EMC® Smarts®
Service Assurance Manager

Version 9.2

Deployment Guide

P/N 300-999-852
REV 01

EMC²

Copyright © 1996 - 2013 EMC Corporation. All rights reserved. Published in the USA.

Published March, 2013

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.

EMC^2, EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (https://support.emc.com).

CONTENTS

Preface

Chapter 1 Overview of the Service Assurance Deployment Process

EMC Smarts architecture 20
Deployment process 22
Stage 1: Designing the deployment 23
Stage 2: Installing and configuring the components 24
Stage 3: Validating the deployment 24
Stage 4: Monitoring and tuning the deployment 25
Before you begin checklist 25

Chapter 2 Gathering Basic Information

Document the deployment 28
The solution architecture diagram 28
The deployment build guide 29
Gather information required to size Service Assurance 29
Determine how many devices and applications will be managed 29
Estimate quantities and types of Service Assurance clients 30
Estimate the number of underlying EMC Smarts domains 30
Determine requirements for installing software 30
Define requirements for integrating existing software with EMC Smarts 30
Gather network security information 31
Firewalls between parts of the EMC Smarts deployment 31
Other network features affecting EMC Smarts 32
Basic Service Assurance information checklist 33

Chapter 3 Designing the Service Assurance Deployment

Determine the Service Assurance configuration 36
Standard Global Manager configuration 38
Basic hierarchical Global Manager configuration 39
Hierarchical aggregation Global Manager configuration 40
Hierarchical presentation Global Manager configuration 41
Hierarchical mesh Global Manager configuration 42
Hardware for Service Assurance platform equipment tiers 43
Web Console and Business Dashboard requirements 45
Java Web Start requirements for Web Console 46
Web browser requirements for the Smarts Adapter for UIM/O 47
Deploying Global Managers and Domain Managers 47
Adding information to the solution architecture diagram and build guide 47
Essential EMC Smarts applications 48
Locating EMC Smarts Brokers 48
Locating license servers 49
Locating the Global Managers in the network 50
Considering security and firewalls 51
Considering failover (high availability) configurations 51

Designing acceptance tests 52
Initial design checklist 53

Chapter 4 Designing for Topology Operations

Designing topology synchronization 56
Understanding topology synchronization 56
Planning for topology synchronization 57
Designing topology groupings 58
Designing topology operations checklist 58

Chapter 5 Planning for Notifications

Overview of notification processing 60
Notifications from the Adapter Platform 61
Configuring notifications for clients 61
Designing trap processing 62
Recommended trap processing design 62
Trap forwarding 64
Trap processing performance 65
Filtering traps in trap_mgr.conf 65
Using trap manager hook scripts 65
Adding new topology for received traps 65
Looking up host names 65
Batching traps 66
Configuring trapd.conf file to manage flooding of traps 66
Planning for notifications checklist 66

Chapter 6 Designing for EMC Smarts Users

Determining the Global Manager users 68
Choose a console for types of users 68
Deployment considerations for client consoles 69
Users and security 69
Designing user profiles 71
Designing notification lists 71
Restricting console operations 72
Designing consoles 72
Planning for tools and tool deployment 72
EMC Smarts user design checklist 73

Chapter 7 Deploying Business Impact Processing

Designing business impact processing 76
Business processes 76
Business topology 76
Weights: Impacts of issues in the topology 77
Importing information 77
The structure of import files 78
Business Impact Manager design checklist 79

Chapter 8 Deploying Service Assurance

General installation/deployment guidelines 82
EMC Smarts licensing 82
EMC Smarts Service Assurance installation and configuration 83

Configure security 83
Deploy Business Impact Manager 84
Deploy user configurations 85
Configuring the Java clients to use a proxy server 86

Chapter 9 Validating Your Deployment (Acceptance Testing)

Validation techniques 90
Initial validation 90
Validating topology operations 90
Validating users and capabilities 90

Chapter 10 Monitoring the Performance of Your Deployment

Assessing performance 92
Monitor Global Manager memory usage 92
Monitoring Global Manager queues 93
Monitoring with heartbeat notifications 96
Monitoring topology synchronization 97
Periodic dump threads and dump clients 99
Monitoring tools 101
Monitoring escalation 101
Improving performance 104
Refine notification lists for users 104
Reassess configuration of topology synchronization 104
Refine set of trap and syslog notifications sent to Global Manager 104
Increase worker threads for escalation 105
Upgrade server hardware 106
Redesign the Service Assurance deployment 106

Appendix A Design and Deployment Checklists

Before you begin checklist 108
Basic information checklist 109
EMC Smarts design checklist 110
Designing topology operations checklist 110
Planning for notifications checklist 111
EMC Smarts client and user design checklist 112
Business Impact Manager design checklist 113

Appendix B Service Assurance Manager Benchmarks

Service Assurance Manager benchmarking deployment 116
Number of traps Service Assurance Manager can successfully process 120
Number of syslog messages a given Smarts syslog parser can successfully handle per second 123
Syslog adapter using --file option 123
Syslog adapter using --tail option 123
Number of escalation policies that can successfully be processed using multiple threads 124
Service Assurance Manager 8.0 performance benchmarking 126
Environment 126
Scenario 126
Results 127
Time to log in to the SAM 8.0 server 127
Environment 127

Scenario 127
Results 128
Default Global Console heap size allocation 128
Environment 128
Recommendations 128
SAM 8.1 latency performance benchmark testing 129
Environment 129
Scenarios 130
Results 130
Amount of memory SAM 8.1 requires to process notifications 131
Environment 131
Test scenarios 132
Results on a Linux AS5 platform 133

Index

FIGURES

	Title	Page
1	EMC Smarts Service Assurance Manager deployment architecture	22
2	Stages in an EMC Smarts deployment phase	23
3	Standard Service Assurance configuration	38
4	Basic hierarchical Service Assurance configuration	39
5	Hierarchical aggregation Service Assurance configuration	40
6	Hierarchical presentation Service Assurance configuration	41
7	Hierarchical Mesh Service Assurance configuration	42
8	Flow of event information in a basic Global Manager configuration	60
9	Recommended trap processing design	63
10	Example of a deployment with a proxy server	86
11	Global Manager queues information example	93
12	Global Manager Topology Synchronization Information Example	97
13	Global Manager threads information example	99
14	Global Manager client information example	100
15	Output of the Dumpstatistics option for tools	101
16	Output of the Dumpstatistics option for escalation	102
17	Service Assurance Manager deployment with one trap adapter and no trap exploder	117
18	Service Assurance Manager deployment with one trap adapter and one trap exploder	118
19	Service Assurance Manager deployment with two trap adapters and one trap exploder	119

TABLES

	Title	Page
1	Before you begin checklist	25
2	Service Assurance information checklist	33
3	Determining the required Service Assurance configuration	36
4	Typical hardware for the equipment tiers	43
5	Required disk space for the Service Assurance Manager Server	44
6	Required disk space for the Service Assurance Manager console	44
7	CPU, memory, data disk space requirements	45
8	Initial design checklist	53
9	Designing topology operations checklist	58
10	Planning for notifications checklist	66
11	Examples of EMC Smarts user categories	68
12	Client and user design checklist	73
13	Business Impact Manager design checklist	79
14	Parameters for Properties files	87
15	Before you begin checklist	108
16	Basic Service Assurance information checklist	109
17	Initial design checklist	110
18	Designing topology operations checklist	110
19	Planning for notifications checklist	111
20	EMC Smarts client and user design checklist	112
21	Business Impact Manager design checklist	113
22	Service Assurance Manager benchmarking hardware	116
23	One trap adapter, no trap exploder, ASL script disabled	120
24	One trap adapter, with trap exploder, ASL script disabled	121
25	One trap adapter, with trap exploder, ASL script enabled	121
26	One trap adapter, with trap exploder, ASL script enabled	121
27	Two trap adapters, with trap exploder, ASL script enabled	122
28	SAM 8.0 Syslog adapter using the --file option benchmarking results	123
29	SAM 8.0 Syslog adapter using the --tail option benchmarking results	123
30	Escalation policy benchmark results	124
31	Performance benchmarking hardware configuration	126
32	Time to log in to the SAM 8.0 server benchmarking hardware configuration	127
33	Logging into the SAM 8.0 server benchmark	128
34	Default Global Console heap size allocation benchmarking hardware configuration	128
35	Latency results for SAM-to-SAM notification processing	130
36	Latency results to complete an initial topology synchronization between an IP domain manager and a SAM domain manager	130
37	SAM 8.1 Memory benchmarking results	133

PREFACE

As part of an effort to improve its product lines, EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your EMC technical support professional if a product does not function properly or does not function as described in this document.

NOTICE

This document was accurate at publication time. Go to EMC Online Support (https://support.emc.com) to ensure that you are using the latest version of this document.

Purpose

The *EMC Smarts Service Assurance Manager Deployment Guide* describes how to deploy the EMC Smarts Service Assurance Manager and related components of the EMC Smarts Service Assurance Manager.

This guide is intended as a comprehensive resource for deployment tasks. In many cases, you will be referred to other EMC Smarts documents for specific procedures and configuration tasks, such as:

- *EMC Smarts Installation Guide for SAM, IP, ESM, MPLS, VoIP, and NPM Managers* — This document that accompanied your software product contains detailed hardware requirements, platform requirements, and installation procedures.
- *EMC Smarts Service Assurance Manager Configuration Guide* — This document contains information about the Global Manager Administration Console.
- *EMC Smarts System Administration Guide* — This document contains detailed administration procedures for the EMC Smarts deployment.

Audience

This guide is intended for the following audiences:

- Network and system administrators who aid in designing and maintaining EMC Smarts deployments
- Integrators and network consultants who aid in designing EMC Smarts deployments and then install, validate, and tune the deployments
- Systems engineers who design EMC Smarts deployments
- EMC Smarts Professional Services personnel who design, install, validate, and tune EMC Smarts deployments
- EMC Smarts Global Services personnel who respond to inquiries, problems, and issues that arise during EMC Smarts deployments

EMC Smarts Service Assurance Manager installation directory

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

- For Windows, this location is: C:\InCharge\‹*product*›.

On Windows operating systems, this product is, by default, installed to: C:\InCharge\SAM\smarts. This location is referred to as *BASEDIR*/smarts.

Optionally, you can specify the root of BASEDIR to be something other than C:\InCharge (on Windows), but you cannot change the ‹*product*› location under the root directory.

The *EMC Smarts System Administration Guide* provides more information about the directory structure of EMC Smarts software.

EMC Smarts Service Assurance Manager

The EMC Smarts Service Assurance Manager includes the following products:

- Service Assurance Manager (Global Manager), includes Business Impact Manager (BIM) and Failover System
- Global Console
- Business Dashboard
- Companion UI (CUI)
- SAM Native Adapters:
  - Service Assurance Manager Notification Adapters (E-Mail Notifier Adapter, Script Notifier Adapter, SNMP Trap Notifier Adapter, Log File Notifier Adapter)
  - Adapter Platform (Adapter Platform server, Syslog Adapter, SNMP Trap Adapter, sm_ems command-line interface)
  - EMC Smarts Adapter for UIM/O
  - XML Adapter

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:

- *Smarts Foundation EMC Data Access API (EDAA) Programmer Guide*
- *EMC Smarts Common Information Model (ICIM) 1.11 Reference for Non-Service Assurance Manager Products*
- *EMC Smarts Common Information Model (ICIM) Reference for Service Assurance Manager*
- *EMC Smarts Foundation Documentation Portfolio*
- *EMC Smarts System Administration Guide*
- *EMC Smarts ASL Reference Guide*
- *EMC Smarts Perl Reference Guide*
- *EMC Smarts Dynamic Modeling Tutorial*
- *EMC Smarts MODEL Reference Guide*
- *FlexNet Publisher License Administration Guide*

EMC Smarts Service Assurance Manager documentation

The following documents are relevant to users of the EMC Smarts Service Assurance Manager:

- *EMC Smarts Service Assurance Manager Release Notes*
- *EMC Smarts Open Source License and Copyright Information*
- *EMC Smarts Service Assurance Manager Introduction*
- *EMC Smarts Installation Guide for SAM, IP, ESM, MPLS, VoIP, and NPM Managers*
- *EMC Smarts Service Assurance Manager Deployment Guide*
- *EMC Smarts Service Assurance Manager Configuration Guide*
- *EMC Smarts Service Assurance Manager Operator Guide*
- *EMC Smarts Service Assurance Manager Dashboard Configuration Guide*
- *EMC Smarts Business Impact Manager User Guide*
- *EMC Smarts Failover System User Guide*
- *EMC Smarts Service Assurance Manager Notification Adapters User Guide*
- *EMC Smarts Service Assurance Manager Adapter Platform User Guide*
- *EMC Smarts Adapter for Unified Infrastructure Manager/Operations Configuration Guide*
- *EMC Smarts XML Adapter User Guide*
- *EMC Smarts Notification Module User Guide*
- *EMC Smarts Service Assurance Manager Troubleshooting Guide*
- *EMC Smarts Companion UI Online Help*

- *EMC Smarts Companion UI Open Source License and Copyright Information for GPLv3 as Included with a Distribution of SLES 11*
- *EMC Smarts Service Assurance Manager Documentation Portfolio*

The *EMC Smarts Documentation Catalog* provides documentation resources for other EMC Smarts products.

Suggestions for searching PDF files

You can search across multiple PDF files by using the Adobe Acrobat Reader software:

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

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

Conventions used in this document

EMC uses the following convention for special notices:

NOTICE

NOTICE is used to address practices not related to personal injury.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities • URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages Used in procedures for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • What the user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis, for example, a new term • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for specific user input, such as commands
`Courier italic`	Used in procedures for: • Variables on the command line • User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections — the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

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 appears for Windows and an open circle appears for UNIX.

Manager

Unless otherwise specified, the term 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 information about EMC products, go to EMC Online Support at:

https://support.emc.com

Technical support — Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to:

`techpubcomments@emc.com`

CHAPTER 1
Overview of the Service Assurance Deployment Process

This chapter provides an overview of the deployment process for the Service Assurance portions of the EMC Smarts architecture and contains the following information:

- EMC Smarts architecture 20
- Deployment process 22
- Before you begin checklist 25

EMC Smarts architecture

The EMC® Smarts® is a fundamental part of all EMC Smarts deployments. As shown in Figure 1 on page 22, it consists of the following components and relationships:

- The Global Manager (Service Assurance Manager) is a manager of managers that provides end-to-end views of the status of your managed IT environment. The Global Manager consolidates and abstracts topology, analysis, and events from various sources, including EMC Smarts Domain Managers such as EMC Smarts IP Availability Manager and EMC Smarts IP Performance Manager. In addition, the Global Manager serves as an integration point for the EMC Smarts products.
- The EMC Smarts Business Impact Manager allows you to create and relate business-level objects, such as departments, services, and customers, to applications and network infrastructure. The Business Impact Manager can then determine the business impact of infrastructure failures on business services and their users. The Business Impact Manager is installed as part of the Global Manager. The Business Impact Manager has undergone an architectural change, that is, the analysis has been separated into separate processes.
- The EMC Smarts Global Console is a user interface that provides information about EMC Smarts analysis: root-cause failures, notifications, topology, and even business views. The Global Console also provides configuration and administration controls. This functionality can also be provided to a web browser by using a Java applet as the EMC Smarts Web Console. Both the Web Console and Global Console connect directly to the Global Manager.

  NOTICE

  The Global Console, Web Console and Business Dashboard are installed from a separate CD.

- The Business Dashboard is a flexible, business-oriented alternative to the Web Console. With the Business Dashboard, a collection of EMC Smarts analysis data can be displayed with important data from other sources on an organization's web page. Each component of EMC Smarts data is referred to as a Dashboard viewlet, which connects directly to the Global Manager.

- EMC Smarts Service Assurance Adapters (SAM Adapters), EMC Smarts Service Assurance Manager Adapter Platform (Adapter Platform) and EMC Smarts Adapter for Unified Infrastructure Manager/Operations (EMC Smarts Adapter for UIM/O) provide EMC Smarts software with additional sources of data when monitoring a network. The Adapter Platform normalizes and consolidates data from adapters before it is passed to the Service Assurance Manager. This guide covers deployment of the Adapter Platform and specifically describes two adapters that send data to the Adapter Platform: the EMC Smarts SNMP Trap Adapter and the EMC Smarts Syslog Adapter. The *EMC Smarts Adapter for Unified Infrastructure Manager/Operations Configuration Guide* provides deployment information, configuration instructions, and client tool examples on the EMC Smarts Adapter for UIM/O.
- EMC Smarts Companion UI (Companion UI) supplements existing functionality in SAM, EMC Smarts Network Configuration Manager (NCM) and Watch4net Automated Performance Grapher (APG). The Companion UI is a VMware®-based appliance, that is, a virtual machine. In a solution, one Companion UI is deployed for one presentation SAM server. The *EMC Smarts SolutionPack Installation and Configuration Guide* provides detailed information about installing and configuring the Companion UI.

Products such as the IP Manager and the Network Protocol Manager include Domain Managers that discover or monitor objects for a particular domain.

The Domain Managers use Codebook Correlation Technology™ to perform root-cause analysis for the related domain. These domains forward topology and events to the Global Manager, which in turn provides this information in a consolidated and abstracted manner to the Global Console.

Design and deployment of the underlying Domain Managers is not covered in this guide. Individual product deployment guides provide information on deploying underlying Domain Managers.

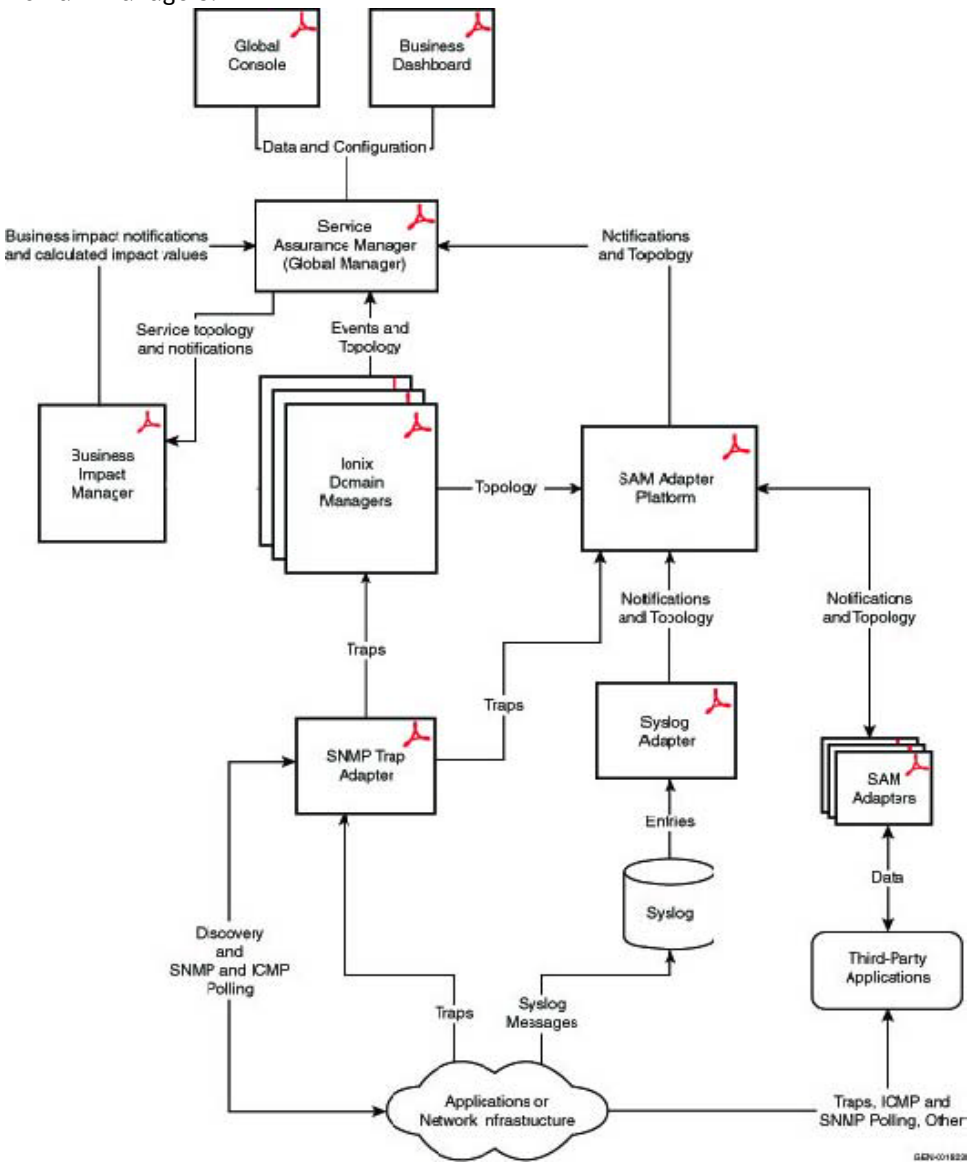

Figure 1 EMC Smarts Service Assurance Manager deployment architecture

Deployment process

EMC Smarts software is deployed in phases. In the first phase, the domains that feed the EMC Smarts Service Assurance Manager are deployed. If there are many types of underlying EMC Smarts domain, each type can be deployed in a separate phase. The sequence of these initial phases is determined by the priorities of the organization and the complexity of each domain.

An EMC Smarts Service Assurance Manager is typically deployed in the initial phases with these underlying EMC Smarts domains, but only in a simple, standard configuration. This deployment is intended to only validate the operation of the EMC Smarts domains and is described in the deployment guide for that domain.

After the initial phases, the production design of the EMC Smarts Service Assurance Manager configuration is deployed. This deployment phase is covered in this guide and includes the design, installation, and configuration of EMC Smarts Service Assurance Manager. The *EMC Smarts Installation Guide for SAM, IP, ESM, MPLS, VoIP, and NPM Managers* and the *EMC Smarts Service Assurance Manager Configuration Guide* provide specific installation and configuration details.

As shown in Figure 2 on page 23, each phase in the deployment process can be further divided into four distinct stages:

- Stage 1: Designing the deployment
- Stage 2: Installing and configuring the components
- Stage 3: Validating the deployment
- Stage 4: Tuning and maintaining the deployment to improve performance

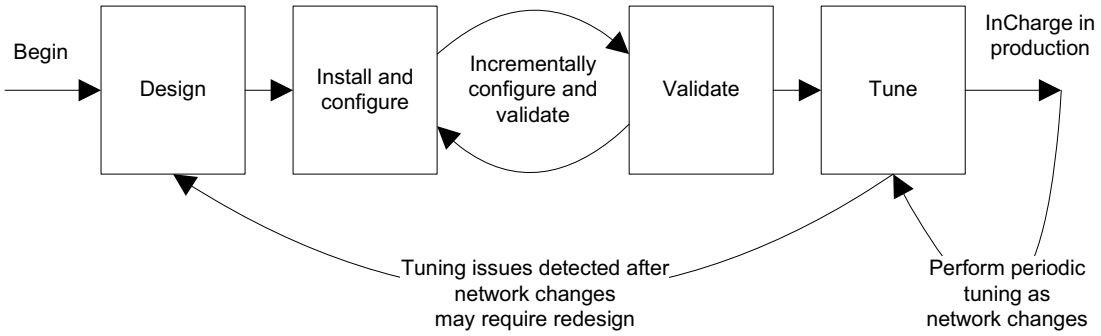

Figure 2 Stages in an EMC Smarts deployment phase

During each stage of deployment, it is vitally important that you document all aspects of the deployment that could be required to recreate, troubleshoot, and reconfigure the installation. "Document the deployment" on page 28 provides information on creating a document for this information.

Stage 1: Designing the deployment

Designing an EMC Smarts deployment consists of gathering necessary information and then using the information to develop a plan for the deployment.

Gathering the information is a process that involves both the designer and the network and application administrators. The administrators must provide details of their organization's network and application infrastructure. Chapter 2, "Gathering Basic Information," describes this process.

This information is then applied to develop a deployment design that will properly support the infrastructure. The design should also account for projected growth of the network and applications.

While the deployment is being designed, all plans, decisions, and supporting information must be documented. Typically, this documentation takes the form of a solution architecture diagram and a deployment build guide. In their preliminary stages, this diagram and guide may be used in the response to an organization's request for proposal (RFP) or request for quote (RFQ).

After a deployment is contracted, the information in the solution architecture diagram and deployment build guide can be refined and verified.

Design is covered in the following chapters:

- Chapter 3, "Designing the Service Assurance Deployment,"
- Chapter 4, "Designing for Topology Operations,"
- Chapter 5, "Planning for Notifications,"
- Chapter 6, "Designing for EMC Smarts Users,"
- Chapter 7, "Deploying Business Impact Processing," (optional)

Normally, this design process is performed by EMC Smarts Professional Services or an EMC Smarts partner with extensive input from the organization's IT staff.

Stage 2: Installing and configuring the components

After the design is complete, the next stage is installing and configuring EMC Smarts components. This stage is usually broken down into deployment phases where related components are installed and configured together. Each installed and configured phase is validated individually as described in the next stage of the deployment process. This phased approach eases troubleshooting.

Many organizations have specific procedural requirements that must be met before and during installation of new software products in their production environments. These requirements might include lab installations with performance validations and pre-production deployments. Lab configurations usually require the use of a testbed that is configured and equipped similarly to the production environment. Before deployment in the production environment, acceptance tests may be performed. After the production deployment, the lab or testbed may be used to test upgrades and, if required, patches.

Though this guide cannot cover organization-specific requirements, it does provide guidelines that may aid you in responding to these conditions.

Chapter 8, "Deploying Service Assurance," provides information on installing and configuring the software. The deployment information, including software locations and all configuration choices, should be recorded in the deployment build guide. Normally, this stage is performed by the purchaser of the deployment and either EMC Smarts Professional Services or an EMC Smarts partner.

Stage 3: Validating the deployment

Validating the deployment ensures that all installed components are operational and can communicate with each other as required and that the appropriate components can properly discover and poll the network.

Logical segments of EMC Smarts software are usually installed and validated to ease troubleshooting. After all individual segments are installed and validated, the complete deployment must be validated from end to end. Included in this overall validation could be acceptance tests that demonstrate the functionality of the installation. Criteria for acceptance tests should be defined during the design stage. The execution of these acceptance tests and the results are usually included in an installation or build report.

Chapter 9, "Validating Your Deployment (Acceptance Testing)," provides information on validating components. Normally, this stage is performed by the purchaser of the deployment and either EMC Smarts Professional Services or an EMC Smarts partner.

Stage 4: Monitoring and tuning the deployment

After EMC Smarts software is deployed and validated, you must ensure that it is operating at an optimal level. Tuning is the process of adjusting the configuration to improve performance; it should be performed after all components are installed and validated.

Chapter 10, "Monitoring the Performance of Your Deployment," describes the process of monitoring a deployment for tuning. Normally, this stage is initially performed by either EMC Smarts Professional Services or a partner and the knowledge is transferred to the purchasing organization's staff. As the network and application infrastructure grows, the organization's staff will take on the task of tuning the deployment to handle this growth. If the infrastructure growth is extensive, the original design may no longer be sufficient and redesigning the deployment may be required.

Throughout the life of the deployment, administrators should repeat this stage to maintain the deployment.

Before you begin checklist

Before you begin an EMC Smarts deployment, you must meet the requirements described in Table 1 on page 25. Each chapter in this guide includes a checklist. Appendix A, "Design and Deployment Checklists," contains all available checklists.

Table 1 Before you begin checklist

Complete	Requirement	Description
❑	Possess an understanding of the architecture and capabilities.	At a minimum, you must understand the concepts and architecture described in the following documents: • *EMC Smarts Service Assurance Manager Introduction* • *EMC Smarts Service Assurance Manager Configuration Guide* • *EMC Smarts Service Assurance Manager Adapter Platform User Guide* • *EMC Smarts System Administration Guide* • *EMC Smarts Installation Guide for SAM, IP, ESM, MPLS, VoIP, and NPM Managers* that accompanied your product • EMC Smarts user guides related to the underlying domains in your deployment
		To improve your understanding, attend training courses offered by EMC Education Services and Development. Typically, deployment requires the knowledge equivalent to what is provided in the training courses on: • Global Manager • EMC Smarts Service Assurance Manager Adapter Platform • Any underlying EMC Smarts domain that will be part of your deployment.

Table 1 Before you begin checklist (continued)

Complete	Requirement	Description
❑	Obtain contact information for the deployment team.	The contact list should include titles, responsibilities, and contact methods for all team members.
❑	Get non-disclosure requirements and negotiate an agreement.	Be aware of the requirements of the non-disclosure agreements that are in place for the deployment.
❑	Develop schedules and set milestones for early deliverable.	Scheduling a software deployment varies based on the size and scope of the deployment and the organization's requirements. Typical milestones might include: • Initial project meeting to define the deployment scope • Purchase of EMC Smarts software • Project development begins • Installation in test environment complete • Testing complete • Installation in production environment complete • EMC Smarts goes live Additional information on scheduling is beyond the scope of this guide.

CHAPTER 2
Gathering Basic Information

This chapter describes the first step in designing a Service Assurance deployment, that is, gathering the organizational and network-related information that is required to develop a successful design. This chapter also describes how to document the information that you gather and your resulting design in a solution architecture diagram and a deployment build guide. This chapter contains the following information:

- Document the deployment 28
- Gather information required to size Service Assurance 29
- Determine requirements for installing software 30
- Gather network security information 31
- Other network features affecting EMC Smarts 32
- Basic Service Assurance information checklist 33

Document the deployment

The most useful way to document the managed environment and the design of your deployment is to create a solution architecture diagram and record implementation details in a deployment build guide.

The solution architecture diagram

Based on the complexity of the deployment, a solution architecture diagram may actually be a set of diagrams documenting various levels of the architecture.

The diagram relates the information that you gather on the environment to both physical and logical choices for your architecture in an easily understood manner. This diagram illustrates your design choices and will be an important part of the review and approval process for your design.

The solution architecture diagram should always include:

- Necessary information about the managed network and application environment
- A logical representation of the EMC Smarts components that will be installed
- Locations for each component including the name and IP address of the host and the geographical location of the host
- Connections between components and the ports that are used for communications
- Connections, including port numbers, between EMC Smarts components and external sources such as networks and third-party software products

This diagram cannot be completed until the design is complete and deployed, so recommendations for adding information to the diagram appear throughout this guide.

The deployment build guide

To record the specifics of the managed environment and the deployment design and implementation, create a document called a deployment build guide. As with the solution architecture diagram, this chapter describes information that you should add to the deployment build guide. The deployment build guide should include the complete design and all installation specifications, validation results, and tuning activities.

Start the deployment build guide by recording all the information that you gather on the network, some of which may come from your planning of IP Availability Manager or IP Performance Manager installations. As you continue the deployment process, this guide will include recommendations for adding other information to the deployment build guide.

Gather information required to size Service Assurance

To determine the size and configuration of Service Assurance, including the number of Global Managers and the equipment required to support them, gather the following information:

- The number of devices and applications to manage
- The number of Service Assurance clients
- The number of underlying managed domains

Determine how many devices and applications will be managed

One of the most important factors in determining the size of a Service Assurance deployment is quantity of devices and applications that must be managed. Estimate all managed devices in a network once. Include the following types of devices:

- All routers, including virtual routers
- All switches
- Hosts that will be managed, including virtual machines
- Any other devices that will be managed

All managed applications should also be counted: if an application consists of component processes, count each process.

EMC Corporation has determined that even in well-managed large networks, administrators can underestimate the quantities of devices and applications that they intend to manage.

If possible, validate the estimated quantities for some portion of the network. If the estimate does not match, try to obtain an improved estimate for another part of the network and then validate again. If improving the estimates is not possible, determine the percentage error and use it to adjust all the other estimates.

As a final step in estimating the quantities of managed devices and applications, account for growth of the organization's network and for planned application deployments.

Your Service Assurance deployment design should account for network growth over the expected life of the design. The network growth rate will relate to the vertical market environment and the organization's plans, so the organization must provide you with growth estimates.

Information on the future deployment of business applications may be more difficult to obtain. For example, an organization's budgetary plans may provide useful information, but may not be available to you. Interviews with IT staff may be your best source of information.

Estimate quantities and types of Service Assurance clients

The number of Service Assurance clients will affect Global Manager configuration and equipment choices. The clients can include any of the following:

- Users of the Global Console, Web Console, and Business Impact Manager.
- EMC Smarts Adapters that connect to the Global Manager directly, including the Service Assurance Manager Notification Adapters: SNMP Trap Notifier Adapter, Script Notifier Adapter, and Log File Notifier Adapter.

Estimate the number of underlying EMC Smarts domains

Estimating the number of underlying domains requires that you size these domains following EMC Smarts Professional Services recommendations or by following the process described in the underlying domain's deployment guide, such as the *EMC Smarts IP Manager Deployment Guide*.

Determine requirements for installing software

Most organizations define some criteria for installing new software on their network. At a minimum, this might include software testing requirements. In addition, plan on using a staging area before deploying. By staging the deployment, you can maintain a clean software distribution that does not include unwanted files, changes, and logs.

Define requirements for integrating existing software with EMC Smarts

Integrating third-party software products with EMC Smarts software requires the use of EMC Smarts adapters. Most integrations do one of the following:

- Exchange information. Determine the exact type of information (topology or events) that must be exchanged with any third-party software products.
- Allow access to the third-party software from the Global Console. Some EMC Smarts adapters provide this capability as the Server Tools functionality. For example, both the EMC Smarts Adapter for InfoVista and EMC Smarts Adapter for Concord eHealth allow users at the Global Console to access reports from these products.

EMC Corporation has developed many adapters for use with EMC Smarts software that may already support the integration requirements of your deployment. Itemize the integration requirements so that the suitability of existing EMC Smarts adapters can be assessed by EMC Smarts Professional Services. This guide specifically covers the integration capabilities provided by the EMC Smarts SNMP Trap Adapter (Receiver). This guide also addresses the basic information for other adapters.

Gather network security information

Determine the level of security for the network that EMC Smarts software will monitor so that the software can be configured to a corresponding level of security. For example, the security needs of a network in a financial, defense, or health care vertical market may be greater than in the manufacturing vertical market. Enumerate security preferences, such as the use of passwords, encrypted password storage, and encrypted communications to guide you when configuring security capabilities.

Firewalls between parts of the EMC Smarts deployment

Appropriate Service Assurance components must be able to receive traps and communicate with other EMC Smarts components. Certain TCP and UDP ports will need to be opened in the firewalls to facilitate these communications. "Considering security and firewalls" on page 51 provides additional information.

Other network features affecting EMC Smarts

Other network features may affect the EMC Smarts deployment design. Consider the following questions when gathering information on the network:

- Does the organization require failover capabilities in network software?
- Is there an “out-of-band” network just for management information? For example, are certain Ethernet ports just for management information? This is typical with some deployments in the financial and military or defense vertical markets.
- What are the issues related to network latency, bandwidth, and speed available for network management traffic?

Basic Service Assurance information checklist

Use the checklist shown in Table 2 on page 33 to aid in gathering information for your Service Assurance design. Each chapter in this guide includes a checklist. "Design and Deployment Checklists" on page 107 contains all available checklists.

Table 2 Service Assurance information checklist

Complete	Task	Description	Related documentation
❑	Determine the number of managed devices and applications in the environment.	Document all quantities in the deployment build guide.	"Determine how many devices and applications will be managed" on page 29
❑	Estimate potential growth in quantity of managed devices.	The deployment must support potential network growth. Estimate the growth over a specific time period. Document the calculations in the deployment build guide.	"Determine how many devices and applications will be managed" on page 29
❑	Get the organization's testing/acceptance requirements.	Your design may be required to meet test and acceptance requirements. Obtain any specifications that cover integration testing, user acceptance testing, and operational acceptance testing. You may be required to write an installation or deployment report that follows an organization's particular standards.	"Determine requirements for installing software" on page 30
❑	Describe the organization's requirements for installing new software.	Document these requirements and how the design meets them in the deployment build guide.	"Determine requirements for installing software" on page 30
❑	List the products that currently monitor the network and will be integrated with the EMC Smarts deployment.	Document the products (including version) in the deployment build guide.	"Define requirements for integrating existing software with EMC Smarts" on page 30
❑	Describe the network security.	Describe security features that will affect the deployment. Document the security features in the deployment build guide.	"Gather network security information" on page 31
❑	List any other network requirements or features that may affect the EMC Smarts deployment.	Document the features in the deployment build guide.	"Other network features affecting EMC Smarts" on page 32

CHAPTER 3
Designing the Service Assurance Deployment

This chapter provides guidelines for producing an initial design of the Service Assurance deployment by using the information you have gathered on the organization's application and network environment. This chapter contains the following information:

- Determine the Service Assurance configuration 36
- Web Console and Business Dashboard requirements 45
- Essential EMC Smarts applications 48
- Locating the Global Managers in the network 50
- Considering security and firewalls 51
- Considering failover (high availability) configurations 51
- Designing acceptance tests 52
- Initial design checklist 53

Determine the Service Assurance configuration

A single Global Manager can support most Service Assurance deployments, but there are cases where multiple Global Managers are required. Use Table 3 on page 36 to determine the Service Assurance configuration required to support your managed environment. The first three columns in Table 3 on page 36 identify the quantities that will determine the Service Assurance configuration:

- The Managed devices and applications column details the total quantity that will be managed in the Service Assurance deployment. This number should include estimates for growth.
- The Underlying domains column details the total number of domains that will feed topology and events to the Service Assurance deployment.
- The Concurrently connected clients column details the total number of Service Assurance clients that will be connected concurrently, including Global Consoles, Web Consoles, Dashboard viewlets, and any adapters that connect to the Global Manager directly.

Table 3 Determining the required Service Assurance configuration

Using the quantities for the managed environment			Use this Service Assurance configuration		
				Global Manager platform equipment tier	
Managed devices and applications	Underlying domains	Concurrently connected clients	Global Manager configuration	Aggregation layer (if used)	Presentation layer (if used)
0 to 2,000 (small deployment)	≤ 10	≤100	Standard	Small	
		›100	Basic Hierarchical	Small	Medium
	⩾ 10	≤ 100	Standard	Medium	
		›100	Basic hierarchical	Medium	Medium
2,001 to 6,000 (medium deployment)	≤ 10	≤ 100	Standard	Small	
		›100	Basic hierarchical	Small	Medium
	⩾10	≤ 100	Standard	Medium	
		›100	Basic hierarchical	Medium	Medium
6,001 to 12,000 (large deployment)	≤ 10	≤ 100	Basic hierarchical	Medium	Medium
		›100	Hierarchical presentation	Large	All medium
	⩾10	≤ 100	Hierarchical aggregation	All medium	Large
		›100	Hierarchical Mesh	All large	All Large

Table 3 Determining the required Service Assurance configuration (continued)

Using the quantities for the managed environment			Use this Service Assurance configuration		
Managed devices and applications	Underlying domains	Concurrently connected clients	Global Manager configuration	Global Manager platform equipment tier	
				Aggregation layer (if used)	Presentation layer (if used)
12,001 to 20,000 (extra large deployment)	≤ 10	≤ 100	Basic hierarchical	Large	Large
		›100	Hierarchical presentation	Extra large	All large
	⇧10	≤ 100	Hierarchical aggregation	All large	Extra large
		›100	Hierarchical mesh	All extra large	All extra large
Above 20,000 devices			EMC Smarts deployments in this range are currently in production. Consult with EMC Smarts Professional Services for aid in this design.		

The columns in the Use this Service Assurance configuration section define recommendations to support the deployment:

- The Global Manager configuration column details the recommendation for the number of Global Managers, either a single one (Standard) or multiple (any of the hierarchical configurations). If a hierarchical configuration is recommended, the specific structure is also defined. The Global Manager configurations are explained in the following sections.
- The Global Manager platform equipment tier columns define the hardware that is required to support the Global Managers. For hierarchical configurations, two layers of Global Managers are defined: an aggregation layer and a presentation layer. Domain Managers connect to a Global Manager at the aggregation layer while Service Assurance clients connect to a Global Manager at the presentation layer. The table indicates the equipment required to support each layer. "Hardware for Service Assurance platform equipment tiers" on page 43 provides definitions for the equipment tiers.

Table 3 on page 36 defines Service Assurance configurations for coherent deployments. It is possible to create separate deployments in a managed environment that do not feed events and topology to a single, central Global Manager. If this is the case, each of these deployments must be treated separately with distinct Global Manager configurations based on unique quantities for first three columns in Table 3 on page 36.

Standard Global Manager configuration

A single Global Manager, as shown in Figure 3 on page 38, should be able to handle many typical deployments. This is called the Standard Configuration. In this configuration, there are no links that introduce high latency between the main EMC Smarts components.

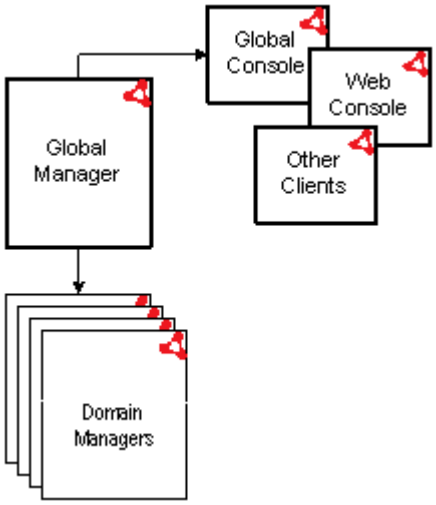

Figure 3 Standard Service Assurance configuration

Basic hierarchical Global Manager configuration

Deployments with either large numbers of clients or large numbers of domains can usually be supported using the basic hierarchical configuration shown in Figure 4 on page 39. This hierarchical configuration separates the topology synchronization processing (aggregation) from the client processing (presentation) to improve performance.

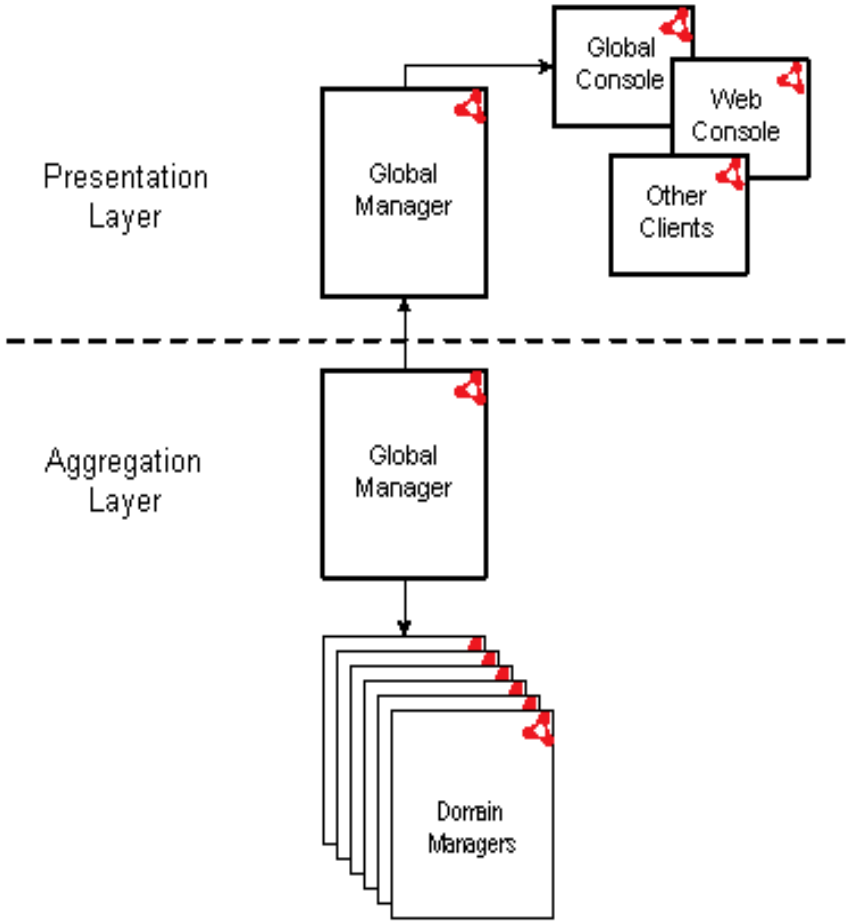

Figure 4 Basic hierarchical Service Assurance configuration

Hierarchical aggregation Global Manager configuration

The hierarchical aggregation configuration shown in Figure 5 on page 40 uses multiple Global Managers at the aggregation layer to address these configurations:

- More than 10 underlying domains: It is recommended that each Global Manager at the aggregation layer supports a maximum of 10 EMC Smarts domains. The example in Figure 5 on page 40, therefore, would support 20 domains.
- Geographically separate groups of domains: Using a standard Global Manager configuration would force some domains to connect to a Global Manager over a high-latency link. Additional Global Managers at the aggregation layer would be located in the same geographical area as the domains.

To ensure that acknowledgements and ownership of notifications are properly propagated, Service Assurance clients should connect only to the Global Manager at the presentation layer.

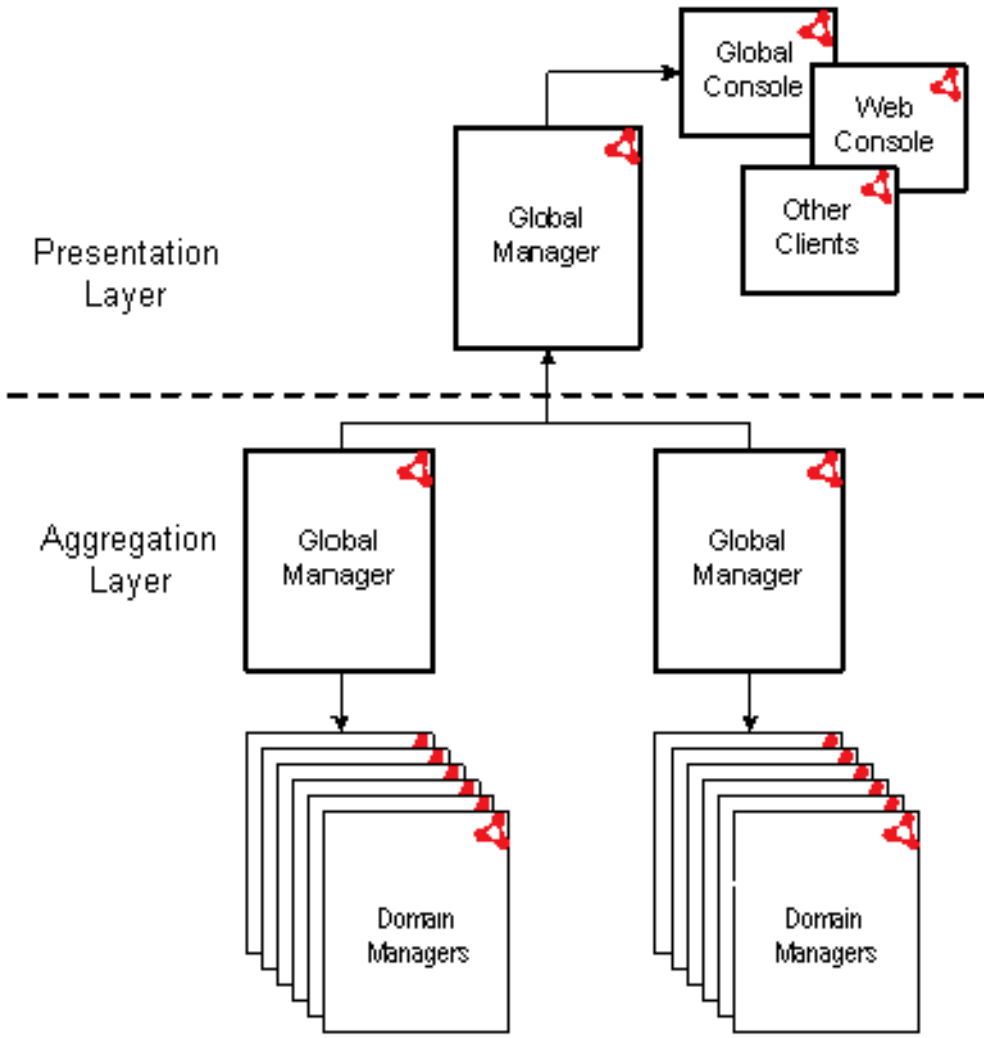

Figure 5 Hierarchical aggregation Service Assurance configuration

Hierarchical presentation Global Manager configuration

The hierarchical presentation configuration shown in Figure 6 on page 41 uses multiple Global Managers at the presentation layer to address these configurations:

- More than 100 Service Assurance clients that connect concurrently: This includes Global Consoles, Web Consoles, Dashboard viewlets, and any adapters that connect to the Global Manager directly. It is recommended that each Global Manager at the presentation layer supports up to 100 concurrently connected clients. The example in Figure 6 on page 41, therefore, would support 200 clients.
- Geographically separate groups of clients: In these cases, using a standard Global Manager configuration would force a group of clients to connect to a Global Manager over a high-latency link. Additional Global Managers at the presentation layer would be located in the same geographical area as the groups of clients.

To ensure that acknowledgements and ownership of notifications are properly propagated, some customization may be required.

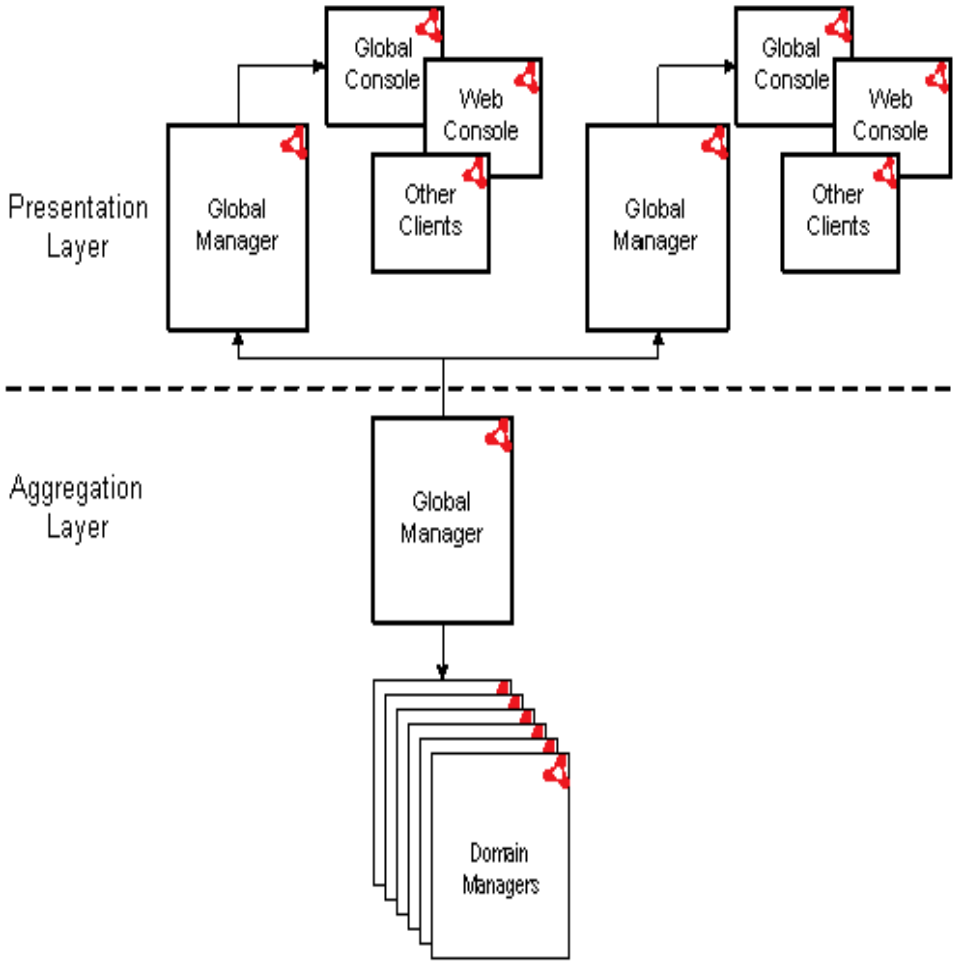

Figure 6 Hierarchical presentation Service Assurance configuration

Hierarchical mesh Global Manager configuration

The hierarchical mesh configuration shown in Figure 7 on page 42 is used to support the largest, most complex deployments that require multiple Presentation Global Managers and multiple Aggregation Global Managers.

As with the other hierarchical configurations, it is recommended that each additional Aggregation Global Manager should support up to 10 underlying domains and each additional Presentation Global Manager should support up to 100 simultaneously connected clients.

Deploying the hierarchical mesh configuration requires the aid of EMC Smarts Professional Services.

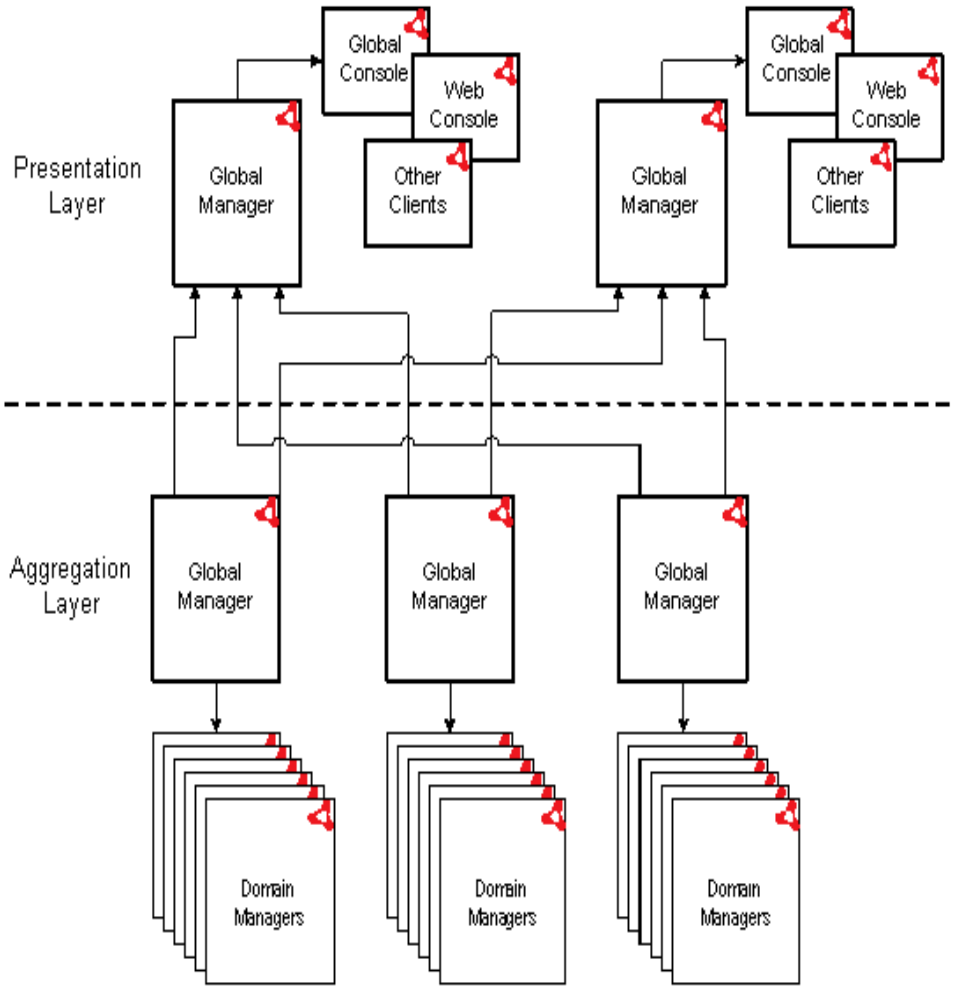

Figure 7 Hierarchical Mesh Service Assurance configuration

Hardware for Service Assurance platform equipment tiers

Typical hardware for the equipment tiers and operating systems is listed in Table 4 on page 43.

Table 4 Typical hardware for the equipment tiers

Operating system	Platform equipment tier			
	Small (1–2 low end CPUs, 2 GB RAM)	Medium (2 CPUs, 4 GB RAM)	Large (2 CPUs, 4 GB RAM)	Extra large (4 CPUs, 4–8 GB RAM)
Red Hat Enterprise Linux	Any vendor, 1-2 Xeon 2.8 GHz	Any vendor, 2 Xeon 2.8 GHz	Any vendor, 2 Xeon 2.8 GHz	Any vendor, 4 Xeon 3.0 GHz
Solaris	UltraSPARC-T1 (T2000)	Ultra-SPARC-T2 (T5220)	SPARC64 VII (M4000)	SPARC64 VII (M4000)

It is recommended that you work with Sun Microsystems, Inc. to determine models that meet or exceed these requirements in addressing your business needs.

NOTICE

Xeon processors vary a lot across models. The Xeon 3GHz 5160 is rated at about 16 at SpecInt for speed while the 2.93GHz Xeon X5670 is rated at about 37 for speed. If you require a specific speed processor, EMC recommends using the SpecInt (SPEC Integer) benchmark published by SPEC both for assessing the CPU speed and throughput ratings. SPEC is an organization of computer industry vendors dedicated to developing standardized benchmarks and publishing reviewed results.

CPUs may have multiple cores and cores may have multiple hardware threads. Cores and threads may share CPU resources with other cores and threads. This makes assessment of CPU capacity difficult. EMC recommends using the benchmark published by SPEC for assessing expected relative CPU performance. CPU2006 is the current version of the CPU component benchmark suite from SPEC.

The results are broken out by reported metric for:

- **CPU Speed:** You can access this benchmark from:

http://www.spec.org/cpu2006/results/cint2006.html

- **CPU Throughput:** You can access this benchmark from:

http://www.spec.org/cpu2006/results/rint2006.html

While cint runs one copy of the benchmark, rint runs as many copies as there are threads in the machine.

CPU speed and the number of CPUs available affect performance. You can get a sense of the expected performance by comparing the CPU speed directly to the speed ratings of the CPUs on which the benchmark is based. The current strategy for assessing the number of effective CPUs in a machine is to divide the throughput rating by the speed rating. In general, our software benefits more from faster CPU speed, than the comparable addition of more CPUs. Thus, it is preferred to have 1 CPU rated at 20, than 2 rated at 10.

If you are installing more than one product, then your system must meet the products total memory and data disk space requirements.

Table 5 on page 44 lists disk space requirements for installing the Service Assurance Manager Server.

Table 5 **Required disk space for the Service Assurance Manager Server**

Platform	Disk space
Solaris 10	730 MB
Red Hat Enterprise Linux AS 5	475 MB
Windows Server 2008 R2	2.7 GB

Table 6 on page 44 lists disk space requirements for installing the Service Assurance Manager console.

Table 6 **Required disk space for the Service Assurance Manager console**

Platform	Disk space
Solaris 10	485 MB
Red Hat Enterprise Linux AS 5	365 MB
Windows Server 2008 R2, Windows Vista, and Windows XP	1.6 GB

Table 7 on page 45 lists minimum requirements for CPUs, memory, and data disk space that is used by the software for writable files such as logs, repository files, and output files.

Table 7 CPU, memory, data disk space requirements

EMC Smarts product	CPUs	Memory (RAM)	Data disk space
Service Assurance Manager	2	512 MB	100 MB each
Global Console	1	256 MB	Not applicable
Global Console, Web Console, and Business Dashboard	1	256 MB	150 MB
EMC Smarts Service Assurance Manager Adapter Platform	1	512 MB	100 MB each
Syslog Adapter	1	256 MB	50 MB each
SNMP Trap Adapter	1	256 MB	50 MB each
EMC Smarts Adapter for UIM/O	1	256 MB	50 MB
XML Adapter	1	256 MB	See the following Note.

NOTICE

CPU means the number of physical CPU cores and not the number of CPUs displayed when hyperthreading is activated.
For the XML Adapter, disk space depends on the size of the output that you want to export. Each output file can range from 0 to 200 MB. If, for example, the XML Adapter communicates with 5 Service Assurance Global Managers and each output file is 200 MB, then 1 gigabyte (GB) is required.

If you are installing multiple products, then your system must meet the products' total memory and data disk space requirements. Table 5 on page 44 defines required disk space and Table 7 on page 45 lists CPU, memory, and data disk space requirements.

Web Console and Business Dashboard requirements

The EMC Smarts Global Console can also be deployed as either:

- EMC Smarts Web Console

  or

- EMC Smarts Business Dashboard

NOTICE

Tomcat Servlet Engine version 6.0.30 or above is installed as part of the Web Console/Business Dashboard. The Tomcat Servlet Engine serves the files used by the Web applications.

Users who display Business Dashboard viewlets in a Web browser, or want to use the Web Console, require the following software:One of the following Web browsers:

- Internet Explorer 6.0 with Service Pack 1, 7.0, 8.0, 9.0
- Firefox 2, 3.5, 3.6, 4.0, and 5.0
- Google Chrome 12.0
- Safari 3.0
- Browser that supports Java JRE 1.6.x

NOTICE

Only browsers running on Linux, Solaris, and Microsoft Windows operating systems are supported.

- For Windows, the browser must be JavaScript enabled.

  Also, a list of specific versions of browsers that are supported on different platforms can be found in the Oracle website.

- Java 2 Platform Standard Edition 6.0, which includes the 1.6.x Web browser plug-in must be installed on the client workstation.

  If you are running any version prior to 1.6, when you try to start up and run the Dashboard or web console, the applets inside the sample views that are shipped with the software do not display properly.

  If your client workstation does not have this version of JRE, you must obtain it and install it:

  For all platforms, download it from the Sun Microsystems website.

  If the JRE is not installed on your client workstation, you will be prompted to install it when you launch the Web Console or Business Dashboard.

NOTICE

In SAM 9.2, Red Hat Enterprise Linux AS 5, Windows Server 2008 R2, and Solaris 10, uses JRE version 1.6.0.26.

Java Web Start requirements for Web Console

Java Web Start is optional and can be invoked when the Web Console is executed.

Java Web Start must be installed and configured on the client machine:

- For UNIX, contact your system administrator to install and configure Java Web Start or follow the instructions provided in the Sun Microsystems website.
- For Windows, on the client machine, depending upon the type of browser, after you launch the Web Console, use an automatic installer or follow the instructions provided in the Sun Microsystems website.

  To launch the Java Web Start version of the Web Console and use the automatic installer, point your browser to the following URL:

  `http://<Tomcat_host>:8080/webconsole/globalConsoleCached.html`

  where ‹*Tomcat_host*› refers to the name of the machine where the Tomcat Servlet Engine is installed and 8080 refers to the default port for Tomcat.

The *EMC Smarts Service Assurance Manager Dashboard Configuration Guide* provides information about using the Web Console.

NOTICE

When launching the Web Console from the browser (using the globalConsoleCached.html and JRE 1.6.13), an exception may be raised stating that the JAR resource in the JNLP file is not signed by the same certificate.
Launch the Web Console from http://‹servername›:8080/webconsole/ to download the correct (JRE 1.6_25) version.

Web browser requirements for the Smarts Adapter for UIM/O

The following web browsers are required for the Smarts Adapter for UIM/O:

- Internet Explorer 6.0 with Service Pack 1, 7.0, and 8.0
- Firefox 3.5, 3.6, 4.0, and 5.0

Only browsers running on Windows operating systems are supported. For Windows, the browser must be JavaScript enabled. The *EMC Smarts Adapter for Unified Infrastructure Manager/Operations Configuration Guide* describes a recommended configuration task for both browsers and Firefox addons.

In addition, Java Standard Edition 6.0, which includes the 1.6.x Web browser plug-in, must be installed on the client workstation. If your client workstation does not have the JRE 1.6, you must obtain it and install it. For all platforms, download it from the Oracle website, http://www.oracle.com.

NOTICE

The Deployment Guide may not be updated for patch or service pack releases. For the most up-to-date information about a release, refer to the Release Notes.

Deploying Global Managers and Domain Managers

Deploy the Global Manager separately from the Domain Managers. After deployments become large, run the Service Assurance Adapter Platform separately from the Global Manager. Avoid running any other applications on these platforms to ensure that you do not adversely affect the performance of the EMC Smarts applications.

Adding information to the solution architecture diagram and build guide

Include the Global Managers and related applications on the architecture solution diagram. Draw boxes to show the platforms and include the names, locations, and, when applicable, port numbers for all the EMC Smarts components.

For the deployment build guide, document equipment choices and create a table that documents EMC Smarts components on each host.

Essential EMC Smarts applications

Every EMC Smarts installation must include an EMC Smarts Broker and a license server. Both are critical components, without either, the installation will not work. Usually, you install these two components together, in a single location. All of the EMC Smarts applications must be able to communicate with both applications.

Locating EMC Smarts Brokers

The Broker manages a registry of EMC Smarts server applications. When an EMC Smarts application needs to connect with another application, it gets the necessary information from the Broker. Periodically, the Broker pings the applications in its registry to determine whether they are still active.

Server applications, such as the Global Manager, IP Availability Manager, and Adapter Platform, are maintained in the Broker's registry. Applications such as the Global Console connect to the Broker to determine the location of registered applications.

With most installations, the Broker is typically installed on the same machine as the license server. All EMC Smarts applications must be able to access the Broker. By default, the Broker uses port 426.

It is recommended that you use a DNS alias to a machine for the Broker so that it is easier to redirect other EMC Smarts applications if it becomes necessary to move.

Add the following information for the Broker to the deployment build guide:

- Name of host where the Broker will reside
- Port number to be used by the Broker (the default is 426.)
- Information for the DNS broker alias

Installations use more than one Broker for failover configurations. Multiple Brokers require special consideration to synchronize lists of registered servers. If you determine that your installation needs more than one Broker, consult with the EMC Smarts Professional Services organization.

Locating license servers

The license servers authenticate EMC Smarts applications. Without proper licensing, the applications will not work correctly. Licenses control access to applications and features as well as limit the number of systems discovered.

It is recommended that the license server runs on the same host as the Broker—all applications need to be able to access the license server. However, an existing FLEXnet License Server may be used if it meets the EMC Smarts specifications. If there is more than one Broker for the installation, there may be good reason to use more than one license server.

In certain cases, you may use multiple license servers in your network for the same reasons that you split a topology or choose different locations for the Domain Managers: geographical requirements, corporate organizational requirements, or the network's security design. Multiple license servers would require the corresponding number of licenses.

When multiple license servers are used, each domain will use blocks of licenses from a specific license server. Therefore, the license blocks must be divided among multiple licenses and the appropriate license must be deployed to each license server. Then, the Domain Managers that rely on a specific license server will have the appropriate licenses to support the systems that they must manage.

Collect the following information for each license server and add it to the deployment build guide:

- The host ID of the computer where the FLEXnet license server is running
- The expected installation path to BASEDIR/smarts/bin on your broker (or to the EMC Smarts installation from which you want to run the license server)
- The operating system of the computer
- The port number used

The *EMC Smarts System Administration Guide* provides additional information.

Locating the Global Managers in the network

In choosing locations for the platforms that support the Global Managers, there may be restrictions on locations that are not related to network and application efficiency. Locating the Global Managers may be based on the following considerations:

- Geographical requirements: An organization may require that all EMC Smarts Managers be based in a single Network Operations Center (NOC). Other organizations might have the Global Manager in the NOC and underlying Domain Managers located in regional data centers.
- Corporate organizational requirements: For example, organizations that have a distributed business management might require that components be located to match the organization's structure.

Communication between applications is influenced by network speed and latency. Avoid configurations that require a Global Manager to receive information from other applications across WAN links with high latency. Instead, consider placing Global Managers on both ends of high-latency WAN links. The hierarchical Global Manager configurations can be used to eliminate high-latency link issues for these situations:

- Domain Managers that connect to a Global Manager over a high-latency link: Use the basic hierarchical, hierarchical aggregation, or hierarchical mesh configuration to place a Global Manager on the same side of the high-latency link as the Domain Managers.
- A group of clients that connect to a Global Manager over a high-latency link: Use the basic hierarchical, hierarchical presentation, or hierarchical mesh configuration to place a Global Manager on the same side of the high-latency link as the group of clients.

After you determine the configuration and locations of the Global Managers, add system names and Global Manager names to your solution architecture diagram. Define IP addresses and dedicated port numbers when needed. Establish a host naming convention and a Global Manager naming convention before you settle on any names.

Considering security and firewalls

Based on the security information you obtained earlier, you must plan design solutions so that the applications can function properly in the network's security environment:

- Configure security policies (rules) to enable a one-way connection from the FLEXnet license server and the Broker to the various applications.

  For communication between Managers across firewalls, plan on opening a hole in the firewall for the EMC Smarts communications. Certain UDP and TCP ports must be opened for proper communications:

  - Broker: Port 426
  - License Manager: Port 1744
  - Manager: one port each, which can be configured
  - Adapters, including SNMP Trap Adapter and Syslog Adapter. The *EMC Smarts Service Assurance Manager Adapter Platform User Guide* provides details.

- Consider proxy servers when communicating with EMC Smarts applications that reside behind firewalls. Using a proxy server reduces the number of firewall ports that need to be opened to one firewall port. "Configuring the Java clients to use a proxy server" on page 86 provides additional information on the procedure to configure the Java clients to use a proxy server.

- If access lists are used, plan on deploying the IP addresses of hosts that include Managers to the access list of devices that will be managed. EMC Smarts applications must have full access to browse the MIBs of the devices. The *EMC Smarts IP Manager User Guide* lists the specific MIBs. Depending on the network size and complexity, this may require scheduling to obtain support from the organization's network personnel.

- You must have a listing of SNMP versions and related security parameter values that are used by specific devices in the organization's network. Due to security concerns, it may not be appropriate to include them in the deployment build guide.

Security levels and EMC Smarts applications

In addition, consider the level of security to configure EMC Smarts applications. The security mechanisms support various levels of user authentication, and also authentication and encrypted communication between applications. Ensure that you understand the capabilities described in the *EMC Smarts System Administration Guide* and then choose a level of security that is appropriate for the deployment.

Considering failover (high availability) configurations

EMC Smarts offers a failover product that works in conjunction with the Global Manager. You can find information about configuring a failover solution in the *EMC Smarts Failover System User Guide*. For additional help, consult with EMC Smarts Professional Services.

Designing acceptance tests

Acceptance tests may be required for different portions of EMC Smarts functionality. Be aware of the requirements and develop acceptance criteria with the aid of the network administrators and other organization personnel. Include all necessary acceptance tests in the deployment build guide for the deployment. At a minimum, you must develop completion criteria that the organization's project managers approve. Document these completion criteria in the deployment build guide and use them in validation.

Initial design checklist

Document your initial overall design of the deployment.

Each chapter in this guide includes a checklist. "Design and Deployment Checklists" on page 107 contains all the available checklists.

Table 8 Initial design checklist

Complete	Task	Description	Related documentation
❑	Determine the configuration of the Service Assurance deployment.	Determine Service Assurance configuration, including the number of Global Managers and how they are organized in the deployment.	"Determine the Service Assurance configuration" on page 36
❑	Determine the hardware required to support the Service Assurance deployment.	For each Global Manager in the Service Assurance configuration, determine the type of equipment required to support it.	"Hardware for Service Assurance platform equipment tiers" on page 43
❑	Determine the location of the EMC Smarts Broker.	Choose a location for the Broker that will support the deployment.	"Locating EMC Smarts Brokers" on page 48
❑	Determine the location of the License Server.	Choose a location for the License Server that will support the deployment.	"Locating license servers" on page 49
❑	Determine the location of the EMC Smarts Global Managers.	Choose locations for the one or more Global Managers that will support the deployment.	"Locating the Global Managers in the network" on page 50
❑	Determine if the Service Assurance Manager must be configured with failover.	Determine if a high-availability configuration is required.	"Considering failover (high availability) configurations" on page 51
❑	Determine if acceptance tests are required.	Acceptance tests may be required for different portions of functionality. Be aware of the requirements and develop acceptance criteria.	"Designing acceptance tests" on page 52

CHAPTER 4
Designing for Topology Operations

This chapter describes considerations for importing and organizing topology. The *EMC Smarts Service Assurance Manager Configuration Guide* provides additional details about the methods for importing and organizing topology.

This chapter contains the following information:

- Designing topology synchronization 56
- Designing topology groupings 58
- Designing topology operations checklist 58

NOTICE

The *EMC Smarts Business Impact Manager User Guide* and the *EMC Smarts Adapter for Alcatel-Lucent 5620 SAM EMS Installation Guide* provide information on business and application topology.

Designing topology synchronization

A Global Manager imports topology from configured underlying EMC Smarts domains. The topology sources are specified in the Global Manager's ics-default.xml file located in the BASEDIR/smarts/conf/ics directory. A Global Manager synchronizes its topology whenever any of the following occurs:

- The Global Manager is started.
- A reconfiguration to an underlying server is required.
- The analysis domain is reconnected after a connection loss or a restart.
- The analysis domain performs a full or incremental discovery, or rediscovers an object; or a manual discovery is initiated.

Understanding topology synchronization

The Global Manager maintains an abstracted representation of the topology discovered by the Domain Managers and the topology from the Adapter Platform. When the Domain Manager completes a discovery, the Global Manager resynchronizes the topology from the Domain Managers.

In all cases, the Global Manager does a full topology synchronization, even when only one device is added to an underlying domain. This means that topology synchronization is most efficient when it occurs less frequently, but with larger sets of changed devices. The Global Manager resynchronizes topology after a single device discovery, a pending discovery, or a full rediscovery.

Another important aspect of topology synchronization is synchronization between the Global Manager and the Adapter Platform. The Adapter Platform is typically configured to contain some of the topology from the IP Availability Manager. With this topology, the Adapter Platform matches events to devices managed in the IP Availability Manager. So, the Adapter Platform often contains all of the IP Availability Manager topology and also some topology created based on events received as traps, or from other sources.

When the Global Manager synchronizes topology with the Adapter Platform, the topology inherited from the IP Availability Manager, as well as the topology created due to events from traps or other sources, is synchronized. This means that a Global Manager may synchronize topology twice when a change is made to a device in IP Availability Manager: once with the IP Availability Manager and then once with the Adapter Platform.

The Adapter Platform does not contain all of the topology from the IP Availability Manager. So, the synchronization it performs with the Global Manager for the topology from IP Availability Manager is less performance-intensive than the synchronization between the Global Manager and the IP Availability Manager.

Topology synchronizations in the Global Manager may be done concurrently. If there are a number of underlying domains that need to synchronize topology at the same time, the Global Manager may synchronize these at the same time. In most cases, this gives optimal performance. However, if there are many concurrent topology synchronizations, the performance of the Global Manager may be adversely affected. Also, topology synchronizations overall may take longer due to the contention caused by many concurrent threads updating the Global Manager repository.

Planning for topology synchronization

If there is a large number of underlying domains and the topology of each changes frequently, it is likely that the Global Manager will often be required to perform multiple concurrent topology synchronizations.

To optimize Global Manager performance, it is best to reduce the number of small discoveries, and instead have a smaller number of larger discoveries. Also, if full rediscoveries are scheduled, Global Manager performance can be improved by staggering the times of full rediscoveries of the underlying domains.

Finally, if you expect a constantly changing network with a large number of discoveries, you may want to reconsider your Global Manager configuration. If you are not already planning on implementing a hierarchical aggregation Global Manager configuration or hierarchical mesh Global Manager configuration, consider using one of these configurations. "Determine the Service Assurance configuration" on page 36 provides additional information.

Designing topology groupings

Topology grouping is a visual tool that does not affect Global Manager performance. Groups help organize topology elements. The groups can represent resources, geographical areas, business areas, or entities of other organizations.

Think about groups in terms of the business needs: Is your topology divided by region? Does the topology correspond to business units? Is the topology allocated by customer or function?

Groups are organized into parent and child groups. A group can contain both members and child groups. When planning group organization, you should try to restrict each group to hold only members or only child groups. The Map Console will not show the members of a group if the group also contains child groups.

There are two ways to create groups. The first method uses matching patterns entered through a Global Console. This method creates *selective groups*. The second method relies on a file to determine group hierarchy and membership. This method creates hierarchical groups.

For selective groups, you must define three properties to organize your groups: matching criteria, priority, and target classes. You define hierarchical groups and their members by using files. One file contains groups and child groups, which can also include members or links to files that contain members.

The *EMC Smarts Service Assurance Manager Configuration Guide* provides additional information about topology groupings.

Designing topology operations checklist

Table 9 on page 58 contains the designing topology operations checklist. Each chapter in this guide includes a checklist. "Design and Deployment Checklists" on page 107 contain all available checklists.

Table 9 Designing topology operations checklist

Complete	Task	Description	Related documentation
❑	Review topology synchronization issues.	Ensure that you understand how topology synchronization affects Global Manager performance.	"Understanding topology synchronization" on page 56
❑	Plan topology synchronizations.	To optimize Global Manager performance, it is best to reduce the number of topology synchronizations where possible.	"Planning for topology synchronization" on page 57
❑	Plan topology groups.	Topology grouping is a visual tool that doe not affect Global Manager performance. Groups help organize topology elements.	"Designing topology groupings" on page 58

CHAPTER 5
Planning for Notifications

This chapter covers the basics of notification handling for the Global Manager, the Adapter Platform, and clients of the Global Manager. The design and deployment of notification handling is an iterative process. Start with a basic configuration and make adjustments based on results. The chapter contains the following information:

- Overview of notification processing 60
- Designing trap processing 62
- Trap processing performance 65
- Planning for notifications checklist 66

Overview of notification processing

In a typical configuration of the Global Manager, there are two types of information sources: the Adapter Platform, which handles information from third-party applications, and the underlying EMC Smarts Domain Managers. After the Global Manager processes the information, it forwards the results to end users through clients such as the Global Console or an adapter that sends information to third-party applications.

Figure 8 on page 60 is an example of a basic Global Manager configuration. The lines show how information enters through either an IP Availability Manager (or other underlying Domain Managers) or an Adapter Platform, is passed to a Global Manager, and finally is passed to an end user or application. Note that the figure does not show the connection that transports topological information between the IP Availability Manager and the Adapter Platform.

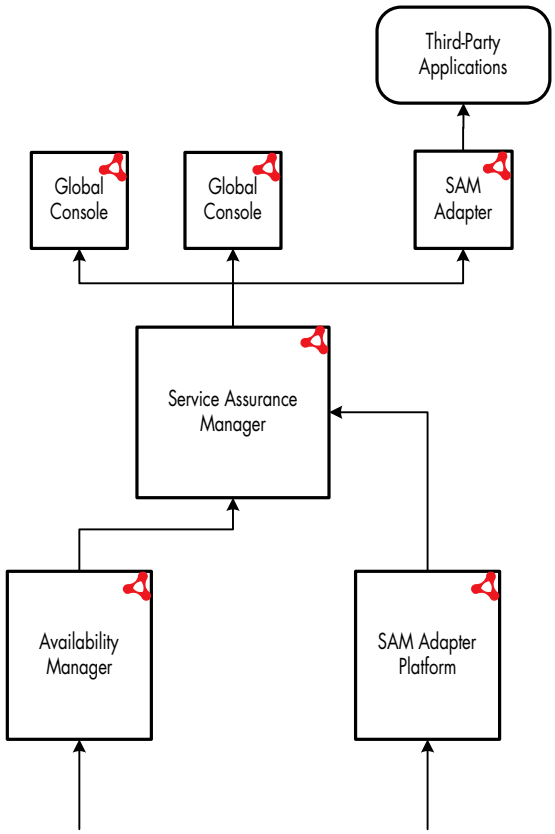

Figure 8 Flow of event information in a basic Global Manager configuration

The Adapter Platform imports topology and event information from third-party applications and prepares it for use by the Global Manager; data from third-party applications should not be sent directly to the Global Manager. The Adapter Platform creates notifications from the incoming information and can even associate these notifications to topology information supplied by another EMC Smarts application. This connection is not shown in Figure 8 on page 60.

Efficient notification processing presents the most important information to the operator who uses a Global Console. One of the goals of deployment design is to plan to optimize the flow of notifications sent through the system. This improves the performance of the deployment and the performance of the operators who use EMC Smarts products.

Notifications from the Adapter Platform

Ensure that only the notifications that are useful to Global Console operators reach the Global Manager. When EMC Smarts software is first deployed, a strategy to reduce and control the flow of notifications from the Adapter Platform to the Global Manager temporarily is used. This is to prevent any notifications from the Adapter Platform from being forwarded to the Global Manager by using notification list filtering. Use the Global Console to connect directly to the Adapter Platform and observe the notifications there; evaluate the notifications. Then, modify the notification list to forward useful notifications to the Global Manager.

Configuring notifications for clients

Notification lists are the mechanism that clients use to subscribe to and receive notifications from a Global Manager. Smaller notification lists produce better performance in both the client and the Global Manager.

Notification lists are associated with user profiles—one list per profile. When designing notification lists for consoles, attempt to design them not to typically include more than 500 active notifications.

It is more efficient to prevent a notification from being sent to an EMC Smarts client by using a notification list than to use a filter to remove the notification from the Notification Log view at a client's console. These filters do not curtail the output of notification lists and do not provide any performance benefit to the Global Manager.

"Designing notification lists" on page 71 provides information about configuring clients to receive notifications.

Designing trap processing

Any EMC Smarts deployment should include a plan to handle trap processing. This section covers one method.

Recommended trap processing design

Figure 9 on page 63 presents the recommended design for trap processing. This configuration relies on two instances of the SNMP Trap Adapter to handle trap processing and performs well under the stress of high volumes of traps.

Instance 1 of the adapter is configured as a trap exploder that receives and forwards traps. In this configuration, the trap exploder does not process traps into notifications or use any ASL scripts for sophisticated trap processing. The trap exploder's sole function is to filter and forward traps as follows:

- IP Availability Manager-required traps are forwarded to IP Availability Managers.
- Traps required for other EMC Smarts domains are forwarded to the proper domain.
- Other useful traps are forwarded to Instance 2 of the SNMP Trap Adapter for processing into notifications. These traps provide meaningful or actionable information to the administrators of the network.
- Traps may be received in SNMP v1, v2c, or v3 formats; however, they are forwarded on to the Adapter Platform or to other Domain Managers as SNMP v1 or v2c only.
- Other traps that administrators do not consider useful can be forwarded to an unused port to reduce overhead at the SNMP Trap Adapter. Normally, the adapter logs an error message for each trap that is not forwarded. This practice eliminates the processing associated with error message logging. Another benefit is that forwarding these traps to a location permits you to set up a tool to collect the traps.

Instance 2 of the SNMP Trap Adapter performs the typical processing of traps into notifications and may invoke ASL scripts for advanced processing. This instance does not forward any traps.

The two SNMP Trap Adapter instances are invoked using different configurations. The following is the typical configuration when the instances are on a single host:

- The trap exploder, Instance 1 of the SNMP Trap Adapter, uses BASEDIR/smarts/local/conf/trapd/trapd.conf. This trapd.conf file includes trap forwarding statements and indicates the port to use when listening for traps.
- Instance 2 of the SNMP Trap Adapter processes traps into notifications and uses BASEDIR/SAM/smarts/local/conf/icoi/trapd.conf and BASEDIR/SAM/smarts/local/conf/icoi/trap_mgr.conf. This instance of the trapd.conf file does not include trap forwarding statements. (If the Trap Adapter was installed as a service, these traps will be forwarded to the Adapter Platform by default). The trap_mgr.conf file includes definitions for all traps that will be forwarded as notifications.

NOTICE

The trapd.conf files installed with IP Availability Manager and IP Performance Manager can be configured to permit the IP Availability Manager and the IP Performance Manager to forward traps by using their built-in trap receivers. This capability should not be used

except for testing deployments or when migrating traps from multiple locations to one destination. The *EMC Smarts Service Assurance Manager Adapter Platform User Guide* provides a complete description of the SNMP Trap Adapter.

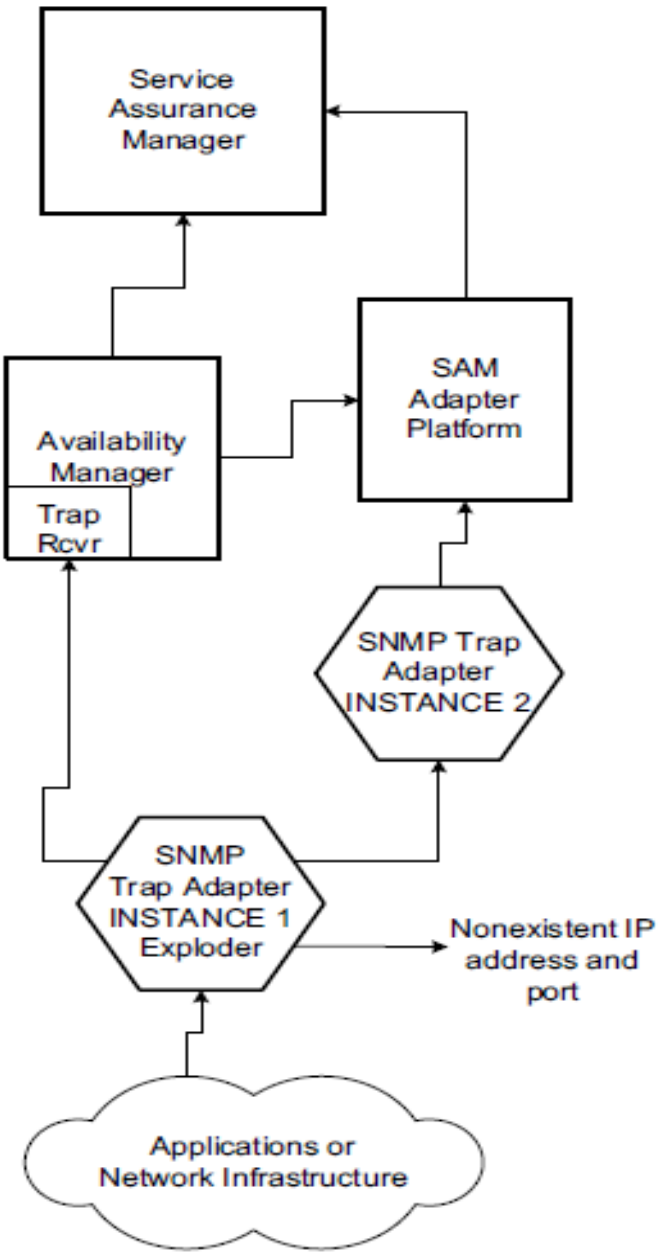

Figure 9 Recommended trap processing design

Batching to improve performance

In a deployment where a high frequency of traps is expected, plan on using the batching capability of the SNMP Trap Adapter to improve performance of clients that process the notifications. The BATCH_NOTIFY_INTERVAL in the trap_mgr.conf configuration file determines the length of the interval. It may be necessary to fine tune this value under the typical trap load, so plan on monitoring the client performance and adjusting this value.

Listening for traps

The SNMP Trap Adapter configured as the trap exploder should listen for traps on the standard port that network devices use when forwarding traps. Typically, this is port 162.

If another application is listening for traps at this port, the application should be moved and the trap exploder can forward needed traps to the application's new location. The SNMP Trap Adapter is usually a better choice for forwarding traps than other third-party software applications because the adapter can be limited to the forwarding function.

The second instance of the SNMP Trap Adapter can listen on any port, but the trap exploder must be configured to forward the appropriate traps to this port.

Trap forwarding

When defining trap forwarding, traps required by underlying EMC Smarts domains are already defined in the trapd.conf file. The only change that you must configure is to specify the specific destination hosts and ports where the domains listen for traps. This change is only necessary in the BASEDIR/smarts/local/conf/trapd/trapd.conf file for the trap exploder instance of the SNMP Trap Adapter (Receiver).

Always try to minimize the volume of traps that are forwarded from the trap exploder to the second instance of the SNMP Trap Adapter. If network administrators do not use the traps, discard them by sending them to a nonexistent address. Judicious forwarding of traps can reduce or eliminate stressful trap processing loads. Deployment administrators should be aware of how traps are discarded so that they can choose to discard other useless traps as their experience with the network grows.

Each trap that is transformed into a notification will require an entry in the trap_mgr.conf file. If an entry does not exist, the trap should not be forwarded to the second instance of the SNMP Trap Adapter.

Traps and notifications

When configuring the SNMP Trap Adapter to process traps, you must decide which traps should become notifications and how the notifications should appear at the Global Console. This is defined in the trap_mgr.conf configuration file which is described in the *EMC Smarts Service Assurance Manager Adapter Platform User Guide*.

Advanced trap processing by using ASL scripts

The *EMC Smarts Service Assurance Manager Adapter Platform User Guide* provides specific information on advanced SNMP Trap integration. Typically, the ASL scripts (hook scripts) will extract event text or other important information from network sources to populate the user defined fields in the notification.

Include as much of the ASL processing definition as possible in the deployment build guide. When the ASL scripts are complete, the code should be documented in the deployment build guide.

Trap processing performance

The following processing issues impact performance directly:

- Filtering traps in trap_mgr.conf
- Using trap manager hook scripts
- Adding new topology for received traps
- Looking up host names
- Batching traps

Consider them as you design and configure trap processing in your environment.

Filtering traps in trap_mgr.conf

The number of traps that are processed and specified in trap_mgr.conf is a factor in performance. The smaller the number of received traps processed as events, the higher the overall rate of processing.

With trap receiver you can import subsets of traps into the Adapter Platform. Large numbers of traps may be sent by network devices, but typically only a small subset are important to network operators. For example, link up and down traps are sent frequently by devices but are not typically important to display to the network operator since the IP Availability Manager already supplies authentic, root-cause problems related to these traps.

Using trap manager hook scripts

With the trap manager, you can customize trap processing by using an ASL hook script. Typically, this additional processing causes performance to be slightly slower than if no ASL hook scripts are used.

Adding new topology for received traps

The trap manager allows the Adapter Platform to create a topology element if a trap is received for an element that is not already in the topology. The performance declines whenever a large percentage of the traps processed result in topology creation and increases whenever all traps relate to existing topological elements.

Looking up host names

The trap processing logic may require a reverse host name lookup to find the logical name of the element sending a trap based on the IP address. The speed of this host name lookup in a particular customer's environment may have a significant impact on trap processing rates. For example, if a customer uses a DNS host lookup and a large number of traps are received for devices not listed in DNS, DNS time-outs may significantly impact trap processing performance.

Batching traps

Configuring the trap manager to batch traps for the same notification results in improved performance due to the reduced number of notifications and updates that need to be processed.

The trap manager has an option to batch traps whenever multiple traps for the same notification are received within a configurable number of seconds. For example, if 50 identical traps are received from a single device in 2 seconds, the result is 50 updates to the same notification. All clients receiving these notifications would receive 50 updates to the notification. Setting a two-second batch interval, however, means that there is only one update to the notification, even though 50 traps were sent.

Configuring trapd.conf file to manage flooding of traps

In order to avoid the trap receiver and trap exploder from crashing due to flooding of traps, the default values for the QUEUE_LIMIT_MEGS and QUEUE_LIMIT_SECONDS parameters in the trapd.conf file are set to 200 MB and 480 seconds, respectively. On reaching the limits set using these parameters, the sm_trapd queue starts discarding traps. An alert, *trapdDiscardingTraps*, which is defined in the trap_mgr.conf file is triggered when the queue starts discarding traps. If and when a device is identified as the source of a trap storm, information about it is logged in the sm_trapd log file.

The *EMC Smarts Service Assurance Manager Adapter Platform User Guide* provides details on the parameters in the trapd.conf file.

Planning for notifications checklist

Before discovering the network, the requirements mentioned in Table 10 on page 66 must be completed. Each chapter in this guide includes a checklist. "Design and Deployment Checklists" on page 107 contains all available checklists.

Table 10 Planning for notifications checklist

Complete	Task	Description	Related documentation
❑	Determine which notifications are important for EMC Smarts operators.	Ensure that only the notifications that are useful to Global Console operators reach the Global Manager.	"Overview of notification processing" on page 60
❑	Configure notification lists.	Notification lists are the mechanism that clients use to subscribe to and receive notifications from a Global Manager.	"Configuring notifications for clients" on page 61
❑	Design trap processing.	Any EMC Smarts deployment should include a plan to handle trap processing.	"Recommended trap processing design" on page 62

CHAPTER 6
Designing for EMC Smarts Users

This chapter describes how best to use EMC Smarts clients to satisfy the needs of operators, including providing appropriate access rights to monitoring and administration functions. This chapter contains the following information:

- Determining the Global Manager users 68
- Choose a console for types of users 68
- Designing user profiles 71
- EMC Smarts user design checklist 73

Determining the Global Manager users

When designing your deployment, you must determine who will use the Global Consoles and for what purposes. Initially, it is not necessary to define individual users — consider, instead, broad functional categories of users with similar needs and characteristics.

Name these categories of users, and list their requirements in the deployment build guide. Using specific position names as categories can make the process easier. For example, you might create one or more of these categories: application engineer, application administrator, application support specialist, network engineer, network administrator, network support specialist, NOC manager, NOC operator, LAN administrator, and IT management. After you choose categories, determine the typical EMC Smarts-related duties that are performed by the individuals in these categories. You may find that as you list the duties, you have to expand or combine certain categories. After the categories are completely defined, list them and their specific needs in the deployment build guide.

For example, Table 11 on page 68 defines two typical user categories.

Table 11 Examples of EMC Smarts user categories

User category	Description of needs or duties
Network engineer	Administers and maintains local and wide area networks and related hardware. Monitors daily activity, enforces licensing agreements, and provides front line support, including both software and hardware support. • Needs monitoring access to all EMC Smarts domains. • Needs to see all important traps, notifications, and network outages.
Local area network administrator supporting a customer of a Service Provider	Directs the daily operational availability of the hardware and software systems required to support facility operations. Directs and oversees scheduled testing and review of hardware and software to ensure that potential problems are identified as soon as possible. Analyzes, evaluates, and builds cost effective LAN solutions that leverage resources and technology to meet business requirements. Designs, creates, and distributes user documentation relating to installation of software: • Needs monitoring access to the EMC Smarts domain that supports the customer, but not access to any other domains. • Needs to see all important traps, notifications, and network outages for the customer's EMC Smarts domain.

Choose a console for types of users

EMC Smarts consoles have different capabilities. For each type of user and their equipment, determine which console is most appropriate:

- Global Console can provide all functions and all views to operators. Administrators can restrict console operations so that operators have only the specific functionality that they require.
- Web Console can also provide all functions and all views that are available from the Global Console, except for the Topology Browser Console. As with the Global Console, most console operations can be restricted.

- Business Dashboard provides access to the results of EMC Smarts analysis through one or more Dashboard viewlets implemented as Java applets. A Dashboard viewlet is a Java application that shows a particular aspect of Service Assurance information such as notifications, containment, or maps. Dashboard viewlets provide operational functions, but no administrative functions.

A Global Console, a Web Console, and a Business Dashboard (on a single browser using one Java Virtual Machine) showing similar content, all use the same resources. This is regardless of the number of viewlets displayed by the Business Dashboard.

Deployment considerations for client consoles

Each of the available Client Consoles has different deployment considerations.

Typically, the Global Console is deployed by installing EMC Smarts software on the user's system, but the Global Console can also be accessed by using the X Windows server software on the user's system when the Global Console runs on a remote UNIX server. If using X Windows, the client that displays the application over X does not require Java.

The Web Console can be deployed in two ways:

- As a typical Java applet where the HTML and JAR files are located on the host where the Business Dashboard is installed. When a user opens the Web Console, the JAR files are loaded over the network to the user's Web browser.
- By using Java Web Start, which integrates the Web Console with the user's desktop. This enables users to start the Web Console directly from their desktop without using a Web browser.

With both methods, the JAR files are loaded over the network during the initial connection but then saved to the user's local system. Subsequent connections use the cached files to optimize the use of network resources.

An installation of the Business Dashboard includes the Tomcat Servlet Engine, which is necessary to serve the JAR files and map layout servlet. The servlet engine is installed as the EMC Smarts Servlet Engine service on the system where you install the Business Dashboard. If you wish to serve Dashboard viewlets from a different Web server, you must install the Business Dashboard to obtain the necessary files.

Users and security

When defining the functional categories for users, consider the EMC Smarts security implementation. For each user category, determine the following:

- List the specific EMC Smarts managers that must be accessed by the users in the category. If necessary, divide a category. For example, in a category of network administrators, some users require access to managers in Asia, other users require access to managers in Europe, and still other users require access to all managers. In this case, the network administrators category could be divided into three categories to match these user needs while maintaining tighter security.
- Determine what functionality must be accessed by users in this category. EMC Smarts software is designed so that users can be classified into levels that define access:
  - All - A level where users can access all Global Console functionality available for one or more EMC Smarts Managers, if their user profile permits it.

- Monitor - A level where users can access only Global Console monitoring functionality, not administrative functionality, at one or more EMC Smarts Managers, if their user profile permits it.
- Ping - A level normally reserved for EMC Smarts processes, where processes will ping hosts where other EMC Smarts processes are installed to determine if the hosts are running.
- None - A level that specifically excludes access to the Global Console.

These types of security access are defined in the *serverConnect.conf* file located in the BASEDIR/smarts/local/conf directory on the servers where EMC Smarts software is installed.

Password configurations

Determine how you will configure EMC Smarts passwords. You can do any or all of the following:

- Allow the host operating system to validate users. This method provides the highest level of security and is easy to manage because it relies on the security implementation that is already in place. There are two variations: Any valid user can access one of the EMC Smarts levels (All, Monitor, Ping), or specific users can access a specific EMC Smarts security level. Defining specific access requires more maintenance because you must list the user names in the configuration files (serverConnect.conf and clientConnect.conf). The *EMC Smarts System Administration Guide* defines methods to configure and to secure access for these files.
- Specify unique passwords for individual users. Consider this method only when there are very few users because it requires a high level of maintenance.

  NOTICE

  This is less secure than permitting the host to validate users. Plan on using the EMC Smarts password encryption mechanism. The *EMC Smarts System Administration Guide* provides information on using this encryption mechanism.

- Specify a common user name with a common password. This method is the least secure, but very easy to maintain. The *EMC Smarts System Administration Guide* provides information on using this.

  NOTICE

  You can combine these methods, for example, you could restrict administration (All) capabilities to specific users validated by the operating system. In addition, you could provide Monitor level abilities to a general EMC Smarts user named, Monitor.

Designing user profiles

The functional grouping of users and their requirements form the basis of EMC Smarts user profiles. These profiles combine access to notification lists, console operations, custom console layouts, and specific tools.

Create a profile for each category of user that you must support. Groups of users with similar needs can then be assigned the same user profile.

If needed, you can further customize a generic user profile by copying it and then modifying it for more specific needs. For example, an administration user profile could be customized for less experienced administrators by restricting access to some administrative console operations and tools. Other possible user profiles could include regional or customer-specific consoles.

Designing notification lists

A notification list determines the events that are forwarded to a user. Essentially, the list filters the notifications that are sent from the Global Manager and can be assigned to one or more users. The lists can be organized by:

- Business units
- Geographical regions
- Groups of resources

For example, a notification list can be defined to allow only notifications from the subnetworks devoted to a specific ISP customer to reach the Global Console of the customer's network administrator. "Configuring notifications for clients" on page 61 provides additional information.

Restricting console operations

Most operations that can be performed at the Global Console can be individually enabled or disabled in the user profiles. When used with the security levels, restricting console operations can fine tune user access to functionality and further protect the deployment.

For example, consider two users who have the Monitor security level that allows access to Global Console monitoring functionality, but not administrative functionality, for an EMC Smarts Manager. You can further restrict one of the viewers to see only the Summary view and the IP Network Map while restricting the other user to the Notification Log and the Topology Browser.

Designing consoles

Specific console layouts can be saved and then automatically provided to match user needs at the Global Console, Web Console, and Business Dashboard. Layouts including the Notification Log view, Map view, Summary view, Status Table view, Notification Properties view, and Containment view can be used to create a custom console layout for Global Console and Web Console users. These same views can be provided to Business Dashboard users as Dashboard viewlets. Another view, the Topology Browser view, can be used only in layouts for Global Console and Web Console users.

Filtering the Notification Log view at the Global Console or Web Console does not affect Global Manager performance, but it can be used to create a simpler, more useful Notification Log view for groups of users. Properly configured filters can be used to customize notification consoles for groups of users. In addition, specific views can be automatically provided to match user needs.

Topology Browser views (not Map views) are more resource intensive than others. Carefully consider the information that operators require before including the Topology Browser view in a console layout.

All console layouts are created by using the Global Console and then deployed by copying them to appropriate directories on the Global Manager.

Planning for tools and tool deployment

Many types of client and server tools can be designed and developed for users. Typically, creating tools will require different levels of programming skills based on their complexity. Plan to have individuals with the appropriate skills available to create the tools. In addition, determine which tools should be available to which users through their user profiles.

EMC Smarts user design checklist

Each chapter in this guide includes a checklist. "Design and Deployment Checklists" on page 107 contains all available checklists.

Table 12 Client and user design checklist

Complete	Task	Description	Related documentation
❑	Define functional groups for users.	List the broad functional groups that users will belong to and then define the needs and duties of each group. Add this information to the deployment build guide.	"Determining the Global Manager users" on page 68
❑	Choose appropriate EMC Smarts clients for users.	EMC Smarts consoles have different capabilities. For each type of user and their equipment, you must determine which console is most appropriate. Document your choices in the deployment build guide.	"Choose a console for types of users" on page 68
❑	Define how you will implement security for user groups.	When defining the functional categories for users, consider EMC Smarts's security implementation. Document the implementation in the deployment build guide.	"Users and security" on page 69
❑	Define the notification lists in EMC Smarts user profiles.	The Notification List determines which notifications are sent from the Global Manager to the EMC Smarts client. Carefully define the Notification Lists, as they can affect Global Manager performance. Document the notification list provided for each user profile in the deployment build guide.	"Designing notification lists" on page 71
❑	Design console operations access.	Most operations that can be performed at the EMC Smarts clients can be individually enabled or disabled in the user profiles. Document the access provided for each user profile in the deployment build guide.	"Designing consoles" on page 72
❑	Design console layouts.	Specific console layouts can be saved and automatically provided to match user needs at the Global Console, Web Console, and Business Dashboard. Document the console layout used in each user profile in the deployment build guide.	"Designing consoles" on page 72
❑	Design client and server tools.	Define the necessary tools that users must access and the levels of programming skills required to create the tools. Document this information and tools provided by each user profile in the deployment build guide.	"Planning for tools and tool deployment" on page 72

CHAPTER 7
Deploying Business Impact Processing

The EMC Smarts Business Impact Manager calculates the impact of infrastructure and application events on business services. This chapter describes the choices that must be made when designing business impact processing, and contains the following information:

- Designing business impact processing 76
- Importing information 77
- Business Impact Manager design checklist 79

NOTICE

The *EMC Smarts Business Impact Manager User Guide* provides complete configuration details.

Designing business impact processing

Business Impact Manager uses information from a Global Manager in conjunction with user-defined criteria to calculate the business impacts of infrastructure and application events.

To design business impact processing, you must consider at least these things:

- Business processes to model
- Topology of the business processes and the underlying infrastructure
- Impact (weight) caused by issues in the topology

Business processes

When designing a Business Impact Manager implementation, consider the company's business processes. Try to answer the following questions:

- Which business processes exist?
- Which business processes depend on applications or network devices?

A business process represents functions and operations that support the internal business activities of an enterprise. In the manufacturing sector, for example, business processes include order processing, production planning, accounting, fulfillment and so on. Each of these may depend on different applications in order to operate as well as on network devices to pass information among and between applications and processes.

In a hypothetical situation, an order comes into a company and is processed. The processing involves entering the order into a Customer Relationship Management (CRM) application and simultaneously sending it to accounting for approval and fulfillment for shipping. Underlying accounting is a financials application and underlying fulfillment is a logistics application. Fulfillment begins preparing the order for shipping but cannot release the order until it receives approval from accounting. After accounting approves the order, it notifies both fulfillment and ordering through a financials application that updates both the logistics and CRM applications. If a network device that supports the financials application exhibits problems, the fulfillment and CRM applications will not receive necessary updates, which has an impact on the business.

Try to map these dependencies in preparation for creating business topologies and weighing the impacts of problems on these processes. In the hypothetical situation, updating the logistics application, after accounting approves the order, may be a higher priority than updating the CRM application.

Business topology

After you have established the business processes, create a business topology to model them. The Business Impact Manager has special elements to describe business processes, including ServiceOffering, BusinessProcess, ServiceSubscriber, Customer, and others. The Business Impact Manager provides flexibility in classifying elements, but be consistent in your choices.

The names of these business elements or entities must be unique. The Global Manager assigns prefixes to the names of business elements to identify elements, but their display names do not include this prefix. When importing topology, remember to specify these names. When using a custom adapter to import topology, the element names must include the prefix.

The relationship between elements in the business topology is defined as part of an element's definition. The next step in designing the business topology is to define relationships between business elements and other elements. As you do this, consider the following:

- How do infrastructure and other events affect business elements?
- What are the critical elements upon which the business elements rely?

In the case of an internet service provider (ISP), ServiceOffering represents the connections between a customer and the ISP. ServiceSubscriber represents a customer. You can make other choices. In this example, a ServiceOffering is a specific contract to provide a connection to a ServiceSubscriber. The business topology gets linked to infrastructure elements through more definitions. The critical elements associated with a ServiceOffering are the customer routers. While the ISP edge routers are important, a ServiceOffering does not necessarily have to be associated with a particular router. When events affect customer routers, the ServiceOffering is directly affected. These routers are associated as members of the ServiceOffering.

NOTICE

As the ISP does not manage the customer routers, the ISP has only ICMP access to those routers. The ISP could choose to represent those routers as hosts.

Weights: Impacts of issues in the topology

Weight values are optionally set to indicate the magnitude of a problem but are not necessary to a Business Impact Manager installation. When a root-cause event occurs, the Business Impact Manager sums the weight of the affected elements for a total impact, which is displayed with the root-cause: the greater the impact number, the greater the problem. You can assign weights by class, instance name, and instance. Wildcards can be used when you assign weights by instance.

There is no particular method to use to assign weights. Weights measure the relative importance of aspects such as revenue, contractual requirements, reputation, cost and so on.

One strategy to follow when assigning weights is to develop weights from parent classes, through subclasses, down to particular instances. If there is any overlap, the instance weight is used before the class weight, which is used before the parent class weight.

Importing information

There are two methods of importing Business Impact Manager information: import files and manually through a Global Console:

- For small Business Impact Manager configurations, such as demonstrations and test installations, manual configuration using the Topology Builder Console is a reasonable choice. Objects created through this console are stored only in the topology. You cannot modify elements not created with the Topology Builder or that are discovered by the underlying Domain Manager. Before you can use this console, you must set correct user privileges and permissions.
- For larger deployments, creating import files can ease the process. Import files provide an additional benefit in that they act as backup to the business topology.

Do not use both methods to modify business topology. Elements added through one method are not modifiable by the other.

The structure of import files

When creating import files, consider the structure of the files. Especially when defining relationships, multiple files may be used to define one-to- many relationships. Business elements can be comprised of members and subscribers. Separate files may exist for these files or may exist for all of the different business elements.

The import files define business elements and service data, related infrastructure, and lists of related elements. The lists of related elements can be very useful when constructing relationships among large groups of elements, subscribers, or applications.

The *EMC Smarts Business Impact Manager User Guide* provides additional information.

Business Impact Manager design checklist

Each chapter in this guide includes a checklist. "Design and Deployment Checklists" on page 107 contains all available checklists.

Table 13 Business Impact Manager design checklist

Complete	Task	Description	Related documentation
❑	Define relevant business processes.	Determine the business processes to model. Record your choices in the deployment build guide.	"Designing business impact processing" on page 76
❑	Create a business topology.	Create business elements to model the business processes. Define the expectations of the model and the events it should analyze. Document your topology in the deployment build guide.	"Business topology" on page 76
❑	Assign weights to the elements of the topology.	This step is optional but allows you to assign a value to measure the impact of events that occur in the topology. The weights of impacted elements are summed in the root-cause event. Record your choices in the deployment build guide.	"Weights: Impacts of issues in the topology" on page 77
❑	Choose a method for importing information.	If the topology is small, use the Topology Builder Console. For larger topologies, use one or more topology files. Record your choice in the deployment build guide.	"Importing information" on page 77

CHAPTER 8
Deploying Service Assurance

This chapter describes important design considerations when installing EMC Smarts software. This chapter should be considered as a supplement, not a replacement, for the installation guides included with the EMC Smarts Service Assurance Manager software and for the *EMC Smarts System Administration Guide*. This chapter contains the following information:

- General installation/deployment guidelines 82
- Configuring the Java clients to use a proxy server 86

NOTICE

The *EMC Smarts Installation Guide for SAM, IP, ESM, MPLS, VoIP, and NPM Managers* provides detailed installation procedures and the *EMC Smarts System Administration Guide* provides detailed configuration procedures.

General installation/deployment guidelines

Many organizations have strict rules for deploying enterprise-level software which may include deployment of staging areas whenever possible. A staging area is a copy of the installation.

Never install a deployment during a normal production shift; instead, choose a period of low utilization for installation. If a test environment is not available, create a staging area (an environment that mirrors production) and make and test all changes in the staging area.

If possible, install the deployment in stages to reduce the size and complexity of each step in the overall process. Doing so makes debugging easier. "Validating Your Deployment (Acceptance Testing)" on page 89 provides information on verifying each stage of the deployment.

EMC Smarts licensing

EMC Smarts licensing comes in two different forms: temporary and permanent. Temporary licensing is file-based, where each application uses a special license file. Permanent licensing uses a licensing server.

Information to collect for permanent licensing

For permanent licensing, EMC Smarts products rely on FLEXnet licensing software and a license file provided by EMC Smarts.

The FLEXnet software is installed with the EMC Smarts software, but a license file from EMC Corporation is also required. The license file must be installed by using the *install_license* script. The *EMC Smarts System Administration Guide* provides additional information on using this script. To generate the license, EMC Corporation needs the following information for the computer where the FLEXnet license server is running:

- The host ID
- The operating system
- The installation path to BASEDIR/smarts/bin on your Broker (or to the EMC Smarts installation from which you want to run the license server).

Send this information to the licensing@emc.com e-mail address to ensure that an appropriate license is available for the deployment.

Locating the license server

It is best to install the license server on the same host as the Broker. Both the license server and the Broker need to be able to communicate with various EMC Smarts applications.

Multiple license servers

When multiple license servers are used, each Domain Manager will use blocks of volume licenses from a specific license server. Therefore, the license blocks must be divided among multiple licenses; the appropriate license must be deployed to each license server.

Then, Availability Managers that rely on a specific license server will have the appropriate volume licenses to support the systems they manage. If multiple license servers are used, ensure that EMC Corporation is aware of the configuration details.

EMC Smarts Service Assurance installation and configuration

As you install EMC Smarts applications, configure each application before installing the next application. The following installation order is recommended:

1. Broker (normally installed and configured when the first EMC Smarts components are installed)
2. A Global Console to use to verify the installation of other applications
3. Underlying domains (IP Availability Manager, IP Performance Manager, Network Protocol Manager and so on) with associated adapters
4. Global Manager which includes Business Impact Manager, if licensed
5. Adapter Platform
6. EMC Smarts Adapters (that use Adapter Platform as a server)
7. Global Consoles for end users

To ease troubleshooting of the initial deployment, install limited segments of the deployment first. Always configure and validate a segment before installing the next segment.

By default, all EMC Smarts applications are installed as services and are started immediately after installation. During deployment, you should set the services to start manually until your installation, configuration, and validation are complete.

Setting environment variables

Setting inappropriate values for environment variables is a common cause of post-installation problems. The *EMC Smarts System Administration Guide* provides detailed descriptions of the EMC Smarts environment variables, including methods for setting them.

Configure security

Usernames and passwords

For initial validation, use the default administration user name and password (*admin* and *changeme*). Once you have validated your installation, change the default administration user name and password.

Security configuration files and a Secure Broker

A Secure Broker is a broker that requires authentication for EMC Smarts applications to use it. By default, the Broker is not secure. Universal access is provided to all EMC Smarts applications. By changing the password associated with the Broker, you configure a Secure Broker. EMC Smarts applications need to use the username and password to register with the Broker and also to connect to the Broker to determine the location of server applications.

Use and guard your EMC Smarts secret phrase

EMC Smarts components are installed using a default secret phrase. This phrase is used to encrypt passwords used in authentication and to encrypt communications between EMC Smarts components.

It is recommended that you take advantage of the added level of security provided through the secret phrase and its related security mechanisms. To do this, you must change the secret phrase using **sm_rebond** and make it consistent at all your EMC Smarts installation sites. Due to the sensitive and vital nature of this secret phrase, store and guard the phrase as you would do with the root passwords of the most sensitive servers in your network.

Under certain circumstances, the loss of the secret phrase can force extensive reconfigurations and require reinstallations of all EMC Smarts components.

Propagating the secret phrase

The recommended method to change the secret phrase for an EMC Smarts deployment is to run the sm_rebond utility in each installation area. Under certain circumstances, you can use one machine to create and encrypt the different configuration files as well as the imk.dat file.

It is best to do the propagation before the other applications are started. If you want to propagate the files and the server has started but there are no SNMP V3 passwords in use, replace the connection configuration files. If you are using SNMP V3 passwords, you will need to replace the connection configuration files, delete the existing repository files (**.rps*), and restart the server.

An alternative to propagating all of the files is to propagate imk.dat, then run it for each file requiring security on each installation:

```
sm_edit --noedit ‹filename›
```

Deploy Business Impact Manager

Deploying the Business Impact Manager consists of the following tasks:

- Modify weights by changing weights.conf and applying the change using the ASL script, ICS_RemoteConfig.asl. Unless the weights are modified, each element will have the same value.
- Import business data by using the Topology Builder Console.

Deploy user configurations

Deploying the user configurations consists of these tasks:

- Configure access to the software by adding user names and passwords to the security files (clientConnect.conf and serverConnect.conf). The *EMC Smarts System Administration Guide* provides additional information. These files define the security level for each user, including user capabilities, servers that can be accessed, and passwords.
- Configure users:
  - Deploy and configure tools. Server tools must be copied to the server where the Service Assurance Manager is installed and client tools must be copied to all systems where Global Consoles are installed.
  - Create console configurations and then save them on the server where the Global Manager is installed. Create each console configuration by opening the Global Console and then arranging the layout and customizing preferences. Save the customized console.
  - Define customized Notification Lists.
  - Create user profiles that associate individual users with access to specific tools, consoles, and a Notification List.

    The *EMC Smarts Service Assurance Manager Configuration Guide* provides configuration information.

Configuring the Java clients to use a proxy server

You can configure the Web Console, Business Dashboard, Dashboard viewlets, and Global Console to use a proxy server to communicate with EMC Smarts applications (for example, Availability Manager and Global Manager) that reside behind a firewall. Using a proxy server reduces the number of firewall ports that need to be opened to one firewall port.

For example, Figure 10 on page 86 illustrates a typical deployment where a web server acts as the proxy server.

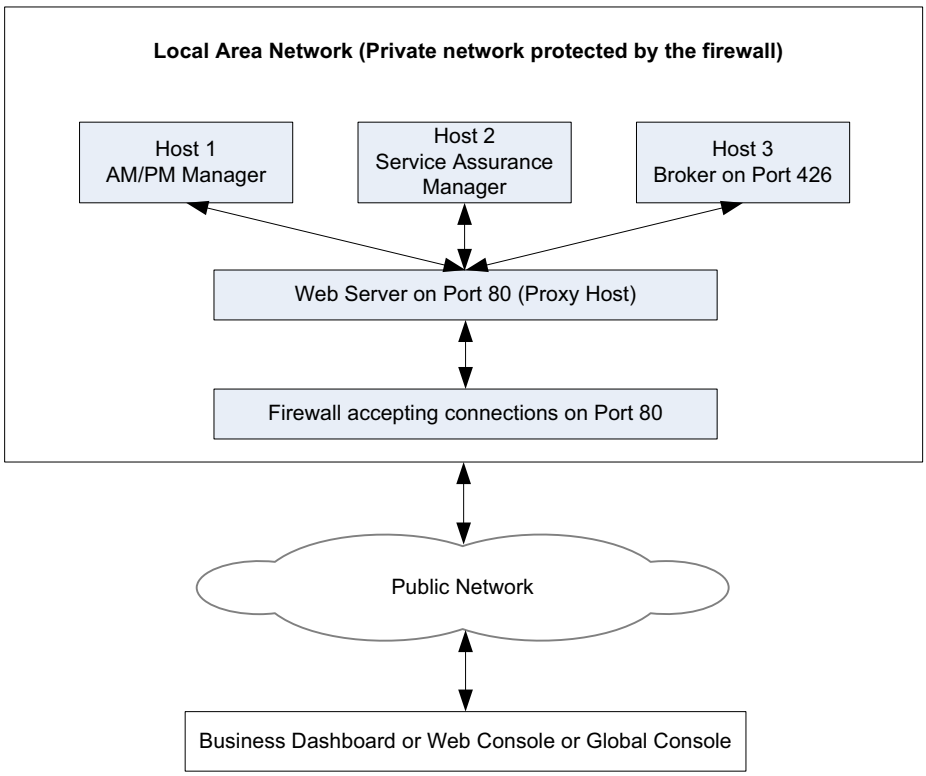

Figure 10 Example of a deployment with a proxy server

Before you configure these Java clients, ensure that the web server that will act as the proxy satisfies the requirement. "Requirement for proxy server" on page 88 provides additional information.

To configure the Java clients to use a proxy server, perform the following steps:

1. Your firewall administrator needs to open Port 80 on the firewall to enable communications between the client and a web server.

   Through a series of requests, the client communicates with a web server and that web server acts as the proxy to the EMC Smarts applications.

2. Use the sm_edit utility to add the Proxy properties described in Table 14 on page 87 to the appropriate properties file:

- If you are using the Web Console (or Java Web Start), modify the webconsole.properties file in the BASEDIR/smarts/tomcat/webapps/webconsole directory.
- If you are using the Business Dashboard and viewlets, modify the dashboard.properties file in the BASEDIR/smarts/tomcat/webapps/templates directory.
- If you are using the Global Console, modify the BASEDIR/smarts/conf/console/properties.conf file on the machine where the Global Console is running.

Table 14 Parameters for Properties files

Parameter	Value
com.smarts.proxyHost=	"‹Host name where the web server is running›"
com.smarts.proxyPort=	‹Port number of the web server›
com.smarts.proxyAuthString=	"‹Authentication string if authentication is required›"
com.smarts.proxyAdditionalHeaders =	"‹Additional HTTP Requests if any ›"

The parameters, com.smarts.proxyAuthString and com.smarts.proxyAdditionalHeaders, are optional.

For com.smarts.proxyAuthString, the value would depend upon the type of authentication scheme the web server supports. For example, if it supports Basic Access Authentication Scheme, and if the Username is "User" and Password is "Password", the value would be:

```
"Proxy-Authorization: basic VXNlcjpQYXNzd29yZAo=\r\n"
```

Then, string "VXNlcjpQYXNzd29yZAo=" would be the base64 encoding of, "User:Password".

For com.smarts.proxyAdditionalHeaders, the value could be any valid HTTP request like this, for example:

```
▼"GET http://www.example.com HTTP/1.1 \r\nHost: example.domain.com
  \r\n"▲
```

3. Modify the proxy connection settings in Internet Explorer. (This step is not applicable for the Global Console.)
   - Open Internet Explorer and select Internet Options dialog from the Tools menu.
   - In the Internet Options dialog, select the Connections tab.
   - In the Connections tab, click the **LAN Settings** button.
   - In the Local Area Network (LAN) Settings dialog, in the Proxy server section, select the checkbox for "Use a proxy server for your LAN", and specify an IP address for the proxy server and port number. Click **OK** to save your changes.
   - Click **OK** in the Internet Properties dialog.

   NOTICE

   If you are using Java Web Start, it has a similar dialog in which you can specify the address for the proxy server and the port number.

Requirement for proxy server

The connection time-out mechanism implemented in the EMC Smarts Service Assurance Manager makes use of a standard TCP feature known as urgent data or out-of-band data.

The HTTP proxy used with this code must pass the out-of-band data as is.

The RFCs that define the correct operation of HTTP proxies require that the proxies provide a transparent network connection. However, they do not explicitly mention support of out-of-band data.

Because of the large number of HTTP proxy implementations in use, EMC Corporation is not in a position to indicate which, if any, fail to correctly pass out-of-band data.

Please check if the proxy server you are considering passes out-of-band data. Contact EMC Global Services if the proxy server does not pass TCP urgent data.

If the proxy server fails to pass out-of-band data, connections will time out within about three minutes. The Global Manager or Domain Manager log will contain a message of the following form:

```
CI-E-EFLOWKEEPALIVESEND-Flow closing due to missed keepalive
```

CHAPTER 9
Validating Your Deployment (Acceptance Testing)

This chapter describes how to ensure that your deployment operates as intended and contains the following information:

- Validation techniques 90

Validation techniques

When validating, begin by dividing the deployment into manageable, logical segments. Ensure that each of the segments function properly and then perform end-to-end testing. Check data flow and then check the accuracy of the data itself. Validate as much as possible before discovering the topology so that you reduce complexity.

Initial validation

To start validation, ensure that the Broker has access to all the EMC Smarts Managers and that the EMC Smarts Manager processes are registered and running. Use **brcontrol** to list the processes registered with the Broker and their status. The *EMC Smarts System Administration Guide* provides additional information.

Validating topology operations

After the topology of the network is discovered, it must be provided to the Global Manager from underlying domains or third-party software and then from the Global Manager to the Global Consoles. You must validate the topology before and after each exchange.

If the deployment includes an IP Availability Manager, ensure that you verify the topology at the Domain Manager. The *EMC Smarts IP Manager Deployment Guide* provides additional information.

Validating trap processing

Use **sm_snmp** to generate traps to the SNMP Trap Adapter (Receiver) functioning as the trap exploder.

Ensure that you send traps that cause notifications from the underlying analysis servers as well as from the SAM Adapter configuration. Create a trap for each trap processing statement in the *trap_mgr.conf* file. If ASL scripting is included in the trap processing, ensure that the scripts function as intended.

Validating users and capabilities

To validate users and their capabilities, test each user profile. Create a temporary user for each of your user profiles. Log onto the Global Console as each user in turn and review the associated console capabilities including access to console operations, access to tools, configuration of the notification list, and layout of the console. Ensure that the capabilities match your expectations for each user profile.

CHAPTER 10
Monitoring the Performance of Your Deployment

This chapter describes methods for identifying performance problems in a Service Assurance deployment and the steps that you can take to correct these problems. This chapter contains the following information:

- Assessing performance 92
- Improving performance 104

NOTICE

Regardless of the size of your deployment, always monitor its performance. If tuning is required, remember that tuning your deployment is an ongoing process that should be performed regularly.

Assessing performance

For the Global Manager, assess performance by checking the resources required by EMC Smarts processes against the operating system limits and by reviewing EMC Smarts performance metrics. The following methods helps you access performance:

- Monitoring Global Manager memory usage
- Monitoring Global Manager queues
- Monitoring EMC Smarts software with heartbeat notifications
- Monitoring topology synchronization performance
- Monitoring clients and threads
- Monitoring tool execution
- Monitoring escalation

Monitor Global Manager memory usage

Monitor memory usage by the Global Manager to ensure that it does not exceed the operating system's memory limit and as a way to detect possible performance issues.

Although monitoring memory usage should be part of a routine, these symptoms require immediate investigation:

- Fast memory growth after the Global Manager has reached a steady state, such as hundreds of megabytes of memory growth in a few hours. The steady state is typically reached sometime after startup: Be conservative and allow a few hours after startup before assuming that the Global Manager is in the steady state.
- Consistent memory growth over time, never reaching a plateau. This is relevant only if the overall growth of the Global Manager process is also hundreds of megabytes of memory, after reaching steady state.

Establish and monitor a baseline for process memory size

Under normal conditions with the Global Manager process memory size well within the operating system limit, monitor memory size to establish a baseline that can be used to detect performance issues early.

For UNIX systems, this can be automated by creating a script that periodically executes **ps -l -p proclist** and writes the output to a file. For Windows, similar functionality can be created using a batch file run by the Task Scheduler.

Investigating the values from the file helps to understand the memory growth pattern over time. Once the pattern is understood, it can be used to establish a baseline value for the memory usage of a running Global Manager.

The baseline value is then used to establish a memory usage threshold. Using a monitoring script, an alert can be generated if the memory used by the Global Manager ever exceeds the threshold.

Once the alert is received, investigate the queue growth before attempting to remedy the condition.

Monitoring Global Manager queues

Global Manager queue growth is often a cause of the Global Manager memory growth. Memory growth and queue growth are expected occasionally, due to large bursts of information processed. However, consistent growth of queues may indicate a problem.

The dump queues command

To monitor queues, EMC Smarts software provides a dump queues command that uses this syntax:

```
▼dmctl -s ‹server_name› exec dmdebug --queues
   -output=‹file_name›▲
```

where:

- ‹*server_name*› is the name of the Global Manager.
- ‹*file_name*› is the name of the file where the output is written, such as queues-010105-0100.out (where 010105 is January 1, 2005, and 0100 is 1 AM). The file is written to the *smarts/local/logs* directory.

```
SubscriberFE                                          [0 worker(s)]
Current size 0, max  369; processed entries  21843
Size          0-0.00            0-0.00            0-0.00            0-0.00
Flow      21839+0.06        21803+0.05        21446+0.05        17895+0.06
Late     0.0000+0.00       0.0000+0.00       0.0000+0.00       0.0000+0.00
```

Average over 3 minutes | Average over 30 minutes | Average over 300 minutes | Average over 3000 minutes

Figure 11 Global Manager queues information example

Analyzing queue size

The queues output lists many queues. Monitor these queues closely:

- SM_PropertyPoller: This queue contains client requests for property subscriptions and data to be sent to clients as responses to property subscriptions. For example, in the Global Manager, notifications are sent to clients through property subscriptions. So, the property poller queue contains notifications to be sent to clients.
- SubscriberFE: This queue represents a connection the Global Manager server makes to another server, where the Global Manager server subscribes to information in another server. For example, there is a SubscriberFE queue for each underlying Domain Manager. For each queue, the first line provides name of the queue and the number of workers.

NOTICE

Subscriber Front Ends like this one never have any workers, since they are not used as normal server queues.

The second line provides the current size, and an exact maximum size. Also included is the total number of entries “processed.” To be exact, this is the total number of entries that were pulled from the queue.

The next three rows provide averages. Each column reports values averaged over a different time interval: The last 3 minutes, 30 minutes, 300 minutes, and 3000 minutes. Each row reports a different value:

- Size is the queue size, that is, the average of what “Current size” reports for this moment.
- Flow is the number of queue entries processed, that is, the average of what “Current processed entries” reports.
- Late is the delay between when an entry in a timer queue was scheduled to run and when it actually ran. The Late row is always all 0's for a server queue.

Each of the averages listed in the Size, Flow, and Late rows have two fields separated by a sign (+ or -):

- The first field is the average value of the variable over the appropriate period. The queue in Figure 11 on page 93 has processed many entries -- almost 22,000 -- and once, even held 369 entries; but that was apparently an isolated peak some time in the past, since its average Size over all the reported intervals is 0: Most of the time, the queue is empty. On the other hand, the Flow value over the last 3 minutes is very close to the value for processed entries (21,843). If we look at this value averaged over longer and longer periods, we see that the longer the period, the smaller the value. This is what we would expect if the rate at which elements are flowing through this queue is roughly constant. Always keep in mind that Size grows and shrinks, while Flow can only grow.
- The second field is the average rate of change in the first value, per second. In the example above, the Size is shrinking slowly: -0.0 indicates fewer than 0.01 entries per second. On the other hand, the Flow is increasing at a steady rate: between 0.05 and 0.06 entries per second (3.0 and 3.6 entries per minute) are flowing through this queue. Again, because Flow can only increase, the second field is always non-negative. Note that the rates are virtually the same across all time periods, which is consistent with the analysis in the previous paragraph that entries are flowing through at a slow, steady rate.

Guidelines for monitoring queues

Follow these guidelines:

- The queues output lists many queues. Queues that should be monitored more closely are SubscriberFE and SM_PropertyPoller.
- Current size information is the most important to monitor. For most queues, this value is often 0. Some queues, such as ICS_ActionScheduler, almost always have non-zero values for current size. If the current size is not zero and grows over several successive dump queues commands, it should be further analyzed.

Establish and monitor a baseline for queue size

Under normal conditions, monitor queue size to establish a baseline that can be used to detect performance issues early.

To set up automatic monitoring of queues, a script may be written to parse the output of the dump queues. The script could specifically parse the current size for each queue. For example, if the current size exceeds a specified threshold, then alert someone to investigate this further. Some queues would need to be exempt from the threshold, such as ICS_ActionScheduler.

A possible threshold for queues would be 10,000. However, this setting would depend on your specific needs. It would also be possible to set the threshold lower. If you set a lower threshold, you should do more monitoring to determine whether or not there is a consistent queue growth.

If queue growth happens consistently over time, further investigation is recommended.

Monitoring with heartbeat notifications

To monitor the throughput of notifications by a Global Manager, you can implement a mechanism to create heartbeat notifications. These notifications are generated in Domain Managers or Adapter Platforms on a periodic basis. The notifications are propagated from the lower level domains to the Global Manager or through multiple levels of Global Manager.

A custom adapter listens for these notifications at the Global Manager. The custom adapter monitors the rate of receipt of the notifications from each of the underlying domains.

If a notification is not received within a predetermined period after it is generated, appropriate personnel are alerted automatically for further investigation. This investigation would entail determining if there is a consistent delay from either of the following:

- One particular EMC Smarts domain: If the delay is from one domain, analyze the domain. Consult the appropriate deployment guide for the domain.
- All EMC Smarts domains: If the delay is from all domains, analyze the Global Manager. Analyze the Global Manager memory usage, queues, and topology synchronizations as described in this chapter.

The heartbeat notifications *must* be filtered so that they are not displayed in the Global Consoles. This would add an unnecessary burden to the Global Console users. Configure the notifications lists to eliminate the heartbeat notifications.

EMC Smarts Global Services or Professional Services can aid in deploying heartbeat notifications for monitoring.

Monitoring topology synchronization

Topology synchronization information is written to the Global Manager log file. This information indicates when the Global Manager starts and ends a topology synchronization with an underlying domain. Topology synchronization is one of the more resource intensive aspects of processing in the Global Manager. Therefore, it is important when analyzing Global Manager performance to determine the following information about topology synchronizations:

- How many topology synchronizations occur over a period?
- How frequently are individual domains synchronizing topology?
- How long did each topology synchronization take?

The log file contains a significant amount of information. To extract only the topology synchronization information, use **grep**. The gnu version of **grep** provides an option to return lines above and below the matching line, which is very helpful in this case.

For example, using **/opt/gnu/bin/grep -C1 "Topology Synch"** will return an output similar to the sample in the Figure 12 on page 97. Note that the "-C1" option returns one line above and one line below the matched line.

```
[06-Jul-2004 13:57:27+522ms EST]
ICS-W-ics-event-processing.asl: 'IC-AM1' - Topology Synchronization has
    started
--
[06-Jul-2004 14:01:46+737ms EST]
ICS-W-ics-topology-sync.asl: IC-AM1' - Topology Synchronization has
    ended
--
[06-Jul-2004 14:13:46+172ms EST]
ICS-W-ics-topology-sync.asl: 'IC-AM2' - Topology Synchronization has
    ended
--
[06-Jul-2004 14:42:10+151ms EST]
ICS-W-ics-event-processing.asl: 'IC-AM1' - Topology Synchronization has
    started
--
[06-Jul-2004 14:42:57+528ms EST]
ICS-W-ics-topology-sync.asl: 'IC-AM1' - Topology Synchronization has
    Ended
```

Figure 12 Global Manager Topology Synchronization Information Example

Figure 12 on page 97 shows two topology synchronizations from IC-AM1, an IP Availability Manager, during the log file monitoring period:

- Started on July 6 at 13:57 and then ended at 14:01.
- started on July 6 at 14:42 and then ended at 14:42.

Another topology synchronization was in progress from IC-AM2 which ended at 14:13. The start time is not included in the log file.

A slow topology synchronization may not be an issue. Since topology synchronization occurs in the background, its duration does not necessarily impact general processing of the Global Manager. If there is a long latency in the connection from Global Manager to the

underlying domain that it is synchronizing with, the topology synchronization may take longer and yet not affect Global Manager. Also, topology synchronization will take longer for larger topologies than smaller topologies.

However, it may be a symptom that the Global Manager is overloaded and is trying to process more than it can handle. If the Global Manager is overloaded, you may see long topology synchronization times, slow performance when viewing information in the console, and possibly even delays in processing notifications.

Since topology synchronization is one of the Global Manager processing tasks, improving control of topology synchronization may be a way to improve performance.

For example, it is possible to make small topology changes on an underlying domain at a high frequency or larger topology changes at a low frequency on an underlying domain. Global Manager performance will be better if a low frequency of larger topology changes are made. The frequency of topology changes can be controlled partially in underlying domains by adjusting the periodic and full discovery intervals. Also, an EMC Smarts administrator can initiate discoveries. The frequency of these on-demand discoveries may affect Global Manager performance; minimize them when possible.

Another topology synchronization consideration is the number of underlying domains communicating with Global Manager and the concurrency of the topology synchronization of these. If there are a large number of domains, there is an increased likelihood of concurrent topology synchronizations of underlying domains. Since topology synchronization is one of the more intensive processes done by Global Manager, having multiple topology synchronizations concurrently may cause console performance and notification throughput to slow down.

Periodic dump threads and dump clients

EMC Smarts provides a number of different monitoring utilities, including:

- Listing thread information from a running EMC Smarts server
- Listing attached clients for a running EMC Smarts server

Listing thread information

These can be executed using the command:

```
▼dmctl -s ‹server_name› exec dmdebug dumpThreads
   -output=‹file_name›▲
```

The output of threads is shown in Figure 13 on page 99.

```
t@1     dmboot                                          RUNNING
t@5     Tracked Variable Manager                        RUNNING  Tracki
ables every 0.483248091497799 sec OOR 5.80298835530771E-09
t@6     SM_LicenseManager #1                            RUNNING
t@7     DX_MonitorControlObserver #1                    RUNNING
t@8     DX_Correlator #1                                RUNNING
t@9     DX_CorrelationTimer #1                          RUNNING
t@10    DX_CorrelatorEngine #1                          RUNNING
t@11    DX_MonitorControl #1                            RUNNING
t@12    DX_Dispatcher #1                                RUNNING
t@13    DX_ModelObserver #1                             RUNNING
t@14    DX_ModelManagerProcessor #1                     RUNNING
t@18    ICS_ActionScheduler::ICS-ActionScheduler #1     RUNNING
t@19    ICS Auto Action Timer #1                        RUNNING
t@20    SubscriberFE-EventTimer #1                      RUNNING
t@21    ICS-Impact-Driver                               RUNNING
t@22    SubscriberFE-EventTimer #1                      RUNNING
t@23    SM_PropertyPoller #1                            RUNNING
t@24    HistoryOnlyAccessor polling #1                  RUNNING
t@25    monitored_property propagate retry              RUNNING
```

Figure 13 Global Manager threads information example

Figure 13 on page 99 lists running threads in the server. This gives an idea of the type of processing done in the server. This is often used in conjunction with a **pstack** output. This information is not typically monitored, but often used in investigating issues.

Listing client information

This command displays a list of all clients attached to an EMC Smarts server, including adapters.

▼`dmctl -s` *‹server_name›* `exec dmdebug dumpClients -output=`*‹file_name›*▲

Figure 14 on page 100 shows an example of dump clients output.

```
Remote Clients [21-Oct-2004 10:05:05 GMT] -
  Id     User        Privs Host                   PID     Since                     Description
---------------------------------------------------
  5      AdministratorPING+READNETVISTAM42-SAM   unknown  18-Oct-2004 11:08:21 GMT  IC Console
   Input:
    Socket 101, mode Read; 0/2048 bytes, lasterr 0, no timeout
    Desktop1:2180 -> 192.6.17.52:3073
    00000000 00000006 526F7574 65720000 000B5243 4C4F4E35 30303938 30000000
   Output:
    Socket 101, mode Write; 0/2048 bytes, lasterr 0, no timeout
    Desktop1:2180 -> 192.6.17.52:3073
    02000000 0D000000 06526F75 7465726B 436F6E6E 65637469 6F6E4F62 6A656374
  10     root        ALL   server1                1061    18-Oct-2004 11:08:21 GMT  ASL
  Input:
   Socket 151, mode Read; 0/2048 bytes, lasterr 0, no timeout
   Server1:2180 -> server2:63325
   00000048 00000014 4943535F 4E6F7469 66696361 74696F6E 4C697374 0000000E
  Output:
   Socket 151, mode Write; 0/2048 bytes, lasterr 0, no timeout
   Server1:2180 -> server2:63325
   28284950 20416464 72657373 2050726F 626C656D 293A202E 312E332E 362E312E
```

Figure 14 Global Manager client information example

The example in Figure 14 on page 100 shows one console client, “IC Console”, and one ASL client, “ASL”. The output gives the time the connection was initiated, the privileges associated with the connection, and the host the client runs on.

Monitoring tools

Tools are programs that can be invoked automatically by the Global Manager or through the Global Console by an operator. The tools should be monitored using the following EMC Smarts utility to ensure that they do not adversely affect performance:

▼`dmctl -s` *‹server_name›* `invoke ICS_ActionManager::ICS-ActionManager dumpStatistics` *‹string› filename*▲

Figure 15 on page 101 illustrates the output of the Dumpstatistics option for tools.

```
Synchronous Action Statistics [28-Dec-2004 03:46:22 PM Eastern Standard Time]
Summary :
 number_of_sync_actions_executed    :           7
 max_execution_time                 :          30.6950 secs
 avg_execution_time                 :           9.3593 secs

 number_of_successfull_sync_actions :           5
 number_of_timeout_sync_actions     :           2
 number_of_failed_sync_actions      :           0

Details :
 Action ID : ICS-ServerProgramAction-Sample - Close Trouble Ticket
  No of times invoked         :           1
  Maximum Execution time      :           1.3720 secs
  Average Execution time      :           1.3720 secs
  No of Successful Execution  :           1
  No of Timeouts              :           0
  No of Failures              :           0

 Action ID : ICS-AutoProgramAction-Set UserDefined1
  No of times invoked         :           1
  Maximum Execution time      :           0.5410 secs
  Average Execution time      :           0.5410 secs
  No of Successful Execution  :           1
  No of Timeouts              :           0
  No of Failures              :           0

 Action ID : ICS-ServerProgramAction-Sample - Open Trouble Ticket
  No of times invoked         :           1
  Maximum Execution time      :           1.8720 secs
  Average Execution time      :           1.8720 secs
  No of Successful Execution  :           1
  No of Timeouts              :           0
  No of Failures              :           0

 Action ID : ICS-AutoProgramAction-Esc-Paging
  No of times invoked         :           4
  Maximum Execution time      :          30.6950 secs
  Average Execution time      :          15.4325 secs
  No of Successful Execution  :           2
  No of Timeouts              :           2
  No of Failures              :           0
```

Figure 15 Output of the Dumpstatistics option for tools

The tools monitored by the utility include all synchronous tools (server and automatic). Asynchronous tools (client tools), which are client tools invoked by users from the Global Console, are not monitored using this utility.

Monitoring escalation

EMC Smarts software provides a utility that can monitor escalation, including statistics for each escalation level in every escalation path configured in the Global Manager. The syntax is as follows:

```
▼dmctl -s ‹server_name› invoke
   ICS_AutoActionManager::ICS-AutoActionManager dumpStatistics
   ‹string›fileName▲

 [20-May-2003 05:49:05 PM EDT]
 Escalation Statistics (All times are in seconds)
 OVERALL:
 number_of_executed_jobs  :          122
 total_execution_time     :      10.0430
 max_execution_time       :       0.3024
 avg_execution_time       :       0.0823
 total_queue_delay        :       5.0068
 max_queue_delay          :       0.2419
 avg_queue_delay          :       0.0410

 LEVEL Policy1-Path1-0:
 number_of_executed_jobs  :            4
 total_execution_time     :       0.8123
 max_execution_time       :       0.2856
 avg_execution_time       :       0.2031

 LEVEL Policy1-Path1-1:
 number_of_executed_jobs  :            0
 total_execution_time     :       0.0000
 max_execution_time       :       0.0000
 avg_execution_time       :       0.0000

 LEVEL Policy1-Path1-2:
 number_of_executed_jobs  :            0
 total_execution_time     :       0.0000
 max_execution_time       :       0.0000
 avg_execution_time       :       0.0000

 LEVEL Policy2-Path2-0:
 number_of_executed_jobs  :          118
 total_execution_time     :       9.2307
 max_execution_time       :       0.3024
 avg_execution_time       :       0.0782
```

Figure 16 Output of the Dumpstatistics option for escalation

Figure 16 on page 102 shows the sample output for the Dumpstatistics option for escalation. The following values are included as a total for all escalation paths and for each level in each path:

- Number of executed jobs. A job is a scheduled notification that matched a policy-path and that repeats for each level in that path. The number of executed jobs is indicated for all levels in all paths (overall) and for each level.
- Total, maximum, and average execution times. Execution time indicates how long the tool actions defined for a job took in seconds. If the maximum execution time at a level far exceeds the average, it may indicate an issue for a particular tool.
- Total, maximum, and average queue delays for all the escalation paths is also displayed. These delays are not related to execution time. Delay is difference between when a job was scheduled to execute and when the job was actually executed. A positive value for the delay indicates that the system is not keeping up with scheduled escalation. Delay may be caused by the overall load on the server, number of notifications, policies, paths, or levels. When a burst of notifications make it into a path, all jobs are scheduled to execute at the same time in a future time, so delay may increase due to these bursts.

"Increase worker threads for escalation" on page 105 provides information on improving escalation performance.

Improving performance

If the performance metrics indicate a performance degradation, do the following to improve performance:

- Refine notification lists for users
- Reassess the topology synchronization configuration
- Refine sets of trap, syslog notifications sent to SAM
- Increase worker threads used for escalation
- Redesign your deployment to deploy additional Global Managers in a hierarchical Service Assurance configuration

Refine notification lists for users

A Notification List determines the events that are forwarded to a user. Essentially, the list filters the notifications that are sent from the Global Manager and can be assigned to one or more users.

In large Service Assurance deployments, significant numbers of notifications are active at any given time. Displaying all of these notifications to all users may adversely affect the performance of the system. Instead, create notification lists that are focused on providing groups of users with only the notifications that are specifically useful to them. This approach will improve performance and may also simplify users' tasks by allowing them to focusing their efforts on appropriate notifications.

Reassess configuration of topology synchronization

"Designing for Topology Operations" on page 55 describes the frequency of topology changes that can be controlled partially in underlying domains by adjusting the periodic and full discovery intervals. If the pace of change in your network and application infrastructure can be supported with longer discovery intervals, increasing the interval may also improve Global Manager performance.

Because discovery can also initiated manually, another method to improve performance may be to reduce the frequency of manual discoveries. Reassess access to unplanned discovery and remove it from the user profiles of administrators who do not require it. For administrators who retain access to manual discovery, educate them on the effect of initiating an unscheduled discovery.

Refine set of trap and syslog notifications sent to Global Manager

The notifications that result from analysis by EMC Smarts Managers typically contribute very little to the notification load on Global Manager. During analysis, EMC Smarts Managers reduce the load by analyzing numerous traps and polling results to produce a notification for each root cause rather than the symptoms.

Do not forward traps to the Global Manager when they are already sent to a Domain Manager for analysis. Use the first match in trap forwarding configuration file, if possible. If you have not implemented batching at the EMC Smarts SNMP Trap Adapter, do so.

"Designing trap processing" on page 62 provides additional information.

Increase worker threads for escalation

If the performance metrics indicate a performance degradation in Escalation, add worker threads to improve performance. Increasing threads to 10 worker threads is acceptable, but keep in mind that more threads may require additional or more capable CPUs. Be conservative when increasing threads, since this action may reduce the overall resources that are available and affect performance. Increase the value in the ics-default.xml file. The *EMC Smarts Service Assurance Manager Configuration Guide* provides information on reconfiguring the Global Manager.

NOTICE

Once increased, the number of worker threads cannot be reduced unless the server is restarted.

Upgrade server hardware

Improve the capabilities of the equipment where the EMC Smarts process is installed by using a faster CPU or reinstall EMC Smarts software on a more capable equipment. Before resorting to this hardware upgrade, consider the previous options.

Redesign the Service Assurance deployment

If, after attempting the other methods to improve performance, there are still issues, then determine if your actual deployment has exceeded the original deployment design.

If this is the case, you may have to deploy additional Global Managers in a hierarchical Service Assurance configuration. To start redesigning a deployment, return to "Designing the Service Assurance Deployment" on page 35.

APPENDIX A
Design and Deployment Checklists

This appendix gather the checklists that are included throughout this guide into a single, easily accessible location. The checklists included in this chapter are:

- Before you begin checklist 108
- Basic information checklist 109
- EMC Smarts design checklist 110
- Designing topology operations checklist 110
- Planning for notifications checklist 111
- EMC Smarts client and user design checklist 112
- Business Impact Manager design checklist 113

Before you begin checklist

Table 15 Before you begin checklist

Complete	Requirement	Description
❑	Possess an understanding of the EMC Smarts architecture and capabilities.	At a minimum, you must understand the concepts and EMC Smarts architecture described in the following documents: • *EMC Smarts Service Assurance Manager Introduction* • *EMC Smarts Service Assurance Manager Configuration Guide* • *EMC Smarts Service Assurance Manager Adapter Platform User Guide* • *EMC Smarts System Administration Guide* • *EMC Smarts Installation Guide for SAM, IP, ESM, MPLS, VoIP, and NPM Managers* that accompanied your software product • EMC Smarts user guides related to the underlying EMC Smarts domains in your deployment
		To improve your understanding, attend EMC Smarts training courses offered by EMC® Education Services and Development. Typically, deployment requires the knowledge equivalent to what is provided in the training courses on: • EMC Smarts Global Manager • EMC Smarts Service Assurance Manager Adapter Platform • Any underlying EMC Smarts domain that will be part of your deployment.
❑	Obtain contact information for the deployment team.	The contact list should include titles, responsibilities, and contact methods for all team members.
❑	Get nondisclosure requirements and negotiate an agreement.	Be aware of the requirements of the non-disclosure agreements that are in place for the EMC Smarts deployment.
❑	Develop schedules and set milestones for early deliverable.	Scheduling a software deployment varies based on the size and scope of the deployment and the organization's requirements. Typical milestones might include: • Initial project meeting to define the deployment scope • Purchase of EMC Smarts software • Project development begins • Installation in test environment complete • Testing complete • Installation in production environment complete • EMC Smarts goes live Additional information on scheduling is beyond the scope of this guide.

Basic information checklist

Table 16 Basic Service Assurance information checklist

Complete	Task	Description	Related documentation
o	Determine the number of managed devices and applications in the environment.	Document all quantities in the deployment build guide.	"Determine how many devices and applications will be managed" on page 29
o	Estimate potential growth in quantity of managed devices.	The EMC Smarts deployment must support potential network growth. Estimate the growth over a specific time period. Document the calculations in the deployment build guide.	"Determine how many devices and applications will be managed" on page 29
o	Get the organization's testing/acceptance requirements.	Your design may be required to meet test and acceptance requirements. Obtain any specifications that cover integration testing, user acceptance testing, and operational acceptance testing. You may be required to write an installation or deployment report that follows an organization's particular standards.	"Determine requirements for installing software" on page 30
o	Describe the organization's requirements for installing new software.	Document these requirements and how the design meets them in the deployment build guide.	"Determine requirements for installing software" on page 30
o	List the products that currently monitor the network.	Document the products (including version) in deployment build guide.	"Define requirements for integrating existing software with EMC Smarts" on page 30
o	Describe the network security.	Describe security features that will affect the deployment. Document the security features in the deployment build guide.	"Gather network security information" on page 31
o	List any other network requirements or features that may affect the deployment.	Document the features in the deployment build guide.	"Other network features affecting EMC Smarts" on page 32

EMC Smarts design checklist

Table 17 Initial design checklist

Complete	Task	Description	Related documentation
o	Determine the configuration of the Service Assurance Deployment.	Determine Service Assurance configuration, including the numbers of Global Managers and how they are organized in the EMC Smarts deployment.	"Determine the Service Assurance configuration" on page 36
o	Determine the hardware required to support the Service Assurance Deployment.	For each Global Manager in the Service Assurance configuration, determine the type of equipment required to support it.	"Hardware for Service Assurance platform equipment tiers" on page 43
o	Determine the location of the EMC Smarts Broker	Choose a location for the EMC Smarts Broker that will support the EMC Smarts deployment.	"Locating EMC Smarts Brokers" on page 48
o	Determine the location of the License Server	Choose a location for the EMC Smarts License Server that will support the EMC Smarts deployment.	"Locating license servers" on page 49
o	Determine the location of the EMC Smarts Global Manager(s)	Choose locations for the one or more EMC Smarts Global Manager(s) that will support the EMC Smarts deployment.	"Locating the Global Managers in the network" on page 50
o	Determine if Service Assurance Manager must be configured with failover.	Determine if a high availability configuration is required.	"Considering failover (high availability) configurations" on page 51
o	Determine if acceptance tests are required.	Acceptance tests may be required for different portions of EMC Smarts functionality. Be aware of the requirements and develop acceptance criteria.	"Designing acceptance tests" on page 52

Designing topology operations checklist

Table 18 Designing topology operations checklist

Complete	Task	Description	Related documentation
o	Review topology synchronization issues.	Ensure that you understand how topology synchronization affects Global Manager performance.	"Understanding topology synchronization" on page 56
o	Plan topology synchronizations.	To optimize Global Manager performance, it is best to reduce the number of topology synchronizations where possible.	"Planning for topology synchronization" on page 57
o	Plan topology groups.	Topology grouping is a visual tool that does not affect Global Manager performance. Groups help organize topology elements.	"Designing topology groupings" on page 58

Planning for notifications checklist

Table 19 Planning for notifications checklist

Complete	Task	Description	Related documentation
o	Determine which notifications are important for EMC Smarts operators.	Ensure only the notifications that are useful to Global Console operators reach the Global Manager.	"Overview of notification processing" on page 60
o	Configure Notification Lists.	Notification lists are the mechanism that clients use to subscribe to and receive notifications from a Global Manager.	"Configuring notifications for clients" on page 61
o	Design Trap Processing.	Any EMC Smarts deployment should include a plan to handle trap processing.	"Recommended trap processing design" on page 62

EMC Smarts client and user design checklist

Table 20 EMC Smarts client and user design checklist

Complete	Task	Description	Related documentation
o	Define functional groups for users.	List the broad functional groups that users will belong to and then define the needs and duties of each group. Add this information to the deployment build guide.	"Determining the Global Manager users" on page 68
o	Choose appropriate EMC Smarts clients for users.	EMC Smarts consoles have different capabilities. For each type of user and their equipment, you must determine which console is most appropriate. Document your choices in the deployment build guide.	"Choose a console for types of users" on page 68
o	Define how you will implement security for user groups.	When defining the functional categories for users, consider EMC Smarts's security implementation. Document the implementation in the deployment build guide.	"Users and security" on page 69
o	Define the notification lists in EMC Smarts user profiles.	The Notification List determines which notifications are sent from the Global Manager to the EMC Smarts client. Carefully define the notification lists as they can affect Global Manager performance. Document the notification list provided for each user profile in the deployment build guide.	"Designing notification lists" on page 71
o	Design console operations access.	Most operations that can be performed at the EMC Smarts clients can be individually enabled or disabled in the user profiles. Document the access provided for each user profile in the deployment build guide.	"Designing consoles" on page 72
o	Design console layouts.	Specific console layouts can be saved and then automatically provided to match user needs at the Global Console, Web Console, and Business Dashboard. Document the console layout used in each user profile in the deployment build guide.	"Designing consoles" on page 72
o	Design client and server tools.	Define the necessary tools that users must access and the levels of programming skills required to create the tools. Document this information and tools provided by each user profile in the deployment build guide.	"Planning for tools and tool deployment" on page 72
o	Define functional groups for users.	List the broad functional groups that users will belong to and then define the needs and duties of each group. Add this information to the deployment build guide.	"Determining the Global Manager users" on page 68

Business Impact Manager design checklist

Table 21 Business Impact Manager design checklist

Complete	Task	Description	Related documentation
o	Define relevant business processes.	Determine the business processes to model.	"Designing business impact processing" on page 76
o	Create a business topology.	Create business elements to model the business processes. Define the expectations of the model and the events it should analyze.	"Business topology" on page 76
o	Assign weights to the elements of the topology.	This step is optional but allows you to assign a value to measure the impact of events that occur in the topology. The weights of impacted elements are summed in the root-cause event.	"Weights: Impacts of issues in the topology" on page 77
o	Choose a method for importing information.	If the topology is small, use the Topology Builder Console. For larger topologies, use one or more topology files.	"Importing information" on page 77

APPENDIX B
Service Assurance Manager Benchmarks

This appendix contains Service Assurance Manager (SAM) benchmarking statistics and includes the following information:

- Service Assurance Manager benchmarking deployment 116
- Number of syslog messages a given Smarts syslog parser can successfully handle per second..... 123
- Number of escalation policies that can successfully be processed using multiple threads 124
- Service Assurance Manager 8.0 performance benchmarking 126
- Time to log in to the SAM 8.0 server 127
- Default Global Console heap size allocation 128
- SAM 8.1 latency performance benchmark testing 129
- Amount of memory SAM 8.1 requires to process notifications 131

Service Assurance Manager benchmarking deployment

The following software and hardware configuration was used in conducting the SAM 9.1 benchmarking:

- Software
  - Service Assurance Manager 9.1
  - Availability Manager-Performance Manager 9.1
- Hardware

  Table 22 on page 116 defines the hardware used:

Table 22 Service Assurance Manager benchmarking hardware

Hardware	Operating system	CPU quantity	CPU Model	Memory
PowerEdge R310 - Dell	Red Hat Enterprise Linux Server release 5.8	4	Intel(R)Xeon(R)CPU X3450 @ 2.67GHz	16 GB

The following three setups were used to deploy the Service Assurance Manager components and domain manager.

- Figure 17 on page 117 shows Service Assurance Manager deployment with one trap adapter and no trap exploder
- Figure 18 on page 118 shows Service Assurance Manager deployment with one trap adapter and one trap exploder
- Figure 19 on page 119shows Service Assurance Manager deployment with two trap adapters and one trap exploder

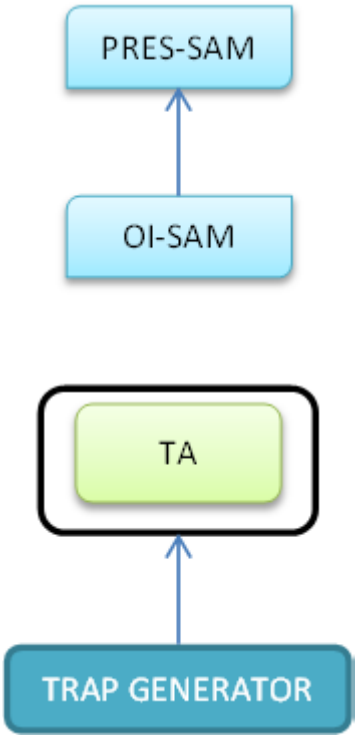

Figure 17 Service Assurance Manager deployment with one trap adapter and no trap exploder

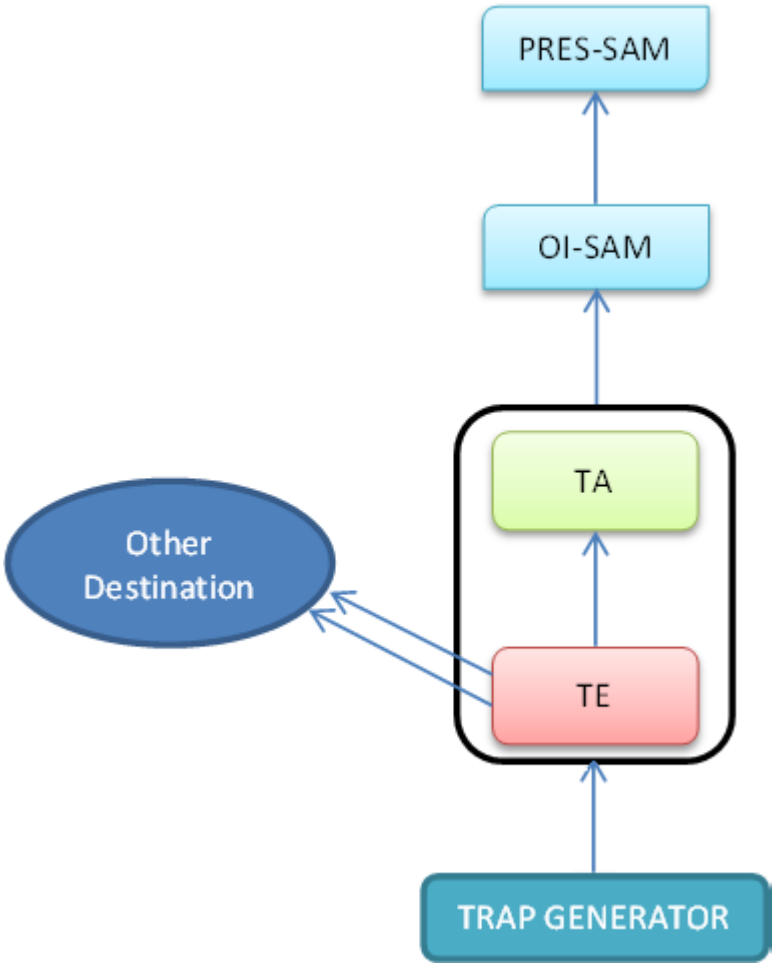

Figure 18 Service Assurance Manager deployment with one trap adapter and one trap exploder

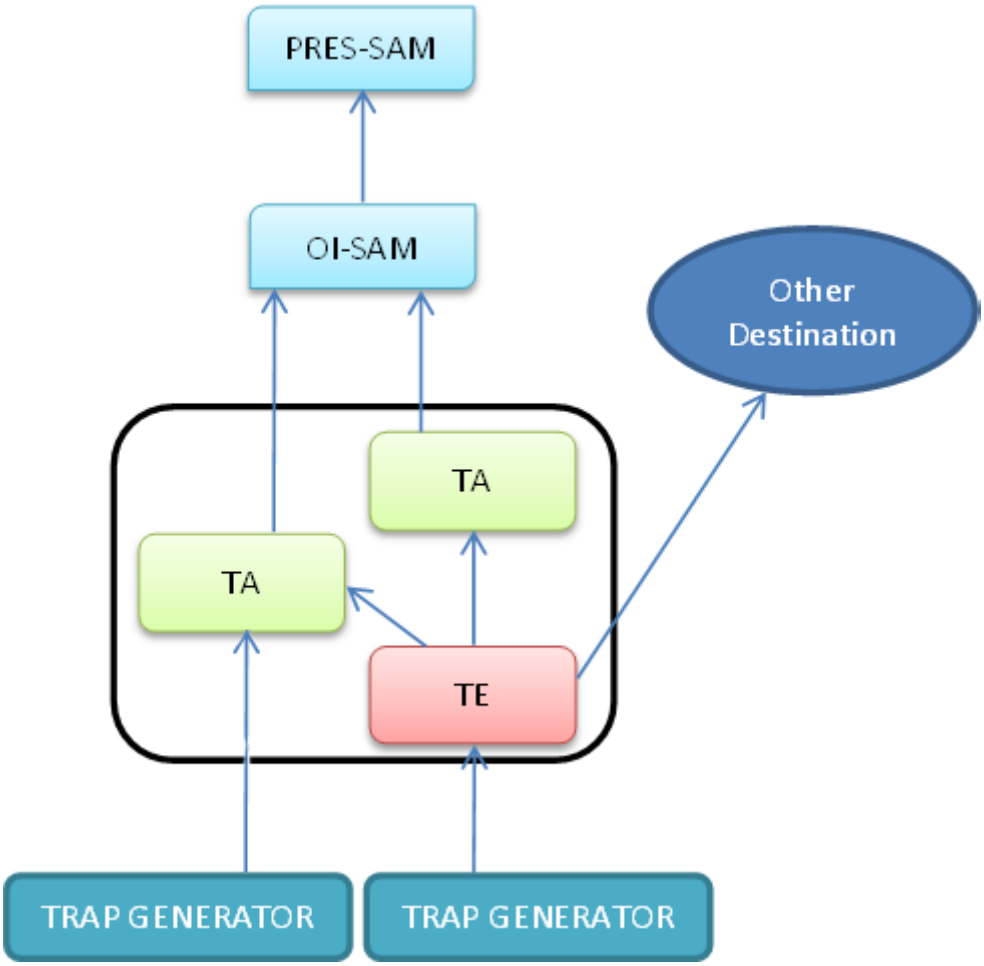

Figure 19 Service Assurance Manager deployment with two trap adapters and one trap exploder

Number of traps Service Assurance Manager can successfully process

This benchmark defines how many traps per second the Service Assurance Manager can successfully process. The following points were considered when setting up the benchmark environment:

- The benchmark is based on receiving a mix of new notifications from multiple traps
- The trap adapter will be tested with, and without ASL scripts in use
- The trap adapter is flooded using a trap generator with traps ranging from 2000 to 10000 TPM to reach it's optimum limit
- SAM servers, Trap Exploder, Trap Adapter are running on the same machine.

The following configurations were tested:

- One trap adapter, no trap exploder, ASL script disabled
- One trap adapter, with trap exploder, ASL script disabled
- One trap adapter, with trap exploder, ASL script enabled
- One trap adapter, with trap exploder, ASL script enabled, with Queue limits settings
- Two trap adapters, with trap exploder, ASL script enabled

One trap adapter, no trap exploder, ASL script disabled

The procedure for benchmarking one trap adapter with ASL script disabled included:

- Generating consistent inflow of traps each minute directly to Trap Adapter.
- Without Queue Limits set
- Match=ALL

Table 23 on page 120 defines the results for a one trap adapter, no trap exploder, ASL script disabled, with 10 threads configuration.

Table 23 One trap adapter, no trap exploder, ASL script disabled

	Trap exploder	Trap adapter	Threads	ASL script
Trap rate	NA	100 T/s	10	NA

Legend
T/s – Traps per second

One trap adapter, with trap exploder, ASL script disabled

The procedure for benchmarking one trap adapter with a ASL script disabled included:

- Generating consistent inflow of traps each minute directly to Trap Exploder.
- Trap Exploder forwards traps to Trap Adapter.
- Without Queue Limits set
- Match=ALL

Table 24 on page 121 defines the results for a one trap adapter, with trap exploder, ASL script disabled, with 10 threads configuration.

Table 24 One trap adapter, with trap exploder, ASL script disabled

	Trap exploder	Trap adapter	Threads	ASL script
Trap rate	133.33T/s	100 T/s	10	NA

Legend
T/s – Traps per second

One trap adapter, with trap exploder, ASL script enabled

The procedure for benchmarking one trap adapter with a ASL script enabled included:

- Generating consistent inflow of traps each minute directly to Trap Exploder.
- Trap Exploder forwards traps to Trap Adapter.
- Without Queue Limits set
- Match=ALL

Table 25 on page 121 defines the results for a one trap adapter, no trap exploder, ASL script disabled, with 10 threads configuration.

Table 25 One trap adapter, with trap exploder, ASL script enabled

	Trap exploder	Trap adapter	Threads	ASL script
Trap rate	133.33 T/s	33.33 T/s	10	Yes

Legend
T/s – Traps per second

One trap adapter, with trap exploder, ASL script enabled, with Queue Limit settings

The procedure for benchmarking one trap adapter with a ASL script enabled included:

- Generating consistent inflow of traps each minute directly to Trap Exploder.
- Trap Exploder forwards traps to Trap Adapter.
- With Queue Limits set to 200 MB
- Match=ALL

Table 26 on page 121 defines the results for a one trap adapter, no trap exploder, ASL script disabled, with 10 threads configuration.

Table 26 One trap adapter, with trap exploder, ASL script enabled

	Trap exploder	Trap adapter	Threads	ASL script
Trap rate	133.33 T/s	50 T/s	10	Yes

Legend
T/s – Traps per second

Two trap adapters, with trap exploder, ASL script enabled

The procedure for benchmarking one trap adapter with a ASL script enabled included:

- Generating consistent inflow of traps each minute directly to Trap Exploder and Trap Adapter.
- Trap Generator forwards traps to Trap Exploder.
- Trap Exploder forwards traps to first Trap Adapter.
- Another instance of Trap Generator forwards traps directly to second Trap Adapter.
- Without Queue Limits set
- Match=ALL

Table 27 on page 122 defines the results for a two trap adapters, with trap exploder, ASL script enabled, with 10 threads configuration.

Table 27 Two trap adapters, with trap exploder, ASL script enabled

	Trap exploder	Trap adapter 1	Trap adapter 2	Threads	ASL script
Trap rate	133.33 T/s	33.33 T/s	33.33 T/s	10	Yes

Legend
T/s – Traps per second

Number of syslog messages a given Smarts syslog parser can successfully handle per second

The following configurations were benchmarked:

- Syslog adapter using the --file option
- Syslog adapter using the -- tail option

Syslog adapter using --file option

When a file containing all syslog entries was processed, both the SAM aggregate server and the SAM presentation server approximately 55 notifications per second in a SAM 8.0 environment.

Table 28 on page 123 defines the results for a syslog adapter using the --file option configuration.

Table 28 SAM 8.0 Syslog adapter using the --file option benchmarking results

	Syslog Adapter	Adapter Platform (OI)	SAM 8.0 aggregate server	SAM 8.0 presentation server
Syslog Rate	57.18 S/s	57.18 S/s	56.88 N/s	55.59 N/s

Legend
S/s – Syslogs per second
N/s – Notifications per second

Syslog adapter using --tail option

When a file was appended with 33,333 syslog entries a second, both the SAM aggregate server and the SAM presentation server processed 58.65 notifications per second in a SAM 8.0 environment.

Table 29 on page 123 defines the results for a syslog adapter using the --tail option configuration.

Table 29 SAM 8.0 Syslog adapter using the --tail option benchmarking results

	Syslog Adapter	Adapter Platform (OI)	SAM 8.0 aggregate server	SAM 8.0 presentation server
Syslog Rate	58.69 S/s	58.69 N/s	58.65 N/s	58.65 N/s

Legend
S/s – Syslogs per second
N/s – Notifications per second

Number of escalation policies that can successfully be processed using multiple threads

When 50,000 traps were sent to the trap adapters, the rate to compute the escalation of notifications at the Service Assurance Manager presentation server was measured for 5, 10, 20, and 50 escalation policies. For each escalation policy:

- The following path and levels were used: 1 path and 1 level, 5 paths and 3 levels, and 10 paths and 6 levels
- Each notification matches exactly one policy per path

NOTICE

Each escalation, path, and level action consisted of running a unique shell script that created an empty file in the local file system.

Table 30 on page 124 defines the escalation policy benchmarking results for SAM 8.0.

Table 30 Escalation policy benchmark results

Configuration policy/ path/ level	Adapter Platform (OI) N/s	SAM-Aggregate N/s	SAM-Presentation N/s	Elapsed time in seconds	Queue growth	Memory start in KB	Memory end in KB	Delay in completion in seconds	Notification average escalation time in seconds	Time to load XML file
5/1/1	37.48	35.41	13.94	3587	2	264448	419432	0	0.1987	3.6
10/1/1	41.91	40.23	13.14	3805	6	283944	438064	0	0.2016	3.6
20/1/1	42.37	40.52	12.42	4025	5	283376	433032	0	0.2108	3.8
50/1/1	40.00	37.94	12.53	3990	2	268864	422168	0	0.207	4.3
5/5/3	38.70	37.06	5.48	9131	896	274960	516096	2 min.	121.0753	5.3
10/5/3	32.98	30.40	5.87	8642	1022	230984	472424	2 min.	121.2158	6.8
20/5/3	37.97	36.34	5.21	9721	836	283760	514104	2 min.	121.1235	19.7
50/5/3	35.46	32.81	5.47	9269	885	238376	478216	2 min. 1 sec.	121.0869	69.3
5/10/6	32.24	28.97	2.93	17359	1857	219896	588280	5 min. 2 sec.	325.505	15.2
10/10/6	37.12	32.94	2.38	21350	1686	269768	625400	5 min. 1 sec.	311.5431	47.6
20/10/6	37.12	35.04	2.60	19567	1013	283224	628792	5 min. 13 sec.	305.2938	253
50/10/6	33.94	31.87	2.56	19808	1082	288608	402664	5 min. 2 sec.	305.0976	1146.8

Legend

- * Queue name is ICS Auto Action Time. If no value, the queue file was not created.
- N/s – Notifications per second.
- NG – No queue growth.
- Elapsed Time in Seconds is the time to execute all escalations.

- min. – minutes.
- sec. – seconds.
- Delay in Completion in Seconds is the excess time to complete the execution of the escalations This is measured as the difference between the time the last notification was received by the Service Assurance Manager server and the time the last escalation was completed.
- Time to load XML file defines the time it took to load the escalation policies configuration file into the Service Assurance Manager server. This file is in XML format and is loaded into the server using the sm_config command.

Service Assurance Manager 8.0 performance benchmarking

Service Assurance Manager 8.0 performance benchmarking focused on the following functionality:

- Displaying containment views
- Displaying maps
- Processing notifications
- Performing topology synchronization

Environment

The Service Assurance Manager 8.0 performance benchmarking environment consisted of multiple:

- Hierarchical Service Assurance Manager servers
- Underlying domains (at a minimum four underlying domains, including IP and NPM)
- Trap adapters

Table 31 on page 126 defines the hardware configuration used for the performance benchmarking.

Table 31 Performance benchmarking hardware configuration

Type	Model	Platform	Memory size
Server System	Sun SPARC Enterprise T5220	sun4v/1.1GHz	16256 MB

Scenario

The performance benchmarking scenario included 100 concurrent clients responsible for the same number of notifications, performing different operations, such as taking ownership of an event, releasing ownership of an event, getting properties of an event, getting a map, and getting containment views.

Results

During a typical scenario with 100 concurrent users logged in and with the SAM server receiving 20 event notifications a minute and the console displaying 1,200 event notifications when logged in, a containment view displays in approximately 5 seconds.

Time to log in to the SAM 8.0 server

This section describes the performance benchmark for logging in to the SAM 8.0 server.

Environment

Table 32 on page 127 defines the hardware configuration used in benchmarking the time to log in to the SAM 8.0 server.

Table 32 Time to log in to the SAM 8.0 server benchmarking hardware configuration

Type	Model	Platform	Memory size
Server	Sun Fire V240	sun4v / 1.5 GHz	4 GB
Console	Windows	2003 SP2 / 3 GHz	1 GB

NOTICE

The environment for this benchmark did not include any network latency.

Scenario

This benchmark included the following scenarios:

- 40,000 total notifications with 21 concurrent clients logging in to the SAM server at the same time.
- 40,000 total notifications with 20 concurrent clients logged in and performing operations and 1 client logging in to the SAM server.
- 20,000 total notifications with 21 concurrent clients logging in at the same time.
- 20,000 total notifications with 20 concurrent clients logged in and performing operations and 1 client logging in to the SAM server.
- 40,000 total notifications with 1 client logging into the SAM server.

Results

Table 33 on page 128 defines the benchmarks for logging into the SAM 8.0 server.

Table 33 Logging into the SAM 8.0 server benchmark

Scenario	Benchmark
40,000 total notifications with 21 concurrent clients logging in to the SAM server at the same time.	1 minute and 51 seconds
40,000 total notifications with 20 concurrent clients logged in and performing operations and 1 client logging in to the SAM server.	1 minute and 44 seconds
20,000 total notifications with 21 concurrent clients logging in at the same time.	51 seconds
20,000 total notifications with 20 concurrent clients logged in and performing operations and 1 client logging in to the SAM server.	46 seconds
40,000 total notifications with 1 client logging into the SAM server.	1 minute and 46 seconds

Default Global Console heap size allocation

This section describes the default heap size allocation for the SAM 8.0 Global Console.

Environment

Table 34 on page 128 defines the hardware configuration used to determine the default Global Console heap size allocation.

Table 34 Default Global Console heap size allocation benchmarking hardware configuration

Type	Model	Platform	Processor	Memory size
Server	Windows	2003 SP2	2.66 GHz	2 GB

Recommendations

The SAM 8.0 default heap size allocation for the Global Console is 350 MB. A heap size of 350 MB is sufficient for deployments having 50,000 notifications.

NOTICE

The default heap size allocation for previous versions of SAM was 256 MB.

For the Web Console, the default heap size is 64 MB. A heap size of 64 MB is sufficient for SAM deployments with 15,000 notifications. For deployments with more than 50,000 notifications the heap size should be increased to 350 MB. The *EMC Smarts Service Assurance Manager Dashboard Configuration Guide* provides detailed heap size considerations for the Web Console and Business Dashboard applets.

SAM 8.1 latency performance benchmark testing

SAM 8.1 latency performance benchmark testing consisted of introducing latency in a wide area network environment and testing the:

- Notification processing time between the presentation SAM server and the underlying aggregate SAM domain managers
- Topology synchronization time between the SAM server and the underlying servers

Environment

Router connections governing communication between two hosts were adapted to cause network latency delays of less than 1 millisecond (ms), 100 milliseconds, and 300 milliseconds. Transactions between domain managers were then monitored at each latency level over the router connection.

The specific environment specifications include:

- Host specifications include:
  - 4 GB of RAM
  - 2 CPUs
  - 60 GB of storage
- Static notification set:

  All notifications used in the notification synchronization were included in the aggregate SAM repository. Additional notifications were *not* added during the test.
- Topology synchronization repository specifications include:
  - 1,200 systems
    - Instances — 17,862 (17,859 added 0 deleted)
    - Relationships — 35,306 (79,398 connections)
  - 2,600 systems
    - Instances — 21,841 (21,838 added 0 deleted)
    - Relationships — 43,528 (99,066 connections)
  - 3,600 systems
    - Instances — 26,951 (26,948 added 0 deleted)
    - Relationships — 53,204 (121,016 connections)

Scenarios

Scenario 1

Table 35 on page 130 defines the time required to process notifications between two SAM domain managers, an underlying aggregate SAM server and a top-level presentation SAM server.

Table 35 Latency results for SAM-to-SAM notification processing

Latency	1 ms	100 ms	300 ms
~14500 notifications	1:45	2:37	3:32
~32500 notifications	3:10	4:07	5:00
~35100 notifications	3:46	4:20	5:53

NOTICE

The format for the time measurement is minutes:seconds.

Scenario 2

Table 36 on page 130 defines the time required to complete initial topology synchronization between an IP domain manager (for example, AM and AMPM) and a SAM domain manager.

Table 36 Latency results to complete an initial topology synchronization between an IP domain manager and a SAM domain manager

Latency	1 ms	100 ms	300 ms
~1200 systems	2:35:735	2:39:962	2:51:916
~2600 systems	4:20:484	4:21:969	4:55:593
~3600 systems	4:41:862	4:53:813	5:40:516

NOTICE

The format for the time measurement is minutes:seconds:milliseconds.

Results

Latency did have an effect on processing time. However, the number of objects being monitored and the complexity of the repository structure did greatly influence the processing time.

Amount of memory SAM 8.1 requires to process notifications

This benchmark determines the amount of memory SAM 8.1 requires to process notifications.

Environment

The benchmark environment consisted of:

- SAM server version 8.1 running on:
  - LinuxAS5 physical machine
  - Red Hat Enterprise Linux Server release 5, Kernel 2.6.18-8.el5 on an x86_64, with 4 GB of memory
  - 4 Intel® Xeon™ CPU 3.60GHz
- Console version 8.1 running on:
  - Windows 2003 Standard Edition SP2, with 1 GB of memory
  - Intel® Xeon™ CPU 3.00GHz, 2.99GHz
- IP servers version 7.0.3 started on:
  - Sun T5220 Solaris 10, with 16 GB of memory, 40 1167 MHz SUNW,UltraSPARC-T2 Virtual CPUs

Test scenarios

Test scenarios included the following four cases:

- Case 1 — Configured an AM/PM server with approximately 2000 devices as an underlying Service Assurance Manager domain. Generated 40,000 notifications as rapidly as possible. Measured the memory of the SAM server from initialization until all notifications were processed.
- Case 2 — Repeated Case-1 with 30 simulated consoles attached.
- Case 3 — Configured 2 AM/PM servers with approximately 2000 devices, configured 2 AM servers with approximately 1000 devices as underlying Service Assurance Manager domains. Generated 90,000 notifications as rapidly as possible. Measured the memory of the SAM server from initialization until all notifications were processed.
- Case 4 — Repeated Case 3 with 30 simulated consoles attached.

Results on a Linux AS5 platform

Table 37 on page 133 defines the results on a Linux AS5 platform.

Table 37 SAM 8.1 Memory benchmarking results

Initialization:	477580 KB
Topology synchronization (27,000 instances, 30,000 relationships) from Case 1 and Case 2:	202240 KB
Topology synchronization (50,000 instances, 59,000 relationships) from Case 3 and Case 4:	315704 KB
43,000 notifications from Case 1:	159084 KB 3.69 KB per notification.
43,000 notifications with 30 consoles from Case 2:	414136 KB Amount of overhead per console for 43,000 notifications equals 8501.73 KB, or 197.7 bytes per notification.
96,000 notifications from Case 3:	343044 KB 3.57 KB per notification.
96,000 notifications with 30 Consoles from Case 4:	598388 KB Amount of overhead per console for 96,000 notifications equals 8511.47KB, or 88.66 bytes per notification.

INDEX

A
adapter
    Syslog 21
Adapter Platform 60, 83
adapters 30
    Concord eHealth 30
    InfoVista 30
aggregation layer 36
architecture
    general 20
Architecture Solution Diagram 47
ASL 65
ASL scripts 64
authentication 83, 87

B
Basic Hierarchical configuration 39
batching traps 63, 66
before you begin checklist 25, 108
BIM 75
brcontrol 90
Broker 48
    default port 48, 51
    Secure 83
Build Guide 23, 29, 47
Business Dashboard 20, 69
Business Impact Manager 20, 75

C
checklists
    before you begin 25, 108
    discovery design 66, 111
clientConnect.conf 70
clients 30, 36, 69
Concord eHealth 30
configurations 38 to 42
    Basic Hierarchical 39, 50
    hierarchical aggregation 40, 50, 57
    hierarchical mesh 42, 50, 57
    hierarchical presentation 41
    password 70
    security 83
    Service Assurance 83
    standard 38
Consoles
    Web 41
consoles
    clients 69
    defining 72
    designing 72
    layouts 72
    restricting operations 72

D
dashboard.properties 87
deploy
    Domain Managers 47
    Global Manager 47
Deployment
    Build Guide 23, 29, 47
deployment
    design description 23
    installation and configuration description 24
    tools 72
    tuning description 25
    validation description 24
deployment process
    designing 23
    installing and configuring 24
    validating 24
designing
    consoles 72
discovery design checklists 66, 111
DNS 65
Domain Manager 21
    deploy 47
domains
    underlying 36
dump queues 93

E
environment variables 83

F
failover capabilities 32
filtering traps 65
Firewalls 86
firewalls 31
FLEXnet 49

G
Global Console 68
    function 20
Global Console users
    categories 68
    security 69
Global Manager 20
    deploy 47

H
hardware 43
heartbeat notifications 96
hierarchical aggregation configuration 40, 57
hierarchical configuration 37
hierarchical mesh configuration 42, 57

Hierarchical Presentation Configuration 41
hook script 65
host name lookup 65
HTTP 87

I

ICS_ActionScheduler 96
InfoVista 30
installation
    Service Assurance 83
integrating existing software 30

J

Java applets 69

L

latency 32, 41, 50, 97
layers
    aggregation 36
    presentation 36
layouts 72
License Manager
    port 51
licensing
    configurations 49
    generating a license 82
    multiple license servers 49
    permanent 82

M

memory 92
MIBs 51
monitoring utilities 99

N

network
    firewall 31
    security 31
    size 36
network management 32
notifications
    heartbeat 96
    lists 61, 71, 104
    processing 60 to 61
    syslog 104
    traps 104

O

out of band 32, 88

P

password configuration 70, 83
performance
    guidelines 92
    improving 104
    trap processing 65 to 66
ports 51
    Broker 48, 51
    License Manager 51
presentation layer 36
profiles 71
properties.conf 87
proxy server 51, 86

Q

queue
    growth 93
    size 93

S

SAM Adapter Platform
    function 21
secret phrase 84
Secure Broker 83
security 31, 51, 69
    clientConnect.conf 70
    configuration 83
    Global Console users 69
    serverConnect.conf 70
    users 69, 83
Server Tools 30
serverConnect.conf 70
Service Assurance Adapters 21
Service Assurance configuration 83
Service Assurance installation 83
Service Assurance Manager 20
    Global Console function 20
Service Assurance Manager Benchmarks 115
SM_PropertyPoller 93, 96
sm_snmp 90
SNMP Trap Adapter 51, 62
software
    integrating third-party 30
Solution Architecture Diagram 23, 28, 50
Standard Configuration 38
SubscriberFE 93, 96
syslog
    notifications 104
Syslog Adapter 51
    function 21

T

TCP 31, 51
tool deployment 72
topology 65
    browser views 72
    grouping 58
    synchronization 39, 56 to 57, 97, 104
    validating 90
trap
    exploder 62
    notifications 104
trap processing
    adding new topology 65
    ASL scripts 62, 64

batching 63
filtering 65
host name lookup 65
notifications 64
performance 65
ports 64
recommended design 62
validation 90
trap_mgr.conf 65, 90
trapd.conf 64

U

UDP 31, 51
underlying domains 36
user profiles 71, 90
users 69
validation 90

V

validation 90
viewlets 41, 69
views
Topology Browser 72

W

Web Consoles 41, 68
webconsole.properties 87