple interfaces (one physical and the others virtual) that share the same MAC address. "Enabling the creation of a physical connection for a shared MAC address" on page 4-16 clarifies the use of this parameter.
ForceAgentAddressInListSysPattern <name pattern> Example (in tpmgr-param.conf file): • ForceAgentAddressInListSysPattern * For any discovered device whose Name attribute string matches * (all discovered devices will match this filter), the IP address held in its SNMP agent's AgentAddress attribute will be added to the list of IP addresses held in its SNMP agent's AgentAddressList attribute.	*	Identifies via a name pattern which devices should have the IP address held in their SNMP agent's AgentAddress attribute added to the list of IP addresses held in their SNMP agent's AgentAddressList attribute. The AgentAddress attribute holds the IP address used by the initial ICMP poll to reach the device, and the AgentAddressList attribute holds a list of IP addresses discovered on the device. In general (but not always), rules are already in place that result in the AgentAddress IP being added to the AgentAddressList. The ForceAgentAddressInListSysPattern parameter provides a means to specify which devices should be forced to have their AgentAddress IP address added to their AgentAddressList.
CDPDeviceOIDs <sysObjectID1>|<sysObjectID2>|<sysObjectIDn> Example (in tpmgr-param.conf file): • CDPDeviceOIDs ~.1.3.6.1.4.1.2467.4.*|~.1.3.6.1.4.1.9.9.368.4.* Where: sysObjectID1 and sysObjectID2 together represent the Cisco CSS 11000 and 11500 Series Content Services Switches. For a Cisco CSS 11000 or 15000 Series Content Services Switch, no CDP connections are created for the interfaces on the switch. Conversely, for any discovered device other than a Cisco CSS 11000 or 15000 Series Content Services Switch, CDP connections are created for the interfaces on the switch.	(Identified in example)	Identifies which Cisco devices to which Cisco Discovery Protocol (CDP) connections should NOT be made. This parameter is used to exclude network connections to Cisco devices (such as CSS 11000 and 11500 Series Content Services Switches) that have a management interface. Because CDP is used for some network management functions, any person on a directly connected segment to a Cisco CSS 11000 or 11500 series switch (for example) is able to perform CDP actions, actions that might include the gathering of sensitive information about the switch. This information could be used to design attacks against the switch. For this reason, CDP connections should not be made to Cisco devices having a management interface.

Example of configuring interface-matching filters

This section presents an example of configuring interface-matching filters that will exclude any interface having “Ethernet” in its description during the discovery of an Adtran Atlas 830 Router device, which has a sysObjectID of .1.3.6.1.4.1.664.1.430.

Be aware that your matching patterns and the retrieved interface descriptions are not case-sensitive because the IP Manager transforms both of them to upper-case letters before comparing them with one another.

To configure the interface-matching filters for the Adtran Atlas 830 Router device:

1. Perform a MIB walk of the device to obtain a list of interfaces for the device.
2. Inspect the interfaces and attempt to form a matching pattern for the interfaces that you want to exclude from the discovered topology.

   For example, look at their IfType (.1.3.6.1.2.1.2.2.1.3) values and ifDescr (.1.3.6.1.2.1.2.2.1.2) values and determine if the interfaces that you want to exclude all have the same type or all have the same string in their descriptions.

   In this procedure, the matching pattern “~*Ethernet*” will be used, a pattern that appears in interface ifDescr descriptions such as “Ethernet0” and “FastEthernet1/0.”
3. To obtain the values (interface-matching patterns) for the device, go to the *BASEDIR/smarts/bin* directory of the IP Management Suite installation area and issue the following commands:

```
dmctl -s <IP Manager instance name> [-b <broker>] invoke
   ICF_TopologyManager::ICF-TopologyManager findParameter
   IFIndexPattern-.1.3.6.1.4.1.664.1.430

dmctl -s <IP Manager instance name> [-b <broker>] invoke
   ICF_TopologyManager::ICF-TopologyManager findParameter
   IFDescrPattern-.1.3.6.1.4.1.664.1.430

dmctl -s <IP Manager instance name> [-b <broker>] invoke
   ICF_TopologyManager::ICF-TopologyManager findParameter
   IFTypePattern-.1.3.6.1.4.1.664.1.430
```

   Note: The option -b <broker> is used to specify the host and port (<host:port>) location of the Broker if the IP Manager is attached to a Broker other than "localhost:426."

   These commands retrieve the interface-matching patterns (if any) from the table defined for the ICF-TopologyManager. The table stores an entry as a key-value pair, and the preceding three commands supplied three keys.

   If blank values are returned, start with the following default values:

   - IFIndexPattern-.1.3.6.1.4.1.664.1.430 ~*
   - IFDescrPattern-.1.3.6.1.4.1.664.1.430 ~*
   - IFTypePattern-.1.3.6.1.4.1.664.1.430 *

4. Whatever values you start with (the default values or the values obtained from the **dmctl** commands), change those values to exclude the interfaces having "Ethernet" in their descriptions.

   The following values will be used in this example:

   - IFIndexPattern-.1.3.6.1.4.1.664.1.430 ~*
   - IFDescrPattern-.1.3.6.1.4.1.664.1.430 ~*Ethernet*
   - IFTypePattern-.1.3.6.1.4.1.664.1.430 ~*

   No interface descriptions can match the "~*" value, and only interface descriptions NOT containing the string "Ethernet" can match the "~*Ethernet*" value.

5. Issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

6. Add the following two lines to the file:

   ```
   IFDescrPattern-.1.3.6.1.4.1.664.1.430 ~*Ethernet*
   IFTypePattern-.1.3.6.1.4.1.664.1.430 ~*
   ```

7. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

8. To force the ICF-TopologyManager to read the *tpmgr-param.conf* file, issue the following command:

   ```
   dmctl -s <IP Manager instance name> [-b <broker>] invoke
      GA_Driver::TopologyManager-Parameter-Driver start
   ```

9. To confirm that your configuration have been read by the ICF-TopologyManager, issue the following commands:

   ```
   dmctl -s <IP Manager instance name> [-b <broker>] invoke
      ICF_TopologyManager::ICF-TopologyManager findParameter
      IFDescrPattern-.1.3.6.1.4.1.664.1.430

   dmctl -s <IP Manager instance name> [-b <broker>] invoke
      ICF_TopologyManager::ICF-TopologyManager findParameter
      IFTypePattern-.1.3.6.1.4.1.664.1.430
   ```

10. Examine the returned values and confirm that they are the same as the ones that you added to the *tpmgr-param.conf* file in Step 6.

Note: The following steps verify that the interface-matching filters are configured properly.

---

11. Attach the Domain Manager Administration Console to the IP Manager. ("Procedure for opening the Polling and Thresholds Console" on page 5-2 provides instructions on opening the Domain Manager Administration Console.)

12. In the topology tree, delete every instance of the Adtran Atlas 830 Router device. To delete a device, right-click its name and then select **Delete** from the pop-up menu.

    When you delete a device, all of its components (including interfaces) and all of its connections are also deleted.

13. To rediscover the device, select **Topology > Discover All**.

14. In the topology tree, expand the **Router** class to see the Adtran Atlas 830 Router instances that the IP Manager just rediscovered.

15. Choose an Adtran Atlas 830 Router instance, expand its **ComposedOf** relationship, expand the **Interface** class, and select an interface instance to display its **Attributes** tab in the right panel.
16. In the **Attributes** tab, verify that the Description attribute does not contain the "Ethernet" string.
17. Repeat Steps 15 and 16 for each interface instance of each Adtran Atlas 830 Router instance.

Optimizing the discovery of ERX devices

The discovery of Juniper ERX devices running JUNOS 6.x and higher might take longer than what is acceptable because the IP Manager, by default, discovers all interfaces on an ERX device. Depending on the deployment, discovering all interfaces on an ERX device is not always desirable, especially when speed of discovery is an issue.

Note: The term "Juniper ERX device" or just "ERX device," as used here, represents an instance of any discovered Juniper ERX main or virtual router. "Enabling the concatenation of InterfaceAlias fields in ERX devices" on page 2-24 describes ERX main routers and ERX virtual routers.

The solution to this problem is to use the parameters in Table 4-3 to configure the IP Manager to discover fewer interfaces on the discovered ERX devices. The solution is based on the filtering of ERX device interfaces by type and description to prevent (exclude) undesirable interfaces from being discovered.

The parameters in Table 4-3 all belong to the tpmgr-param.conf file, and all but the last two are also described in "Other interface-limiting parameter types in the tpmgr-param.conf file" on page 4-6. The last two parameters are the standard interface-matching filters described at the beginning of this chapter.

Table 4-3 Parameters for optimizing the discovery of ERX devices

Parameter name	Filter type	Default value	Description
ERXIfExcludeSysPattern <name pattern> Example: • ERXIfExcludeSysPattern *network* Any discovered Juniper ERX device whose Name attribute string matches "network" (not case-sensitive) will be subjected to the following two filters: ERXIfExcludeTypePattern ERXIfExcludeDescrPattern	Inclusive	* (By default, all discovered Juniper ERX devices will match this filter.)	Identifies via a name pattern which Juniper ERX devices are to be subjected to the following two filters: • ERXIfExcludeTypePattern • ERXIfExcludeDescrPattern Any discovered ERX device whose Name attribute string matches the name pattern specified for this parameter will be subjected to these two filters.
ERXIfExcludeTypePattern <interface type pattern> Example: • ERXIfExcludeSysPattern *network* • ERXIfExcludeTypePattern 1|23|126|134 Where: Interface types are 1 => other (generic), 23 => ppp, 126 => ip, and 134 => atmSubInterface. For any discovered Juniper ERX device whose Name attribute string matches "network" (not case-sensitive), any interface having an ifType value other than 1, 23, 126, or 134 will be created.	Exclusive	1|23|126|134	A filter that restricts the creation of interfaces (ports) for the Juniper ERX devices matching the ERXIfExcludeSysPattern filter. The filtering is based on a specific interface type pattern. The ERXIfExcludeTypePattern filter is an exclusive-type filter: Any interface type matching the filter is excluded, and any interface type not matching the filter is included.

Table 4-3 Parameters for optimizing the discovery of ERX devices (continued)

Parameter name	Filter type	Default value	Description
ERXIfExcludeDescrPattern <interface description pattern> Example: • ERXIfExcludeSysPattern *network* • #ERXIfExcludeDescrPattern *ATM*<0-20>/<0-20>.<1800-900000>| *LOOPBACK<200-10000>|*PPPOE* (commented out in the tpmgr-param.conf file) For any discovered Juniper ERX device whose Name attribute string matches "network" (not case-sensitive), any interface having an ifDescr description matching any of these patterns will NOT be created, and any interface having an ifDescr description NOT matching any of these patterns will be created.	Exclusive	~*	A filter that restricts the creation of interfaces (ports) for the Juniper ERX devices matching the ERXIfExcludeSysPattern filter. The filtering is based on a specific interface description pattern. Like the ERXIfExcludeTypePattern filter, the ERXIfExcludeDescrPattern filter is an exclusive-type filter: Any interface description matching the filter is excluded, and any interface description not matching the filter is included.
IFTypePattern-<ERX sysObjectID> <interface type pattern> Example: • IFTypePattern-.1.3.6.1.4.1.4874.1.1.1.1.1 ~80 Where: sysObjectID is the Unisphere ERX-1440 Router; and Interface type is 80 => atmLogical. Note that in this example all ifType values will match ~80 except ifType 80.	Inclusive	*	A filter (standard filter) that further restricts the creation of interfaces (ports) for a Juniper ERX device based on a specific interface type pattern. Any interface type matching the filter is included, and any interface type not matching the filter is excluded. **Note:** In the case of an ifType exclude/include conflict, the ERXIfExcludeTypePattern filter has higher priority than the IFTypePattern filter.
IFDescrPattern-<ERX sysObjectID> <interface description pattern> Example: • IFDescrPattern-.1.3.6.1.4.1.4874.1.1.1.1.1 ~*Logical Interface* Where: sysObjectID is the Unisphere ERX-1440 Router. Note that in this example all ifDescr descriptions will match ~*Logical Interface* except those that contain a "Logical Interface" string.	Inclusive	~*	A filter (standard filter) that further restricts the creation of interfaces (ports) for a Juniper ERX device based on a specific interface description pattern. Any interface description matching the filter is included, and any interface description not matching the filter is excluded. When this filter is used, the associated IFTypePattern filter is typically set to "~*". **Note:** In the case of an ifDescr exclude/include conflict, the ERXIfExcludeDescrPattern filter has higher priority than the IFDescrPattern filter.

Procedure for optimizing the discovery of ERX devices

To optimize the discovery of Juniper ERX devices:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Consult Table 4-3 and set all or any of the following parameters to optimize the discovery of Juniper ERX devices in your environment:

   ```
   ERXIfExcludeSysPattern
   ERXIfExcludeTypePattern
   ERXIfExcludeDescrPattern
   IFTypePattern
   IFDescrPattern
   ```

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=tpmgr-param.conf
   ```

Enabling the creation of a physical connection for a shared MAC address

The IP Manager perceives each interface found in the ifTable of a discovered device as being a physical interface, meaning that the IP Manager does not distinguish between physical and virtual interfaces. Thus, when the IP Manager detects multiple interfaces (one physical and the others virtual) sharing the same MAC address, it perceives the condition as an error and subsequently removes the "LayeredOver/Underlying" relationships between the interfaces and the MAC address—relationships that the IP Manager had created initially for each interface that it found in the ifTable. Without a LayeredOver/Underlying relationship between the physical interface and the MAC address, the IP Manager will not create the network connection for the physical interface during the post-processing phase of the discovery process.

By identifying the virtual interfaces for a device in the IfVirtualTypePattern parameter, a user changes the behavior of the IP Manager in the following way: The IP Manager will not create LayeredOver/Underlying relationships between these interfaces and the underlying MAC address. Accordingly, the LayeredOver/Underlying relationship between the physical interface and the underlying MAC address will not be removed, and the network connection for the physical interface will be created during the post-processing phase of the discovery process.

The IfVirtualTypePattern parameter is described in "Other interface-limiting parameter types in the tpmgr-param.conf file" on page 4-6.

Procedure for enabling the creation of a physical connection for a shared MAC address

To enable the IP Manager to create a physical connection for a shared MAC address:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command to open the *tpmgr-param.conf* file:

   ```
   sm_edit conf/discovery/tpmgr-param.conf
   ```

2. Set the following parameter to the sysObjectID of the target device and to the ifType values of the target virtual interfaces:

   ```
   IfVirtualTypePattern-<sysObjectID> <interface type pattern>
   ```

   For example:

   ```
   ifVirtualTypePattern-.1.3.6.1.4.1.9.1.452 135|136
   ```

3. Save the *tpmgr-param.conf* file. The modified version of the file is saved to the *BASEDIR/smarts/local/conf/discovery* directory.

4. Issue the following command from *BASEDIR/smarts/bin* to make your changes take effect:

   ```
   sm_tpmgr -s <IP Manager instance name> [-b <broker>]
      --load-conf=tpmgr-param.conf
   ```

5

Using the Polling and Thresholds Console

This chapter introduces the Polling and Thresholds Console, defines groups and settings, and provides instructions on using the Polling and Thresholds Console to work with groups and settings. It consists of the following sections:

- Opening the Polling and Thresholds Console 5-2
- Understanding groups and settings 5-5
- Working with groups and settings 5-7

Opening the Polling and Thresholds Console

The Global Console is a collection of many console views, two of which are the Domain Manager Administration Console and the Polling and Thresholds Console. You access the Polling and Thresholds Console from the Domain Manager Administration Console.

User accounts and passwords

Attaching the Global Console to a Domain Manager (such as IP Availability Manager) or to a Global Manager requires an EMC Smarts user account. The default administration account is username admin and password changeme, and the default operator account is username oper and password oper.

For users logged on with an administration account, the Global Console includes a Configure menu, which enables administrators to open the Domain Manager Administration Console and the Polling and Thresholds Console, to open other console views such as the Global Manager Administration Console, to edit Map Icons, and so on.

For information about changing the password for the default administration account (recommended) and configuring access privileges, see the *EMC Smarts System Administration Guide*. For information about configuring permissions to perform specific console operations, see the *EMC Smarts Service Assurance Manager Configuration Guide*.

Procedure for opening the Polling and Thresholds Console

To open the Polling and Thresholds Console:

1. Start the Global Console.
   - On a Windows system, select **Start > Programs > InCharge 7> EMC Smarts Global Console**.
   - On a UNIX system, go to the *BASEDIR/smarts/bin* directory in the EMC Smarts Service Assurance Manager (Global Manager) installation area and enter **sm_gui**.

   The **Attach Manager** dialog box opens, an example of which is shown in Figure 5-1.

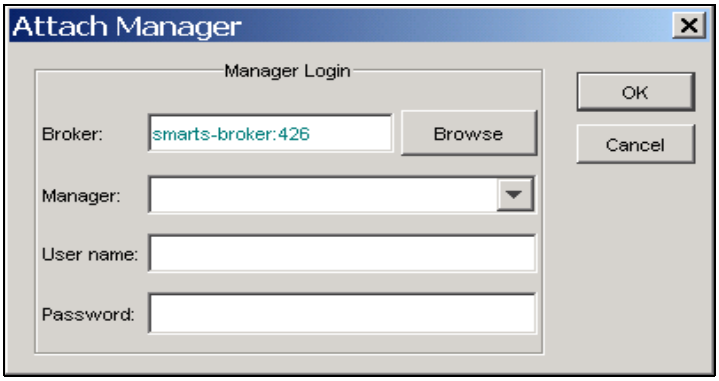


Figure 5-1 Attach Manager dialog box

2. In the dialog box,
   - Ensure that the EMC Smarts Broker for your deployment appears in the **Broker** text box.
   - Click the **Manager** drop-down list or the **Browse** button to display a list of active (running) Smarts Managers, and from that list select an IP Manager application (for example, INCHARGE-AM) in your deployment as the Smarts Manager to which you want to connect.
   - Enter your login username and password. (The default administration login is username *admin* and password *changeme*.)
   - Click **OK**.

   The **Topology Browser Console** opens.
3. In the **Topology Browser Console**, choose **Configure > Domain Manager Administration Console**.

   The **Domain Manager Administration Console** opens.
4. In the **Domain Manager Administration Console**, choose **Configure > Polling and Thresholds**.

   The **Polling and Thresholds Console** opens, an example of which is shown in Figure 5-2.

   In the example display, the **Polling and Thresholds Console** is attached to an IP Availability Manager application named INCHARGE-AM.

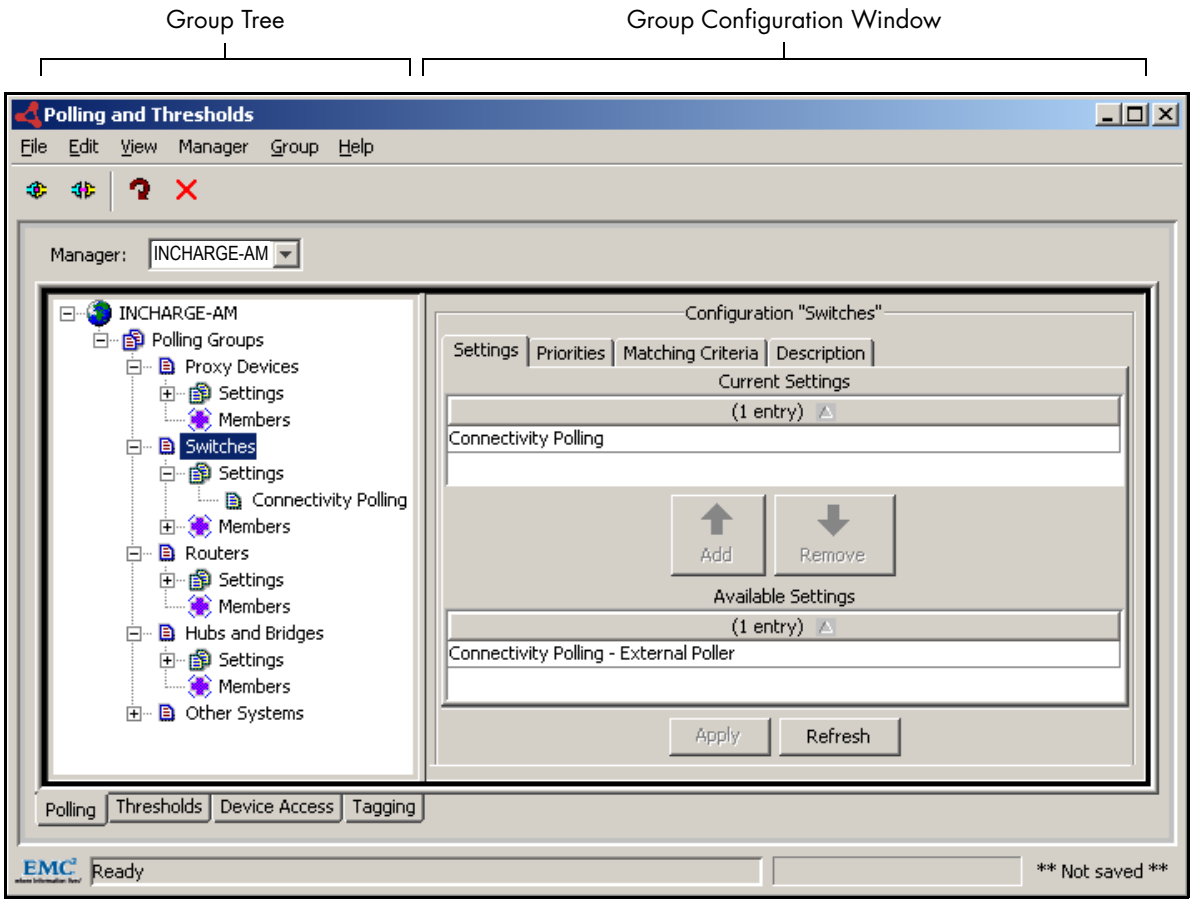


Figure 5-2 Polling and Thresholds Console—example

Layout of the Polling and Thresholds Console

When attached to IP Availability Manager or IP Performance Manager, the Polling and Thresholds Console contains four tabs:

- Polling

  Enables administrators to control the polling of the managed objects for monitoring purposes.

- Thresholds

  Enables administrators to set thresholds for the polled data.

- Device Access

  Enables administrators to specify credentials (such as login credentials) for device access.

- Tagging

  Enables administrators to create IP tagging filters.

Note: When attached to other Domain Managers, the Polling and Thresholds Console contains the Polling and Thresholds tabs, or just the Polling tab, or just the Thresholds tab. The Device Access and Tagging tabs are unique to IP Availability Manager and IP Performance Manager.

The Polling and Thresholds Console is divided into two panels.

- The left panel displays the icon for the Domain Manager in the upper-left corner and provides one, two, or four tabs, depending on the Domain Manager to which the Polling and Thresholds Console is attached. Selecting a tab shows what groups, settings, and members are available on the tab.
- The right panel remains blank until a group, setting, or member is selected in the left panel. When an item is selected in the left panel, the right panel displays additional information regarding that item.

Polling and Thresholds Console toolbar buttons

The toolbar of the Polling and Thresholds Console provides quick access to the commands described in Table 5-1.

Table 5-1 Polling and Thresholds Console toolbar buttons

Button	Description
	Attach to a Smarts Manager
	Detach from a Smarts Manager
	Reconfigure a group
	Delete the selected item

Understanding groups and settings

A group contains one or more settings, and a setting contains one or more parameters. Managed objects (systems discovered in the managed environment) become members of a group based on the matching criteria defined for the group.

Figure 5-3 illustrates the relationships between managed objects, groups, and settings.

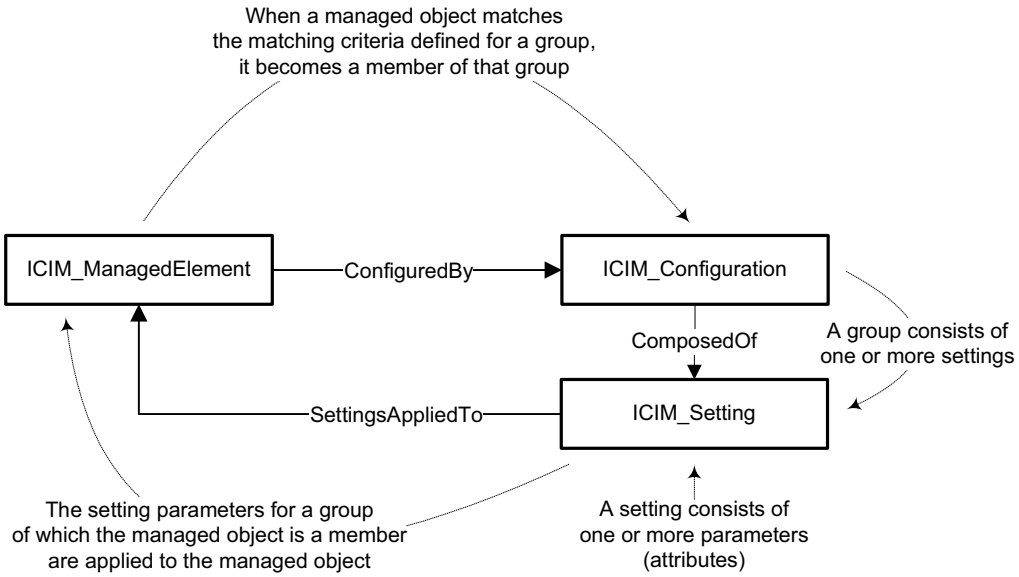


Figure 5-3 Relating managed objects to groups and settings

- The EMC Smarts ICIM core class for group (in the sense of "groups and settings") is ICIM_Configuration. This class is the parent of all group classes used by IP Manager or any other EMC Smarts Domain Manager.
- The ICIM core class for setting is ICIM_Setting. This class is the parent of all setting classes used by IP Manager or any other EMC Smarts Domain Manager.
- The ICIM core class for managed object is ICIM_ManagedElement. This class is the parent of all managed object classes used by IP Manager or any other EMC Smarts Domain Manager.

Default groups and settings

The Polling, Thresholds, Device Access, and Tagging tabs of the Polling and Thresholds Console show the default polling groups, the default threshold groups, the default device access groups, and the default IP tagging filter groups shipped with the IP Management Suite software. The tabs also identify the matching criteria defined for each group, the default settings associated with each group, and the default parameter values associated with each setting.

Matching criteria

The matching criteria for a group are the attribute values defined for the target class associated with the group. For polling and threshold groups, the target class is Port, Interface, or UnitaryComputerSystem (such as a Switch, Router, Hub, or Bridge). For

device access and IP tagging filter groups, the target class is UnitaryComputerSystem.

Any managed object meeting the matching criteria for a group becomes a member of the group. When a managed object becomes a member of a group, all parameter values defined for the group are applied to the managed object.

Prioritized lists of groups

Within each group category (polling groups, threshold groups, device access groups, and IP tagging filter groups), groups can be arranged in a high-to-low priority. A managed object is compared against the highest priority group first and becomes a member of the first group for which there is a criteria match.

A managed object may become a member of different groups in different group categories, but a managed object may only be a member of one group within a particular group category.

Working with groups and settings

Using the Polling and Thresholds Console, you can change the parameter values for a group. You can also:

- Add settings to a group.
- Remove settings from a group.
- Change the matching criteria defined for a group.
- Create new groups.
- Prioritize the order of groups within a group category.

How managed objects are assigned to groups

When the IP Manager performs discovery, it automatically assigns each managed object to one or more groups based on:

- The matching criteria defined for a group.
- The priority of a group, which determines membership when an object meets the matching criteria for more than one group within a particular group category.

A managed object may be a member of one and only one group within a particular group category.

Modifying the properties of a group

A group's matching criteria and priority determine which managed objects become members of the group.

When a group is selected in the left panel of the Polling and Thresholds Console, four tabs appear in the right panel of the console:

- Settings
- Priorities
- Matching Criteria
- Description

Modifying the properties in the Settings, Priorities, or Matching Criteria tab changes the configuration of the group. When you finish editing the properties of a group, click the Apply button to save the changes, and then click the Reconfigure toolbar button or select Reconfigure from the Group menu to make the configuration changes take effect.

Adding or removing settings

A group's settings determine what parameters are applied to the managed objects that are members of the group.

The Settings tab is divided into two sections: Current Settings and Available Settings. The Current Settings section lists the settings that are applied to the group. The Available Settings section lists additional available settings.

To add or remove a setting:

1. In the **Settings** tab, select a setting in the **Available Settings** list or in the **Current Settings** list.
2. Click **Add** to move an available setting to the **Current Settings** list, or click **Remove** to move a current setting to the **Available Settings** list.
3. Click the **Apply** button to save your changes.
4. Click the **Reconfigure** toolbar button to make your changes take effect.

Modifying the parameters of a setting

The parameters of a setting are changed in one of two ways:

- By choosing a value from a drop-down menu
- By entering a value in a Value field or adjusting a slider bar representing a range of values

Changing the parameters of a setting

To change the parameters of a setting:

1. In the left panel of the **Polling and Thresholds Console**, select the setting to display the parameters of the setting in the right panel of the console.
2. Change the value of a parameter using one of the following methods:

   For a drop-down menu, click the menu and select a value.

   For a slider bar presentation,

   - Type a value into the **Value** field and press **Enter** or
   - Select the slider bar and drag its handle with the mouse to change the value or select the slider bar and use the arrow keys to move its handle to change the value.
3. Click the **Apply** button to save your changes.
4. Click the **Reconfigure** toolbar button to make your changes take effect.

Restoring the default values of a setting

To restore the default values of a setting:

1. In the left panel of the **Polling and Thresholds Console**, select the setting to display the parameters of the setting in the right panel of the console.
2. Click **Restore Defaults**.
3. Click the **Reconfigure** toolbar button to make your changes take effect.

Modifying the priority of groups

In any group category, you can arrange the groups in a high-to-low priority. A candidate member object is compared against the highest priority group first and becomes a member of the first group for which there is a criteria match.

The Priorities tab shows groups in the order of their priority, from highest to lowest.

To change the priority of a group:

1. In the **Priorities** tab, select the group for which you want to change the priority.
2. Click the up or down arrow to change its position relative to the other groups.
3. Click the **Apply** button to save your changes.
4. Click the **Reconfigure** toolbar button to make your changes take effect.

Editing matching criteria

A group's matching criteria, which appear at the top of the Matching Criteria tab, are defined using the attributes of the target class associated with the group. Each matching criterion has three fields: Name, Description, and Value.

- Name identifies the name of the target class attribute.
- Description is the description of the target class attribute.
- Value is the value of the target class attribute, which can be any combination of text, integers, and wildcards. The Value field for a matching criterion is not case-sensitive.

As an example of a matching criterion for target class UnitaryComputerSystem, if you specify "Router" for the Type attribute, only routers become members of the group; if you specify "R1*" for the Name attribute, only routers having names beginning with the string R1 become members of the group. Appendix A, "Wildcard Patterns," discusses wildcard syntax.

When a managed object matches a group's target class and matches all of the attributes defined for the target class, the managed object becomes a member of the group.

Adding or removing matching criteria

To add or remove matching criteria:

1. In the **Disabled Criteria** area of the **Matching Criteria** tab, select a matching criterion.
2. Click **Enable** to make the criterion active, moving it to the top of the **Matching Criteria** tab.

   Use **Disable** to deactivate the criterion, moving it to the **Disabled Criteria** area of the **Matching Criteria** tab.
3. If you are adding a matching criterion, type a matching pattern in the **Value** field.
4. Click the **Apply** button to save your changes.
5. Click the **Reconfigure** toolbar button to make your changes take effect.

Changing the value of a matching criterion

To change the value of a matching criterion:

1. In the **Matching Criteria** tab, select the string in the **Value** field or double-click the **Value** field to highlight the current value.
2. Type the new value.
3. Click the **Apply** button to save your changes.
4. Click the **Reconfigure** toolbar button to make your changes take effect.

The IP Manager compares a managed object against the matching criteria of the group with the highest priority. If an object matches all the criteria, the IP Manager adds the object as a member of the group. Otherwise, the IP Manager compares the object against the matching criteria of the group with the second highest priority, and so on.

Creating new groups

You can create additional groups within any group category.

Creating a new group enables you to customize the settings for a group of managed objects. You can use two methods to create a new group:

- Copy an existing group. The new group contains the same settings and thresholds as the original group. Matching criteria are not copied.
- Create an empty group. The new group does not contain any settings or members. You must add settings and matching criteria, and set the priority of the new group.

After you create a new group, use the procedures previously described in this chapter to add settings and matching criteria and to set the priority of the new group.

Copying an existing group

To copy an existing group:

1. In the left panel of the **Polling and Thresholds Console**, right-click the group that you want to copy.
2. Select **Copy** from the pop-up menu to display the **Copy Group** dialog box.
3. In the dialog box, type a name and an optional description for the new group and click **OK**.

   The new group contains the same settings and thresholds as the group you copied.
4. Edit the settings, matching criteria, and priority of the new group. Change the value of any parameters as necessary.
5. Click the **Reconfigure** toolbar button to make your changes take effect.

Creating an empty group

To create an empty group:

1. In the left panel of the **Polling and Thresholds Console**, right-click the group for which you want to create a new group.
2. Select **New Group** from the pop-up menu to display the **New Group** dialog box.
3. In the dialog box, type a name and an optional description for the new group and click **OK**.
4. Add settings and matching criteria, and set the priority of the new group. Change the values of any thresholds or parameters as necessary.
5. Click the **Reconfigure** toolbar button to make your changes take effect.

6

Customizing Polling

This chapter describes the default polling groups and settings for IP Availability Manager and IP Performance Manager. It consists of the following sections:

- Viewing the Polling tab 6-2
- Understanding polling groups and settings 6-4
- Default polling groups and settings for IP Availability Manager 6-5
- Default polling groups and settings for IP Performance Manager 6-8
- Enabling IPv6 address polling 6-12

Viewing the Polling tab

Figure 6-1 shows the Polling tab of a Polling and Thresholds Console connected to an IP Availability Manager instance named INCHARGE-AM. The right panel of the console identifies "Connectivity Polling" as the default setting for the default polling group named "Switches."

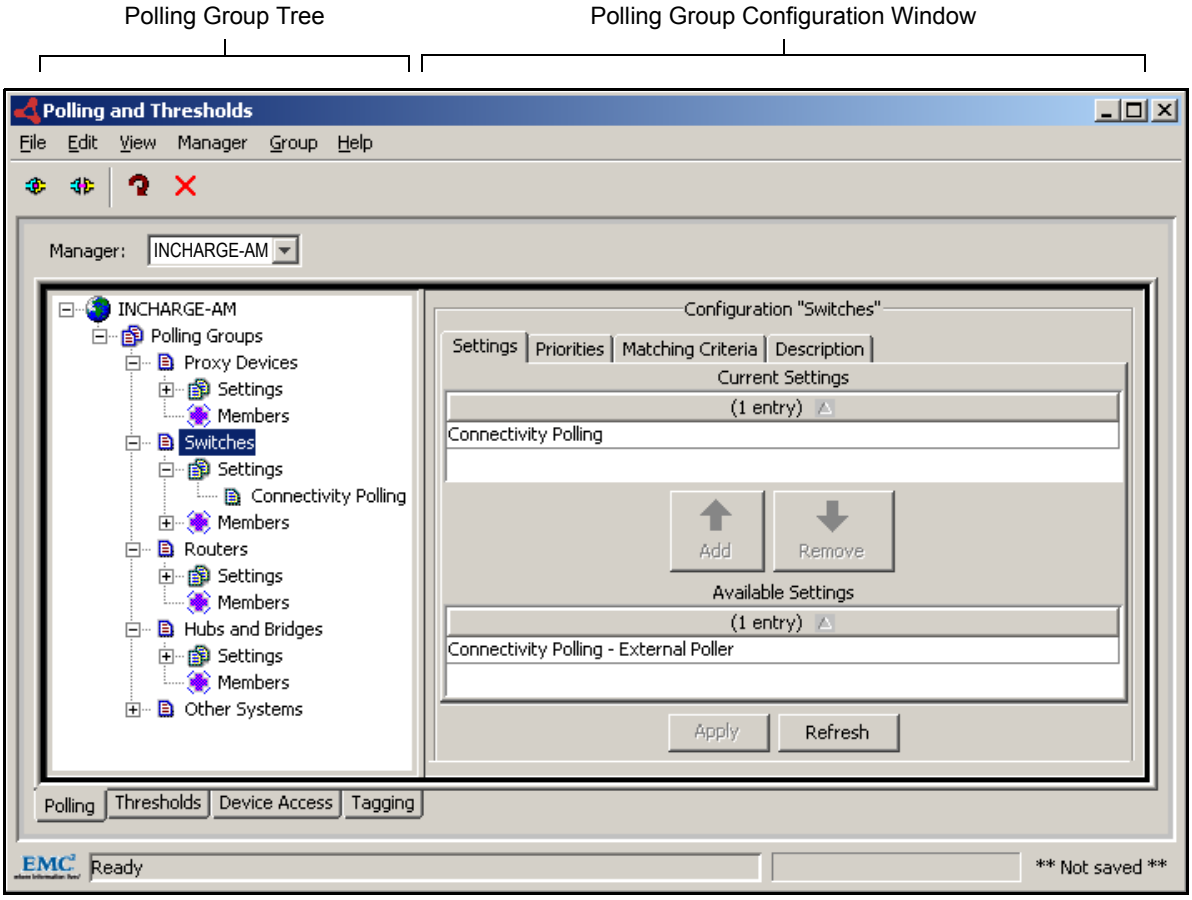


Figure 6-1 Polling tab for IP Availability Manager

Figure 6-2 shows the Polling tab of a Polling and Thresholds Console connected to an IP Performance Manager instance named INCHARGE-PM. The right panel of the console identifies "Connectivity Polling" and "Environment Polling" as two of the four default settings for the default polling group named "Routers." The other two default settings for Routers are "Performance Polling - Ports and Interfaces" and "Performance Polling - Processor and Memory."

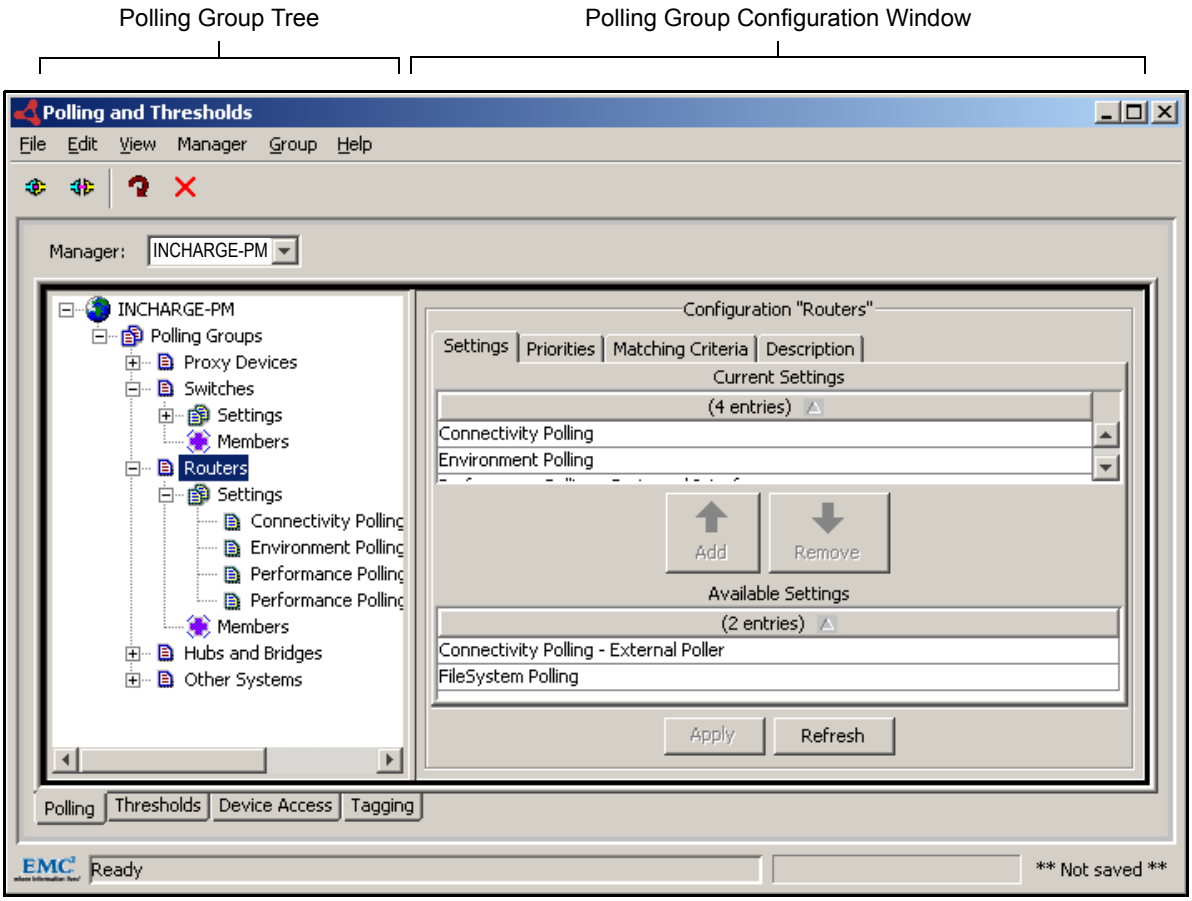


Figure 6-2 Polling tab for IP Performance Manager

Understanding polling groups and settings

IP Availability Manager and IP Performance Manager monitor the network by sending Internet Control Message Protocol (ICMP) polls and Simple Network Management Protocol (SNMP) polls. The results of this polling are then compared to threshold values that define acceptable and unacceptable levels of connectivity or performance. The thresholds, in conjunction with traps, are used to diagnose the failed or degraded objects that interrupt network connectivity or adversely affect network performance.

IP Availability Manager and IP Performance Manager use settings to assign polling and threshold parameters to groups of managed objects. From a polling and threshold perspective, a setting is a collection of parameters common to a particular type of analysis (for example, connectivity polling or environment polling). The parameters defined for the settings of the polling and threshold groups define the management policies for IP Availability Manager and IP Performance Manager.

Modifying polling and polling groups

The *EMC Smarts IP Management Suite Deployment Guide* provides extensive guidelines on when to make changes to existing polling groups and when to create new polling groups. Before modifying any polling groups, you should consult the deployment guide.

Procedures for modifying existing polling groups and creating new ones are given in "Working with groups and settings" on page 5-7.

Default polling groups and settings for IP Availability Manager

Through the default polling groups for IP Availability Manager, you control the ICMP and SNMP polling of the managed network objects in your network. The polled data serves as input to IP Availability Manager's connectivity correlation analysis of switches, routers, hubs, bridges, and other systems.

Appendix B, "Polling for Analysis," provides a description of ICMP and SNMP polling for correlation analysis.

Default polling groups

IP Availability Manager provides five default polling groups:

- Proxy Devices
- Switches
- Routers
- Hubs and Bridges
- Other Systems

Table 6-1 lists all of the default polling groups, their matching criteria, and their settings. The matching-criteria target class for polling groups is UnitaryComputerSystem.

Table 6-1 Default IP Availability Manager polling groups

Polling group	Matching criteria	Default settings
Proxy Devices	UnitaryComputerSystem attributes: • IsProxy = TRUE	Connectivity Polling - External Poller
Switches	UnitaryComputerSystem attributes: • Type = Switch	Connectivity Polling
Routers	UnitaryComputerSystem attributes: • Type = Router	Connectivity Polling
Hubs and Bridges	UnitaryComputerSystem attributes: • Type = Hub or Bridge	Connectivity Polling
Other Systems	UnitaryComputerSystem attributes: • No attributes	Connectivity Polling

The Other Systems polling group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the other polling groups become members of the Other Systems polling group.

Default polling settings

IP Availability Manager provides the following default polling settings:

- Connectivity Polling
- Connectivity Polling - External Poller

These settings are mutually exclusive, meaning that only one or the other (but not both) can be specified for a polling group at any given time.

The following additional default polling settings are available to IP Availability Manager Extension for IPv6:

- Connectivity Polling for IPv6 Addresses
- Connectivity Polling for IPv6 Addresses - External Polling

A feature license controls whether IP Availability Manager Extension for IPv6 is available for configuration. When the feature license is installed, the IPv6 polling settings are available to all IP Availability Manager polling groups. "Enabling IPv6 address polling" on page 6-12 describes these settings and how to apply them to a polling group.

Connectivity Polling setting

The Connectivity Polling setting configures connectivity monitoring of discovered systems. System connectivity is monitored using a combination of ICMP (ping) requests for IP status and SNMP requests for interface, port, and system status.

Table 6-2 lists the Connectivity Polling setting parameters.

Table 6-2 Default values for the Connectivity Polling setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: ENABLED	Enables or disables connectivity polling.
PollingInterval	30 to 3600 seconds Default: 240 seconds	Time between successive connectivity polls.
Retries	0 to 10 retries Default: 3	Number of retry connectivity polls to perform when the initial poll fails.
Timeout	10 to 10000 milliseconds Default: 700 milliseconds	Amount of time to wait for the poll response before the first poll request times out. The timeout value doubles for each successive retry. For Timeout=700 msec (0.7 sec) and Retries=3: • 0.7 seconds for first retry • 1.4 seconds for second retry • 2.8 seconds for third retry

Connectivity Polling - External Poller setting

The Connectivity Polling - External Poller setting determines the initial states of the instrumentation data to be analyzed by IP Availability Manager. (No ICMP or SNMP polling is performed.) The analysis results are collected by an external process for test purposes.

Note: You can create an external poller using the EMC Smarts Software Development Kit or EMC Smarts Perl API.

Table 6-3 lists the Connectivity Polling - External Poller setting parameters.

Table 6-3 Default values for the Connectivity Polling - External Poller setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: DISABLED	Enables or disables the connectivity analysis of data collected by an external process for test purposes.
InitialStatus	UP, UNKNOWN Default: UNKNOWN	Determines the desired initial state for the stored instrumentation objects. Initial status value can be UP or UNKNOWN.
InstrumentCards	TRUE, FALSE Default: FALSE	Determines whether stored instrumentation objects are created for card objects. If TRUE, stored instrumentation objects are created for card objects. If FALSE, no stored instrumentation objects are created for card objects.
InstrumentIP	TRUE, FALSE Default: TRUE	Determines whether stored instrumentation objects are created for IP objects. If TRUE, stored instrumentation objects are created for IP objects. If FALSE, no stored instrumentation objects are created for IP objects.
InstrumentNetwork-Adapters	TRUE, FALSE Default: TRUE	Determines whether stored instrumentation objects are created for interface and port objects. If TRUE, stored instrumentation objects are created for interface and port objects. If FALSE, no stored instrumentation objects are created for interface and port objects.
InstrumentSNMP-Agents	TRUE, FALSE Default: TRUE	Determines whether stored instrumentation objects are created for SNMP agent objects. If TRUE, stored instrumentation objects are created for SNMP agent objects. If FALSE, no stored instrumentation objects are created for SNMP agent objects.

Default polling groups and settings for IP Performance Manager

Through the default polling groups for IP Performance Manager, you control the ICMP and SNMP polling of the managed network objects in your network. The polled data serves as input to IP Performance Manager's fault and performance correlation analysis of ports and interfaces, processors and memory, disks and filesystems, fans and power supplies, and voltage and temperature sensors.

Appendix B, "Polling for Analysis," provides a description of ICMP and SNMP polling for correlation analysis.

Default polling groups

IP Performance Manager provides five default polling groups:

- Proxy Devices
- Switches
- Routers
- Hubs and Bridges
- Other Systems

Table 6-4 lists all of the default polling groups, their matching criteria, and their settings. The matching-criteria target class for polling groups is UnitaryComputerSystem.

Table 6-4 Default IP Performance Manager polling groups

Polling group	Matching criteria	Default settings
Proxy Devices	UnitaryComputerSystem attributes: • IsProxy = TRUE	Connectivity Polling - External Poller
Switches	UnitaryComputerSystem attributes: • Type = Switch	Connectivity Polling Environment Polling Performance Polling - Ports and Interfaces Performance Polling - Processor and Memory
Routers	UnitaryComputerSystem attributes: • Type = Router	Connectivity Polling Environment Polling Performance Polling - Ports and Interfaces Performance Polling - Processor and Memory
Hubs and Bridges	UnitaryComputerSystem attributes: • Type = Hub or Bridge	Connectivity Polling
Other Systems	UnitaryComputerSystem attributes: • No attributes	Connectivity Polling FileSystem Polling Performance Polling - Processor and Memory

The Other Systems polling group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the other polling groups become members of the Other Systems polling group. For example, host systems become members of the Other Systems polling group.

Default polling settings

IP Performance Manager provides the following default polling settings:

- Connectivity Polling (availability type setting)
- Connectivity Polling - External Poller
- Environment Polling
- FileSystem Polling
- Performance Polling - Ports and Interfaces
- Performance Polling - Processor and Memory

Connectivity Polling is an availability type polling setting that is described in "Connectivity Polling setting" on page 6-6. Availability data plays an important part in fault and performance analysis.

Connectivity Polling - External Poller is a testing type polling setting that is described in "Connectivity Polling - External Poller setting" on page 6-7. No ICMP or SNMP polling is performed when using this setting.

Environment Polling setting

The Environment Polling setting configures polling intervals used to monitor system components such as the power supply, fan, voltage sensor, and temperature sensor objects.

Table 6-5 lists the Environment Polling setting parameters.

Table 6-5 Default values for the Environment Polling setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: ENABLED	Enables or disables environment polling.
PollingInterval	30 to 3600 seconds Default: 240 seconds	Time between successive environment polls.
Retries	0 to 10 retries Default: 3	Number of retry environment performance polls to perform when the initial poll fails.
Timeout	10 to 10000 milliseconds Default: 700 milliseconds	Amount of time to wait for the poll response before the first poll request times out. The timeout value doubles for each successive retry. For Timeout=700 msec (0.7 sec) and Retries=3: • 0.7 seconds for first retry • 1.4 seconds for second retry • 2.8 seconds for third retry

FileSystem Polling setting

The FileSystem Polling setting configures polling intervals used to monitor a host system's filesystem. Although available to all polling groups, this setting is a valid choice only for the Other Systems polling group.

Table 6-6 lists the FileSystem Polling setting parameters.

Table 6-6 Default values for the FileSystem Polling setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: ENABLED	Enables or disables filesystem polling.
PollingInterval	30 to 3600 seconds Default: 240 seconds	Time between successive filesystem performance polls.
Retries	0 to 10 retries Default: 3	Number of retry filesystem performance polls to perform when the initial poll fails.
Timeout	10 to 10000 milliseconds Default: 700 milliseconds	Amount of time to wait for the poll response before the first poll request times out. The timeout value doubles for each successive retry. For Timeout=700 msec (0.7 sec) and Retries=3: • 0.7 seconds for first retry • 1.4 seconds for second retry • 2.8 seconds for third retry

Performance Polling - Ports and Interfaces setting

The Performance Polling - Ports and Interfaces setting configures polling intervals used to monitor the performance of managed ports and interfaces.

Table 6-7 lists the Performance Polling - Ports and Interfaces setting parameters.

Table 6-7 Default values for the Performance Polling - Ports and Interfaces setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: ENABLED	Enables or disables port and interface performance polling.
PollingInterval	30 to 3600 seconds Default: 240 seconds	Time between successive polls for interfaces and trunk ports.
PollingIntervalAccess-Ports	1200 seconds	Performance polling interval for access ports.
Retries	0 to 10 retries Default: 3	Number of retry performance polls to perform when the initial poll fails.
Timeout	10 to 10000 milliseconds Default: 700 milliseconds	Amount of time to wait for the poll response before the first poll request times out. The timeout value doubles for each successive retry. For Timeout=700 msec (0.7 sec) and Retries=3: • 0.7 seconds for first retry • 1.4 seconds for second retry • 2.8 seconds for third retry

Performance Polling - Processor and Memory setting

The Performance Polling - Processor and Memory setting configures polling intervals used to monitor a system's processor and associated memory objects.

Table 6-8 lists the Performance Polling - Processor and Memory setting parameters.

Table 6-8 Default values for the Performance Polling - Processor and Memory setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: ENABLED	Enables or disables processor and memory performance polling.
PollingInterval	30 to 3600 seconds Default: 240 seconds	Time between successive processor and memory performance polls.
Retries	0 to 10 retries Default: 3	Number of retry processor and memory performance polls to perform when the initial poll fails.
Timeout	10 to 10000 milliseconds Default: 700 milliseconds	Amount of time to wait for the poll response before the first poll request times out. The timeout value doubles for each successive retry. For Timeout=700 msec (0.7 sec) and Retries=3: • 0.7 seconds for first retry • 1.4 seconds for second retry • 2.8 seconds for third retry

Enabling IPv6 address polling

For IP Availability Manager Extension for IPv6, the following additional polling settings are available for polling managed systems having IPv6 addresses configured:

- Connectivity Polling for IPv6 Addresses
- Connectivity Polling for IPv6 Addresses - External Polling

These settings are available to all IP Availability Manager polling groups. By default, they are not contained in any of these groups.

You should apply the Connectivity Polling for IPv6 Addresses setting to any polling group that has at least one system having at least one IPv6 address configured. This setting works in conjunction with the Connectivity Polling setting, so the Connectivity Polling setting should also be applied to the target polling group.

If you are using an external poller instead of the one included within the EMC Smarts IP Management Suite, you may apply the Connectivity Polling for IPv6 Addresses - External Polling setting to a polling group. If you apply this setting, you must remove the Connectivity Polling for IPv6 Addresses setting and the Connectivity Polling setting from the polling group.

Connectivity Polling for IPv6 Addresses setting

The Connectivity Polling for IPv6 Addresses setting configures connectivity monitoring of IPv6 addresses configured on discovered systems. The connectivity is monitored using IPv6 ICMP requests for IP status.

Table 6-9 lists the Connectivity Polling for IPv6 Addresses setting parameters. (The parameters, defaults, and allowed values are identical to the Connectivity Polling parameters described in "Connectivity Polling setting" on page 6-6.)

Table 6-9 Default values for the Connectivity Polling for IPv6 Addresses setting

Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: ENABLED	Enables or disables IPv6 ICMP connectivity polling.
PollingInterval	30 to 3600 seconds Default: 240 seconds	Time between successive connectivity polls.
Retries	0 to 10 retries Default: 3	Number of retry connectivity polls to perform when the initial poll fails.
Timeout	10 to 10000 milliseconds Default: 700 milliseconds	Amount of time to wait for the poll response before the first poll request times out. The timeout value doubles for each successive retry. For Timeout=700 msec (0.7 sec) and Retries=3: • 0.7 seconds for first retry • 1.4 seconds for second retry • 2.8 seconds for third retry

To use the Connectivity Polling for IPv6 Addresses setting, you need to add the setting to a polling group. Accessing the Polling and Thresholds Console and adding a setting to a group are described in "Opening the Polling and Thresholds Console" on page 5-2 and "Working with groups and settings" on page 5-7.

Connectivity Polling for IPv6 Addresses - External Poller setting

The Connectivity Polling for IPv6 Addresses - External Poller setting determines the initial states of the instrumentation data to be analyzed by IP Availability Manager. (No IPv6 ICMP polling is performed.) The analysis results are collected by an external process for test purposes.

Note: You can create an external poller using the EMC Smarts Software Development Kit or EMC Smarts Perl API.

Table 6-10 lists the Connectivity Polling for IPv6 Addresses - External Polling setting parameters.

Table 6-10 Default values for the Connectivity Polling for IPv6 Addresses - External Polling setting

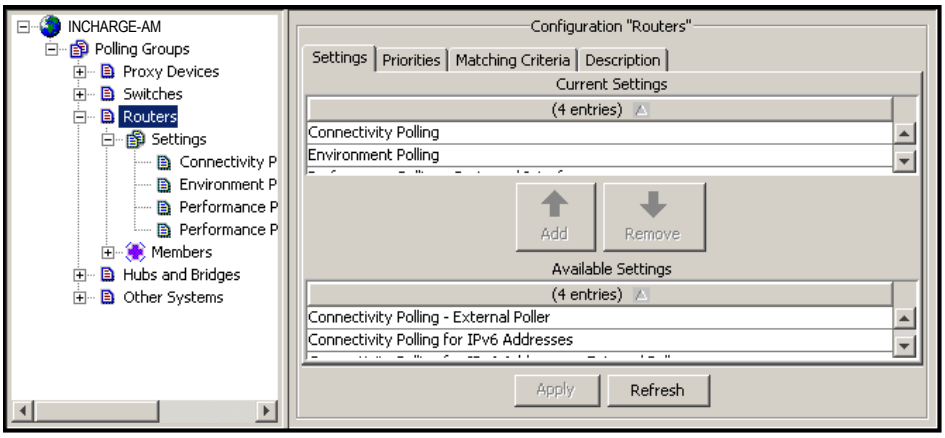
Parameter	Default value	Description
AnalysisMode	ENABLED, DISABLED Default: DISABLED	Enables or disables the connectivity analysis of data collected by an external process for test purposes.
InitialStatus	UP, UNKNOWN Default: UNKNOWN	Determines the desired initial state for the stored instrumentation objects. Initial status value can be UP or UNKNOWN.
InstrumentIPv6s	TRUE, FALSE Default: TRUE	Determines whether stored instrumentation objects are created for IPv6 objects. If TRUE, stored instrumentation objects are created for IPv6 objects. If FALSE, no stored instrumentation objects are created for IPv6 objects.

To use the Connectivity Polling for IPv6 Addresses - External Polling setting, you need to add the setting to a polling group. Accessing the Polling and Thresholds Console and adding a setting to a group are described in "Opening the Polling and Thresholds Console" on page 5-2 and "Working with groups and settings" on page 5-7.

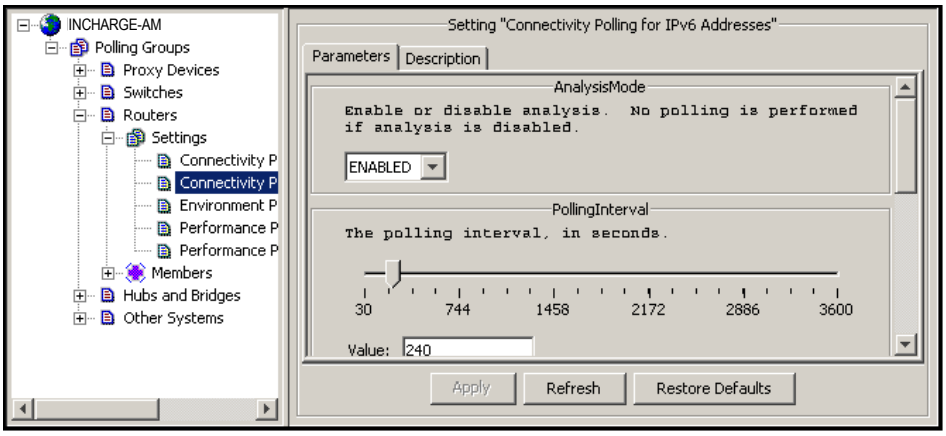
Example of applying an IPv6 polling setting

The following procedure demonstrates how to use the Polling and Thresholds Console to apply the Connectivity Polling for IPv6 Addresses setting to the default polling group named Routers.

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager Extension for IPv6, click the **Polling** tab to display the polling group tree.
2. In the polling group tree, click the **+** signs to expand the polling group tree as shown in the following display, then click **Routers** to display the configuration window and the **Settings** tab.



3. In the **Available Settings** list of the configuration window, select **Connectivity Polling for IPv6 Addresses** and click **Add** to move the **Connectivity Polling for IPv6 Addresses** setting to the **Current Settings** list.
4. Click the **Apply** button to save your changes.
5. In the polling group tree, click **Connectivity Polling for IPv6 Addresses** to display its setting configuration window.



6. In the setting configuration window, consult Table 6-9 on page 6-12 and enter your parameter value choices.
7. Click the **Apply** button to save your changes.
8. Optional: In the polling group tree, click **Routers** to display the configuration window.

9. Optional: In the configuration window, click the **Matching Criteria** tab to display the **Matching Criteria** tab page.
10. Optional: In the **Matching Criteria** tab page, specify additional matching criteria for the **Routers** group and click the **Apply** button to save your changes. (The procedure for editing matching criteria is given in "Editing matching criteria" on page 5-9.)

    The IPv6 addresses configured on any router that becomes a member of the **Routers** group will automatically be polled in accordance to the values that you specified in Step 6.
11. Click the **Reconfigure** toolbar button to make your changes take effect.

7

Customizing Polled Data Thresholds

This chapter describes the default threshold groups and settings for IP Availability Manager and IP Performance Manager. It consists of the following sections:

- Viewing the Thresholds tab 7-2
- Understanding threshold groups and settings 7-4
- Default threshold groups and settings for IP Availability Manager 7-5
- Default threshold groups and settings for IP Performance Manager 7-12
- Using a policy setting to control the managed state of interfaces 7-22
- Using a policy setting to ensure appropriate VLAN assignments 7-25
- Using a policy setting to prevent VLAN impacts 7-28

Viewing the Thresholds tab

Figure 7-1 shows the Thresholds tab of a Polling and Thresholds Console connected to an IP Availability Manager instance named INCHARGE-AM. The right panel of the console identifies "Connectivity" as the default setting for the default "System Resource Group" named "Switches."

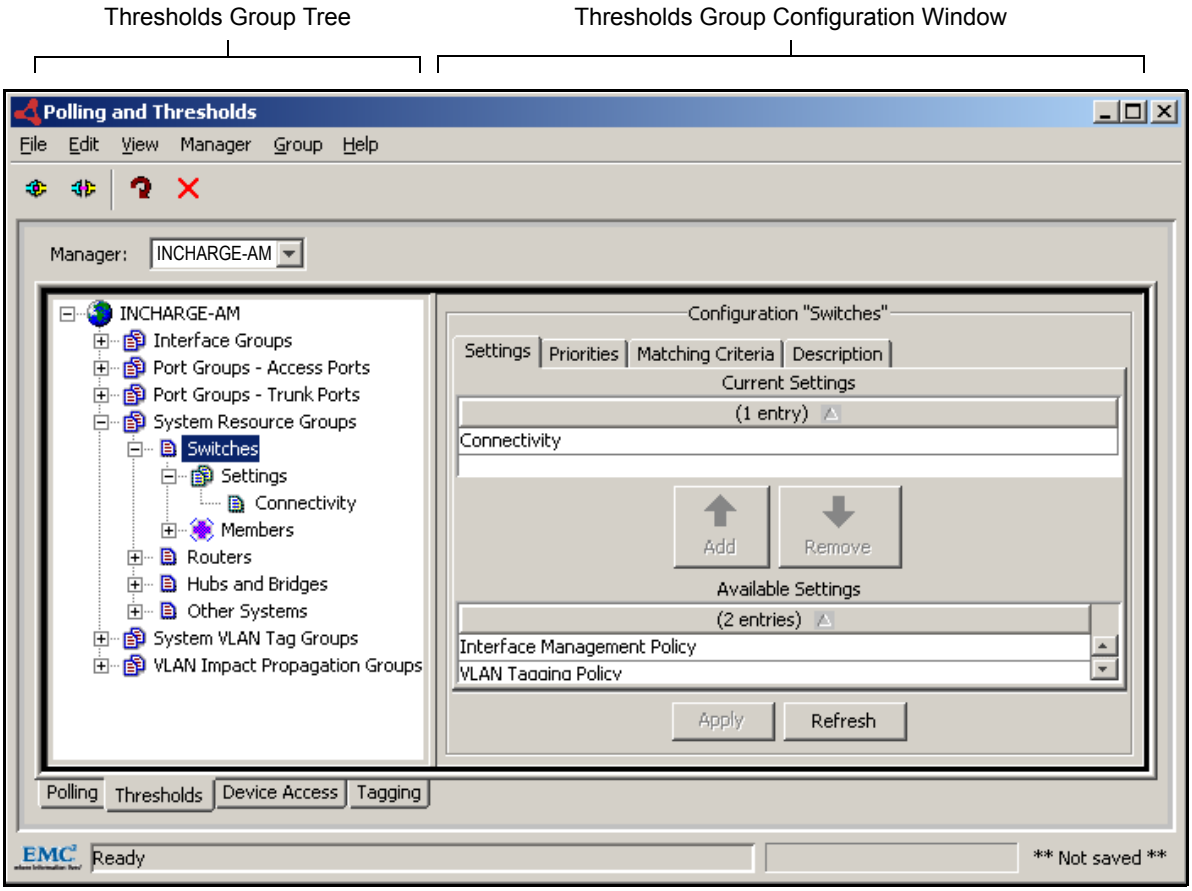


Figure 7-1 Thresholds tab for IP Availability Manager

Figure 7-2 shows the Thresholds tab of a Polling and Thresholds Console connected to an IP Performance Manager instance named INCHARGE-PM. The right panel of the console identifies "Ethernet Interface/Port Performance" as the default setting for the default "Port Groups - Trunk Ports" group named "1 Gb Ethernet."

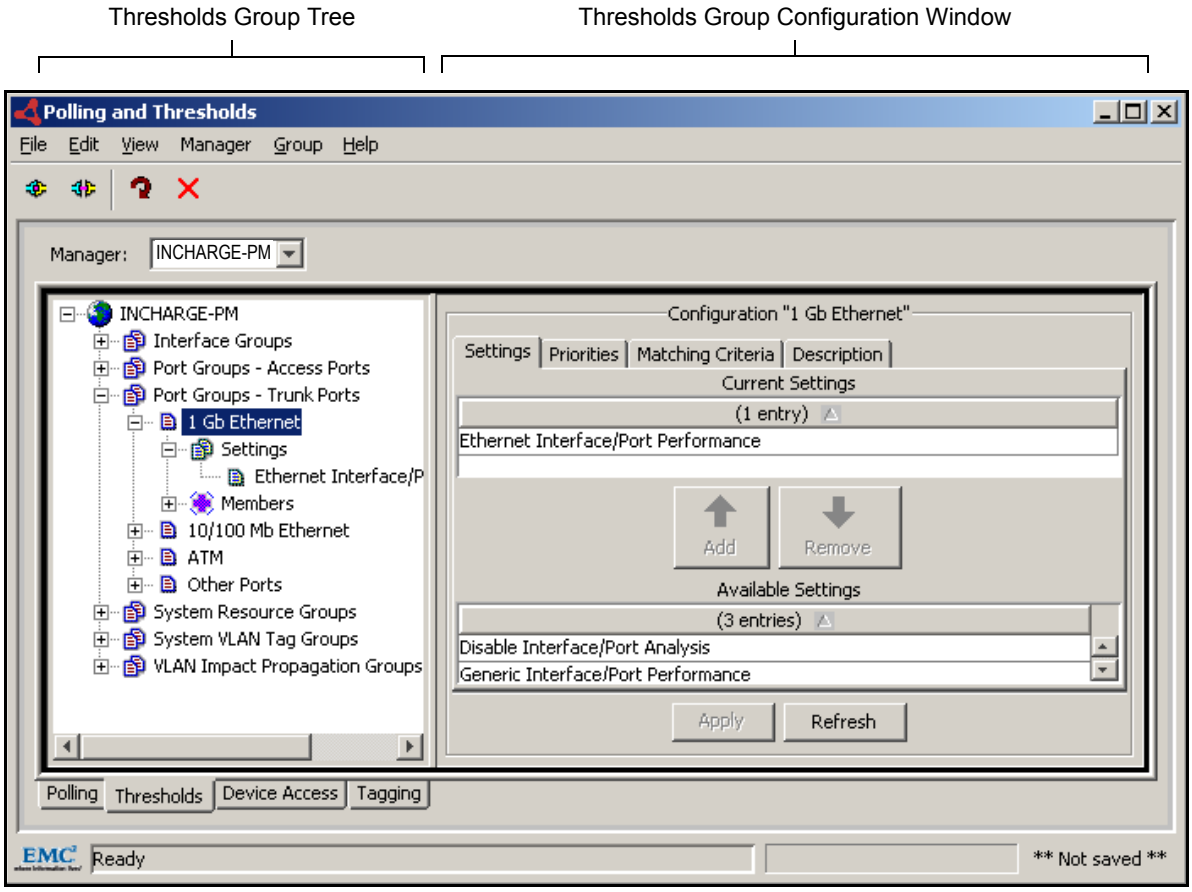


Figure 7-2 Thresholds tab for IP Performance Manager

Understanding threshold groups and settings

IP Availability Manager and IP Performance Manager monitor the network by sending ICMP and SNMP polls. The results of this polling are then compared to threshold values that define acceptable and unacceptable levels of connectivity or performance. The thresholds, in conjunction with traps, are used to diagnose the failed or degraded objects that interrupt network connectivity or adversely affect network performance.

IP Availability Manager and IP Performance Manager use settings to assign polling and threshold parameters to groups of managed objects. From a polling and threshold perspective, a setting is a collection of parameters common to a particular type of analysis (for example, connectivity polling or environment polling). The parameters defined for the settings of the polling and threshold groups define the management policies for IP Availability Manager and IP Performance Manager.

Modifying threshold values and threshold groups

The *EMC Smarts IP Management Suite Deployment Guide* provides extensive guidelines on when to make changes to existing threshold groups and when to create new threshold groups. Before modifying any threshold groups, you should consult the deployment guide.

Procedures for modifying existing threshold groups and creating new ones are given in "Working with groups and settings" on page 5-7.

Default threshold groups and settings for IP Availability Manager

The default threshold categories, groups, and settings for IP Availability Manager are accessible via the Thresholds tab of the Polling and Thresholds Console. Through these groups and settings, you set thresholds for the data collected by the ICMP and SNMP polling. IP Availability Manager compares the polled data against the thresholds to determine fault conditions and to perform fault and connectivity correlation analysis.

Default threshold categories and groups

IP Availability Manager provides the following six default categories of threshold groups, each of which contains default threshold groups:

- Interface Groups
- Port Groups - Access Ports
- Port Groups - Trunk Ports
- System Resource Groups
- System VLAN Tag Groups
- VLAN Impact Propagation Groups

Interface Groups

Threshold groups for interfaces are used to configure parameters for interface analysis. Thresholds are determined by the interface's media type (for example, Ethernet or ATM) and its role (primary, backup, or dial-on-demand).

Table 7-1 lists the default interface threshold groups, their matching criteria, and their settings. The matching-criteria target class for interface threshold groups is Interface.

Table 7-1 Default IP Availability Manager Interface Groups

Interface Group	Matching criteria	Default settings
1 Gb Ethernet	Interface attributes: • MaxSpeed = 1,000,000,000 • Type = ETHER or CSMACD	No default settings
10/100 Mb Ethernet	Interface attributes: • Type = ETHER or CSMACD	No default settings
ATM	Interface attributes: • Type = ATM	Interface/Port Flapping
Token Ring	Interface attributes: • Type = TOKEN	No default settings
ISDN Physical Interface	Interface attributes: • InterfaceCode = ISDNPHYSICAL	Interface/Port Flapping
ISDN B Channel	Interface attributes: • InterfaceCode = ISDNBCHANNEL	Backup Interface Support
ISDN D Channel	Interface attributes: • InterfaceCode = ISDNDCHANNEL	Interface/Port Flapping

Table 7-1 Default IP Availability Manager Interface Groups (continued)

Interface Group	Matching criteria	Default settings
Serial	Interface attributes: • Type = Serial or FrameRelay	Interface/Port Flapping
FDDI	Interface attributes: • Type = FDDI	No default settings
Backup	Interface attributes: • Type = ISDN	Backup Interface Support
Dial-On-Demand	Interface attributes: • Type = PPP or SLIP	Dial-On-Demand Interface Support
Other Interfaces	Interface attributes: • No attributes	Interface/Port Flapping

The Other Interfaces threshold group does not contain any matching-criteria attributes and has the lowest priority. Interfaces that do not match the criteria for the other interface threshold groups become members of the Other Interfaces group.

Port Groups - Access Ports and Trunk Ports

Threshold groups for ports are used to configure parameters for port analysis. Thresholds are determined by the port's media type (for example, Ethernet or ATM) and its role (access or trunk).

An access port is a switch port that is connected to a host. Access ports are unmanaged by default and are not associated with the access ports groups. An access port automatically becomes managed if the host that it is connected to is managed or if the port is explicitly managed. A trunk port is a switch port that is connected to a switch, router, hub, or bridge.

Table 7-2 shows the default access and trunk port threshold groups, their matching criteria, and their settings. By default, no settings are applied to the port threshold groups. The matching-criteria target class for port threshold groups is Port.

Table 7-2 Default IP Availability Manager Port Groups - Access Ports and Trunk Ports

Port Group	Matching criteria	Default settings
1 Gb Ethernet	Port attributes: • Max Speed = 1,000,000,000 • Type = ETHER or CSMACD	No default settings
10/100 Mb Ethernet	Port attributes: • Type = ETHER or CSMACD	No default settings
ATM	Port attributes: • Type = ATM	No default settings
Other Ports	Port attributes: • No attributes	No default settings

The Other Ports group does not contain any matching-criteria attributes and has the lowest priority. Access ports that do not match the criteria for the other access ports groups become members of the Other Ports group.

System Resource Groups

Threshold groups for system resources are used to configure parameters for system analysis. The sensitivity of the system thresholds is determined by the role of the system.

Table 7-3 shows the default system resource threshold groups, their matching criteria, and their settings. The matching-criteria target class for system resource threshold groups is UnitaryComputerSystem.

Table 7-3 Default IP Availability Manager System Resource Groups

System Resource Group	Matching criteria	Default settings
Switches	UnitaryComputerSystem attributes: • Type = Switch	Connectivity
Routers	UnitaryComputerSystem attributes: • Type = Router	Connectivity
Hubs and Bridges	UnitaryComputerSystem attributes: • Type = Hub or Bridge	Connectivity
Other Systems	UnitaryComputerSystem attributes: • No attributes	Connectivity

The Other Systems group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the other system resource groups become members of the Other Systems group. For example, host systems become members of the Other Systems group.

System VLAN Tag Groups

Threshold groups for VLAN tagging are used to ensure that systems are assigned to their appropriate VLAN. These groups are described in"Using a policy setting to ensure appropriate VLAN assignments" on page 7-25.

Table 7-4 shows the default system VLAN tag groups, their matching criteria, and their settings. The matching-criteria target class for system VLAN tag groups is UnitaryComputerSystem.

Table 7-4 Default IP Availability Manager System VLAN Tag Groups

System VLAN Tag Group	Matching criteria	Default settings
Switches	UnitaryComputerSystem attributes: • Type = Switch	VLAN Tagging Policy
Other Systems	UnitaryComputerSystem attributes: • No attributes	No default settings

The Other Systems group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the Switches system VLAN tag group become members of the Other Systems group.

VLAN Impact Propagation Groups

Threshold groups for VLAN impact propagation are used to prevent device impact propagation in certain (user-specified) VLANs, such as management VLANs and default VLANs. These groups are described in"Using a policy setting to prevent VLAN impacts" on page 7-28.

Table 7-5 shows the default VLAN impact propagation groups, their matching criteria, and their settings. The matching-criteria target class for VLAN impact propagation groups is VLAN.

Table 7-5 Default IP Availability Manager VLAN Impact Propagation Groups

VLAN Impact Propagation Group	Matching criteria	Default settings
VLANs	VLAN attributes: • CreationClassName = VLAN • VLANKey = <user-defined value>	Problem Impact propagation policy over VLANs
Other VLANs	VLAN attributes: • No attributes	No default settings

The Other VLANs group does not contain any matching-criteria attributes and has the lowest priority. VLANs that do not match the criteria for the VLANs group become members of the Other VLANs group.

Default threshold settings

IP Availability Manager provides the following default threshold settings:

- Backup Interface Support (by default, contained in the Interface Group - ISDN B Channel and Interface Group - Backup)
- Connectivity (by default, contained in all System Resource Groups and available for all System VLAN Tag Groups)
- Dial-on-Demand Interface Support (by default, contained in the Interface Group - Dial-on-Demand)
- Interface Management Policy (by default, not contained in any group but available for all System Resource Groups and all System VLAN Tag Groups)
- Interface/Port Flapping (by default, contained in Interface Groups for ATM, ISDN Physical Interfaces, ISDN D Channel, Serial, and Other Interfaces)
- Problem Impact propagation policy over VLANs (by default, contained in the VLAN Impact Propagation Group named VLANs and available in the VLAN Impact Propagation Group named Other VLANs)
- VLAN Tagging Policy (by default, contained in the System VLAN Tag Group named Switches and available in the System VLAN Tag Group named Other Systems and in all System Resource Groups)

"Using a policy setting to control the managed state of interfaces" on page 7-22 introduces the "Interface Management Policy" setting and describes how to apply it to a system resource group.

"Using a policy setting to ensure appropriate VLAN assignments" on page 7-25 introduces the "VLAN Tagging Policy" setting and describes how to use it to create a system VLAN tag group.

"Using a policy setting to prevent VLAN impacts" on page 7-28 introduces the "Problem Impact propagation policy over VLANs" setting and describes how to use it to create a VLAN impact propagation group.

Backup Interface Support threshold setting

The Backup Interface Support threshold setting configures analysis for interfaces used as backup. When an interface is identified as a backup, the connectivity failure diagnosis is modified for it as follows:

- The Down notification is not generated if the interface is down.
- The BackupActivated notification is generated if the interface comes up.
- The ExceededMaximumUptime notification is generated if the interface stays up too long.

Table 7-6 lists the Backup Interface Support threshold setting parameters.

Table 7-6 Default values for the Backup Interface Support threshold setting

Parameter	Default value	Description
MaximumUptime	0	Maximum length of time, in seconds, that the interface may be up before the ExceededMaximumUptime notification is generated. If the value of this parameter is 0, the ExceededMaximumUptime event is disabled.

Connectivity threshold setting

The Connectivity threshold setting configures connectivity threshold parameters for network adapters (ports and interfaces). It also controls:

- The use of bridging connectivity for root-cause analysis.
- The analysis of systems that repeatedly restart, and are thus considered unstable. (Chapter 4, "Diagnosis of Unstable Objects," in the *EMC Smarts IP Availability Manager User's Guide* provides information about how IP Availability Manager concludes that a system is unstable.)
- Whether SystemRestarted traps are sent to the EMC Smarts Service Assurance Manager Adapter Platform (Adapter Platform).

Table 7-7 lists the Connectivity threshold setting parameters.

Table 7-7 Default values for the Connectivity threshold setting

Parameter	Default value	Description
CorrelationUseBridgingMode	ENABLED, DISABLED Default: ENABLED	Enables or disables the use of bridging connectivity for root-cause analysis. Bridging connectivity is not as precise as physical connectivity and this parameter controls the extent to which the analysis relies on bridging connectivity. The parameter is set to ENABLED by default because, under normal conditions, most physical connectivity is discovered and there is sparse bridging connectivity. However, you can set the parameter to DISABLED in cases where there is sparse physical connectivity and heavily meshed bridging connectivity.

Table 7-7 Default values for the Connectivity threshold setting (continued)

Parameter	Default value	Description
EnableGenericOIEvent	ENABLED, DISABLED Default: DISABLED	Applicable to IP Availability Manager only: Enables or disables the sending of a SystemRestarted trap for a system to the Adapter Platform whenever the system is in the UP state for less than the time specified in the MaxUpTimeThreshold parameter. ENABLED activates the sending of a SystemRestarted trap to the Adapter Platform. DISABLED suppresses the sending of a SystemRestarted trap to the Adapter Platform.
MaxUpTimeThreshold	1 to 1436 minutes Default: 60 minutes	Applicable to IP Availability Manager only: Specifies the threshold uptime, in minutes, below which a system is considered to have been restarted. That is, if a system is in the UP state for less than the time specified for this parameter, IP Availability Manager considers the system as having been restarted.
NumberOfBridgedViaThreshold	20	Determines whether a BridgedVia relationship is used in correlation. If a MAC endpoint is BridgedVia less than the number of Ports specified by this threshold, the BridgedVia relationship is used for correlation.
NumberOfBridgesThreshold	20	Determines whether a Bridges relationship is used in correlation. If a Port Bridges less than the number of MAC endpoints specified by this threshold, the Bridges relationship is used for correlation.
RestartTrapThreshold	3	Sets the number of SNMP cold or warm start traps that must be received within the amount of time set by the RestartTrapWindow parameter in order for IP Availability Manager to consider a system unstable. A value of 0 turns off restart analysis.
RestartTrapWindow	900 seconds (15 minutes)	Sets the window of time used to monitor a system's repeated restarts. If the number of start traps meets or exceeds the RestartTrapThreshold during this window of time, the system is considered to be unstable.
TestingNotificationMode	ENABLED, DISABLED Default: ENABLED	Enables or disables the generation of the Down notification for a port or interface. ENABLED activates the diagnosis of a port or interface that is in testing status. DISABLED suppresses the diagnosis of a port or interface that is in testing status.

Dial-On-Demand Interface Support threshold setting

The Dial-on-Demand Interface Support threshold setting configures the analysis for interfaces used as dial-on-demand. When an interface is identified as dial-on-demand, the connectivity failure diagnosis is modified for it as follows:

- The Down notification is not generated if the interface is down.
- The ExceededMaximumUptime notification is generated if the interface stays up too long.

Table 7-8 lists the Dial-On-Demand Interface Support threshold setting parameters.

Table 7-8 Default values for the Dial-On-Demand Interface Support threshold setting

Parameter	Default value	Description
MaximumUptime	7200 seconds (2 hours)	Sets the maximum length of time that the interface may be up before the ExceededMaximumUptime notification is generated. If the value of this parameter is 0, the ExceededMaximumUptime event is disabled.

Interface/Port Flapping threshold setting

The Interface/Port Flapping threshold setting controls the analysis of network adapters (ports and interfaces) that are continually going up and down. Unstable analysis monitors SNMP link down traps to identify a flapping network adapter and then generates a notification to report that it is unstable. (Chapter 4, "Diagnosis of Unstable Objects," in the *EMC Smarts IP Availability Manager User's Guide* provides information about how IP Availability Manager concludes that a network adapter is unstable.)

Table 7-9 lists the Interface/Port Flapping threshold setting parameters.

Table 7-9 Default values for the Interface/Port Flapping threshold setting

Parameter	Default value	Description
LinkTrapThreshold	3	Sets the number of SNMP link down traps that must be received within the LinkTrapWindow in order for IP Availability Manager to consider the interface or port flapping. A value of 0 turns off flapping analysis.
LinkTrapWindow	300 seconds (5 minutes)	Sets the window of time used to monitor flapping analysis of a port or interface. If the number of link down traps meets or exceeds the LinkTrapThreshold during this window of time, the interface or port is considered to be flapping.

Default threshold groups and settings for IP Performance Manager

The default threshold categories, groups, and settings for IP Performance Manager are accessible via the Thresholds tab of the Polling and Thresholds Console. Through these groups and settings, you set thresholds for the data collected by the ICMP and SNMP polling. IP Performance Manager compares the polled data against the thresholds to determine fault conditions and to perform fault and performance correlation analysis.

Default threshold categories and groups

IP Performance Manager provides the following six default categories of threshold groups, each of which contains default threshold groups:

- Interface Groups
- Port Groups - Access Ports
- Port Groups - Trunk Ports
- System Resource Groups
- System VLAN Tag Groups
- VLAN Impact Propagation Groups

Interface Groups

Threshold groups for interfaces are used to configure parameters for interface analysis. Thresholds are determined by the interface's media type (for example, Ethernet, ATM) and its role (primary, backup, or dial-on-demand).

Table 7-10 lists the default interface threshold groups, their matching criteria, and their settings. The matching-criteria target class for interface threshold groups is Interface.

Table 7-10 Default IP Performance Manager Interface Groups

Interface Group	Matching criteria	Default settings
1 Gb Ethernet	Interface attributes: • Max Speed = 1,000,000,000 • Type = ETHER or CSMACD	Ethernet Interface/Port Performance
10/100 Mb Ethernet	Interface attributes: • Type = ETHER or CSMACD	Ethernet Interface/Port Performance
ATM	Interface attributes: • Type = ATM	Generic Interface/Port Performance Interface/Port Flapping
Token Ring	Interface attributes: • Type = TOKEN	Generic Interface/Port Performance
ISDN Physical Interface	Interface attributes: • InterfaceCode = ISDNPHYSICAL	Generic Interface/Port Performance Interface/Port Flapping
ISDN B Channel	Interface attributes: • InterfaceCode = ISDNBCHANNEL	Backup Interface Support
ISDN D Channel	Interface attributes: • InterfaceCode = ISDNDCHANNEL	Interface/Port Flapping

Table 7-10 Default IP Performance Manager Interface Groups (continued)

Interface Group	Matching criteria	Default settings
Serial	Interface attributes: • Type = Serial or FrameRelay	Generic Interface/Port Performance Interface/Port Flapping
FDDI	Interface attributes: • Type = FDDI	Generic Interface/Port Performance
Backup	Interface attributes: • Type = ISDN	Backup Interface Support
Dial-On-Demand	Interface attributes: • Type = PPP or SLIP	Dial-On-Demand Interface Support
Other Interfaces	Interface attributes: • No attributes	Generic Interface/Port Performance Interface/Port Flapping

The Other Interfaces threshold group does not contain any matching-criteria attributes and has the lowest priority. Interfaces that do not match the criteria for the other threshold groups become members of the Other Interfaces threshold group.

Port Groups - Access Ports and Trunk Ports

Threshold groups for ports are used to configure parameters for port analysis. Thresholds are determined by the port's media type (for example, Ethernet, ATM) and its role (access or trunk).

An access port is a switch port that is connected to a host. Access ports are unmanaged by default and are not members of the access ports groups. An access port automatically becomes managed if the host that it is connected to is managed or if it is explicitly managed. A trunk port is a switch port that is connected to a switch, router, hub, or bridge.

Table 7-11 shows the default access and trunk port threshold groups, their matching criteria, and their settings. The matching-criteria target class for port threshold groups is Port.

Table 7-11 Default IP Performance Manager Port Groups - Access Ports and Trunk Ports

Port Group	Matching criteria	Default settings
1 Gb Ethernet	Port attributes: • Max Speed = 1,000,000,000 • Type = ETHER or CSMACD	Ethernet Interface/Port Performance
10/100 Mb Ethernet	Port attributes: • Type = ETHER or CSMACD	Ethernet Interface/Port Performance
ATM	Port attributes: • Type = ATM	Generic Interface/Port Performance
Other Ports	Port attributes: • No attributes	Generic Interface/Port Performance

The Other Ports threshold group does not contain any matching-criteria attributes and has the lowest priority. Access ports that do not match the criteria for the other access ports groups become members of the Other Ports group.

System Resource Groups

Threshold groups for system resources are used to configure parameters for system analysis. The sensitivity of the system thresholds is determined by the role of the system.

Table 7-12 shows the default system resource threshold groups, their matching criteria, and their settings. The matching-criteria target class for system resource threshold groups is UnitaryComputerSystem.

Table 7-12 Default IP Performance Manager System Resource Groups

Port Resource Group	Matching criteria	Default settings
Switches	UnitaryComputerSystem attributes: • Type = Switch	Connectivity Environment Processor and Memory
Routers	UnitaryComputerSystem attributes: • Type = Router	Connectivity Environment Processor and Memory
Hubs and Bridges	UnitaryComputerSystem attributes: • Type = Hub or Bridge	Connectivity
Other Systems	UnitaryComputerSystem attributes: • No attributes	Connectivity Environment FileSystem Processor and Memory

The Other Systems group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the other system resource groups become members of the Other Systems group. For example, host systems become members of the Other Systems group.

System VLAN Tag Groups

Threshold groups for VLAN tagging are used to ensure that systems are assigned to their appropriate VLAN. These groups are described in"Using a policy setting to ensure appropriate VLAN assignments" on page 7-25.

Table 7-13 shows the default system VLAN tag groups, their matching criteria, and their settings. The matching-criteria target class for system VLAN tag groups is UnitaryComputerSystem.

Table 7-13 Default IP Performance Manager System VLAN Tag Groups

System VLAN Tag Group	Matching criteria	Default settings
Switches	UnitaryComputerSystem attributes: • Type = Switch	VLAN Tagging Policy
Other Systems	UnitaryComputerSystem attributes: • No attributes	No default settings

The Other Systems group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the Switches system VLAN tag group become members of the Other Systems group.

VLAN Impact Propagation Groups

Threshold groups for VLAN impact propagation are used to prevent device impact propagation in certain (user-specified) VLANs, such as management VLANs and default VLANs. These groups are described in"Using a policy setting to prevent VLAN impacts" on page 7-28.

Table 7-14 shows the default VLAN impact propagation groups, their matching criteria, and their settings. The matching-criteria target class for VLAN impact propagation groups is VLAN.

Table 7-14 Default IP Performance Manager VLAN Impact Propagation Groups

VLAN Impact Propagation Group	Matching criteria	Default settings
VLANs	VLAN attributes: • CreationClassName = VLAN • VLANKey = <user-defined value>	Problem Impact propagation policy over VLANs
Other VLANs	VLAN attributes: • No attributes	No default settings

The Other VLANs group does not contain any matching-criteria attributes and has the lowest priority. VLANs that do not match the criteria for the VLANs group become members of the Other VLANs group.

Default threshold settings

IP Performance Manager provides the following default threshold settings:

- Backup Interface Support (availability type setting)
- Connectivity (availability type setting)
- Dial-On-Demand Interface Support (availability type setting)
- Interface/Port Flapping (availability type setting)
- Ethernet Interface/Port Performance
- Generic Interface/Port Performance
- Environment
- FileSystem
- Interface Management Policy
- Problem Impact propagation policy over VLANs
- Processor and Memory
- VLAN Tagging Policy

Table 7-15 identifies the default threshold settings for the default threshold groups.

Table 7-15 Groups and default performance threshold settings

Default groups	Default settings
Interface Group - 1 Gb Ethernet Interface Group - 10/100 Mb Ethernet Port Group (Access Port) - 1 Gb Ethernet Port Group (Access Port) - 10/100 Mb Ethernet Port Group (Trunk Port) - 1 Gb Ethernet Port Group (Trunk Port) - 10/100 Mb Ethernet	Ethernet Interface/Port Performance
Interface Group - Token Ring Interface Group - FDDI Port Group (Access Port) - ATM Port Group (Access Port) - Other Ports Port Group (Trunk Port) - ATM Port Group (Trunk Port) - Other Ports	Generic Interface/Port Performance
Interface Group - ATM Interface Group - ISDN Physical Interface Interface Group - Serial Interface Group - Other Interfaces	Generic Interface/Port Performance Interface/Port Flapping
Interface Group - ISDN B Channel Interface Group - Backup	Backup Interface Support
Interface Group - ISDN D Channel	Interface/Port Flapping
Interface Group - Dial-On-Demand	Dial-On-Demand Interface Support
System Resource Group - Switches System Resource Group - Routers	Connectivity Environment Processor and Memory
System Resource Group - Hubs and Bridges	Connectivity
System Resource Group - Other Systems	Connectivity Environment FileSystem Processor and Memory
System VLAN Tag Group - Switches System VLAN Tag Group - Other Systems	VLAN Tagging Policy
VLAN Impact Propagation Group - VLANs VLAN Impact Propagation Group - Other VLANs	Problem Impact propagation policy over VLANs
By default, not contained in any group but available for all System Resource Groups and all System VLAN Tag Groups	Interface Management Policy
By default, not contained in any group but available for all System Resource Groups and all System VLAN Tag Groups	Juniper Environment

The Connectivity threshold setting is an availability type threshold setting that is described in "Connectivity threshold setting" on page 7-9.

"Using a policy setting to control the managed state of interfaces" on page 7-22 introduces the "Interface Management Policy" setting and describes how to apply it to a system resource group.

"Using a policy setting to ensure appropriate VLAN assignments" on page 7-25 introduces the "VLAN Tagging Policy" setting and describes how to use it to create a system VLAN tag group.

"Using a policy setting to prevent VLAN impacts" on page 7-28 introduces the "Problem Impact propagation policy over VLANs" setting and describes how to use it to create a VLAN impact propagation group.

Ethernet Interface/Port Performance threshold setting

The Ethernet Interface/Port Performance threshold setting configures the monitoring of an Ethernet network adapter's performance characteristics. Parameters in this setting control the notification of these faults:

- High Broadcast Rate
- High Collision Rate
- High Discard Rate
- High Queue Drop Rate
- High Error Rate
- High Utilization

One other parameter in this setting determines whether IP Performance Manager analyzes subinterfaces, which are interfaces LayeredOver physical interfaces. For example, a physical interface in a Frame Relay or ATM network is configured with multiple virtual circuits, and each virtual circuit is associated with a subinterface.

Note: A subinterface, similar to an interface, is an instance of the Interface class.

Table 7-16 lists the Ethernet Interface/Port Performance threshold setting parameters.

Table 7-16 Default values for the Ethernet Interface/Port Performance threshold setting

Parameter	Default Value	Description
AnalysisModeOfSubInterface-Performance	ENABLED, DISABLED Default: DISABLED	When enabled, IP Performance Manager analyzes the performance of all subinterfaces of all objects in the threshold group.
BroadcastThreshold	15%	Sets upper threshold for broadcast traffic expressed as a percentage of the total bandwidth.
CollisionThreshold	10%	Sets the upper threshold for collisions expressed as a percentage of the total number of output packets.
DiscardThreshold	5%	Sets the upper threshold for dropped packets expressed as a percentage of the total number of packets.
ErrorThreshold	10%	Sets the upper threshold for packet errors expressed as a percentage of the total number of packets.

Table 7-16 Default values for the Ethernet Interface/Port Performance threshold setting (continued)

Parameter	Default Value	Description
ErrorTrafficThreshold	2	Sets the minimum packet rate for a network adapter expressed as a percentage of total bandwidth. Value can be any number between 0.1 and 100.0. For example, 0.2 is a valid value.
QueueDropThreshold	1%	Indicates the acceptable percentage of packets dropped because of full queues, expressed as a percentage of the total number of packets.
UtilizationThreshold	40%	Sets the upper threshold for link utilization expressed as a percentage of the total bandwidth.

Generic Interface/Port Performance threshold setting

The Generic Interface/Port Performance threshold setting configures the monitoring of a non-Ethernet network adapter's performance characteristics. It includes basic parameters common to all media types such as utilization, errors, broadcast, and packet drops. (These basic parameters are also included in the Ethernet Interface/Port Performance setting, however their default values may be different.) Parameters in this setting control the notification of these faults:

- High Broadcast Rate
- High Discard Rate
- High Error Rate
- High Queue Drop Rate
- High Utilization

Similar to the Ethernet Interface/Port Performance threshold setting, one other parameter in this setting determines whether IP Performance Manager analyzes subinterfaces, which are interfaces LayeredOver physical interfaces.

Table 7-17 list the Generic Interface/Port Performance threshold setting parameters.

Table 7-17 Default values for the Generic Interface/Port Performance threshold setting

Parameter	Default value	Description
AnalysisModeOfSubInterface-Performance	ENABLED, DISABLED Default: DISABLED	When enabled, IP Performance Manager analyzes the performance of all subinterfaces of all objects in the threshold group.
BroadcastThreshold	15%	Sets the upper threshold for broadcast traffic expressed as a percentage of the total bandwidth.
DiscardThreshold	5%	Sets the upper threshold for dropped packets expressed as a percentage of the total number of packets.
ErrorThreshold	10%	Sets the upper threshold for packet errors expressed as a percentage of the total number of packets.

Table 7-17 **Default values for the Generic Interface/Port Performance threshold setting (continued)**

Parameter	Default value	Description
ErrorTrafficThreshold	2	Sets the minimum packet rate for a network adapter expressed as a percentage of total bandwidth. Value can be any number between 0.1 and 100.0. For example, 0.2 is a valid value.
QueueDropThreshold	1%	Sets the acceptable percentage of packets dropped because of full queues, expressed as a percentage of the total number of packets.
UtilizationThreshold	60%	Sets the upper threshold for link utilization expressed as a percentage of the total bandwidth.

Environment threshold setting

The Environment threshold setting configures the monitoring of a system's environmental conditions. System components such as the power supply, fan, voltage sensor, and temperature sensor objects are monitored.

Table 7-18 lists the Environment threshold setting parameters.

Table 7-18 **Default values for the Environment threshold setting**

Parameter	Default Value	Description
RelativeTemperature-Threshold	0 to 100% Default: 10%	Indicates how close the current temperature value can be to the value that triggers an emergency shutdown, expressed as a percentage of the emergency shutdown value. For example, if the shutdown temperature is 50 degrees C and the RelativeTemperatureThreshold is 10%, the OutOfRange event is notified if the temperature exceeds 45 degrees C.
RelativeVoltage-Threshold	0 to 5% Default: 0%	Indicates how close the current voltage value can be to the value that triggers an emergency shutdown, expressed as a percentage of the emergency shutdown value. For example, if the shutdown value is +30V, and the RelativeVoltageThreshold is 10%, the OutOfRange event is notified if the voltage exceeds +27V.

Juniper Environment threshold setting

The Juniper Environment threshold setting configures the monitoring of a Juniper system's environmental conditions. Juniper system components such as the power supply, fan, voltage sensor, and temperature sensor objects are monitored.

Relative to the Environment threshold setting, this setting provides an additional parameter for monitoring the environmental conditions of a Juniper system for temperature sensor objects.

Table 7-19 lists the Juniper Environment threshold setting parameters.

Table 7-19 Default values for the Juniper Environment threshold setting

Parameter	Default Value	Description
HighTemperature-Threshold	0 to 100 degrees C Default: 40 degrees C	Indicates the high temperature threshold in degrees C. This value, in conjunction with the RelativeTemperatureThreshold value determines the temperature that triggers an emergency shutdown. For example, if the HighTemperatureThreshold is 50 degrees C and the RelativeTemperatureThreshold is 10%, the OutOfRange event is notified if the temperature exceeds 45 degrees C.
RelativeTemperature-Threshold	0 to 100% Default: 10%	Indicates how close the current temperature value can be to the value that triggers an emergency shutdown, expressed as a percentage of the emergency shutdown value. For example, if the shutdown temperature is 50 degrees C and the RelativeTemperatureThreshold is 10%, the OutOfRange event is notified if the temperature exceeds 45 degrees C.
RelativeVoltage-Threshold	0 to 5% Default: 0%	Indicates how close the current voltage value can be to the value that triggers an emergency shutdown, expressed as a percentage of the emergency shutdown value. For example, if the shutdown value is +30V, and the RelativeVoltageThreshold is 2%, the OutOfRange event is notified if the voltage exceeds +29.4V.

FileSystem threshold setting

The FileSystem threshold setting configures the performance monitoring of a host system's filesystem. Although available to all threshold groups in System Resource Groups and System VLAN Tag Groups, this setting is contained only in the Other Systems threshold group in System Resource Groups.

Table 7-20 lists the FileSystem threshold setting parameters. The MaxUtilizationPct parameter controls the notification of the HighUtilization fault.

Table 7-20 Default value for the FileSystem threshold setting

Parameter	Default value	Description
MaxUtilizationPct	95%	Sets the upper threshold for a filesystem's utilization expressed as a percentage of the total size currently in use.

Processor and Memory threshold setting

The Processor and Memory threshold setting configures the performance monitoring of a system's processor and its associated memory objects. Parameters in this setting control the notification of these faults:

- High Utilization
- High Backplane Utilization (chassis)
- High Buffer Miss Rate
- High Buffer Utilization
- Fragmentation
- Insufficient Free Memory

Table 7-21 lists the Processor and Memory threshold setting parameters.

Table 7-21 Default values for the Processor and Memory threshold setting

Parameter	Default value	Description
BackplaneUtilizationThreshold	80%	Sets the upper threshold for a switch's backplane utilization expressed as a percentage of the total backplane bandwidth.
FreeMemoryThreshold	15%	Sets the lower threshold for the acceptable amount of free memory as measured by the ratio of free memory to the total memory.
MemoryBufferMissThreshold	10%	Sets the upper threshold for the number of buffer misses expressed as a percentage of the total number of buffer requests.
MemoryBufferUtilization-Threshold	90%	Sets the upper threshold for the number of buffers used expressed as a percentage of the total number of buffers.
MemoryFragmentation-Threshold	5%	Sets the lower threshold for memory fragmentation. The fragmentation value is the ratio of the largest number of contiguous unallocated bytes to the total amount of free memory. For example, a value of 5 indicates that the largest free buffer must be at least 5% of the free memory.
ProcessorUtilizationThreshold	90%	Sets the upper threshold for processor utilization expressed as a percentage of the total capacity of the processor.

Using a policy setting to control the managed state of interfaces

The Interface Management Policy is a system-level setting for controlling the managed state of interfaces. This setting is available to all System Resource Groups and all System VLAN Tag Groups. By default, it is not contained in any of these groups.

If you do not use this setting, all discovered network objects, including interfaces, are automatically managed. (One possible exception is access ports. An access port is only managed when the system to which it is connected is managed.) Interfaces exist on hosts, routers, and Layer 3 switches.

Note: If you are using one of the EMC Smarts IP adapters described in the *EMC Smarts IP Adapters User's Guide*, you must allow the third-party source to control the managed status of network objects. When the adapter's topology reader synchronizes the topology of the IP Manager with the topology of the third-party source, the topology reader must be able to overwrite the managed status of objects in the IP Manager's topology.

You should apply the Interface Management Policy setting to a system resource group or a system VLAN tag group (and set the DefaultManagementStatus parameter to UNMANAGED) if you do not want to manage the interfaces that are part of the system or systems that are members of the group.

Interface Management Policy setting

Table 7-22 lists the Interface Management Policy setting parameters.

Table 7-22 Default values for the Interface Management Policy setting

Parameter	Default value	Description
DefaultManagementStatus	MANAGED	Indicates the default management status of interfaces on systems to which this setting applies. The value can be MANAGED or UNMANAGED. Discovery is required for any change to take effect.

Note: If you set an interface management policy, the policy can be overridden for a particular interface using the manage/unmanage commands available through the Domain Manager Administration Console. Chapter 11, "Working with the Topology," in the *EMC Smarts IP Management Suite Discovery Guide* provides information about managing and unmanaging specific objects in the topology.

To use the Interface Management Policy setting, you need to add the setting to a system resource group or a system VLAN tag group. Accessing the Polling and Thresholds Console and adding a setting to a group are described in "Opening the Polling and Thresholds Console" on page 5-2 and "Working with groups and settings" on page 5-7.

When an interface becomes a member of a system resource group or a system VLAN tag group to which the Interface Management Policy setting is applied, the IP Manager sets the IsManaged attribute of the interface in accordance to the DefaultManagementStatus parameter value:

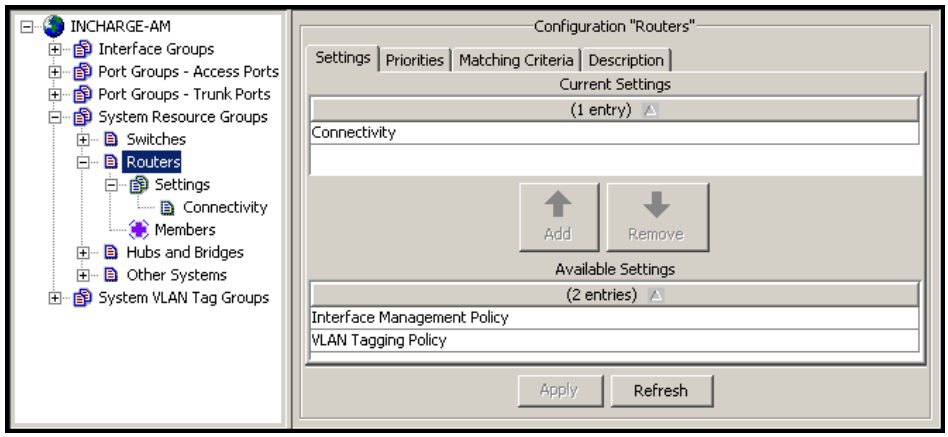
- DefaultManagementStatus = UNMANAGED sets IsManaged to FALSE.
- DefaultManagementStatus = MANAGED sets IsManaged to TRUE.

Similarly, when a system becomes a member of a system resource group or a system VLAN tag group to which the Interface Management Policy setting is applied, the IP Manager sets the IsManaged attribute of the interfaces of the system in accordance to the DefaultManagementStatus parameter value.

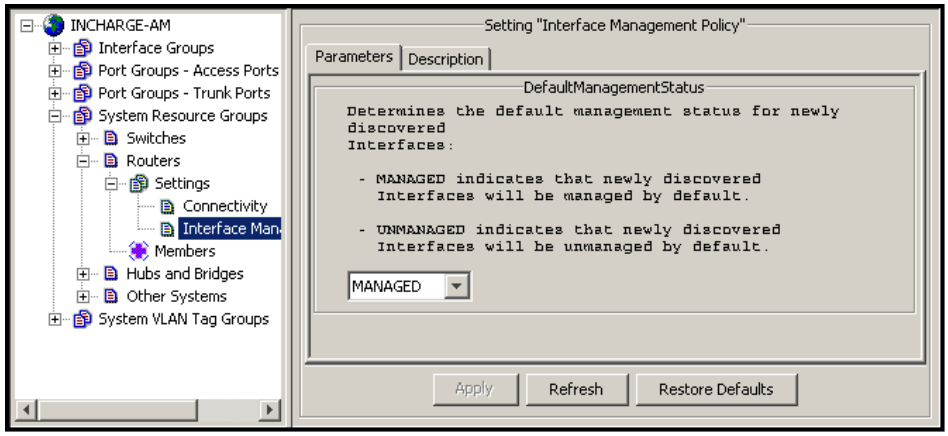
Example of applying the Interface Management Policy setting

The following procedure demonstrates how to use the Polling and Thresholds Console to apply the Interface Management Policy setting to the default system resource group named Routers.

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager or IP Performance Manager, click the **Thresholds** tab to display the thresholds group tree.
2. In the thresholds group tree, click the **+** signs to expand the thresholds group tree as shown in the following display, then click **Routers** to display the configuration window and the **Settings** tab.



3. In the **Available Settings** list of the **Settings** tab, select **Interface Management Policy** and then click **Add** to move the **Interface Management Policy** setting to the **Current Settings** list.
4. Click the **Apply** button to save your changes.
5. In the thresholds group tree, click **Interface Management Policy** to display its setting configuration window.



6. In the setting configuration window, click the drop-down menu and select **MANAGED** or **UNMANAGED** from the menu; **MANAGED** is the default.

7. Click the **Apply** button to save your changes.
8. Optional: In the thresholds group tree, click **Routers** to display the configuration window.
9. Optional: In the configuration window, click **Matching Criteria** to display the **Matching Criteria** tab.
10. Optional: In the **Matching Criteria** tab, specify additional matching criteria for the **Routers** group and click the **Apply** button to save your changes. (The procedure for editing matching criteria is given in "Editing matching criteria" on page 5-9.)

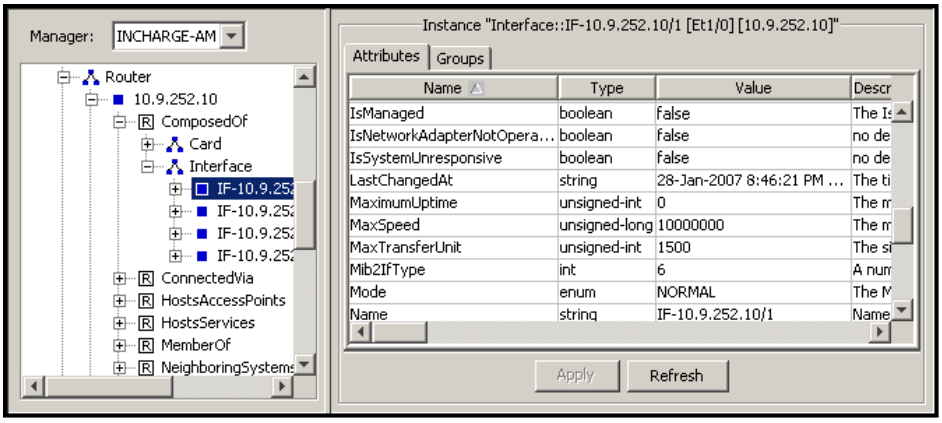
    The IsManaged attribute value of the interfaces of any router that becomes a member of the **Routers** group will automatically be set to the value that you selected in Step 6:

    - DefaultManagementStatus = UNMANAGED will set IsManaged to FALSE.
    - DefaultManagementStatus = MANAGED will set IsManaged to TRUE.
11. Click the **Reconfigure** toolbar button to make your changes take effect.

Verifying interface management policy configuration

To verify that the interface management policy of your Routers group is configured properly:

1. Attach the Domain Manager Administration Console to the target IP Manager. ("Procedure for opening the Polling and Thresholds Console" on page 5-2 provides instructions on opening the Domain Manager Administration Console.)
2. In the console, select **Topology > Discover All** to rediscover all managed systems.
3. In the topology tree of the console, expand the **Router** class to see the router instances that the IP Manager just rediscovered.
4. Choose a router instance that is configured to be a member of the **Routers** group, expand its **ComposedOf** relationship, expand the **Interface** class, and select an interface instance to display its **Attributes** tab in the right panel.



5. In the **Attributes** tab, verify that the value of the IsManaged attribute is set to the expected value.
6. Repeat Steps 4 and 5 for each interface instance of each router instance that is configured to be a member of the **Routers** group.

Using a policy setting to ensure appropriate VLAN assignments

The VLAN Tagging Policy is a system-level setting for ensuring that systems are assigned to their appropriate VLAN. By default, this setting is contained in the System VLAN Tag Group named Switches and available in the System VLAN Tag Group named Other Systems and in all System Resource Groups.

A networked environment may consist of many separate networks, each having many VLANs. One VLAN may have the same name (VLAN ID) as a VLAN in another network. Identically named VLANs do not create problems across different networks; however, when these VLANs are discovered and placed into a single topology, conflicts arise.

If you do not use the VLAN Tagging Policy setting, the IP Manager will combine any discovered VLANs having identical names into a single VLAN of that name; for example, VLAN-1. And systems belonging to the identically named but locally distinct VLANs will be represented in the modeled topology as belonging to VLAN-1. Because of this incorrect representation in the modeled topology, the IP Manager root-cause and impact analysis will produce inaccurate results.

If you do use the VLAN Tagging Policy setting, the IP Manager will distinguish any discovered distinct VLANs having identical names by adding a unique tag to the name of each of the distinct VLANs. And systems belonging to identically named but distinct VLANs will be represented in the topology as belonging to distinct VLANs. Because of this correct representation in the modeled topology, the IP Manager root-cause and impact analysis will produce accurate results.

VLAN tagging scenario

Consider the following scenario:

- In the managed network environment, two switches (SW1, SW2) in one VLAN and two switches (SW2, SW3) in another VLAN are members of two locally distinct VLANs having the same name, VLAN-1.
- Without creating a VLAN tag group for at least one of the VLANs, the IP Manager will combine the VLANs into one VLAN, named VLAN-1, that includes all four switches.
- By creating a VLAN tag group for at least one of the VLANs, the IP Manager will assign SW1 and SW2 to one VLAN, and assign SW2 and SW3 to another VLAN.
- For example, if you create a VLAN tag group having a tag named "Marketing" and having matching criteria that allow SW1 and SW2 to become members of that group, the IP Manager will assign SW1 and SW2 to a VLAN named VLAN-Marketing-1, and assign SW2 and SW3 to a VLAN named VLAN-1.

VLAN tagging naming scheme

Develop a naming scheme for the tags and decide which systems to associated with which tags. Tagged VLANs appear in the topology as:

```
VLAN-<tag>-<VLAN number>
```

VLAN Tagging Policy setting

Table 7-23 lists the VLAN Tagging Policy setting parameters.

Table 7-23 Default values for the VLAN Tagging Policy setting

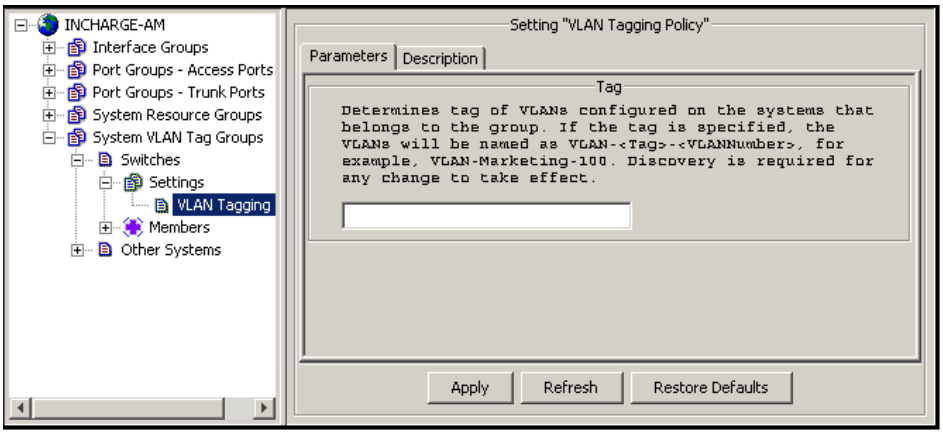
Parameter	Default Value	Description
Tag	Original configuration (no tag)	Determines the tag for the VLANs configured on the systems that belong to a particular system VLAN tag group. If the tag is specified, the VLANs will be named as VLAN-<tag>-<VLAN number>. For example, if you specify a tag, "Marketing," for an originally configured VLAN-1, it appears as VLAN-Marketing-1. Discovery is required for any change to take effect.

To use the VLAN Tagging Policy setting to create a system VLAN tag group, you need to assign a tag to the setting's Tag parameter. Accessing the Polling and Thresholds Console and modifying the parameters in a setting are described in "Opening the Polling and Thresholds Console" on page 5-2 and "Working with groups and settings" on page 5-7.

Example of creating a VLAN tag group

The following procedure demonstrates how to use the Polling and Thresholds Console to create a VLAN tag group for switches by assigning a tag and adding additional matching criteria to the default System VLAN Tag Group named Switches.

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager or IP Performance Manager, click the **Thresholds** tab to display the thresholds group tree.
2. In the thresholds group tree, click the **+** signs to expand the thresholds group tree as shown in the following display, then click **VLAN Tagging Policy** to display its setting configuration window.



3. In the setting configuration window, type a tag value that is meaningful for your configuration and press **Enter**.

Note: For the **Apply** button to be available, you must press the **Enter** key (on your keyboard) after typing or changing the tag value.

4. Click the **Apply** button to save your changes.
5. In the thresholds group tree, click **Switches** to display the configuration window.
6. In the configuration window, click **Matching Criteria** to display the **Matching Criteria** tab.
7. In the **Matching Criteria** tab, specify the additional matching criteria that will allow only the switches belonging to the target VLAN to become members of this group. (The procedure for specifying matching criteria is given in "Editing matching criteria" on page 5-9.)

   Note: Specify matching criteria broad enough so that *all* switches belonging to the target VLAN become members of this group. Otherwise, the IP Manager will create multiple names (multiple VLAN instances) for the target VLAN, which will produce incorrect correlation analysis results.

   For example, if a distinguishing feature for the switches belonging to the target VLAN is that their names all contain the string "sw," set the **Name** attribute to ***sw*** to ensure that only those switches become members of the **Switches** group.

   For any switch that becomes a member of this group, the name of any VLAN to which the switch belongs will be tagged with the value that you specified in Step 3.
8. Click the **Apply** button to save your changes.
9. Click the **Reconfigure** toolbar button to make your changes take effect.

Create a system VLAN tag filter for each group of systems belonging to identically named but locally distinct VLANs.

To create additional system VLAN tag groups for switches, create new groups from the Switches group configured in the preceding procedure and repeat that procedure for the newly created groups. (The procedure for creating new groups is given in "Creating new groups" on page 5-10.) To create system VLAN tag groups for routers, hubs, or bridges, start with the default system VLAN tag group named Other Systems.

Verifying VLAN tag group configurations

To verify that your system VLAN tag groups are configured properly:

1. Attach the Domain Manager Administration Console to the target IP Manager. ("Procedure for opening the Polling and Thresholds Console" on page 5-2 provides instructions on opening the Domain Manager Administration Console.)
2. In the console, select **Topology > Discover All** to rediscover all managed systems.
3. In the topology tree of the console, expand the **VLAN** class to see the tagged VLAN instances (and their member systems) that the IP Manager just created.
4. Verify that the rediscovered switches are members of the appropriate VLANs.

Using a policy setting to prevent VLAN impacts

The "Problem Impact propagation policy over VLANs" is a VLAN-level setting for preventing the generation of impacts of failed devices in a VLAN on other devices in the VLAN. By default, this setting is contained in the VLAN Impact Propagation Group named VLANs and available in the VLAN Impact Propagation Group named Other VLANs.

For two failed devices that are logically connected through a common VLAN, the IP Manager will not only create problem notifications for the failed devices, but will also create impact notifications claiming that each failed device is responsible for the failure of the other failed device. While, in general, this analysis is correct, customers might want to prevent this type of impact propagation in certain VLANs, such as in management VLANs or in default VLANs (for example, in the default VLAN-1 for Cisco switches and routers). The "Problem Impact propagation policy over VLANs" setting allows customers to accomplish this end.

VLAN impact propagation scenario

Figure 7-3 shows three VLANs spanning three subnets.

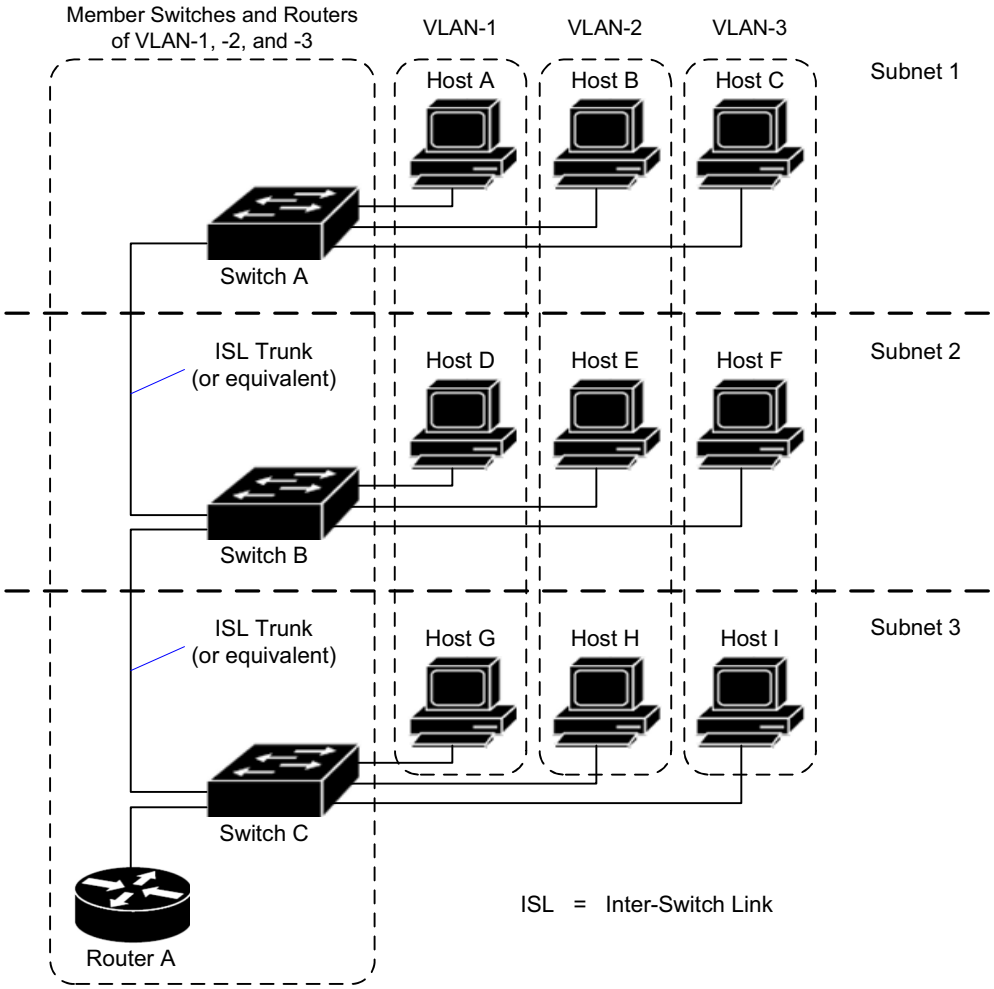


Figure 7-3 VLANs spanning multiple switches and multiple subnets—example

In the figure, Switches A, B, and C provide three separate bridge groups, one for each VLAN, and Router A provides routing for inter-VLAN communications.

If Switch A fails, the IP Manager creates (and the Global Manager imports) the following notifications:

- One Switch A Down (problem) notification
- One Switch A Unresponsive (event) notification
- One TrunkCable DownOrFlapping (event) notification
- Three Cable DownOrFlapping (event) notifications—assuming Hosts A, B, and C are "managed"

The IP Manager uses the events as symptoms to diagnose the Switch A Down problem. The events will appear in the Impact tab of the Notification Properties dialog box for the Switch A Down problem.

Note: Events used as symptoms to diagnose a failure can also be perceived as impacts resulting from the failure.

If Switch B also fails, the IP Manager proceeds as follows:

- Clears its original Switch A Down notification, creates a new Switch A Down notification, and creates a Switch B Unresponsive notification; the Impact tab of the Notification Properties dialog box for the new Switch A Down problem will list an additional impact: Switch B Unresponsive.
- Creates a Switch B Down notification, a Switch B Unresponsive notification, and five TrunkCable/Cable DownOrFlapping notifications; the Impact tab of the Notification Properties dialog box for the Switch B Down problem will list a Switch B Unresponsive impact, five TrunkCable/Cable DownOrFlapping impacts, and a Switch A Unresponsive impact.

The Switch A and B Down notifications claim that each switch is responsible for the associated failure of the other. In this example, the switches have an impact on each other because they are members of the same set of VLANs: VLAN-1, VLAN-2, and VLAN-3.

Creating a VLAN impact propagation group that allows VLAN-1, VLAN-2, and VLAN-3 to become members of the group will prevent the linking of the Switch B Unresponsive impact to the Switch A Down problem, and will prevent the linking of the Switch A Unresponsive impact to the Switch B Down problem.

VLAN impact propagation group matching criteria

The matching criteria for a VLAN impact propagation group are the VLAN class and two of its attributes:

- CreationClassName
- VLANKey

By default, the value of the CreationClassName attribute is set to "VLAN," and the value of the VLANKey attribute is undefined. The value that you specify for the VLANKey attribute determines which VLANs become members of a VLAN impact propagation group.

The value of the VLANKey attribute is included in the value of the Name attribute for a VLAN object in the following way:

```
VLAN-<VLANKey>
```

The value of the VLANKey attribute distinguishes a VLAN from all other VLANs in the managed topology and consists of any combination of the following identifiers:

```
<management domain name> <user-defined VLAN tag> <VLAN number (ID)>
```

Note: The range of the VLAN number depends on the bridging protocol. For the 802.1Q protocol, the range is 1 to 4096.

An example of a VLANKey value consisting of just a management domain name is:

```
VLAN #66
```

An example of a VLANKey value consisting of a user-specified VLAN tag and a VLAN number is:

```
Marketing-1
```

An example of a VLANKey value consisting of just a VLAN number is:

```
1
```

Problem Impact propagation policy over VLANs setting

Table 7-24 lists the "Problem Impact propagation policy over VLANs" setting parameters.

Table 7-24 Default values for the Problem Impact propagation policy over VLANs setting

Parameter	Default Value	Description
ImpactPropagation	ENABLED	Determines whether to allow problems on devices in a VLAN to propagate impacts on other devices in the same VLAN. The value can be ENABLED or DISABLED. • ENABLED allows impact propagation for the VLANs belonging to the VLAN impact propagation group • DISABLED prevents impact propagation for the VLANs belonging to the VLAN impact propagation group **Note:** If a VLAN is not a member of any VLAN impact propagation group, impact propagation is allowed for the VLAN. Discovery is required for any change to take effect.

To use the "Problem Impact propagation policy over VLANs" setting to create a VLAN impact propagation group, you need to change the setting's ImpactPropagation parameter to DISABLED. Accessing the Polling and Thresholds Console and modifying the parameters in a setting are described in "Opening the Polling and Thresholds Console" on page 5-2 and "Working with groups and settings" on page 5-7.

When a VLAN becomes a member of a VLAN impact propagation group having ImpactPropagation = DISABLED, the IP Manager sets the following attribute of the VLAN to FALSE:

```
isImpactPropOverVLANEnabled
```

For a VLAN having isImpactPropOverVLANEnabled = FALSE, impact propagation for the VLAN is prevented.

For a VLAN having isImpactPropOverVLANEnabled = TRUE (the default), impact propagation for the VLAN is allowed.

Creating VLAN impact propagation group objects

Currently, the underlying software does not automatically create the default group objects of the "Problem Impact propagation policy over VLANs" setting, meaning that the group category "VLAN Impact Propagation Groups" does not automatically appear on the Thresholds tab of a Polling and Thresholds Console attached to the IP Manager. Instead, a user must issue an sm_adapter command to manually create the VLAN impact propagation group objects. Once created, the group objects are available every time that the IP Manager is restarted.

To create the default group objects of the "Problem Impact propagation policy over VLANs" setting:

1. Go to the *BASEDIR/smarts/bin* directory in the IP Management Suite installation area and issue the following command on one line:

   ```
   sm_adapter -s <IP Manager instance name> [-b <broker>] -f
      BASEDIR/IP/smarts/conf/discovery/settings.conf
      icf/init-settings.asl
   ```

   Where:

Option	Description
-b <broker>	Alternate Broker host and port (<host:port>) location for the target IP Manager instance
-f <path>	Full path to the settings.conf file
icf/init-settings.asl	Script that parses the settings.conf file

   For example, enter the following command on one line:

   ```
   sm_adapter -s INCHARGE-AM -f
      c:\InCharge7\IP\smarts\conf\discovery\settings.conf
      icf\init-settings.asl
   ```

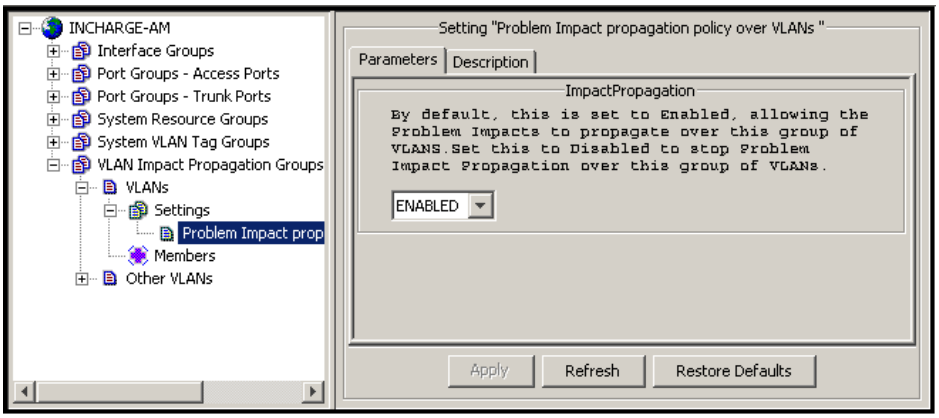
2. When prompted for a username and password, specify an administration account; for example, *admin* and *changeme*.

Example of creating a VLAN impact propagation group

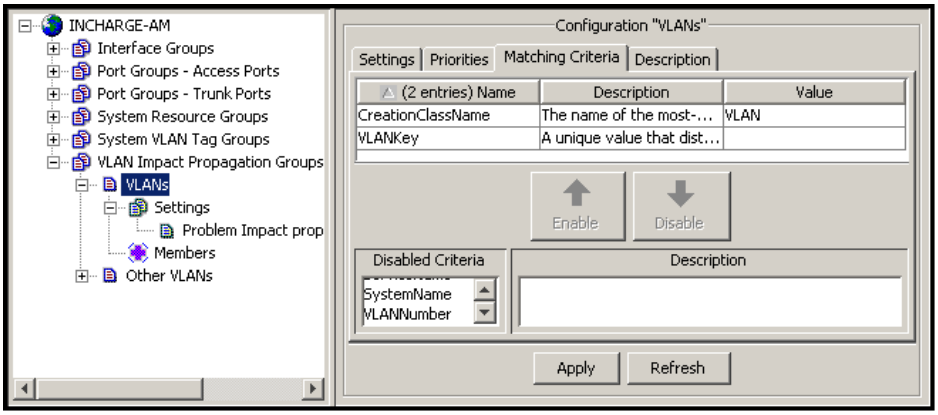
The following procedure demonstrates how to use the Polling and Thresholds Console to create a VLAN impact propagation group by disabling impact propagation and adding matching criteria to the default VLAN Impact Propagation Group named VLANs.

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager or IP Performance Manager, click the **Thresholds** tab to display the thresholds group tree.

2. In the thresholds group tree, click the + signs to expand the thresholds group tree as shown in the following display, then click **Problem Impact propagation policy over VLANs** to display its setting configuration window.



3. In the setting configuration window, click the drop-down menu and select **DISABLED** from the menu; **ENABLED** is the default.
4. Click the **Apply** button to save your changes.
5. In the thresholds group tree, click **VLANs** to display the configuration window.
6. In the configuration window, click **Matching Criteria** to display the **Matching Criteria** tab.
7. In the **Disabled Criteria** list of the **Matching Criteria** tab, select **VLANKey** and then click **Enable** to move the **VLANKey** attribute to the top of the **Matching Criteria** tab.



8. Double-click the value field of the **VLANKey** attribute and type the matching criteria that will allow only the target VLANs to become members of this group. (The procedure for specifying matching criteria is given in "Editing matching criteria" on page 5-9.)

Note: Specify matching criteria broad enough so that *all* VLANs supported by a particular group of switches and routers become members of this group. Otherwise, the IP Manager will continue to propagate impacts over the VLAN or VLANs that do not become a member of this group.

For example, if the VLANs supported by a particular group of switches have **VLANKey** values containing the string "1," "2," or "3" (as in the example given in Figure 7-3 on page 7-28), set the **VLANKey** value to ***1|2|3** to ensure that all of those VLANs become members of the **VLANs** group.

The isImpactPropOverVLANEnabled attribute value of any VLAN that becomes a member of the **VLANs** group will automatically be set to the value that you selected in Step 3:

- ImpactPropagation = DISABLED will set isImpactPropOverVLANEnabled to FALSE.
- ImpactPropagation = ENABLED will set isImpactPropOverVLANEnabled to TRUE.

9. Optional: In the **Matching Criteria** tab, specify additional matching criteria for the **VLANs** group.
10. Click the **Apply** button to save your changes.
11. Click the **Reconfigure** toolbar button to make your changes take effect.

Create a VLAN propagation impact group for each group of VLANs for which you want to disable impact propagation.

To create additional VLAN impact propagation groups, create new groups from the VLANs group configured in the preceding procedure and repeat that procedure for the newly created groups. (The procedure for creating new groups is given in "Creating new groups" on page 5-10.) To create VLAN impact propagation groups for VLANs having a CreationClassName attribute value other than "VLAN," start with the default VLAN impact propagation group named Other VLANs.

Verifying VLAN impact propagation group configurations

To verify that your VLAN impact propagation groups are configured properly:

1. Attach the Domain Manager Administration Console to the target IP Manager. ("Procedure for opening the Polling and Thresholds Console" on page 5-2 provides instructions on opening the Domain Manager Administration Console.)
2. In the console, select **Topology > Discover All** to rediscover all managed systems.
3. In the topology tree of the console, expand the **VLAN** class to see the VLAN instances that the IP Manager just rediscovered.
4. Choose a VLAN instance that is configured to be a member of a VLAN impact propagation group and click its name to display its **Attributes** tab in the right panel.
5. In the **Attributes** tab, verify that the value of the isImpactPropOverVLANEnabled attribute is set to FALSE.
6. Repeat Steps 4 and 5 for each VLAN instance that is configured to be a member of a VLAN impact propagation group.

8

Creating Credential Sets

This chapter describes the default device access groups and settings for IP Availability Manager and IP Performance Manager. It consists of the following sections:

- Viewing the Device Access tab 8-2
- Understanding device access groups and settings 8-3
- Default device access groups and settings 8-4
- Overview of creating a device access group 8-5

Viewing the Device Access tab

Figure 8-1 shows the Device Access tab of a Polling and Thresholds Console connected to an IP Availability Manager instance named INCHARGE-AM. The right panel of the console identifies "CLI Access Setting" as the default setting for the default polling group named "CLI Access Group."

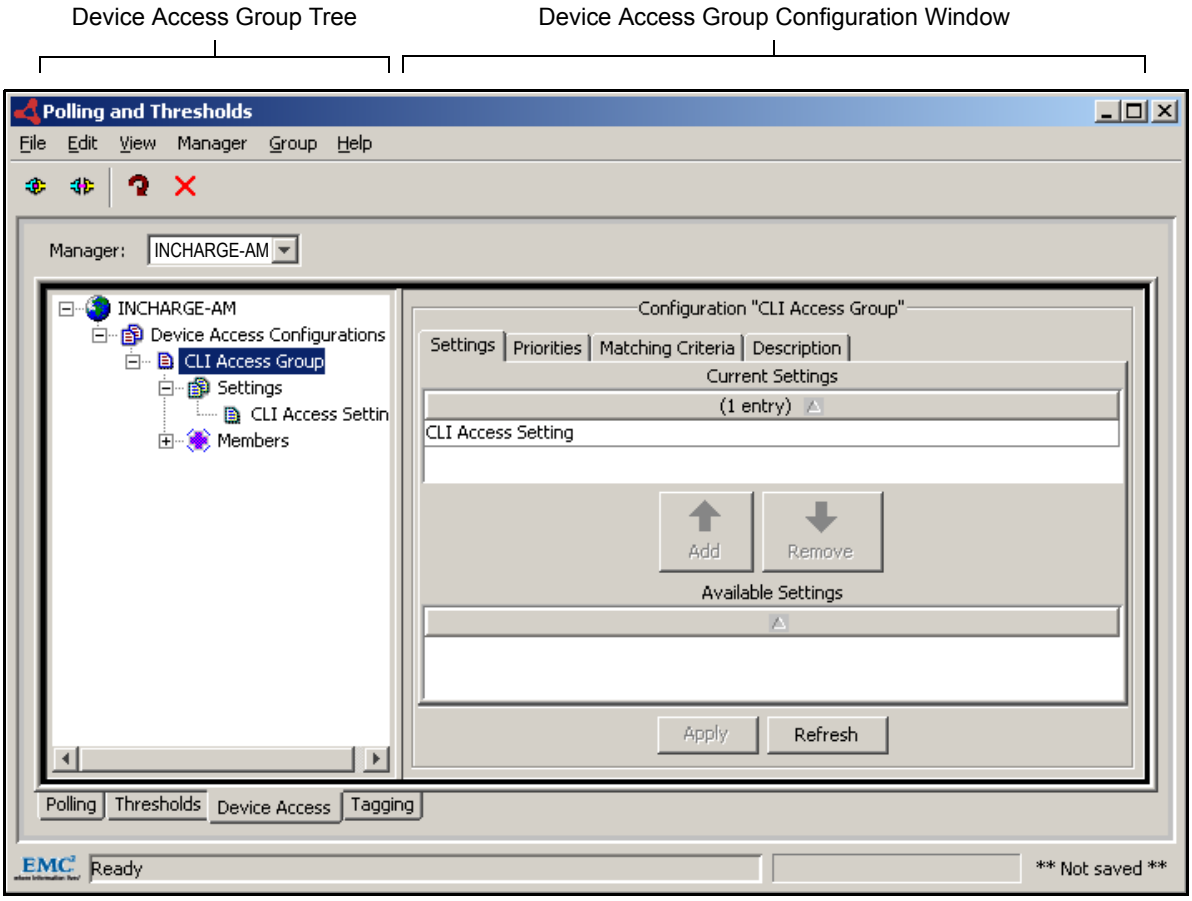


Figure 8-1 Device Access tab for IP Availability Manager

The Device Access tab of the Polling and Thresholds Console connected to an IP Performance Manager, or connected to IP Availability Manager and IP Performance Manager running as a single process, is identical to what is shown in Figure 8-1. The device access group and setting configurations are common to all IP Managers started from the same IP Management Suite installation area.

Understanding device access groups and settings

Because IP Availability Manager is a component of many EMC Smarts deployments, IP Availability Manager has been selected as the central repository for specifying and storing the device access credentials for other Domain Managers—such as Command Line Interface (CLI) login credentials required by Network Protocol Manager for EIGRP and Network Protocol Manager for IS-IS to perform CLI discovery and, optionally, CLI polling. The satellite Domain Managers import the device access credentials as well as initial topology from IP Availability Manager when they add IP Availability Manager as a source.

The device access groups and settings provide a means to create different sets of device access credentials. Currently, the Device Access tab provides only one default group and one default setting, used to create CLI login credential sets. In the future, the Device Access tab will contain additional default groups and settings to be used to create other types of device access credential sets.

Modifying device access values and creating device access groups

Procedures for modifying existing device access groups and creating new ones are given in "Working with groups and settings" on page 5-7.

Default device access groups and settings

Through the default device access groups, you create sets of credentials (such as login credentials) needed to access systems in the managed network environment.

Default device access groups

The IP Manager provides one default device access group named CLI Access Group.

Table 8-1 lists the matching criteria and the default setting for CLI Access Group. The matching-criteria target class for CLI Access Group is UnitaryComputerSystem.

Table 8-1 Default device access groups

Device access group	Matching criteria	Default settings
CLI Access Group	UnitaryComputerSystem attributes: • Name = *	CLI Access Setting

Default device access settings

The IP Manager provides a single default device access setting named CLI Access Setting. Table 8-2 lists the CLI Access Setting parameters.

Table 8-2 Default values for the CLI Access Setting

Parameter	Default Value	Description
AccessProtocol	TELNET SSH1 SSH2 Default: TELNET	Determines the remote-access application to use to establish connections to the managed routing devices that belong to this group. Note that selecting SSH1 or SSH2 requires configuring an SSH client on the Network Protocol Manager host.
LoginID	String of unspecified length Default: null string (empty)	Specifies the username (user ID) for the managed routing devices that belong to this group. There must be an entry for this parameter. If no username is configured on the devices that belong to this group, enter any value; for example, cisco.
Password	String of unspecified length Default: null string (empty)	Specifies the password for the managed Cisco devices that belong to this group. Enter the password value twice to confirm the password value. There must be an entry for this parameter unless the access protocol for this group is SSH1 or SSH2 and passwordless authentication is in effect. Note that IP Availability Manager uses the site key to encrypt the entered password value. As explained in *EMC Smarts System Administration Guide*, the site key is created during the installation of EMC Smarts applications.
Timeout	1 to 496 seconds Default: 10 seconds	Sets the amount of time to wait for a Telnet/SSH response before the Telnet /SSH session request times out.

Overview of creating a device access group

In an environment managed by Network Protocol Manager for EIGRP and Network Protocol Manager for IS-IS, you, as the Network Protocol Manager administrator, must create a CLI access group for each group of managed Cisco routers configured with a different access protocol or a different set of CLI login credentials.

The assumption is that all routers using a particular access protocol are configured with the same set of credentials (and timeout value). If, in fact, there are two groups of routers using a particular access protocol, each configured with its own set of credentials, you would create two CLI access groups for that particular access protocol; for example, create two CLI access groups named Telnet_S1 and Telnet_S2. The matching criteria defined for each of the groups would limit the group's members to the appropriate routers.

The basic steps for creating and configuring CLI access groups are:

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager or IP Performance Manager, click the **Device Access** tab to display the device access group tree.
2. In the device access group tree, use the **CLI Access Group** to create a CLI access group for each group of routers that have a different access protocol or a different set of CLI login credentials. One way to create a new CLI access group is to right-click **CLI Access Group** and select **Copy** from the pop-up menu. (The procedure for creating groups is given in "Creating new groups" on page 5-10.)
3. Edit the parameters as appropriate in each of the CLI access groups.
4. Define matching criteria for each of the CLI access groups to limit each group's membership to the router or group of routers for which the parameters specified in Step 3 apply.

   Start by defining **Vendor** = **Cisco**. Then, define other variables and values to refine the membership; for example, **Name** = **R1***, which would limit the membership to only those routing devices having names beginning with the string R1.
5. Change the priority of the CLI access group as needed. Typically, the default **CLI Access Group** is placed at the bottom of the priority list. (The procedure for changing the priority of groups is given in "Modifying the priority of groups" on page 5-8.)

You modify the properties of a CLI access group in the same way that you modify the properties of a polling or threshold group: through the Settings, Priorities, and Matching Criteria tabs, available when the group name is selected in the left panel of the Polling and Thresholds Console. After customizing the properties of a group, click the Apply button to save the changes and then click the Reconfigure toolbar button or select Reconfigure from the Group menu to make the configuration changes take effect.

Upon adding IP Availability Manager as a source, Network Protocol Manager imports the CLI access groups and the EIGRP or IS-IS object set from IP Availability Manager. For each Cisco router in the object set, Network Protocol Manager searches through the prioritized list of CLI access groups for a router match, finds a match, and then uses the associated access protocol and credentials to log in to the Cisco router and gather EIGRP or IS-IS information from the router. If no match is found, Network Protocol Manager writes an error to its log file.

9

Creating IP Tagging Filters

This chapter describes the default IP tag filter groups and settings for IP Availability Manager and IP Performance Manager. It consists of the following sections:

- Viewing the Tagging tab 9-2
- Understanding the IP tagging feature 9-3
- Understanding tagging filter groups and settings 9-10
- Default tagging filter groups and settings 9-11
- Overview of creating an IP tag filter group 9-15
- Example of creating an IP tag filter group 9-17

Viewing the Tagging tab

Figure 9-1 shows the Tagging tab of a Polling and Thresholds Console connected to an IP Availability Manager instance named INCHARGE-AM. The right panel of the console identifies "IP Tagging Policy" as the default setting for the default tag group named "IP Tag Filter-1."

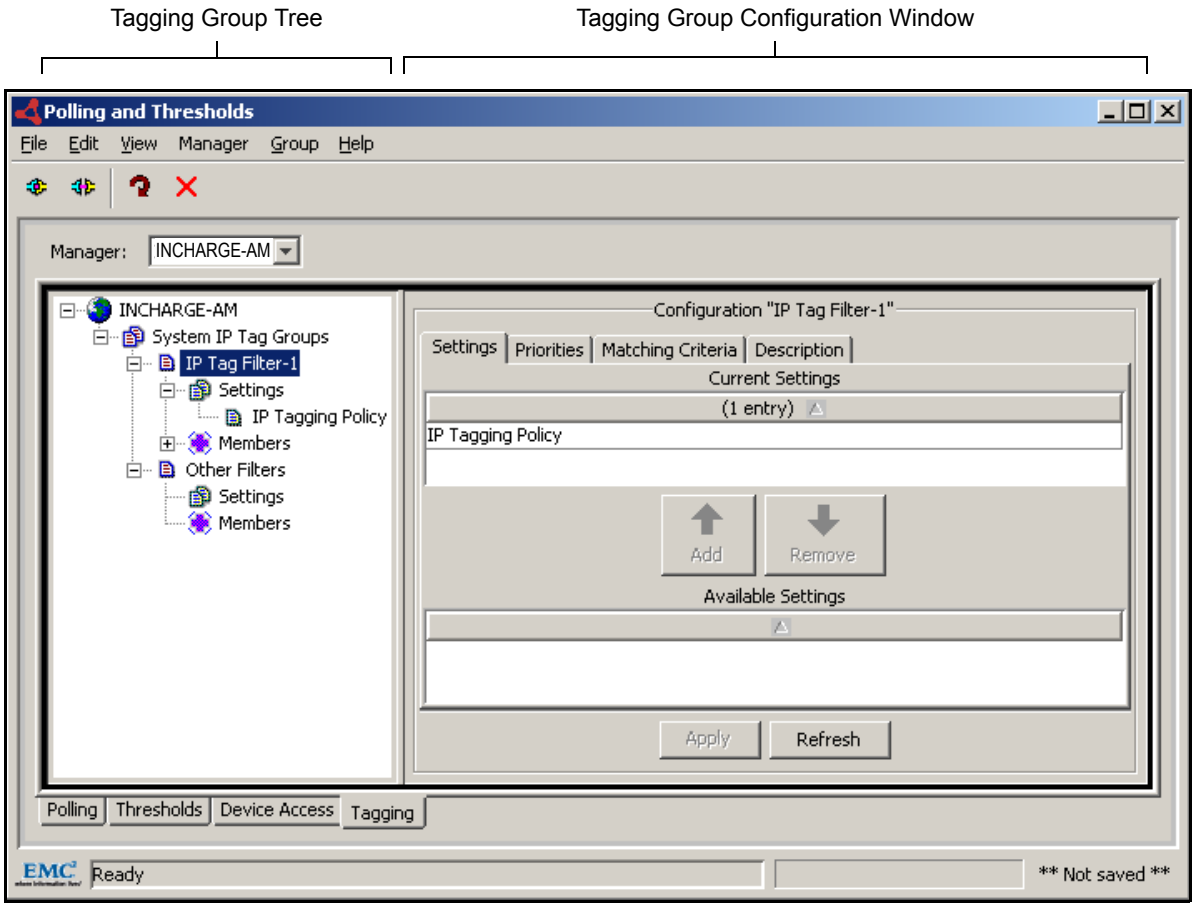


Figure 9-1 Tagging tab for IP Availability Manager

The Tagging tab of the Polling and Thresholds Console connected to an IP Performance Manager, or connected to IP Availability Manager and IP Performance Manager running as a single process, is identical to what is shown in Figure 9-1. The tag group and setting configurations are common to all IP Managers started from the same IP Management Suite installation area.

Understanding the IP tagging feature

In a networked environment, the same IP addresses can be reused across multiple systems (devices) running in separate networks. These IP addresses, known as overlapping IP addresses, are prominent in MPLS VPN environments and in networks centrally managed by Managed Service Providers, where private IP networks of customers often deploy identically numbered IP address spaces. Figure 9-2 is an example of an overlapping IP address in an MPLS VPN network.

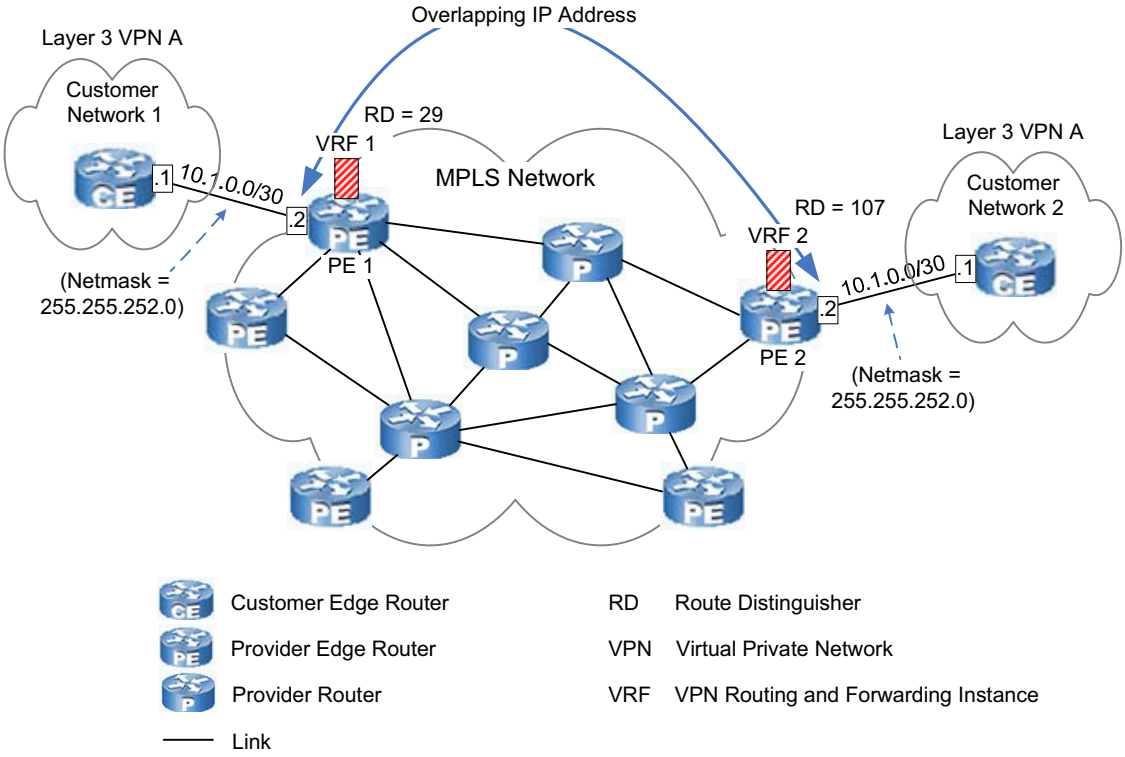


Figure 9-2 Example of an overlapping IP in an MPLS VPN network

Overlapping IP addresses do not create problems across different networks; however, when a service-assurance application for a centrally managed network attempts to discover the overlapping IP addresses, IP address identity problems arise. For example, without the IP tagging feature, the IP Manager will respond as follows when it discovers an overlapping IP address:

- Create a DuplicateIP notification and a DuplicateIP object
- Discard the most recently received overlapping address
- Remove the initially received overlapping address from its repository.

With the IP tagging feature, the IP Manager is able to distinguish between overlapping addresses belonging to different customers by adding a distinguishing tag to their names. That is, instead of using the standard naming scheme of “IP-<IP address>” to name an overlapping IP address instance, the IP Manager uses the “IP-<IP address>/<tag>” naming scheme to name the instance.

IP tagging scenario

Consider the following scenario:

- In the managed network environment, two switches (SW1, SW2) in one customer network and two switches (SW2, SW3) in another customer network are using the the same IP address space.
- Without creating an IP tag group for at least one of the customer networks, the IP Manager will discard the overlapping IP addresses collected from the four switches.
- By creating an IP tag group for at least one of the customer networks, the IP Manager will assign one name to the IP addresses collected from SW1 and SW2, and assign a different name to the IP addresses collected from SW2 and SW3.
- For example, if you create an IP tag group having a "York" tag and matching criteria that allow SW1 and SW2 to become members of the group, the IP Manager will assign the name IP-<IP address>/York to the IP addresses collected from SW1 and SW2, and assign the name IP-<IP address> to the IP addresses collected from SW2 and SW3.

With the IP tagging feature, a user creates an IP tag filter for each group of systems that is using an overlapping address space. Included in each filter is a user-specified tag value and a range of overlapping IP addresses.

Why modeling overlapping IP addresses is important

Through the IP tag filters, the IP Manager is able to create IP objects in the modeled topology that represent the overlapping IP addresses used by locally distinct groups of systems. This capability leads to better modeling of the managed network environment, which in turn leads to more accurate correlation analysis results.

An overlapping IP address is often on a client-side physical interface of a router or switch managed by a Managed Service Provider (MSP). The MSP wants to discover the router and all of its interfaces and IP addresses, but does not want to (and often cannot) discover the client equipment accessible through the overlapping IP address. The inclusion of the overlapping IP addresses in the modeled topology allows the MSP to observe the status of the client equipment associated with the overlapping IP addresses, which enables the MSP to respond immediately to customer outages.

The IP Manager monitors the status of the overlapping IP addresses by periodically sending ICMP and SNMP requests to the systems on which the overlapping IP addresses were discovered.

Overlapping IP addresses versus management IP addresses

An overlapping IP address is typically an address on an edge system's physical interface, not the address—known as a management IP address—bound to the edge system's SNMP agent. (An edge system is a system at the edge of the managed network environment.) An overlapping IP address is typically unreachable in the sense that no routes exist between the host running the IP Manager and the overlapping IP address.

Note: A management IP address is the address that the IP Manager uses to discover information about a system and to monitor the system after it is discovered.

Managed IP objects versus unmanaged IP objects

Even for a pair of overlapping IP addresses that are reachable by the IP Manager, the IP Manager will not be able to monitor them because it will not be able to determine which of the systems bound to those IP addresses is responding to the ICMP or SNMP polls. If one of the overlapping IP address is reachable and the other is not, managing the reachable one and unmanaging the unreachable one would make sense.

Note: In a future release, the IP Manager will support source routing, which will enable a service provider to use a single instance of IP Manager to monitor overlapping IP addresses by using different source routes for each.

By default, an IP tag filter sets all IP objects created for overlapping IP addresses to "unmanaged," and set all IP objects created for management IP addresses to "managed." A managed object is monitored using ICMP and SNMP—unless different access mode is specified—to determine the object's status and connectivity. An unmanaged object is not polled or rediscovered but remains present in the modeled topology.

Chapter 11, "Working with the Topology," in the *EMC Smarts IP Management Suite Discovery Guide* provides more information about managed and unmanaged objects.

Automatic and user-defined IP tagging

The IP tagging feature provides two levels of IP tagging:

- Automatic
- User-defined

By default, no IP tagging occurs. After a user edits the BASEDIR/smarts/conf/discovery/tpmgr-param.conf file and changes the DisableOverlappingIP value to FALSE, automatic IP tagging is available.

Automatic tagging

Automatic IP tagging works just like it did in previous releases of the IP Management Suite:

For every MPLS-related VRF IP address collected for a discovered system, the IP Manager creates a VPNIP object and tags its name with the route distinguisher (RD) tag value associated with the VRF to which the VRF IP address belongs. The IP Manager obtains the RD tag by reading the vendor VPN MIB on the discovered system.

The IP Manager tags the name of the VPNIP object with a route distinguisher value regardless of whether the VRF IP is—or is not—an overlapping IP address.

The IP Manager also creates a corresponding VPNIPNetwork object and tags it with the same value used to tag the VPNIP objects. (The VPNIP objects have a "ConnectedVia" relationship with the VPNIPNetwork object.)

Note: The VPNIP and VPNIPNetwork classes were introduced in IP Management Suite 6.2 as a work-around to solve the issue of duplicate IP objects present in customer networks supporting MPLS VPNs.

If the user edits the BASEDIR/smarts/conf/discovery/tpmgr-param.conf file and changes the UseVPNIP value to FALSE, automatic IP tagging works as just described except that the IP Manager creates tagged IP and IPNetwork objects instead of tagged VPNIP and VPNIPNetwork objects.

Note: Automatic IP tagging is also known as VRF IP tagging.

User-defined tagging

User-defined IP tagging occurs when a user creates an IP tag filter group and specifies a tag value and an overlapping IP address range for the group. The user-specified tag may be applied to both non-VRF overlapping IP addresses and VRF IP addresses, where a non-VRF overlapping IP address is defined as an overlapping IP address that is not part of an MPLS Layer 3 VPN. In addition, users can configure an IP tag filter group to double-tag the name of the objects created for discovered VRF IP addresses, or to replace the route distinguisher tag in the name of these objects with the user-specified tag.

For every IP address that falls within the overlapping IP address range of an IP tag filter group, the IP Manager creates an IP object and tags its name with the user-specified value. The IP Manager also creates a corresponding IPNetwork object and tags it with the same value used to tag the IP objects. (The IP objects have a "ConnectedVia" relationship with the IPNetwork object.)

IP tag naming scheme

The tagged name of a VPNIP, IP, VPNIPNetwork, or IPNetwork object is defined in the object's Name and DisplayName attributes. The tagged name specified in the DisplayName attribute is the name of the object displayed in the Global Console, while the tagged name specified in the Name attribute is the name of the object as known in the IP Manager's repository.

In the tagged naming examples that follow, IP address 10.1.0.2 in Figure 9-1 on page 9-2 is the overlapping IP address.

Naming scheme for automatic tagging

The naming scheme for automatic tagging is based on the route distinguisher (RD) tags obtained from the vendor VPN MIB on the discovered system.

Note: The following discussion assumes DisableOverlappingIP = FALSE.

Tagged VPNIP or IP objects representing VRF IP addresses appear in the topology as:

Name: *IP-<IP address>/<RD tag>*
DisplayName: *<RD tag>:<IP address> [<system containing this IP>]*

Examples:

Name: *IP-10.1.0.2/29*
DisplayName: *29:10.1.0.2 [PE 1]*

Name: *IP-10.1.0.2/107*
DisplayName: *107:10.1.0.2 [PE 2]*

Corresponding tagged VPNIPNetwork or IPNetwork objects appear in the topology as:

Name: *IPNET-<IP network>/<RD tag>*
DisplayName: *<RD tag>:<IP network>*

Examples:

Name: *IPNET-10.1.0.0/29*
DisplayName: *29:10.1.0.0*

Name: *IPNET-10.1.0.0/107*
DisplayName: *107:10.1.0.0*

Naming scheme for user-defined tagging

The naming scheme for user-defined tagging is based on user-specified tags in user-created IP tag filters. The tagged naming examples included in this section are based on the following two IP tag filter groups:

- Router-PE1 containing:
  - Matching criteria that allows only router PE 1 to become a member of the Router-PE1 group
  - User-specified tag = cust1
  - Overlapping IP address range = 10.1.0.2
  - Route distinguisher = enabled (default)
- Router-PE2 containing:
  - Matching criteria that allows only router PE 2 to become a member of the Router-PE2 group
  - User-specified tag = cust2
  - Overlapping IP address range = 10.1.0.2
  - Route distinguisher = enabled (default)

Note: The following discussion assumes DisableOverlappingIP = FALSE and UseVPNIP = FALSE.

Tagged IP objects representing non-VRF IP overlapping addresses appear in the topology as:

Name: *IP-<IP address>/<user-specified tag>*
DisplayName: *<user-specified tag>:<IP address> [<system containing this IP>]*

Corresponding tagged IPNetwork object:

Name: *IPNET-<IP network>/<user-specified tag>*
DisplayName: *<user-specified tag>:<IP network>*

Tagged IP objects representing VRF IP addresses appear in the topology as:

Name: *IP-<IP address>/<user-specified tag>-<RD tag>*
DisplayName: *<user-specified tag>-<RD tag>:<IP address> [<system containing this IP>]*

Examples:

Name: *IP-10.1.0.2/cust1-29*
DisplayName: *cust1-29:10.1.0.2 [PE 1]*

Name: *IP-10.1.0.2/cust2-107*
DisplayName: *cust2-107:10.1.0.2 [PE 2]*

Corresponding tagged IPNetwork object:

Name: *IPNET-<IP network>/<user-specified tag>-<RD tag>*
DisplayName: *<user-specified tag>-<RD tag>:<IP network>*

Examples:

Name: *IPNET-10.1.0.0/cust1-29*
DisplayName: *cust1-29:10.1.0.0*

Name: *IPNET-10.1.0.0/cust2-107*
DisplayName: *cust2-107:10.1.0.0*

The preceding examples demonstrate the double-tagging capability. Because IP address 10.1.0.2 is a VRF IP and because the route distinguisher parameter is enabled in the Router-PE1 and Router-PE2 IP tag filter groups, the IP objects created for IP address 10.1.0.2 are double-tagged.

By disabling the route distinguisher parameter in the filter groups and invoking a rediscovery of the topology, the IP objects for IP address 10.1.0.2 will receive a single tag:

Name: *IP-10.1.0.2/cust1*
DisplayName: *cust1:10.1.0.2 [PE 1]*

Name: *IP-10.1.0.2/cust2*
DisplayName: *cust2:10.1.0.2 [PE 2]*

By disabling the route distinguisher parameter in just one of the groups, say Router-PE1, and performing a rediscovery of the topology, the IP objects for IP address 10.1.0.2 will be named as follows:

Name: *IP-10.1.0.2/cust1*
DisplayName: *cust1:10.1.0.2 [PE 1]*

Name: *IP-10.1.0.2/cust2-107*
DisplayName: *cust2-107:10.1.0.2 [PE 2]*

Shadow IP and IPNetwork objects

When the EnableShadowIP parameter is set to TRUE (the default) in the BASEDIR/smarts/conf/discovery/tpmgr-param.conf file, the IP Manager creates a shadow IP object for each tagged, non-overlapping, non-VRF IP object that it creates, and creates a shadow IPNetwork object for the IPNetwork object to which the IP object is "ConnectedVia." Earlier versions of satellite Domain Managers will import the shadow objects, rather than their tagged counterparts, when importing topology from the IP Manager (specifically, from IP Availability Manager).

Shadow object rationale

Even though a user will not intentionally tag a non-overlapping, non-VRF IP address, it can still happen. (For example, the IP address might just happen to fall within the overlapping IP address range specified for an IP tag filter group.) The creation of a shadow IP object for this IP address, which earlier versions of satellite Domain Managers will import, ensures that the Domain Managers become aware of this IP address.

The IP Manager does not create shadow IP objects for tagged, overlapping, non-VRF IP addresses because earlier versions of satellite Domain Managers expect to import overlapping non-VRF IP addresses as DuplicateIP objects. Because DuplicateIP objects are not useful to the Domain Managers, the absence of shadow IP objects for tagged, overlapping, non-VRF IP addresses ensures that the Domain Managers do not become aware of these IP addresses.

So, from the perspective of earlier versions of satellite Domain Managers, the only shadow IP objects that make sense are the ones for tagged, non-overlapping, non-VRF IP addresses.

Shadow object naming schemes

Shadow IP object names comply with the following naming scheme:

Name: *IP-<IP address>*
Display name: *[Shadow]:<IP address> [<system containing this IP>]*

Where *<IP address>* is the IP address of the tagged, non-VRF IP object that the shadow IP object is shadowing.

Shadow IPNetwork object names comply with the following naming scheme:

Name: *IPNET-<IP network>*
Display name: *[Shadow]:<IP network>*

Where *<IP network>* is the IP network of the tagged, non-VRF IPNetwork object that the shadow IPNetwork object is shadowing.

Shadow object unique characteristics

A shadow IP object is associated with its tagged IP object counterpart via the ShadowOf relationship. Similarly, a shadow IPNetwork object is associated with its tagged IPNetwork object counterpart via the ShadowOf relationship. All shadow objects have an attribute named IsShadow that is set to TRUE.

A shadow IP object has a "ConnectedVia" relationship to a shadow IPNetwork object, and a shadow IPNetwork object has "ConnectedTo" relationships to its shadow IP objects.

Note: Because shadow IP objects are unmanaged, the IP Manager does not include them in its correlation analysis.

Shadow object use by earlier versions of satellite Domain Managers

Because the shadow IP object has the same interface and system layering information as its tagged IP object counterpart, earlier versions of satellite Domain Managers can use Shadow IP objects in the same way that they used untagged IP objects:

To create the correct connections and to determine the hosts that are hosting the IPs.

Understanding tagging filter groups and settings

The tagging filter groups and settings provide a means to create an IP tag filter and specify a unique tag for each group of systems using the same but locally distinct IP addresses. Because the tags enable the IP Manager to distinguish between overlapping addresses belonging to different customers, the IP manager is able to represent the customer-related overlapping IP addresses in the modeled topology.

Each discovered system that meets the defined matching criteria for an IP tag filter group is subjected to a second line of filtering defined by the parameters defined in the group's setting, including a user-specified tag and a range of overlapping IP addresses. In this way, different filtering values can be applied to different groups of systems.

Modifying tagging filter values and creating tagging filter groups

Procedures for modifying existing tagging filter groups and creating new ones are given in "Working with groups and settings" on page 5-7.

Default tagging filter groups and settings

Using the default tagging filter groups for the IP Manager, you create IP tag filter groups to control the naming of the IP addresses gathered from the systems in the managed network environment.

Default tagging groups

The IP Manager provides two default tagging groups:

- IP Tag Filter-1
- Other Filters

Table 9-1 lists the default tagging groups, their matching criteria, and their settings. The matching-criteria target class for tagging groups is UnitaryComputerSystem.

Table 9-1 Default tagging groups

Tagging group	Matching criteria	Default settings
IP Tag Filter-1	UnitaryComputerSystem attributes: • Name = *	IP Tagging Policy
Other Filters	UnitaryComputerSystem attributes: • No attributes	IP Tagging Policy

The Other Filters tag group does not contain any matching-criteria attributes and has the lowest priority. Systems that do not match the criteria for the IP Tag Filter-1 tag group become members of the Other Filters tag group.

Default tagging settings

IP Availability Manager provides a single default tagging setting named IP Tagging Policy.

The IP Tagging Policy setting configures the IP tagging for a system. Table 9-2 lists the IP Tagging Policy setting parameters, and Figure 9-3 shows all of the parameters in a single view.

Table 9-2 Default values for the IP Tagging Policy setting

Parameter	Value	Description
IPTag	String of unspecified length Default: Null string (empty)	Specifies the name to be assigned to this tag. This tag will be applied to the IP addresses falling within the ranges specified for the OverlappingIPAddrOrRange and MgmtIPAddrOrRange parameters.
MgmtIPAddrOrRange	String of unspecified length Default: Null string (empty)	Specifies the IP address or range of IP addresses to be managed by default. You can use pattern matching criteria, described in Appendix A, "Wildcard Patterns." For example: • 192.168.1.2 or 192.168.1.* • 192.168.1.<2-20> • 192.<168-170>.*.*|10.10.<10-20>.* The pipe (|) separator can be used to specify multiple IP ranges.
MonitorMgmtIP	MANAGED, UNMANAGED Default: MANAGED	Determines the management status to be assigned to the IP addresses falling within the range of the MgmtIPAddrOrRange parameter. • MANAGED Indicates that the discovered IP addresses are to be managed. • UNMANAGED Indicates that the discovered IP addresses are are to be unmanaged.
MonitorOverlappingIP	MANAGED, UNMANAGED Default: UNMANAGED	Determines the management status to be assigned to the IP addresses falling within the range of the OverlappingIPAddrOrRange parameter. • MANAGED Indicates that the discovered IP addresses are to be managed. • UNMANAGED Indicates that the discovered IP addresses are are to be unmanaged.

Table 9-2 Default values for the IP Tagging Policy setting (continued)

Parameter	Value	Description
OverlappingIPAddrOrRange	String of unspecified length Default: Null string (empty)	Specifies the IP address or range of IP addresses to be unmanaged by default. You can use pattern matching criteria, described in Appendix A, "Wildcard Patterns." For example: • 192.168.1.2 or 192.168.1.* • 192.168.1.<2-20> • 192.<168-170>.*.*|10.10.<10-20>.* The pipe (|) separator can be used to specify multiple IP ranges.
RouteDistinguisher	ENABLE_RD, DISABLE_RD Default: ENABLE_RD	Determines the level of tagging for VPN-IP (VRF IP) addresses: • ENABLE_RD Indicates that the IP objects created for the VRF IP addresses falling within the range specified for the OverlappingIPAddrOrRange parameter are to be tagged using the route distinguisher from the vendor VPN MIB, in addition to the user-defined tag specified in the IPTag parameter. • DISABLE_RD Indicates that the IP objects created for the VRF IP addresses falling within the range specified for the OverlappingIPAddrOrRange parameter are NOT to be tagged using the route distinguisher from the vendor VPN MIB; only the user-defined tag specified in the IPTag parameter is to be used to tag these IP objects.

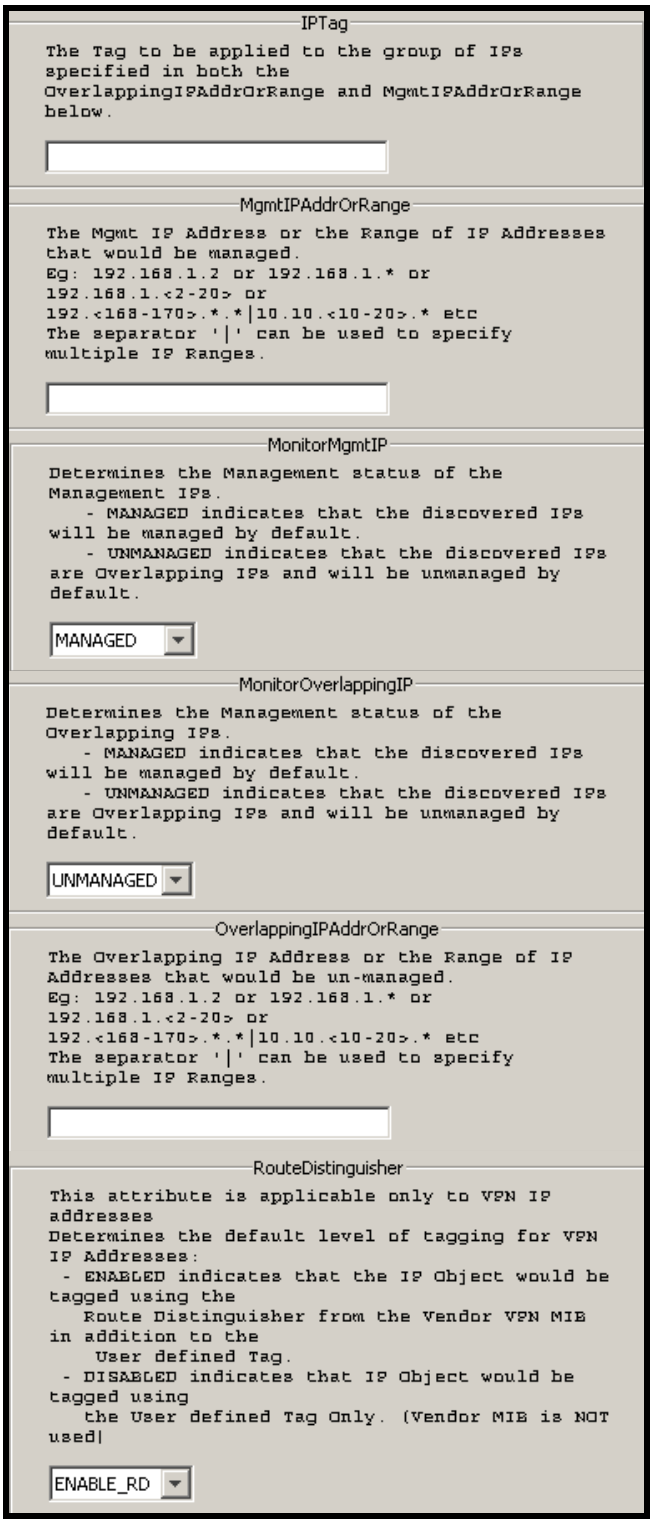


Figure 9-3 Parameters of default setting IP Tagging Policy shown in a single view

Overview of creating an IP tag filter group

An IP tag filter group has two sets of matching criteria:

- System

  The membership class for an IP tag filter group is UnitaryComputerSystem. Only the discovered systems matching the matching criteria (attributes specified for UnitaryComputerSystem) for a particular group can become members of that group.

- IP address range

  The IP address range matching criteria consists of two IP address ranges: one for management IP addresses and one for overlapping IP addresses. Each of the ranges may contain a single IP address. Only the IP addresses collected for the discovered system matching either of the IP address ranges will be tagged in accordance to the user-specified tag for the group.

If a system (device) matches the system criteria for the filter, all IP addresses collected for the discovered system are subject to the management IP address range and the overlapping IP address range defined for the filter. If a system does not match the system criteria for the filter, none of the IP addresses collected for that discovered system are subject to the two IP address ranges defined for the filter.

If an IP address falls within the management IP address range, the user-specified tag for the filter is applied to the IP object created for that IP address. If the IP address does not fall within that range, the user-specified tag for the filter is not applied to the IP object created for that IP address.

Any management IP object created for an IP address matching the management IP address range is MANAGED by default. You can change the value to UNMANAGED by selecting UNMANAGED from the MonitorMgmtIP drop-down menu.

Any management IP address created for an IP address matching the management IP address range is excluded from the overlapping IP address range defined for the IP tag filter group.

If an IP address falls within the overlapping IP address range, the user-specified tag for the filter is applied to the IP object created for that IP address. If the IP address does not fall within that range, the user-specified tag for the filter is not applied to the IP object created for that IP address.

Any overlapping IP object created for an IP address matching the overlapping IP address range is UNMANAGED by default. You can change that value to MANAGED by selecting MANAGED from the MoinitorOverlappingIP drop-down menu.

The RouteDistinguisher parameter (ENABLE_RD or DISABLE_RD; ENABLE_RD by default) for an IP tag filter group is applicable to all VRF IP addresses that fall within the overlapping IP address range defined for the filter. For an enabled RouteDistinguisher parameter, any IP address falling within the range of the overlapping IP address range is tagged with the route distinguisher name associated with the VRF to which that IP address belongs. (The IP Manager gathers route distinguisher information by reading the vendor VPN MIB on the discovered system.) Conversely, if the RouteDistinguisher parameter is disabled, any VRF IP address falling within the overlapping IP address range is NOT tagged with the route distinguisher value.

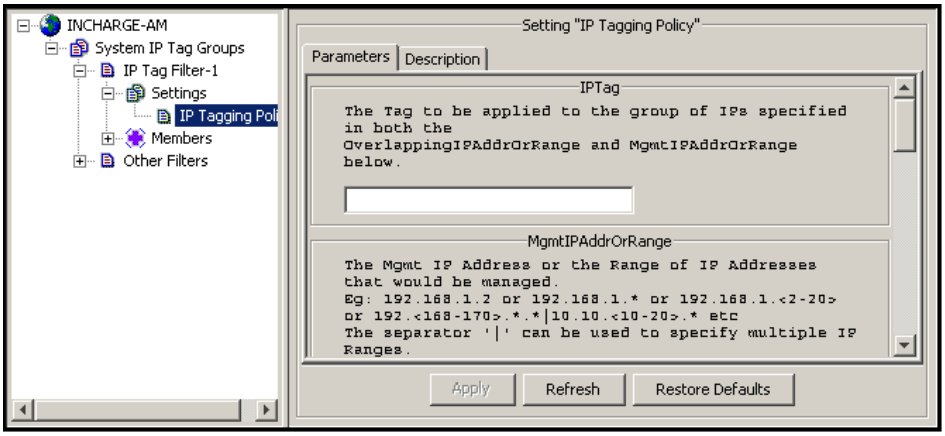
It is important to note that VRF IP addresses NOT falling within the overlapping IP address range will be tagged with the route distinguisher value regardless of the RouteDistinguisher parameter setting (ENABLE_RD, DISABLE_RD). The RouteDistinguisher parameter setting applies only to the VRF IP addresses that fall within the overlapping IP address range defined for the filter.

Single tagging and double tagging of an IP object created for a discovered IP address are both possible. "IP tag naming scheme" on page 9-6 provides descriptions and examples of single-tagged and double-tagged IP objects.

Example of creating an IP tag filter group

The following procedure demonstrates how to use the Polling and Thresholds Console to create an IP tag filter group for routers by assigning values (via the IP Tagging Policy setting) and adding additional matching criteria to the default system IP tag group named IP Tag Filter-1.

1. In a **Polling and Thresholds Console** attached to an IP Availability Manager or IP Performance Manager, click the **Tagging** tab to display the tagging group tree.
2. In the tagging group tree, click the **+** signs to expand the tagging group tree as shown in the following display, then click **IP Tagging Policy** to display its setting configuration window.



3. In the setting configuration window, specify values for the following parameters:
   - IPTag

     For example: USCASJ-ERX-VZ
   - MgmtIPAddrOrRange

     For example: 192.168.35.<1-2>
   - MonitorMgmtIP

     For example: MANAGED
   - MonitorOverlappingIP

     For example: UNMANAGED
   - OverlappingIPAddrOrRange

     For example: 192.168.<35-45>.*|192.168.100.<10-20>
   - RouteDistinguisher

     For example: ENABLE_RD
4. Click the **Apply** button to save your changes.
5. In the tagging group tree, click **IP Tag Filter-1** to display the configuration window.
6. In the configuration window, click **Matching Criteria** to display the **Matching Criteria** tab.

7. In the **Matching Criteria** tab, specify the additional matching criteria that will allow only the routers in the target customer network to become members of this group. (The procedure for specifying matching criteria is given in "Editing matching criteria" on page 5-9.)

   Note: Specify matching criteria broad enough so that *all* routers in the target network become members of this group. Otherwise, the IP Manager will create multiple names (multiple IPNetwork instances) for the target network, which will produce incorrect correlation analysis results.

   For example, if two distinguishing features for the routers in the target customer network is that they are all routers having names beginning with the string "net1," set the **Type** attribute to **Router** and the **Name** attribute to **net1*** to ensure that only those routers become members of the **IP Tag Filter-1** group.

   For any router that becomes a member of this group, the names of all IP objects created for the router IP addresses that fall within the OverlappingIPAddrOrRange range specified in Step 3 will be tagged with the IPTag value that you specified in Step 3.

8. Click the **Apply** button to save your changes.
9. Click the **Reconfigure** toolbar button to make your changes take effect.

To create additional system IP tag groups for systems, create new groups from the group configured in the preceding procedure and repeat that procedure for the newly created groups. (The procedure for creating new groups is given in "Creating new groups" on page 5-10.)

Create an IP tag filter for each group of systems using the same but locally distinct IP addresses.

Verifying IP tag group configurations

To verify that your system IP tag groups are configured properly:

1. Attach the Domain Manager Administration Console to the target IP Manager. ("Procedure for opening the Polling and Thresholds Console" on page 5-2 provides instructions on opening the Domain Manager Administration Console.)
2. In the console, select **Topology > Discover All** to rediscover all managed systems.
3. In the topology tree of the console, expand the IP class to see the tagged IP instances (and their associated systems) that the IP Manager just created.
4. Verify that the rediscovered IPs are associated with the appropriate systems.

A

Wildcard Patterns

This chapter shows you how to use a wildcard pattern in a text field to match a number of objects instead of specifying each object by name. (EMC Smarts programs, such as the Global Console, enable you to specify wildcard patterns in text fields.) It consists of the following sections:

- Types of wildcard patterns A-2

Types of wildcard patterns

A wildcard pattern is a series of characters that are matched against incoming character strings. You can use these patterns when you define pattern matching criteria.

Matching is done strictly from left to right, one character or basic wildcard pattern at a time. Basic wildcard patterns are defined in Table A-1. Characters that are not part of match constructs match themselves. The pattern and the incoming string must match completely. For example, the pattern abcd does not match the input abcde or abc.

A compound wildcard pattern consists of one or more basic wildcard patterns separated by ampersand (&) or tilde (~) characters. A compound wildcard pattern is matched by attempting to match each of its component basic wildcard patterns against the entire input string. For compound wildcard patterns, see Table A-2.

If the first character of a compound wildcard pattern is an ampersand (&) or tilde (~) character, the compound is interpreted as if an asterisk (*) appeared at the beginning of the pattern. For example, the pattern ~*[0-9]* matches any string not containing any digits. A trailing instance of an ampersand character (&) can only match the empty string. A trailing instance of a tilde character (~) can be read as “except for the empty string.”

Note: Spaces are interpreted as characters and are subject to matching even if they are adjacent to operators like “&”.

Table A-1 Basic wildcard patterns

Character	Description
?	Matches any single character. For example, server?.example.com matches server3.example.com and serverB.example.com, but not server10.example.com.
*	Matches an arbitrary string of characters. The string can be empty. For example, server*.example.com matches server-ny.example.com and server.example.com (an empty match).
[set]	Matches any single character that appears within [set]; or, if the first character of [set] is (^), any single character that is *not* in the set. A hyphen (-) within [set] indicates a range, so that [a-d] is equivalent to [abcd]. The character before the hyphen (-) must precede the character after it or the range will be empty. The character (^) in any position except the first, or a hyphen (-) at the first or last position, has no special meaning. For example, server[789-].example.com matches server7.example.com through server9.example.com, but not server6.example.com. It also matches server-.example.com. For example, server[^12].example.com does not match server1.example.com or server2.example.com, but will match server8.example.com.

Table A-1 Basic wildcard patterns (continued)

Character	Description
<n1-n2>	Matches numbers in a given range. Both *n1* and *n2* must be strings of digits, which represent non-negative integer values. The matching characters are a non-empty string of digits whose value, as a non-negative integer, is greater than or equal to *n1* and less than or equal to *n2*. If either end of the range is omitted, no limitation is placed on the accepted number. For example, 98.49.<1-100>.10 matches a range of IP addresses from 98.49.1.10 through 98.49.100.10. Example of an omitted high end of the range: <50-> matches any string of digits with a value greater than or equal to 50. Example of an omitted low end of the range: <-150> matches any value between zero and 150. For a more subtle example: The pattern <1-10>* matches 1, 2, up through 10, with * matching no characters. Similarly, it matches strings like 9x, with * matching the trailing x. However, it does not match 11, because <1-10> always extracts the longest possible string of digits (11) and then matches only if the number it represents is in range.
|	Matches alternatives. For example,"ab|bc|cd" without spaces matches exactly the three following strings: "ab", "bc", and "cd". A | as the first or last character of a pattern accepts an empty string as a match. Example with spaces "ab | bc" matches the strings "ab" and " bc".
\	Removes the special status, if any, of the following character. Backslash (\) has no special meaning within a set ([set]) or range (<n1-n2>) construct.
\\	Removes the special status of the characters (,), [,], <, and >. For example, the pattern \\(A\\) matches the string (A).

Special characters for compound wildcard patterns are summarized in Table A-2.

Table A-2 Compound wildcard patterns

Character	Description
&	"And Also" for a compound wildcard pattern. If a component basic wildcard pattern is preceded by & (or is the first basic wildcard pattern in the compound wildcard pattern), it *must* successfully match. Example: *NY*&*Router* matches all strings which contain NY and also contain Router. Example: <1-100>&*[02468] matches even numbers between 1 and 100 inclusive. The <1-100> component only passes numbers in the correct range and the *[02468] component only passes numbers that end in an even digit. Example: *A*|*B*&*C* matches strings that contain either an A or a B, and also contain a C.
~	"Except" for a compound wildcard pattern (opposite function of &).If a component basic wildcard pattern is preceded by ~, it *must not* match. Example: 10.20.30.*~10.20.30.50 matches all devices on network 10.20.30 except 10.20.30.50. Example: *Router*~*Cisco*&*10.20.30.*~10.20.30.<10-20>* matches a Router, except a Cisco router, with an address on network 10.20.30, except not 10.20.30.10 through 10.20.30.20.

B

Polling for Analysis

This chapter describes how the IP Manager uses a cooperative approach to Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) polling to obtain connectivity, fault, and performance data for its correlation analysis. It consists of the following sections:

- Polling overview B-2
- ICMP poller B-3
- SNMP poller B-4
- ICMP and SNMP polling coordination B-5

Polling overview

After the initial discovery and modeling of the network, all IP products except IP Discovery Manager monitor the status of the discovered systems using ICMP and SNMP polling. The parameters for controlling ICMP and SNMP polling and for setting thresholds for polled data are accessed through the Polling and Thresholds Console.

IP products use IPv4 ICMP and SNMP pollers, which send ICMP and SNMP messages for use with IP version 4. IP Availability Manager Extension for IPv6 also uses an IPv6 ICMP poller, which sends ICMP messages for use with IP version 6.

Note: Currently, no IP products use IPv6 SNMP pollers.

All IP products use IPv4 SNMP to discover the systems in the managed network environment. If a discovered system supports IPv6-knowledgeable MIBs, IP Availability Manager Extension for IPv6 also discovers any IPv6 addresses configured on the system.

Thereafter, all IP products uses IPv4 ICMP and IPv4 SNMP to monitor the system. IP Availability Manager Extension for IPv6 also uses IPv6 ICMP to monitor the IPv6 addresses configured on the system.

The discussions that follow pertain to IPv4 ICMP and SNMP polling.

ICMP poller

The IP Manager uses a high-performance, asynchronous ICMP poller. The ICMP poller uses two asynchronous threads; one thread sends polls, and the other receives polls. Because the send and receive threads operate separately, slow response times or excessive timeouts do not affect the polling rate. Thus, the ICMP poller performs at a consistent polling rate.

Figure B-1 shows the three possible states of a monitored object as determined by its response to an ICMP poll. The states are Up, Notification Pending, and Down.

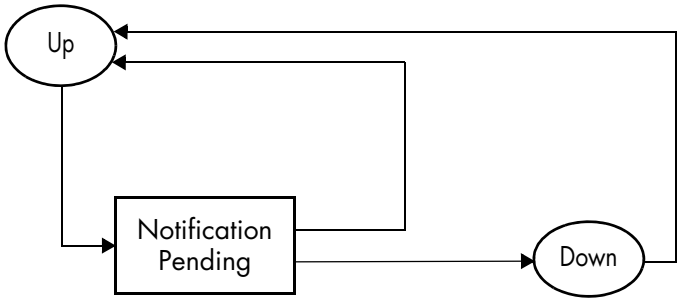


Figure B-1 Possible states of an object during an analysis polling cycle

An object is in the up state until it fails to respond to an ICMP poll. When it fails to respond, the status of the object changes to the Notification Pending state and stays in that state until the IP Manager can determine whether the object is Up or Down.

If the object responds to an ICMP poll before the maximum failure retry count is exceeded for the object, the status of the object changes back to the Up state.

If the maximum failure retry count is exceeded for the object, the status of the object changes to the Down state, and the IP Manager does not poll the object again until the next scheduled polling cycle. An object stays in the Down state until it responds to an ICMP poll. When the object responds, the IP Manager changes the status of the object to the Up state.

This polling sequence is performed for each IP endpoint that is managed.

SNMP poller

The IP Manager uses a synchronous, multi-threaded SNMP poller. By default, the SNMP poller uses ten synchronous polling threads.

The SNMP poller fully supports the SNMPv1 and v2c protocols, and supports authentication and data encryption for the SNMPv3 protocol. With SNMPv1, the correlation model uses 32-bit counters in its correlation analysis. With SNMPv2c or v3, the correlation model uses high-capacity 64-bit counters in its correlation analysis. Using 64-bit counters is critical for performance analysis of high-speed data links because using 32-bit counters might result in wrapping (overflow) of the counters between polls.

Polling for devices with multiple IP addresses is supported because the SNMP poller supports multiple IP addresses for each SNMP agent. The SNMP poller automatically switches to an alternate IP address during failures, thereby ensuring the integrity of the IP Manager correlation analysis during an outage.

Just-in-time polling

The SNMP poller's MIB variable poll list is driven by a Just-In-Time polling algorithm, which ensures that only those MIB variables needed for correlation are polled. For example, if a port monitored for performance data is disabled, or goes down, the SNMP poller automatically removes the relevant MIB variables from the poll list. If the port is re-enabled, or comes back up, the variables are automatically put back onto the MIB poll list.

Request-consolidation polling

Issuing a single SNMP GET request that requests 10 variables is more efficient than issuing 10 GET requests each requesting a single variable. The SNMP poller consolidates as many variables as possible into a single SNMP GET request. The consolidation is not restricted to variables from the same SNMP table. Polling consolidation continually adapts to changes in the MIB variable poll list.

Upon encountering a non-fatal error during polling consolidation, the SNMP poller responds differently to an SNMPv1 agent than to an SNMPv2c or v3 agent for the following reason: Where an SNMPv1 agent stops processing a request upon encountering an error, an SNMPv2c or v3 agent continues processing a request upon encountering an error. An SNMPv2c or v3 agent handles errors on a per-OID basis.

If a non-fatal error is encountered by an SNMPv1 agent during a GET request seeking multiple variables, the SNMP poller suspends the polling of the affected variable because continuing to poll that variable would require the resending of the remainder of the request after receiving the error, which would probably impact the performance of the SNMPv1 agent; the SNMP poller continues to poll the unaffected variables. (An example of an affected variable is one that has become unavailable due to a configuration change.) This behavior enables the SNMP poller to operate efficiently with an SNMPv1 agent during unexpected changes to a device's configuration.

In contrast, if a non-fatal error is encountered by an SNMPv2c or v3 agent during a GET request seeking multiple variables, the SNMP poller continues the polling of the affected variable as well as the unaffected variables.

ICMP and SNMP polling coordination

The IP Manager links its ICMP and SNMP pollers so that the SNMP poller does not send requests to any SNMP agent address that the ICMP poller determines to be unreachable. IP addresses that are unresponsive to ICMP polls are added to a do not poll list, which the SNMP poller checks before sending SNMP requests.

If an SNMP agent address is on the do not poll list, the SNMP poller does not send a request to that address. If an SNMP agent has multiple IP addresses, the SNMP poller checks for each address in the do not poll list. For an address that does not appear in the list, the SNMP poller sends a request to that address. If all addresses for an agent are on the list, the SNMP poller deems the agent as unreachable and temporarily suspends sending SNMP requests to that agent. As soon as an agent's IP address becomes responsive, as determined by the ICMP poller, the address is removed from the list, and SNMP polling resumes.

Index

Numerics

1 Gb Ethernet access ports threshold group 7-6, 7-13
1 Gb Ethernet interface threshold group 7-5, 7-12
1 Gb Ethernet trunk ports threshold group 7-6
10/100 Mb Ethernet access ports threshold group 7-6, 7-13
10/100 Mb Ethernet interface threshold group 7-5, 7-12
10/100 Mb Ethernet trunk ports threshold group 7-6

A

Access
    Port 7-6, 7-13
    Port group 7-6, 7-13
AccessAddressFormat 2-9
AccessProtocol 8-4
Adding or removing matching criteria 5-9
Analysis mode
    Connectivity polling 6-6, 6-12
    Connectivity polling - external poller 6-7, 6-13
    Environment polling 6-9
    Filesystem polling 6-10
    Performance polling - processor and memory 6-10, 6-11
Analysis mode of subinterface performance
    Ethernet interface/port performance 7-17
    Generic interface/port performance 7-18
ATM
    Access ports threshold group 7-13
    Interface threshold group 7-12
    Trunk ports threshold group 7-13
ATM access ports threshold group 7-6
ATM interface group 7-5
ATM trunk ports threshold group 7-6
Attributes
    DisplayName 9-6
    Name 9-6
AutoNameOrder 2-15
autoReprobe_short 2-10

B

Backplane utilization threshold 7-21
Backup interface support setting 7-9
    Maximum uptime 7-9
Backup interface threshold group 7-6
Backup interfaces threshold group 7-13
Broadcast threshold
    Ethernet interface/port performance 7-17
    Generic interface/port performance 7-18
Broker 1-5

C

Changing matching criteria 5-9
Changing priority of a group 5-9
Changing setting parameters 5-8
Cisco Discovery Protocol (CDP) 2-12
CLI Access Group device access group 8-4
CLI Access Setting
    AccessProtocol 8-4
    LoginID 8-4
    Password 8-4
    Timeout 8-4
Collision threshold
    Ethernet interface/port performance 7-17
Configuration overview 1-6
Connectivity polling
    Analysis mode 6-6, 6-12
    Polling interval 6-6, 6-12
    Retries 6-6, 6-12
    Timeout 6-6, 6-12
Connectivity polling - external poller
    Analysis mode 6-7, 6-13
    Initial status 6-7, 6-13
    Instrument card 6-7
    Instrument IP 6-7, 6-13
    Instrument network adapters 6-7
    Instrument SNMP agents 6-7
Connectivity Polling for IPv6 Addresses
    Description 6-12
Connectivity Polling for IPv6 Addresses - External Polling
    Description 6-12
Connectivity setting
    Correlation use bridging mode 7-9
    Number of bridged via threshold 7-10
    Number of bridges threshold 7-10
    Restart trap threshold 7-10
    Restart trap window 7-10

Testing notification mode 7-10
Copying a group 5-10
Correlation use bridging mode 7-9
Creating a group 5-10

D

defaultICMPAutoRetries 2-7, 2-19
defaultICMPAutoTimeout 2-7, 2-19
defaultReadCommunity 2-8
defaultRetries 2-6
defaultSNMPAutoRetries 2-7, 2-19
defaultSNMPAutoTimeout 2-7, 2-19
defaultSNMPPort 2-6
defaultTimeout 2-7, 2-19
Device Access
Settings 8-4
Device Access tab 5-4
Dial-on-demand interface setting 7-6
Maximum uptime 7-11
Dial-On-Demand interfaces threshold group 7-13
DisableOverlappingIP 2-5
Discard threshold
Ethernet interface/port performance 7-17
Generic interface/port performance 7-18
Discovery
Discover All command 4-11, 7-24, 7-27, 7-33, 9-18
discovery.conf 2-6, 2-19
IPSec and IKE tunnels 2-37
name-resolver.conf 2-15
partition.conf 2-16
Short discovery 2-37
tpmgr-param.conf 2-11
user-defined-connections.conf 2-17
discovery.conf 2-6, 2-19
AccessAddressFormat 2-9
autoReprobe_short 2-10
defaultICMPAutoRetries 2-7, 2-19
defaultICMPAutoTimeout 2-7, 2-19
defaultReadCommunity 2-8
defaultRetries 2-6
defaultSNMPAutoRetries 2-7, 2-19
defaultSNMPAutoTimeout 2-7, 2-19
defaultSNMPPort 2-6
defaultTimeout 2-7, 2-19
DisplayNameFormat 2-9
DuplexAssumed 2-10
HostResourcesEnabled 2-9
HSRPEnabled 2-9
ICMPSleepTime 2-7
importExternalTagsConnections 2-10
ipExcludeList 2-6
IPv6DiscoveryEnabled 2-10, 2-11
LogDiscoveryProgress 2-6
MaximumCommunities 2-8
MaximumHostBits 2-8
MinimumDemandGet 2-9
numberProbeThreads 2-7
reprobePeriod_short 2-37
ShowPendingNONSNMP 2-6
VirtualRouterEnabled 2-9
discovery.conf file 2-2, 2-5
DisplayName attribute 9-6
DisplayNameFormat 2-9
Domain Manager Administration Console 5-2, 5-3
Discover All command 4-11, 7-24, 7-27, 7-33, 9-18
DomainSection 3-3
DuplexAssumed 2-10

E

Environment
Relative temperature threshold 7-19, 7-20
Relative voltage threshold 7-19, 7-20
Environment polling 6-9
Environment polling setting
Analysis mode 6-9
Polling interval 6-9
Retries 6-9
Timeout 6-9
Error threshold
Ethernet interface/port performance 7-17
Generic interface/port performance 7-18
Error traffic threshold
Ethernet interface/port performance 7-18
Generic interface/port performance 7-19
Ethernet interface/port performance
Analysis mode of subinterface performance 7-17, 7-18
Broadcast threshold 7-17, 7-18
Collision threshold 7-17
Discard threshold 7-17, 7-18
Error threshold 7-17, 7-18
Error traffic threshold 7-18, 7-19
Queue drop threshold 7-18, 7-19
Utilization threshold 7-18, 7-19

F

FDDI interfaces threshold group 7-6, 7-13
File system
Maximum utilization percentage 7-20
File system polling 6-10
Filesystem polling
Analysis mode 6-10
Polling interval 6-10
Retries 6-10
Timeout 6-10
Free memory threshold 7-21

G

Generic interface/port performance
Analysis mode of subinterface performance 7-18
Broadcast threshold 7-18
Discard threshold 7-18
Error threshold 7-18
Error traffic threshold 7-19
Queue drop threshold 7-19
Utilization threshold 7-19
Global Console 1-5
Overview discussion 5-2
Polling and Thresholds Console 5-4
Type of

Domain Manager administration 5-2
Polling and Thresholds 5-2
Global Manager 1-4, 3-2
Group
Changing priority 5-9
Copying 5-10
Creating 5-10
Properties 5-7
Groups
Definition of 5-5
Device Access 8-4
CLI Access Group 8-4
Polling 6-5, 6-8
Hubs and Bridges 6-5, 6-8
Other Systems 6-5, 6-8
Proxy Devices 6-5, 6-8
Routers 6-5, 6-8
Switches 6-5, 6-8
System resource groups 7-22, 7-25
System VLAN tag groups 7-25
Tagging 9-11
IP Tag Filter-1 9-11
Other Filters 9-11
Threshold
1 Gb Ethernet access ports 7-6, 7-13
1 Gb Ethernet interface 7-5, 7-12
1 Gb Ethernet trunk ports 7-6
10/100 Mb Ethernet access ports 7-6, 7-13
10/100 Mb Ethernet interface 7-5, 7-12
10/100 Mb Ethernet trunk ports 7-6
Access ports 7-13
ATM access ports 7-6, 7-13
ATM interface 7-5, 7-12
Backup interfaces 7-6, 7-13
Dial-On-Demand interfaces 7-13
Dial-on-demand interfaces 7-6
FDDI interfaces 7-6, 7-13
Hubs and Bridges system resources 7-14
Hubs and bridges system resources 7-7
Interface 7-12
ISDN B Channel 7-12
ISDN D Channel 7-12
ISDN Physical interfaces 7-12
Other Ports access ports 7-6, 7-13
Other Ports trunk ports 7-6
Other Systems system resource 7-7, 7-14
Other Systems system VLAN Tag 7-7, 7-14
Other VLANs group 7-8, 7-15
Routers system resources 7-7, 7-14
Serial interfaces 7-13
Serial interfaces group 7-6
Switches system resource 7-7, 7-14
Switches system VLAN Tag 7-7, 7-14
System resource groups 7-7, 7-14
Token Ring interfaces 7-12
Token Ring interfaces group 7-5
Trunk ports 7-6
VLANs 7-8, 7-15
VLAN impact propagation groups 7-28

H

HostResourcesEnabled 2-9
HSRPEnabled 2-9
Hubs and Bridges polling group 6-5, 6-8
Hubs and Bridges system resource threshold group 7-14
Hubs and Bridges threshold group 7-7

I

ICMP
defaultICMPAutoRetries 2-7, 2-19
defaultICMPAutoTimeout 2-7, 2-19
ICMPSleepTime 2-7
ICMP polls 6-4
ICMPSleepTime 2-7
ics 3-2
ics.conf
DomainSection 3-3
importExternalTagsConnections 2-10
Importing
Topology information 1-5
Information flow 1-3
Initial status 6-7, 6-13
Instrument card 6-7
Instrument IP 6-7, 6-13
Instrument network adapters 6-7
Instrument SNMP agents 6-7
Interface Management Policy
Description 7-22
Parameters 7-22
Interface/port flapping setting
Link trap threshold 7-11
Link trap window 7-11
IP product
Purpose 3-v
IP Tag Filter-1 tagging group 9-11
IP Tagging Policy setting
IPTag 9-12
MgmtIPAddrOrRange 9-12
MonitorMgmtIP 9-12
MonitorOverlappingIP 9-12
OverlappingIPAddrOrRange 9-13
RouteDistinguisher 9-13
ipExcludeList 2-6
IPSec and IKE tunnels
Discovering 2-37
IPTag 9-12
IPv6DiscoveryEnabled 2-10, 2-11
ISDN B Channel threshold group 7-12
ISDN Physical interfaces threshold group 7-12

L

Link trap threshold 7-11
Link trap window 7-11
LogDiscoveryProgress 2-6
LoginID 8-4

M

Matching

Pattern A-2
Matching criteria
1 Gb Ethernet access ports threshold group 7-6, 7-13
1 Gb Ethernet interface threshold group 7-5, 7-12
1 Gb Ethernet trunk ports threshold group 7-6
10/100 Mb Ethernet access ports threshold group 7-6, 7-13
10/100 Mb Ethernet interface threshold group 7-5, 7-12
10/100 Mb Ethernet trunk ports threshold group 7-6
Adding or removing 5-9
ATM access ports threshold group 7-6, 7-13
ATM interface threshold group 7-5, 7-12
ATM trunk ports threshold group 7-6
Backup interfaces threshold group 7-6, 7-13
Changing 5-9
CLI Access Group device access group 8-4
Dial-On-Demand interfaces threshold group 7-6, 7-13
FDDI interfaces threshold group 7-6, 7-13
Hubs and Bridges polling group 6-5, 6-8
Hubs and Bridges system resource threshold group 7-7, 7-14
IP Tag Filter- 1 tagging group 9-11
ISDN B Channel threshold group 7-12
ISDN D Channel threshold group 7-12
Other Filters tagging group 9-11
Other Systems polling group 6-5
Other Systems system resource threshold group 7-7
Other Systems system VLAN Tag group 7-7, 7-14
Other VLANs group 7-8, 7-15
Proxy Devices polling group 6-5, 6-8
Routers polling group 6-5, 6-8
Routers system resource threshold group 7-7, 7-14
Serial interfaces threshold group 7-6, 7-13
Switches polling group 6-5, 6-8
Switches system resource threshold group 7-7, 7-14
Switches system VLAN Tag group 7-7, 7-14
Token Ring interfaces threshold group 7-5, 7-12
VLANs group 7-8, 7-15
Maximum uptime 7-9
Backup interface support setting 7-9
Dial-on-Demand interface support setting 7-11
Maximum utilization percentage 7-20
MaximumCommunities 2-8
MaximumHostBits 2-8
Memory buffer miss threshold 7-21
Memory buffer utilization threshold 7-21
Memory fragmentation threshold 7-21
MgmtIPAddrOrRange 9-12
MinimumDemandGet 2-9
MonitorMgmtIP 9-12
MonitorOverlappingIP 9-12

N

Name attribute 9-6
NameFormat 2-15
name-resolver.conf 2-15
NameFormat 2-15
name-resolver.conf file 2-2, 2-5
Network adapter
Sub-interface 7-17
Subinterface 7-18
Number of bridged via threshold 7-10
Number of bridges threshold 7-10
numberProbeThreads 2-7

O

Operator
Wildcard A-2
Other Filters tagging group 9-11
Other Ports threshold group 7-6, 7-13
Other Systems polling group 6-5, 6-8
Other Systems system resource threshold group 7-14
Other Systems threshold group 7-7, 7-14
Other VLANs threshold group 7-8, 7-15
OverlappingIPAddrOrRange 9-13

P

partition.conf 2-16
partition.conf file 2-2, 2-5
Password 8-4
Pattern A-2
Pattern matching A-2
Performance polling - ports and interfaces 6-10
Analysis mode 6-10
Polling interval 6-10
Polling interval access ports 6-10
Retries 6-10
Timeout 6-10
Performance polling - processor and memory 6-11
Analysis mode 6-11
Polling interval 6-11
Retries 6-11
Timeout 6-11
Polling
ICMP 6-4
Settings 6-5, 6-9
for IPv6 6-6
SNMP 6-4, B-4
Polling and Thresholds Console 5-2
Device Access tab 5-4
Layout 5-4
Polling tab 5-4
Tagging tab 5-4
Thresholds tab 5-4
Toolbar buttons 5-4
Polling interval 6-6, 6-12
Environment polling 6-9
Filesystem polling 6-10
Performance polling - ports and interfaces 6-10
Performance polling - processor and memory 6-11
Polling tab 5-4
Port
Access 7-6, 7-13
Trunk 7-6, 7-13
Priority
Changing 5-9
Problem Impact propagation policy over VLANs
Description 7-28

Processor and memory
Backplane utilization threshold 7-21
Free memory threshold 7-21
Memory buffer miss threshold 7-21
Memory buffer utilization threshold 7-21
Memory buffer utilization threshold 7-21
Memory fragmentation threshold 7-21
Processor utilization threshold 7-21
Proxy Devices polling group 6-5, 6-8

Q

Queue drop threshold
Ethernet interface/port performance 7-18
Generic interface/port performance 7-19

R

Relative temperature threshold 7-19, 7-20
Relative voltage threshold 7-19, 7-20
Removing or adding matching criteria 5-9
reprobePeriod_short 2-37
Restart trap threshold 7-10
Restart trap window 7-10
Restoring default values of a setting 5-8
Retries
Connectivity polling 6-6, 6-12
Environment polling 6-9
Filesystem polling 6-10
Performance polling - ports and interfaces 6-10
Performance polling - processor and memory 6-11
RouteDistinguisher 9-13
Routers polling group 6-5, 6-8
Routers system resource threshold group 7-14
Routers threshold group 7-7

S

Security
SNMPv3 3-5
Serial interfaces threshold group 7-6, 7-13
Setting
Changing parameters 5-8
CLI Access Setting device access group 8-4
Definition of 5-5
Hubs and Bridges polling group 6-5
Hubs and Bridges system resource threshold group 7-7
IP Tag Filter-1 tagging group 9-11
Other Filters tagging group 9-11
Other Systems polling group 6-5
Other Systems system resource threshold group 7-7
Proxy Devices polling group 6-5, 6-8
Restoring default values 5-8
Router polling group 6-5
Routers system resource threshold group 7-7
Switches polling group 6-5
Switches system resource threshold group 7-7
Switches system VLAN Tag group 7-7, 7-14
VLANs group 7-8, 7-15
Short discovery 2-37
Configuring 2-37
ShowPendingNONSNMP 2-6
SNMP
defaultReadCommunity 2-8
defaultRetries 2-6
defaultSNMPAutoRetries 2-7, 2-19
defaultSNMPAutoTimeout 2-7, 2-19
defaultSNMPPort 2-6
defaultTimeout 2-7, 2-19
MaximumCommunities 2-8
Polling B-4
Polls 6-4
SNMPv3 security 3-5
Sub-interface 7-17
Subinterface 7-18
Switches polling group 6-5
Switches system resource threshold group 7-14
Switches threshold group 7-7, 7-14
System Resource Groups 7-7, 7-14
Interface Management Policy 7-22
VLAN Tagging Policy 7-25
System VLAN Tag Groups 7-7, 7-14
VLAN Tagging Policy 7-25

T

Tagging
Settings 9-12
Tagging tab 5-4
Testing notification mode 7-10
Threshold
Interface Management Policy 7-22
Problem Impact propagation policy over VLANs 7-28
Settings 7-8, 7-15
VLAN Tagging Policy 7-25
Thresholds tab 5-4
Timeout 8-4
Connectivity polling 6-6, 6-12
Environment polling 6-9
Filesystem polling 6-10
Performance polling - ports and interfaces 6-10
Performance polling - processor and memory 6-11
TM_USEAGENTADDRESS 2-15
TM_USELOOPBACK 2-15
TM_USENONPRIVATEIP 2-15
TM_USEPRIVATEIP 2-15
TM_USESYSNAME 2-15
Token Ring interfaces threshold group 7-5, 7-12
Topology Importer 1-5
tpmgr-param.conf 2-11
tpmgr-param.conf file 2-2, 2-5
trapd.conf file 2-2, 2-5
trap-notify.conf file 2-2, 2-5
Trunk
Port group 7-13
Trunk ports
Definition of 7-6
Group 7-6

U

user-defined-connections.conf 2-17
user-defined-connections.conf file 2-2, 2-5
Utilization threshold
    Ethernet interface/port performance 7-18
    Generic interface/port performance 7-19

V

VirtualRouterEnabled 2-9
VLAN Impact Propagation Groups 7-7, 7-15
    Problem Impact propagation policy over VLANs 7-28
VLAN Tagging Policy
    Description 7-25
VLANs threshold group 7-8, 7-15

W

Wildcard A-2
    Chart of operators A-2