VMware Smart Assurance Service Assurance Manager Dashboard Configuration Guide

VMware Smart Assurance 10.0.0

vmware®

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2019 VMware, Inc. All rights reserved. Copyright and trademark information.

Contents

1 Preface 4

- Purpose 4
- Audience 5
- VMware Smart Assurance Service Assurance Manager installation directory 5
- VMware Smart Assurance Service Assurance Manager 5
- Related documentation 6
- VMware Smart Assurance Service Assurance Manager documentation 6
- Conventions used in this document 7
- Where to get help 8
- Your comments 9

2 Overview 10

3 Architecture of Service Assurance Manager with the dashboard 11

4 Software requirements 13

- Client requirements 13
- Server requirements 14

5 Configuring the VMware Smart Assurance Business Dashboard 15

- Overview 15
- Serving Dashboard viewlets 16
- About user profiles and user accounts 26
- About Dashboard viewlets 27
- HTML syntax for viewlets 29
- URL-addressable viewlets 41
- About the web console 42

6 Examples of Dashboard viewlets 44

- About the sample viewlets 44

7 Troubleshooting 52

- Troubleshooting overview 52
- Debugging the web console 53
- Sun Java Console (Plug-in message log) 56
- Log file location 57

Preface

1

As part of an effort to improve its product lines, VMware 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 VMware representative if a product does not function properly or does not function as described in this document.

Note This document was accurate at publication time. New versions of this document might be released on the VMware online support website. Check the VMware online support website to ensure that you are using the latest version of this document.

This chapter includes the following topics:

- Purpose
- Audience
- VMware Smart Assurance Service Assurance Manager installation directory
- VMware Smart Assurance Service Assurance Manager
- Related documentation
- VMware Smart Assurance Service Assurance Manager documentation
- Conventions used in this document
- Where to get help
- Your comments

Purpose

The VMware Smart Assurance Business Dashboard is a web-based interface that enables customers, operators, and administrators to access information about the status of their managed infrastructure from any standard browser, from any location. This document describes how to configure the VMware Smart Assurance Business Dashboard. Configuration tasks include configuring Tomcat Servlet Engine, saving views in the Global Console as viewlets, writing HTML, and configuring the interaction and operation of dashboard viewlets.

Audience

This document is intended for system or network administrators and IT managers responsible for the configuration and management of the VMware Smart Assurance Business Dashboard. Administrators responsible for deploying web portals that incorporate dashboard viewlets or the web console will also find this document useful.

VMware Smart Assurance Service Assurance Manager installation directory

In this document, the term BASEDIR represents the location where VMware Smart Assurance software is installed:

- For UNIX, this location is: /opt/InCharge/<*productsuite*>.
- For example, on UNIX operating systems, VMware Smart Assurance Service Assurance Manager is, by default, installed to /opt/InCharge/SAM/smarts.

  Optionally, you can specify the root of BASEDIR to be something other than/opt/InCharge (on UNIX) but you cannot change the <*product*> location under the root directory.

  The VMware Smart Assurance System Administration Guide provides more information about the directory structure of VMware Smart Assurance software.

VMware Smart Assurance Service Assurance Manager

The VMware Smart Assurance 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 User Interface (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)
  - XML Adapter
  - VMware Smart Assurance Adapter for UIM/O

Related documentation

In addition to this document, VMware 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.

VMware Smart Assurance documentation

Readers of this guide may find the following related documentation helpful:

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

Note These documents are updated periodically. Electronic versions of the updated manuals are available on the VMware online support website:

docs.vmware.com

VMware Smart Assurance Service Assurance Manager documentation

The following documents are relevant to users of the VMware Smart Assurance Service Assurance Manager:

- VMware Smart Assurance Service Assurance Manager Release Notes
- VMware Smart Assurance Open Source License and Copyright Information
- VMware Smart Assurance Service Assurance Manager Introduction
- VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers

- VMware Smart Assurance Service Assurance Manager Deployment Guide
- VMware Smart Assurance Service Assurance Manager Configuration Guide
- VMware Smart Assurance Service Assurance Manager Operator Guide
- VMware Smart Assurance Service Assurance Manager Dashboard Configuration Guide
- VMware Smart Assurance Business Impact Manager User Guide
- VMware Smart Assurance Failover System User Guide
- VMware Smart Assurance Service Assurance Manager Notification Adapters User Guide
- VMware Smart Assurance Service Assurance Manager Adapter Platform User Guide
- VMware Smart Assurance Adapter for Unified Infrastructure Manager/Operations Configuration Guide
- VMware Smart Assurance XML Adapter User Guide
- VMware Smart Assurance Notification Module User Guide
- VMware Smart Assurance Service Assurance Manager Troubleshooting Guide
- VMware Smart Assurance Companion UI Online Help
- VMware Smart Assurance Companion UI Open Source License and Copyright Information for GPLv3 as Included with a Distribution of SLES 11
- VMware Smart Assurance Service Assurance Manager Documentation Portfolio

  The VMware Smart Assurance Documentation Catalog provides documentation resources for other VMware Smart Assurance products.

Conventions used in this document

VMware uses the following conventions for special notices:

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

Typographical conventions

VMware uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: ■ Names of interface elements ■ 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 ■ 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 (/).

Graphical conventions

Under UNIX, the consoles appear with slight differences. For example, in views that display items in a tree hierarchy such as the Topology Browser, an open circle appears for UNIX.

Manager

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

Where to get help

VMware support, product, and licensing information can be obtained as follows:

Product information

For documentation, release notes, software updates, or information about VMware products, licensing, and service, go to the VMware online support website (registration required) at:

```
docs.vmware.com
```

For technical support, click Get Support or Contact Customer Service on VMware Online Support at support.vmware.com. To open a service request through VMware Online Support, you must have a valid support agreement. Contact your VMware Sales Representative for details about obtaining a valid support agreement or to answer any questions about your account.

Your comments

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

```
docs.vmware.com.
```

Overview

2

The VMware Smart Assurance Business Dashboard displays the results of VMware Smart Assurance analysis in a flexible, web-based user interface. The VMware Smart Assurance Business Dashboard provides the views from the Global Console as individual components called dashboard viewlets. A dashboard viewlet is implemented as a Java applet.

You can configure the display and contents of dashboard viewlets to customize the presentation of VMware Smart Assurance analysis, enabling you to deliver role-based views of the managed topology across all layers of your IT environment, including:

- Business managers can view status summaries for systems and business processes
- Operations staff can view real-time operational status of monitored elements and services
- Users can drill down to see the health and status of related infrastructure, applications, and business services

  Installing the VMware Smart Assurance Business Dashboard provides all of the components necessary to create, configure, and display dashboard viewlets. Chapter 6 Examples of Dashboard viewlets includes a set of ready-to-use viewlets.

  In addition, an installation of the VMware Smart Assurance Business Dashboard includes the web console, which displays the Global Console within a web browser. The web console provides the same functions and features as the Global Console.

Architecture of Service Assurance Manager with the dashboard

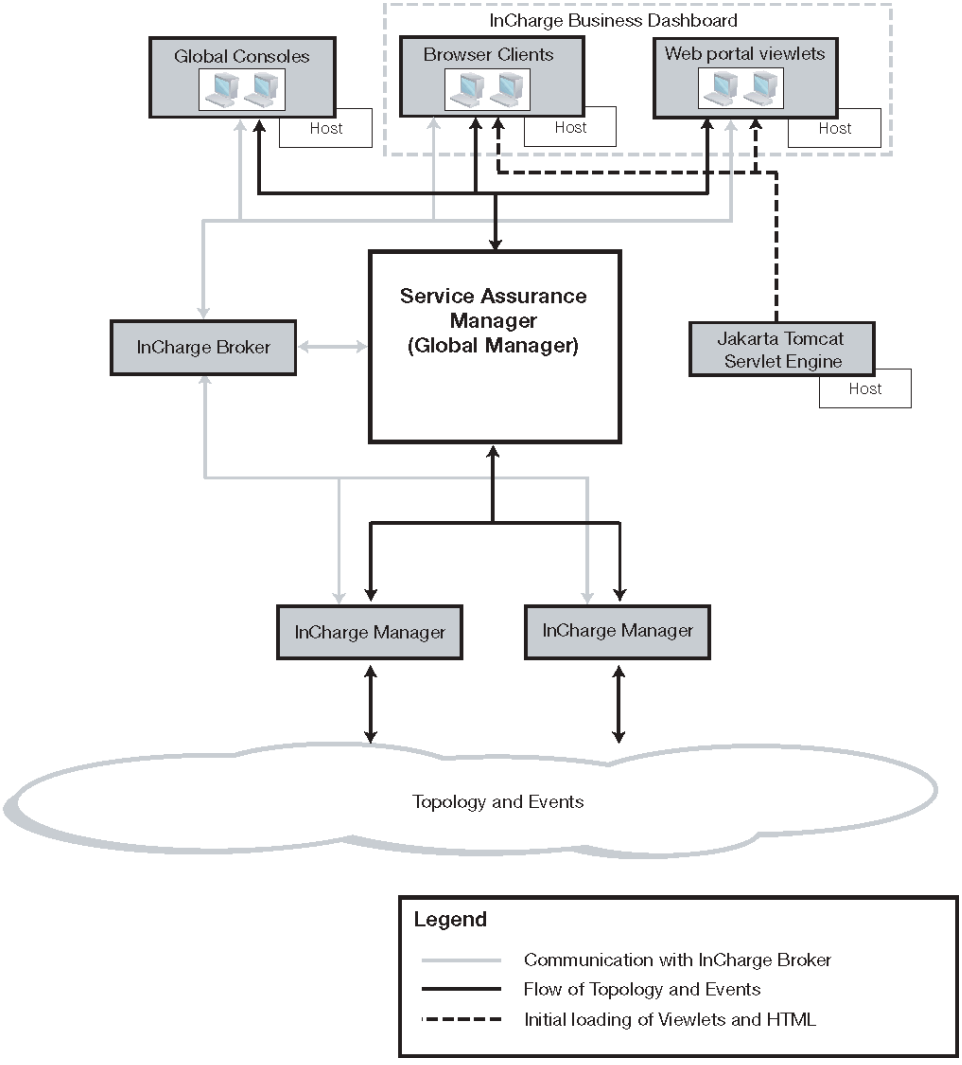
3

While the Global Console is the primary user interface for VMware Smart Assurance software, the VMware Smart Assurance Business Dashboard provides another means of viewing and responding to the analysis provided by VMware Smart Assurance Service Assurance Manager. The VMware Smart Assurance Business Dashboard provides additional features:

- Dashboard viewlets can be displayed in any supported web browser from any location. The web clients do not require an installation of VMware Smart Assurance software.
- Dashboard viewlets can be embedded into a web Portal, combining VMware Smart Assurance' analysis with information from other sources.

  Architecture of Service Assurance with Business Dashboard illustrates how the VMware Smart Assurance Business Dashboard integrates with VMware Smart Assurance Service Assurance Manager.

Figure 3-1. Architecture of Service Assurance with Business Dashboard



- The following components, illustrated in Architecture of Service Assurance with Business Dashboard, are related to the VMware Smart Assurance Business Dashboard:
- Global Manager consolidates topology and event information it receives from data sources such as VMware Smart Assurance Managers. The Global Manager provides this data to clients such as the Global Console or dashboard viewlets.
- VMware Smart Assurance Business Dashboard is a collection of dashboard viewlets that are displayed in a web page. A viewlet can also be embedded within a web portal page or loaded from a dynamically generated URL.
- Tomcat Servlet Engine serves the JAR files that compose dashboard viewlets and the map layout servlet. The servlet engine is installed as a service with the VMware Smart Assurance Business Dashboard. If you are already running a web server, you can use it to serve the necessary files to display dashboard viewlets.

Software requirements

4

VMware Smart Assurance Business Dashboard includes the necessary files and software for serving the files to display dashboard viewlets and the web console.

This chapter includes the following topics:

- Client requirements
- Server requirements

Client requirements

Users who display 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 7.0 and 6.0 with Service Pack 1
  - Firefox 1.5.x
  - Browser that supports Java JRE 1.6.15 or later

    Note Only browsers running on HP-UX,Linux and Solaris operating systems are supported.

    Use Firefox instead of Microsoft IE if you need to use an IPv6 literal address to access the Web Console or Business Dashboard.

- Java 2 Platform Standard Edition 5.0 which includes the 1.6.x web browser Plug-in

  When using Internet explorer with JRE 1.6, exiting the Web Console or Business Dashboard by choosing **File > Exit** does not clear the display on the browser window. The browser window displays whatever was displayed in the window at the time you exited the application. This does not impact the functionality. Instead of exiting the browser, you can either:

- Close the Web browser
- Type a new URL in the Web browser

Server requirements

An installation of the VMware Smart Assurance 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 VMware Smart Assurance Servlet Engine service on the system where you install the VMware Smart Assurance Business Dashboard.

If you wish to serve dashboard viewlets from a different web server, you must install the VMware Smart Assurance Business Dashboard to obtain the necessary files. Deploying viewlets provides a description of these files.

Configuring the VMware Smart Assurance Business Dashboard

5

This chapter includes the following topics:

- Overview
- Serving Dashboard viewlets
- About user profiles and user accounts
- About Dashboard viewlets
- HTML syntax for viewlets
- URL-addressable viewlets
- About the web console

Overview

The configuration tasks for VMware Smart Assurance Business Dashboard depend on how you intend to deploy it. A simple deployment could use the sample viewlets as an VMware Smart Assurance Business Dashboard to display information from a Global Manager. A more complicated deployment could involve creating new HTML pages to display Dashboard viewlets, configuring the display of viewlets, creating an on-demand URL addressable viewlet, or integrating a viewlet with a portal.

The configuration steps necessary to provide these deployments are described in this chapter and involve one or more of the following tasks:

- Configuring the Tomcat Servlet Engine or an alternative web server
- Configuring user accounts on the Global Manager
- Creating saved viewlets with the Global Console
- Writing HTML to display viewlets
- Creating a URL-addressable viewlet

Serving Dashboard viewlets

The Servlet Tomcat Engine is installed into the *BASEDIR*/smarts/tomcat directory on the host where the VMware Smart Assurance Business Dashboard is installed. If you intend to use Tomcat to serve Dashboard viewlets, you should install the VMware Smart Assurance Business Dashboard onto the system from which you want to serve the relevant files. This can be a different host from where the Global Manager, or any other Service Assurance component is installed.

The Servlet Tomcat Engine, its directory structure, and the Tomcat configuration files are not developed by VMware Smart Assurance. As such, the files under this directory, including the files related to the VMware Smart Assurance Business Dashboard, are not intended to be edited using the **sm_edit** utility. Instead, open and edit any files under the *BASEDIR*/smarts/tomcat directory with a text editor.

Tomcat Servlet Engine directories lists relevant subdirectories under *BASEDIR*/smarts/tomcat related to the configuration and running of VMware Smart Assurance Business Dashboard.

Table 5-1. Tomcat Servlet Engine directories

Directory	Description
/conf	Contains Tomcat configuration files, including server.xml
/webapps	Contains directories and files served by Tomcat: ■ /templates includes JAR files, viewlets, configuration files, and HTML files related to the samples described in Chapter 6 Examples of Dashboard viewlets ■ /webconsole includes the files for the web console.

About the VMware Smart Assurance Servlet Engine service

The Servlet Tomcat Engine is installed as a service named the VMware Smart Assurance Servlet Engine. As a service, it starts automatically with the system and runs continuously until it is manually stopped. The VMware Smart Assurance Servlet Engine serves the HTML, JAR, and map layout servlet for both Dashboard viewlets and the web console.

Default service parameters

The VMware Smart Assurance Servlet Engine service is installed with the following options:

```
tC:/InCharge/SAM/smarts/bin/sm_service install --force
--startmode=runonce --description="VMware Smart Assurance Servlet Engine"
--name="ic-business-dashboard" /C:/InCharge/SAM/smarts/bin/sm_tomcat --output start s
```

Starting and stopping the VMware Smart Assurance Servlet Engine

The **sm_service** utility is used to start and stop services. Invoke the following commands from the *BASEDIR*/smarts/bin directory on the host where the service is running:

To stop the VMware Smart Assurance Servlet Engine service:

```
./sm_service stop ic-business-dashboard
```

To start the VMware Smart Assurance Servlet Engine service:

```
./sm_service start ic-business-dashboard
```

The VMware Smart Assurance System Administration Guide provides additional information about services.

Verify that Tomcat is running

After the VMware Smart Assurance Servlet Engine service is installed and started, you can verify whether Tomcat is running by opening the following URL in a web browser: http://<*tomcat_host*>:<*port*>, where <*tomcat_host*> is the system where the VMware Smart Assurance Business Dashboard is installed and <*port*> is the TCP port the Tomcat Servlet Engine is listening on.

You can also access the Tomcat Servlet Engine documentation from the following URL: http://<*tomcat_host*>:<*port*>/tomcat-docs.

Changing the port number of Tomcat

For most installations, Tomcat should not require any additional configuration. In some cases, however, it may be necessary to change the port number on which the Tomcat Servlet Engine is running.

The configuration of Tomcat is determined by the server.xml file located in the *BASEDIR*/smarts/tomcat/conf directory. By default, Tomcat is configured to use TCP port 8080. If this port is already in use on the system where the VMware Smart Assurance Business Dashboard is installed, you can configure Tomcat to use another port.

To change the port number, find the following lines in server.xml:

```
<Connector port="8080" maxHttpHeaderSize="8192"
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
   enableLookups="false" redirectPort="8443" acceptCount="100"
   connectionTimeout="20000" disableUploadTimeout="true" />
```

Change the port number to a suitable number. If Tomcat is currently running, you will need to restart it.

You may also need to change the shutdown server port number. Port number 8005 is the default port for the shutdown server. To change the shutdown server port number, find the following line in the server.xml:

```
<Server port="8005" shutdown="SHUTDOWN">
```

Change the port number to a suitable number.

For more information regarding Tomcat Servlet Engine, see http://tomcat.apache.org

Increasing the heap space

You can increase the heap space for the Java Plug-in and for Java Web Start.

Increasing the Java Plug-in heap space

If the web console or viewlets run out of memory, adjust the maximum amount of heap space that the Java Plug-in uses for dynamic memory allocation. By specifying -Xmx256m as a runtime parameter, for example, you can increase the heap space allocation to 256 MB. The number can be set larger or smaller than 256 MB, depending on the resources available on the client machine running the web browser.

To increase the heap space for the Java Plug-in:

1 Launch the **Java Control Panel**. The Sun website provides instructions (http://java.sun.com).

2 In the **Java Control Panel**, click the **Java** tab.

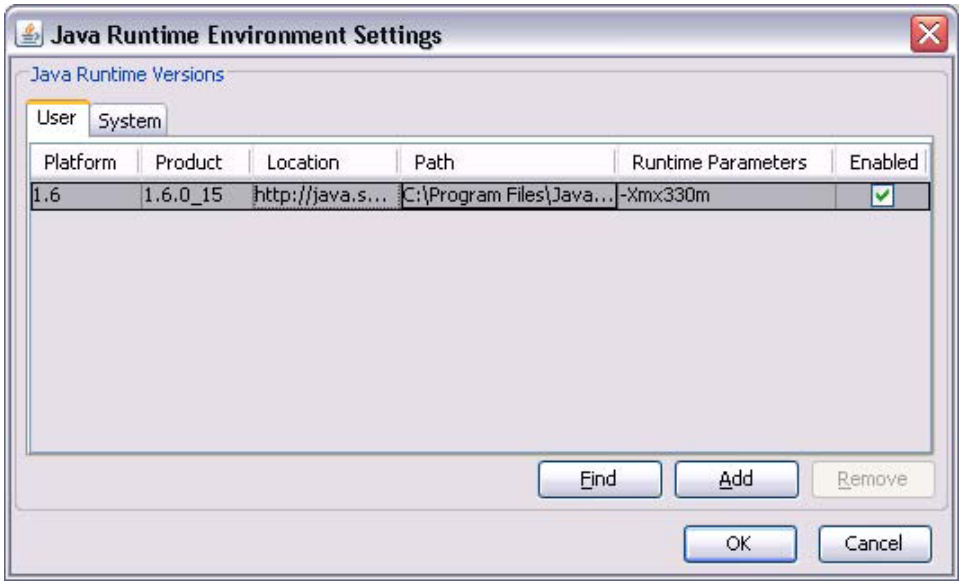
3 In the **Java Runtime Environment Settings** section, click the **View** button.

The **Java Runtime Environment Settings** dialog box appears.

4 In the **Runtime Parameters** field, type the maximum heap-space runtime parameter (-Xmx) and a value for the parameter.

For example: -Xmx330m. Java Runtime Environment Settings illustrates the **Java Runtime Environment Settings** dialog box.

Figure 5-1. Java Runtime Environment Settings



The heap-space runtime parameter specifies heap size in megabytes, for example, -Xmx330m increases the heap space to 330 megabytes.

5 Click **OK**.

6 Click the Java Control Panel **Apply** button.

7 Terminate and restart any web browser running on the host machine.

For the Java Web Start, the heap space defaults to 350 megabytes.

Increasing Java Web Start heap space

To increase the heap space for the Java Web Start:

1 Using a text editor, open the BASEDIR/smarts/tomcat/webapps/webconsole/globalConsoleApp.jnlp file.

2 Locate the following line:

```
<j2se version="1.6.0+" max-heap-size="350M"/>
```

3 Change the max-heap-size variable value.

4 Save and close the file.

Deploying viewlets

Directories under *BASEDIR*/smarts/tomcat/webapps contain the web applications served by the Tomcat Servlet Engine. For example, the sample Dashboard viewlets included with VMware Smart Assurance Business Dashboard are installed to the *BASEDIR*/smarts/tomcat/webapps/templates directory. Each directory under /webapps must contain the JAR, HTML, and configuration files necessary for the web application to run.

You can choose one of the following methods to deploy Dashboard viewlets:

- Modify the template files under *BASEDIR*/smarts/tomcat/webapps/templates. You can also add your own new viewlet files to this directory.
- Create your own viewlets directory under /webapps. You must ensure that the new directory contains the files and directories listed in Files used for viewlets, which can be copied from the templates directory. After you create the new directory and add the necessary files, you need to restart the VMware Smart Assurance Servlet Engine service.

Table 5-2. Files used for viewlets

File or directory	Description
HTML files	HTML files that specify the layout of viewlets, including the JAR files and configuration parameters.
jcommon-0.7.0.jar jfreechart-0.9.3.jar masterpiece.jar tsgdtj50.jar tsgetj50.jar tsglt50.jar tsgltc50.jar tsgmtj50.jar	JAR files necessary for displaying VMware Smart Assurance Business Dashboard viewlets. All JAR files are required.

Table 5-2. Files used for viewlets (Continued)

File or directory	Description
dashboard.properties	[Optional] Properties file to specify parameters that apply to all viewlets. This file must be specified in the HTML using the *com.smarts.propertiesFile* parameter. Specifying additional viewlet properties provides additional information about HTML parameters.
viewlet.properties	[Optional] Property file to specify parameters for a specific viewlet. This file must be specified in the HTML using the *com.smarts.viewletProperties* parameter. Specifying additional viewlet properties provides additional information about HTML parameters.
WEB-INF	Directory indicates that Tomcat should serve the files contained in this directory upon request. This directory is required by Tomcat.

Deploying viewlets with a third-party web server

If you deploy viewlets using a different web server, the JAR files listed in Deploying viewlets with a third-party web server are required. The properties files are optional, but recommended as they simplify configuration of viewlets. Copy the necessary files listed in Deploying viewlets with a third-party web server to the host where the web server is running.

Note If you intend to display Map viewlets, the web server must be capable of executing servlets.

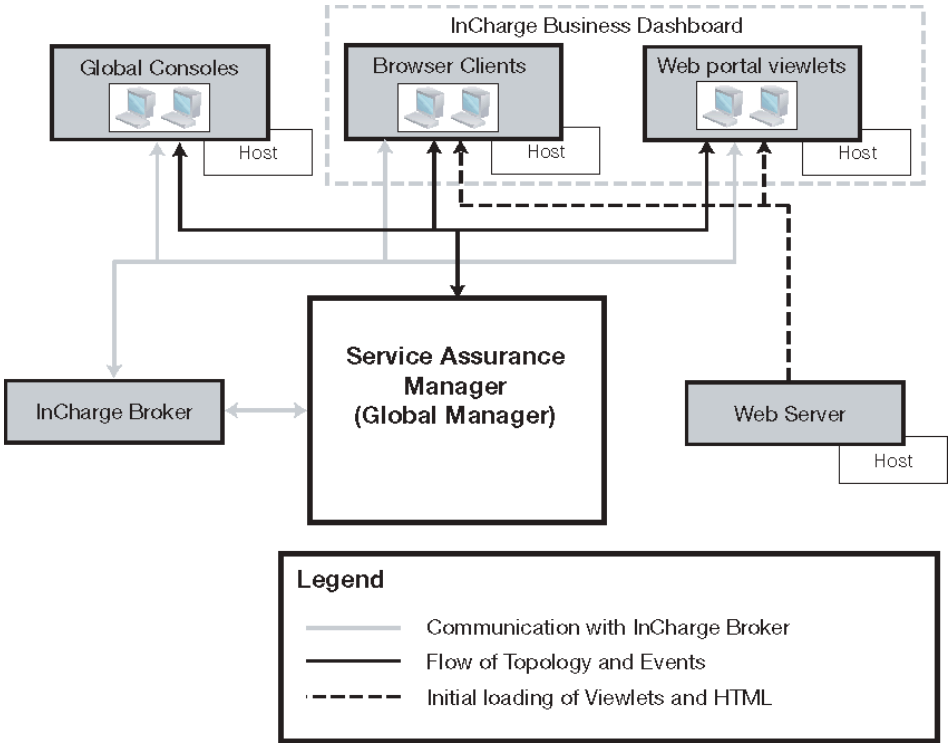
In addition, the files in Additional files needed for third-party web server must be installed on the server.

Table 5-3. Additional files needed for third-party web server

Operating system	Files needed		
Solaris/Linux	libagraphj.so	libjdgraphj.so	libnlayj.so
	libblayj.so	libjhlayj.so	libolayj.so
	libclayj.so	libjmgraphj.so	libslayj.so
	libclidemoj.so	libjolayj.so	libtlayj.so
	libdgraphj.so	libjslayj.so	libtsgltextj.so
	libhlayj.so	libjtlayj.so	libtsgltj.so
	libjagraphj.so	libjutilsj.so	libutilsj.so
	libjclayj.so	libmgraphj.so	
HP-UX	libagraphj.sl	libjdgraphj.sl	libnlayj.sl
	libblayj.sl	libjhlayj.sl	libolayj.sl
	libclayj.sl	libjmgraphj.sl	libslayj.sl
	libclidemoj.sl	libjolayj.sl	libtlayj.sl
	libdgraphj.sl	libjslayj.sl	libtsgltextj.sl
	libhlayj.sl	libjtlayj.sl	libtsgltj.sl
	libjagraphj.sl	libjutilsj.sl	libutilsj.sl
	libjclayj.sl	libmgraphj.sl	

Architecture when using a third-party web server illustrates the flow of information when using a web server other than the VMware Smart Assurance Servlet Engine to serve viewlets.

Figure 5-2. Architecture when using a third-party web server



To deploy the Dashboard viewlets with another web server (for example, a future version of Tomcat 6.x or 7.x), perform the following steps:

Note Version numbers and directory paths in these steps are for example only.

1 Install Tomcat 6.*x* or 7.*x*.

2 Set JAVA_HOME. For example, JAVA_HOME=C:\Program Files\JAVA\jdk1.6.0_15. Add the line to the catalina.bat file in the following location:

- C:\tomcat-6.*x*\bin for Tomcat 6.x.
  - C:\tomcat-7.*x*\bin for Tomcat 7.x.

3 Copy *BASEDIR*\smarts\tomcat\webapps\tss, \webapps\webconsole, and \webapps\templates to the new directory. For example, to the following location:

- C:\tomcat-6.*x*\webapps for Tomcat 6.x.
  - C:\tomcat-7.*x*\webapps for Tomcat 7.x.

4 Create a "lib" folder under the following location:

- C:\tomcat-6.*x*\webapps\tss\WEB-INF for Tomcat 6.x.
  - C:\tomcat-7.*x*\webapps\tss\WEB-INF for Tomcat 7.x.

  and copy the JAR files from *BASEDIR*\smarts\tomcat\webapps\webconsole to this new folder. You need only copy the JAR files that begins with “tsg.”

5 Ensure that *BASEDIR*\smarts\lib is in the system path (used in the library path for UNIX installs) when Tomcat is started. This directory includes libraries that the servlet needs.

Note This only works if the web server is on the same host as the Dashboard or Global Console. If they are not, you must copy libraries from the *BASEDIR*\smarts\lib of the Dashboard or Global Console to the machine where the server is installed. Additional files needed for third-party web server lists these files.

6 Start the Tomcat engine using your standard web server startup command:

```
..\tomcat-6.x\bin\startup.bat
..\tomcat-7.x\bin\startup.bat
```

Configuring the web console, Business Dashboard, Dashboard viewlets, and Global Console to use a proxy server

You can configure the Business Dashboard and Dashboard viewlets to use a proxy server to communicate with VMware Smart Assurance applications (for example, IP Availability Manager and Global Manager) that reside behind a firewall. Using a proxy server, the number of firewall ports that need to be opened is reduced to one.

Before you configure these VMware Smart Assurance Java clients, ensure that the web server that will act as the proxy satisfies the requirement. Requirement for the proxy server describes these requirements.

To configure the VMware Smart Assurance 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 VMware Smart Assurance applications.

2 Use a text editor (not sm_edit) to edit the Proxy properties described in Parameters for properties file in the appropriate properties file (where *BASEDIR* is/opt/InCharge/SAM/smarts or C:\InCharge\SAM\smarts\):

   - 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.

3 Use the **sm_edit** utility to modify the properties.conf file.

   - 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 5-4. Parameters for properties file

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 depends 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"
```

Requirement for the proxy server

The connection time-out mechanism implemented in VMware Smart Assurance 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, VMware, Inc. 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 VMware Smart Assurance Global Services if the proxy server does not pass TCP urgent data.

Encrypted connections

Encrypted connections between Dashboard viewlets and the Global Manager are configured using two environment variables:

- SM_OUTGOING_PROTOCOL — Controls the connections that an VMware Smart Assurance program acting as a client is allowed to initiate.

- SM_INCOMING_PROTOCOL — Controls the connections that an VMware Smart Assurance program acting as a server is allowed to accept.

  The variables are set in the runcmd_env.sh file, which is located in the *BASEDIR*/smarts/local/conf directory. The variables may be set to “0” (unencrypted connection), “1” (encrypted connection), or “1,0” (both encrypted and unencrypted connections).

  The default setting is to use both encrypted and unencrypted connections which means that the Dashboard viewlets and the Global Manager automatically negotiate the connection to the highest level of encryption possible.

  Encryption can be controlled by setting similar variables in the HTML code that displays the Dashboard viewlet. Specifying viewlet parameters in HTML and Using a properties file to specify viewlet parameters provide additional information.

  For these two methods, the following variables are used:

- com.smarts.incoming_protocol — Controls the connections that an VMware Smart Assurance program acting as a server is allowed to accept.
- com.smarts.outgoing_protocol — Controls the connections that an VMware Smart Assurance program acting as a client is allowed to initiate.

  The VMware Smart Assurance System Administration Guide provides detailed information about encrypted connections.

Using SSL (https://)

If your deployment requires the use of Secure Socket Layer (SSL), you need to perform the following steps before you launch the Dashboard with the https protocol:

Note The ssl-howto.html document in the *BASEDIR*/smarts/tomcat/webapps directory on your system provides detailed information. Or, access the document from the website: http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html.

1 Prepare a "self-signed" certificate.

  If you do not purchase a Certificate from a Certificate Authority, you can generate a Certificate.

  For example:

```
java1.5_HOME> /bin/keytool.exe –genkey –alias tomcat –keyalg  RSA – keystore testKeystore
Enter keystore password:  changeit
What is your first and last name?
  [Unknown]: User Name
What is the name of your organizational unit?
  [Unknown]: Engineering
What is the name of your organization?
  [Unknown]: VMware Smart Assurance
What is the name of your City or Locality?
  [Unknown]:  White Plains
What is the name of your State or Province?
  [Unknown]: NY
```

```
What is the two-letter country code for this unit?
  [Unknown]: US
Is CN=User Name, OU=Engineering, O=VMware Smart Assurance, L=White Plains, ST=NY, C=US correct?
  [no]:  yes
Enter key password for <tomcat>
        (RETURN if same as keystore password):<Return>
```

2 Configure Tomcat:

3 In the Tomcat home directory, back up the conf/server.xml file. Then, edit the file.

4 Find the SSL connector element which is commented out (<!-- commented out text -->):

```
<-- Define an SSL HTTP/1.1 Connector on port 8443
-->
<!--
<Connector port="8443"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" debug="0" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
-->
```

5 Uncomment the lines and edit them to include the keystore file that you created. The preceding will look like this:

```
<!-- Define an SSL HTTP/1.1 Connector on port 8443
-->

<Connector port="8443"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
keystoreFile="C:\InCharge\SAM\smarts\tomcat\testKeystore" keystorePass="changeit"
acceptCount="100" debug="0" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
```

6 Then comment out the non-SSL connector used for http, to look like this:

```
<!-- Define a non-SSL HTTP/1.1 Connector on the
port specified during installation  -->
<!--
<Connector port="8080"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443"
acceptCount="100"
debug="0" connectionTimeout="20000"
disableUploadTimeout="true" />
-->
```

7 If the com.smarts.layoutServlet property is defined, ensure that the dashboard.properties and webconsole.properties point to the correct URL so that the maps lay out properly:

```
com.smarts.layoutServlet="https://localhost:8443/tss/
LayoutServer"
```

8 Start the Tomcat engine using the following command:

```
./sm_service start ic-business-dashboard
```

9 In the URL that will launch the web application, replace:

```
http://localhost:8080/
https://localhost:8443/
```

10 Launch the Dashboard by typing the following URL in the browser's Address field:

```
https://localhost:8443/templates
```

The system displays a Security Alert window that provides information about the Certificate being used. On accepting it, a Directory Listing for templates is displayed, as in the case of the non-SSL Dashboard.

About user profiles and user accounts

Users who attach to a Global Manager, whether through the Global Console, the web console, or a web browser, must have a user account. Each user account, in turn, is associated with a user profile. A user profile defines the console operations a user can perform, the tools the user can invoke, and the notifications displayed to the user. The VMware Smart Assurance Service Assurance Manager Configuration Guide provides detailed information about user profiles.

If a user attaches to a Global Manager using both the Global Console and a web browser, the same user profile can apply for both connections.

When you save a viewlet, you can save it to a user's directory or a user profile directory. In both cases, the directory is located in the *BASEDIR*/smarts/local/consoles directory on the host where the Global Manager is running. Dashboard viewlets saved to a user's directory are only accessible to that user. The name of a user's directory is the same as the username. Dashboard viewlets saved to a user profile directory are accessible by any user associated with that user profile. The name of the user profile directory is the same as the user profile name.

You must create the user profiles and user accounts before you can save a viewlet to a user's directory or a user profile.

About Dashboard viewlets

A Dashboard viewlet is a view from the Global Console that you can use as an applet. As such, you can use viewlets in the following ways:

- Create an HTML page that includes one or more viewlets to provide a customized display of VMware Smart Assurance analysis. Because this view is displayed in a web browser, it is accessible from any host with a web browser.
- Provide an application that dynamically generates a URL in response to a notification that displays one or more viewlets in a web browser.
- Embed a viewlet within a web portal application to combine the information from the Global Manager within an existing information view.

  The configuration tasks that must be completed to provide any of the above Dashboard viewlets are similar. After you create a viewlet for a web page, for example, you could also embed that same viewlet within a portal. This enables you to re-use viewlets, assigning them as necessary according the user's role or informational requirements.

  You can also use the Dashboard viewlets located in the *BASEDIR*/smarts/tomcat/webapps/templates directory. The HTML code for each viewlet has been converted to HTML that displays properly in both Netscape and Internet Explorer. Chapter 6 Examples of Dashboard viewlets provides additional information about the sample viewlets. The HTML converter used to create browser-specific HTML is installed as part of Java 2 Platform Standard Edition 6.0.

Global Console views that function as viewlets

The following views from the Global Console can also be displayed as viewlets:

- Notification Log
- Status Table
- Summary View
- Containment
- Notification Properties

  You can display these console elements as viewlets by specifying their parameter name and any configuration parameters in the HTML code. *"HTML syntax for viewlets" on page 35* provides additional information regarding the HTML syntax for viewlets.

Saving views as viewlets

You can configure the views listed below using the Global Console and save them as viewlets:

- Notification Log
- Status Tables

- Summary Views

  Unless noted otherwise, the configuration options provided through the Global Console are retained in the viewlet file.

  To save a view as a viewlet:

  a Configure the view using the Global Console.

    The VMware Smart Assurance Service Assurance Manager Operator Guide provides additional information.

  b Select **View** > **Save View As Viewlet** to open the **Save View As Viewlet** dialog box. You need to provide the following information to save the viewlet:

    - Global Manager where the Dashboard viewlet is to be saved
    - Location where the Dashboard viewlet is to be saved
    - Dashboard viewlet name

  c Select the Global Manager where the viewlet is to be saved. You can save the viewlet to any attached Global Manager.

  d Select the location on the Global Manager where the viewlet is to be saved:

    - **Save for User** saves the viewlet into a user's directory on the selected Global Manager under *BASEDIR*/smarts/local/consoles. The list box lists users for the selected Global Manager. If the user directory under *BASEDIR*/smarts/local/consoles does not exist, it is created. A viewlet saved to a user's directory is only accessible to that user.
    - **Save for User Profile** saves the viewlet into a user profile directory on the selected Global Manager under *BASEDIR*/smarts/local/consoles. The list box lists available user profiles. If the user profile directory does not exist, it is created.

      Any user associated with this user profile can load the viewlet, provided the viewlet is specified by name in the web page loaded by the user.

      Note If a viewlet in the user's directory has the same name as a viewlet in the user profile directory, the viewlet in the user's directory takes precedence.

  e Specify a name for the viewlet. The file naming convention for saved viewlets is as follows:

```
<user_specifed_name><viewlet_type>.icsvm
```

  f Click **OK**.

Displaying maps

Map viewlets are not the same as the maps displayed in a Global Console. Map viewlets do not display a topology tree from which an element can be selected. Use one of the following methods to display elements in a Map viewlet:

- Select Show Map from a menu in another viewlet

- Enable context sharing for the Map viewlet. Context sharing for viewlets provides additional information.
- Save a map, as described below, and display the map by specifying its focus.

Saving maps

The purpose of saving a map is to customize the default map that is displayed for a specific element. A saved map is associated with the element that is the focus of the map. When the focus element is selected to display in a map, the saved map is displayed for users with access to the saved map.

A saved map can include the following information:

- Adjusted scope (for example, expanded nodes)
- Node positions
- Layout type (circular, hierarchical, or manual)
- Filtered classes
- User-defined backgrounds

  To save a map, in the Map Console, select Map > Save Map. Click OK to acknowledge the pop-up message which displays the location of the saved map. The map is saved to the user's directory under *BASEDIR*/smarts/local/consoles/ on the host where the Global Manager is running. The file extension for a saved map is ".icsvm".

  To make a saved map available to other users, you must copy the map file to the shared location *BASEDIR*/smarts/local/consoles.

  The VMware Smart Assurance Service Assurance Manager Operator Guide provides additional information about saved maps.

Displaying a saved map

To display a saved map, you must specify the topological instance that was the focus when the map was saved. Unlike a saved viewlet, it is not necessary to specify the name of the saved map. Properties for map viewlets describes Map viewlets.

HTML syntax for viewlets

This section describes the HTML parameters for loading and configuring Dashboard viewlets. Because the viewlets are Java applets, you must use the HTML tags for displaying applets. The tags vary with the version of HTML.

When you create an HTML page that displays several viewlets, using HTML frames provides control over the layout of the viewlets. In addition, if you want to load two viewlets of the same type, two Notification Logs for example, each viewlet must be specified in a separate HTML page. Then apply Frame tags to display both Notification Log viewlets on the same page.

You can examine and copy the HTML of the samples in the *BASEDIR*/smarts/tomcat/webapps/templates directory. Chapter 6 Examples of Dashboard viewlets describes the viewlets and their associated HTML files.

The HTML code in this section uses elements from the HTML 4.0 Specification.

HTML to display viewlets

Table 5 on page 36 lists the HTML tags and attributes necessary to display a Dashboard viewlet in a web page. Subsequent sections describe attributes that enable you to further configure the viewlet.

An attribute's name and its value must be enclosed by double quotes. When an attribute includes a set of values, the set of values is enclosed by double quotes, as shown in the following example:

```
<APPLET TYPE="applet"
 CODE="com.smarts.webapps.SmWebNotificationsApp.class"
 WIDTH="100%" HEIGHT="100%"
 ARCHIVE="masterpiece.jar,jcommon-0.7.0.jar,\
 jfreechart-0.9.3.jar,tsgdtj50.jar,tsgetj50.jar,\
 tsglt50.jar,tsgltc50.jar,tsgmtj50.jar">
</APPLET>
```

Specifying additional viewlet properties

You can specify additional properties in the HTML code that displays a viewlet, or in a properties file. These properties include parameters for the following features:

- Opening a saved view
- Enabling context sharing
- Setting the focus of a view
- Specifying connection properties

  Additional viewlet properties describes each of these properties.

Specifying viewlet parameters in HTML

Parameters included in the HTML code that displays a viewlet are nested inside the <APPLET> tags and have the following syntax:

```
<param name="name" value="value">
```

In the following example, the com.smarts.context.collectionEntity parameter is set to true to enable context listening for a Notification Log viewlet.

```
<APPLET TYPE="applet"
 CODE="com.smarts.webapps.SmWebNotificationsApp.class"
 WIDTH="100%" HEIGHT="100%"
 ARCHIVE="masterpiece.jar,jcommon-0.7.0.jar,\
 jfreechart-0.9.3.jar,tsgdtj50.jar,tsgetj50.jar,\
```

```
 tsglt50.jar,tsgltc50.jar,tsgmtj50.jar">
 <PARAM NAME="com.smarts.context.collectionEntity"\
VALUE="true">
</APPLET>
```

Using a properties file to specify viewlet parameters

You can use a properties file to specify parameters for a Dashboard viewlet. One or more viewlets can reference the properties file. This simplifies configuration because you can change the parameters within the properties file instead of editing each HTML file.

Two parameters may be used to specify a properties file:

- com.smarts.propertiesFile specifies properties common to all viewlets. Such properties might include the location of the VMware Smart Assurance Broker and the name of the Global Manager.
- com.smarts.viewletProperties specifies properties for a single viewlet. Such properties might enable context sharing or set the focus of a map viewlet.

  The parameters to specify a properties file for the viewlet are nested inside the <APPLET> tags and have the following syntax:

  ```
  <param name=”name” value=”value”>
  ```

  "value" indicates the name of the file containing the properties. You can use any name for a properties file.

  In the following example, the HTML for a Notification Log viewlet references a properties file named dashboard.properties:

  ```
  <APPLET TYPE="applet"
   CODE="com.smarts.webapps.SmWebNotificationsApp.class"
   WIDTH=”100%” HEIGHT=”100%”
   ARCHIVE="masterpiece.jar,jcommon-0.7.0.jar,\
   jfreechart-0.9.3.jar,tsgdtj50.jar,tsgetj50.jar,\
   tsglt50.jar,tsgltc50.jar,tsgmtj50.jar">
   <PARAM NAME="com.smarts.propertiesFile" \
  VALUE ="dashboard.properties">
  </APPLET>
  ```

  Additional viewlet properties and Encrypted connections describes the dashboard properties file. The properties are assigned using the following syntax:

  ```
  parameter_name=”value”
  ```

  Quotes around the value are optional unless you need to have leading or trailing spaces as part of the value.

The following properties file example sets the location of the VMware Smart Assurance Broker and the name of the Global Manager:

```
smarts.remote.broker=host1.smarts.com:426
com.smarts.autoAttachDM=INCHARGE-SA
```

As a result, when a user opens the web page that displays the viewlet, the Attach dialog box includes the information about the Broker and Global Manager. The user simply provides a username and password to connect.

Additional viewlet properties

The following sections describe Dashboard viewlet properties that can be specified either in the HTML displaying the viewlet or in a properties file. Similar properties are grouped together.

Displaying analysis servers and monitoring servers

Parameter for displaying analysis and monitoring servers describes the parameter used to display both analysis servers and monitoring servers in the Broker dialog box.

Table 5-5. Parameter for displaying analysis and monitoring servers

Parameter	Description
com.smarts.attachComponentServers	When set to true, both analysis servers and monitoring servers are displayed in the Broker dialog box. Also, when this parameter is set to true, the functionality that checks for case-insensitive strings of text in domain names is deactivated. **Note** This parameter can also be entered when using the **sm_gui** utility. For example, from <*BASEDIR*> enter: ./sm_gui -Dcom.smarts.attachComponentServers=true

Displaying a saved viewlet

Parameter for displaying a saved viewlet describes the parameter used to specify the name of a saved viewlet.

Table 5-6. Parameter for displaying a saved viewlet

Parameter	Description
com.smarts.savedView	Name of the saved view that is displayed as a viewlet. Note that it is not necessary to include the file extension, ".icsvm", of the viewlet file. Saving views as viewlets provides additional information about saving a viewlet.

Note Tooltips are displayed only for Business Dashboard viewlets that are currently in focus. If you position the cursor over an item in a viewlet that is not in focus, no tooltip appears. To display tooltips in a viewlet other than the focused viewlet, click the second viewlet to cause it to gain focus.

Context sharing for viewlets

Context sharing is when the focus of one viewlet determines the content that is displayed in one or more viewlets on the same HTML page. A simple example illustrates how this might work. When instance context sharing is enabled for a Notification Log viewlet, a user can select a failed node in a Map viewlet and the Notification Log displays the notifications related to the selected map node. The VMware Smart Assurance Service Assurance Manager Operator Guide provides additional information about context sharing.

Parameters for context sharing with viewlets lists the parameters related to context sharing for viewlets and indicates which viewlets the parameters apply to. Setting the value to true enables context sharing.

Table 5-7. Parameters for context sharing with viewlets

Parameter	Description
com.smarts.context.notification	When true, causes the viewlet to listen for notification context changes. You can set this parameter for the following viewlets: ■ Map
com.smarts.context.instance	When true, causes the viewlet to listen for instance context changes. You can set this parameter for the following viewlets: ■ Map ■ Notification Log
com.smarts.context.summary	When true, causes a Notification Log viewlet to display the notifications graphed in the Summary viewlet. ■ Notification Log
com.smarts.context.collectionEntity	When true, causes the viewlet to display information related to the collection entity class. Collection entity classes include the following classes and their subclasses: ■ ServiceOffering ■ ServiceSubscriber ■ HierarchicalGroup ■ SelectiveGroup ■ ApplicationGroup You can set this parameter for the following viewlets: ■ Notification Log ■ Map ■ Summary
com.smarts.launch.globalMg	Name of the Global Manager for context when the viewlet is attached to more than one Global Manager.

Note Notification Properties viewlets always listen for notification context and Containment viewlets always listen for notification and instance context.

Automatic attach for viewlets

Automatic attach (autoAttach) enables you to configure a viewlet so that a user does not need to provide a username or password to view information provided by the Global Manager.

To make the connection to a Global Manager transparent, you must do the following:

- Edit two parameters in the dashboard.properties file (or in the viewlet HTML files).
- Configure the Dashboard credentials identifier (ID). This ID specifies the location of the VMware Smart Assurance Broker, the name of the Global Manager, a username, and a password.

Note The VMware Office of Product Security deems the use of this feature (Automatic attach by storing credentials on disk) as an unsecure mode of operation. Storing credentials on disk amounts to no user authentication. As such, VMware Corporation discourages the use of this feature.

Configure Dashboard viewlets to automatically attach to the Global Manager

The configuration of a Dashboard viewlet to automatically attach to a server involves defining two properties in the dashboard.properties file. Alternatively, the parameters can be defined in the HTML files used to launch the viewlets. If the properties are defined in the HTML files, they must be defined identically in all the viewlet HTML files.

Parameters for automatic attach with viewlets describes the parameters used to configure Dashboard viewlets to automatically attach to the Global Manager.

Table 5-8. Parameters for automatic attach with viewlets

Parameter	Description
com.smarts.viewlet.authhttp	The value of com.smarts.viewlet.authhttp specifies the URL of the servlet that will perform the lookup and retrieval of the stored userid and password credentials. The syntax of this parameter is http://<*tomcat_host*>:8080/SMViewletAuth/ViewletAuth.do?MASK=0 If web server is configured to run over https, then the syntax to be used is: https://<*tomcat_host*>:8443/SMViewletAuth/ViewletAuth.do?MASK=0, where port 8443 is the secure port (the port may differ depending on how the web server is configured). Using SSL (https://) provides additional information.
com.smarts.attach.alias	The value of com.smarts.attach.alias is a unique identifier that labels the credentials to be used by the auto attach mechanism. This unique identifier will have to be defined in a small server database using the administrator Dashboard credentials tool provided in the distribution.

Example of modified dashboard.properties file

An example of a modified dashboard.properties file that uses the parameters described in Example of modified dashboard.properties file is:

```
# This is a Java Properties file, for the GUI running as an Applet.
# Leading and trailing blanks are stripped from parameter values.
# If you have a property which you want to have leading or trailing blanks, enclose that parameter in
double quotes,
# e.g.: some.parameter="  I am a value with leading and trailing blanks  "
com.smarts.viewer.hideToolbarText=true
com.smarts.OemCustomization=NONE
com.smarts.securityEnabled=true
com.smarts.viewlet.authhttp=http://<tomcat_host>:8080/SMViewletAuth/ViewletAuth.do?MASK=0
com.smarts.attach.alias=GUI_ID
```

In this example, the parameter, com.smarts.viewlet.authhttp, refers to the URL of the servlet that will retrieve the credentials (username, password, broker, and Global Manager) from Tomcat.

The parameter, com.smarts.attach.alias, is set to the Dashboard credentials identifier named, GUI_ID. To configure GUI_ID, use the Dashboard credentials administrator's tool.

Configuring the Dashboard credentials identifier

The administrator must define the credentials ID using the Dashboard credentials tool. To configure the Dashboard credentials identifier:

1 Start a web browser specifying the following URL:

```
http://<tomcat_host>:8080/SMViewletAuth
```

2 Type a username and password that will be used in the future to access the Dashboard credential tool (for example, **admin**, **12345**).

3 Click **Submit**.

A message indicating a successful login appears.

Note This step assigns values to two parameters that control access to the tool: configuration.admin.user and configuration.admin.password. Use this username and password whenever you need to access this tool to edit Dashboard credential IDs.

4 Select **Edit Configuration**. The following text appears:

```
# Automatically generated configuration file
# admin credential: DO NOT EDIT 2 fields below
configuration.admin.user=admin
configuration.admin.password=12345
#[connection_alias1]
#    com.smarts.userid=admin
#    com.smarts.password=changeit
#    smarts.remote.broker=10.239.3.92:426
#    com.smarts.autoAttachDM=INCHARGE-SAM
#    com.smarts.clientips=10.239.*;10.7.2.[1-9];10.7.2.11
## end of entry connection_alias1
```

Parameters for automatic attach with viewlets explains the parameters used in this screen.

Table 5-9. Parameters for automatic attach with viewlets

Parameter	Description
connection_alias1	Replace this string of text with the credentials ID (for example, GUI_ID). You must replace this at both the beginning and end of the list.
com.smarts.userid	Username. This name must match a record in the serverConnect.conf file used by the Global Manager. In addition, this user must have an account with the Global Manager.
com.smarts.password	Password.

Table 5-9. Parameters for automatic attach with viewlets (Continued)

Parameter	Description
smarts.remote.broker	Location of the Broker in <*host*>:<*port*> format. For example, smarts-broker:426.
com.smarts.autoAttachDM	Name of the Global Manager, for example INCHARGE-SA.
com.smarts.clientips	Your IP for running the Dashboard applet should be inserted to the "com.smarts.clientips" list of the alias you are using for the Dashboard. For localhost, it might be 127.0.0.1. "com.smarts.clientips" controls which host can run the Dashboard without the need of knowing the connection credentials.

1 Copy the text beginning with the line, #[connection_alias1], and paste it at the end of the file. Uncomment and edit the parameters in these lines in order to create the credentials ID for the autoAttach feature. For example:

```
[GUI_ID]
    com.smarts.userid=oper
    com.smarts.password=oper
    smarts.remote.broker=qa-fancy:426
    com.smarts.autoAttachDM=INCHARGE-SA
    com.smarts.clientips=10.239.*;10.7.2.[1-9];10.7.2.11;127.0.0.1
# end of entry GUI_ID
```

2 Click **Save**.

The **Show Configuration** screen appears and the username and password are scrambled, for example:

```
# start of entry GUI_ID
[GUI_ID]
# The following fields CAN NOT be modified manually
    com.smarts.userid=5f2c442449242f41
    com.smarts.password=5f2c642e4d392041
# end of fields CAN NOT be modified manually
    smarts.remote.broker=qa-fancy:426
    com.smarts.autoAttachDM=INCHARGE-SA
    com.smarts.clientips=10.239.*;10.7.2.[1-9];10.7.2.11;127.0.0.1
# end of entry GUI_ID
```

3 Click **Edit** to return to the **Edit Configuration** screen.

4 Click **Logout** to exit.

By default, the configuration file with all the obfuscated credentials is stored in the directory where tomcat was started. The default filename is **viewletauth.conf**. The administrator can change this behavior and specify a different location. This can be done by specifying the following property:

```
<init-param>
        <param-name>com.smarts.viewletauth.conf</param-name>
  <param-value>file_path</param-value>
      </init-param>
```

5 Start the viewlet. If properly configured, the viewlet should bypass the login dialog box and attach directly to the Global Manager (INCHARGE-SA). (Note, this assumes that the Global Manager is already running.) When values for one or more of the parameters in Parameters for automatic attach with viewlets are not provided, the viewlet displays the login dialog box requesting credentials.

Recovering an administration account user ID and password

To recover an administration account user ID and password:

1 Using a text editor, open the viewletauth.conf file.

Note Theviewletauth.conf file is located in your Service Assurance Manager installation where tomcat is running.

Once the viewletauth.conf file is opened, locate the following lines located near the beginning of the file:

```
# Automatically generated configuration file
# admin credential: DO NOT EDIT 2 fields below
configuration.admin.user=5f24442449242f292d5e572150283341
configuration.admin.password=5f2041214a2a2432315f5641264e4545
```

2 Execute the following command replacing the encoded string value with the encrypted (encoded) configuration.admin.user text string value:

```
java -cp TOMCAT_INSTALL/webapps/SMViewletAuth/WEB-INF/classes
com.smarts.viewletauth.util.SMObfuscatorFactory -s encoded_string
```

3 Execute the following command replacing the encoded string value with the encrypted (encoded) configuration.admin.password text string value:

```
java -cp TOMCAT_INSTALL/webapps/SMViewletAuth/WEB-INF/classes
com.smarts.viewletauth.util.SMObfuscatorFactory -s encoded_string
```

Properties for map viewlets

Parameters for map viewlets describes the parameters which enable you to specify an element as the focus of a map viewlet and to determine what type of map is displayed. The map type must be relevant to the specified element.

Note When launching a map viewlet, you must specify the class of the element and instance name of the element for the map, otherwise the map will be empty. (The map viewlet does not include the topology browser so you cannot select an instance.) However, when the map is used in conjunction with other viewlets, such as the Notification Log viewlet, you can context share to display elements in the map. When you specify the instance in the URL without a map type, the console chooses a default map type for that instance.

Table 5-10. Parameters for map viewlets

Parameter	Description
com.smarts.launch.class	Name of the ICIM class of which the element is an instance.
com.smarts.launch.instance	Display name of the elements.
com.smarts.launch.mapType	Type of map that is displayed. Can include: ■ Physical for Physical Connectivity ■ IP for IP Connectivity ■ VLAN for VLAN Connectivity ■ Membership for Membership ■ Business for Business Services Maps ■ Application for Applications Maps
com.smarts.map.passive	When true, prevents console operations such as Show Map from displaying in the Map viewlet. Instead, actions such as Show Map display in a new Map viewlet.
com.smarts.launch.mapHops	Optional parameter that specifies how many hops the map should display. If a value is not specified, one hop is displayed. (Part of Launching in Context.)
com.smarts.fitInWindow	Value is boolean. When true, this optional parameter forces the Map to fit inside the window no matter how large the map.
com.smarts.map.showAllBusiness	Value is boolean. When true, this optional parameter allows all business topology to be seen in the map and not just the topology that has a direct path to the central map node.

Note You can also use the com.smarts.launch.class and com.smarts.launch.instance parameters for Notification Properties and Containment viewlets.

Configuring map viewlet for dashboard

To configure map viewlets for the dashboard:

1 Open the viewlet.properties.file(com.smarts.ViewletProperties) file located under the **<BASEDIR>**/smarts/tomcat/webapps/templates.

2 Add the following lines:

```
com.smarts.launch.class=<class name>
com.smarts.launch.instance=<instance name>
com.smarts.launch.globalMgr=<SAM Server>
com.smarts.launch.mapType=<Type>
```

3 Save the file.

Enabling client tools

Client tools are programs that can be invoked by an operator using the Global Console, web console, or Business Dashboard. Access to client tools can be controlled by associating or not associating tools to user profiles. Available client tools are displayed in a pop-up menu when an operator right-clicks on a notification or a device. The VMware Smart Assurance Service Assurance Manager Configuration Guide provides additional information about creating client tools.

Note Client tools that require the user to input values are not supported in either the web console or Business Dashboard.

In order for client tools to be invoked, the tool scripts must be placed in a directory accessible from the client machine (where the web browser is running). In an environment where Dashboard users share a network drive, the client tools can be placed in a directory on the network drive, thus avoiding the need to copy the tools onto each client machine.

To enable access to the client tools from the Business Dashboard, add the parameters described in Parameter to enable client tools to the appropriate properties file, for example, dashboard.properties.

Table 5-11. Parameter to enable client tools

Parameter	Description
com.smarts.clientToolsInApplet=true	When set to true, client tools are enabled in the web console or Business Dashboard. If this parameter is specified, and the com.smarts.webconsole.sitemod parameter (see below) is not specified, the Dashboard client tool function gets the client tools from the /actions/client directory in the web browser's default directory. (For IE, this may be Documents and Settings\Desktop. For Firefox, the Firefox installation base directory.) You must put a copy of the tool in this /actions/client path.
com.smarts.webconsole.sitemod	When this parameter is configured in the dashboard.properties file, the Business Dashboard searches for client tools in the directory path(s) specified. For example, com.smarts.webconsole.sitemod=F:/webconsoleSitemod;C:webconsoleSitemod/locl. If this parameter is specified, then com.smarts.clientToolsInApplet=true must also be specified to enable client tools in the web console or Business Dashboard. If this parameter is not specified, but com.smarts.clientToolsInApplet=true is specified, then client tools will be accessed from a default directory (see above).

Example

This example illustrates how to configure a tool named myClientTool.cmd where com.smarts.webconsole.sitemod=Z:/toolsdir. The two parameters would be set as follows:

```
com.smarts.clientToolsInApplet=true
```

com.smarts.webconsole.sitemod=Z:/toolsdir

The client tool must be placed in the /actions/client folder under Z:/toolsdir, for example:

```
Z:/toolsdir/actions/client/myClientTool.cmd
```

The parameter, com.smarts.webconsole.sitemod does not directly specify where the tool file is stored. Instead, it specifies where the /actions/client directory containing the tool file is stored.

If com.smarts.webconsole.sitemod is not specified, the /client/actions folder with the tool must be placed under the web browser's default directory. (For Internet Explorer, this may be Documents and Settings\Desktop. For Firefox, it is the Firefox installation base directory.)

Enabling the com.smarts.user.display property to pass client IP addresses to server scripts

To pass the client IP address instead of hostname to server scripts:

1. For web console, Java Web Start or dashboard, add the following line to the webconsole or dashboard properties file:

   com.smarts.user.display={IP}:0.0

2. To apply this option only to the Java Web Start, add the following property to the globalConsoleApp.jnlp file:
3. **<property name="com.smarts.user.display" value="{IP}:0.0"/>**.
4. For the command line console (sm_gui), add the following command line option:

   ```
   -D com.smarts.user.display={IP}:0.0
   ```

Verifying the configuration

If the correct masterpiece file was used and the com.smarts.user.display property was configured correctly, an entry similar to the following appears in the Java Console or the VMware Smart Assurance console's Java Message log:

Property com.smarts.user.display set to: 128.222.218.149:0.0 (was: {IP}:0.0)

Note The IP address varies depending upon your specific environment.

If the entry is not added to the logs, the configuration is not accurate.

URL-addressable viewlets

A URL-addressable viewlet is a viewlet that is displayed when a web client loads a particular URL. This enables you to dynamically generate a URL to display Map, Notification Properties, and Containment viewlets from within the context of another application. Additional viewlet properties describes the viewlet parameters to configure a URL-addressable viewlet.

To specify viewlet parameters directly in a URL, use the following syntax:

```
http://URL?class=viewlet&parameter_name=value[&...]
```

Class parameter values for URL-addressable viewlets lists values used with the class parameter typically used with URL-addressable viewlets and Separators used in URL-addressable viewlets describes the separators used.

Table 5-12. Class parameter values for URL-addressable viewlets

Parameter	Dashboard viewlets
class	Determines the Dashboard viewlet that is displayed: ■ com.smarts.webapps.SmWebMapApp(Maps) ■ com.smarts.webapps.SmWebNotificationPropertiesApp(Notification Properties) ■ com.smarts.webapps.SmWebContainmentApp(Containment)

Table 5-13. Separators used in URL-addressable viewlets

Separator	Description
?	Separates the filename of the HTML page specifying the JAR files from the parameter/value pairs.
&	Separates parameter/value pairs in the URL.

The following URL is an example of a URL-addressable viewlet that loads a Map viewlet with a specific element class and instance as the focus. It does not specify a map type. The example uses an HTML page included with the viewlet samples. Example of URL-addressable viewlet provides additional information.

```
        t
        http://<tomcat_host>:8080/templates/context.html?
class=com.smarts.webapps.SmWebMapApp&
com.smarts.propertiesFile=dashboard.properties&
com.smarts.launch.class=Router&
com.smarts.launch.instance=lab-gw.example.com s
```

The following URL, which is similar to the preceding, is an example of a URL-addressable viewlet that loads a Map viewlet with a specific element class and instance as the focus, and specifies a map type:

```
          t
           http://<tomcat host>:8080/templates/context.html?
class=com.smarts.webapps.SmWebMapApp&
com.smarts.launch.mapType=IP&
com.smarts.launch.class=Host&
com.smarts.launch.instance=qa-fancy s
```

When necessary, use the **dmctl** utility to list the notifications in the Global Manager and select the one that you want to use for the viewlet:

```
./dmctl -s INCHARGE-SA getInstances ICS_Notification
```

About the web console

The web console provides the same functionality as the Global Console, including the Topology Browser view that is not available as a viewlet. Client requirements defines the client systems requirements.

Users can open the VMware Smart Assurance web console after you complete the following tasks:

- Start the VMware Smart Assurance Business Dashboard service
- Create user profiles and user accounts

  The web console is installed to the *BASEDIR*/smarts/tomcat/webapps/webconsole directory.

Attaching with the web console

Two versions of the web console are available. With both versions, 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.

The first version functions as a typical Java applet. The HTML and JAR files are located on the host where the VMware Smart Assurance 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.

A second version of the web console uses Java Web Start. Java Web Start 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.

Note When opening the web console or a Dashboard viewlet, the JRE on the user's system displays a security warning. Users must select Yes or Always to allow the applet to run.

Using the web console

To display the web console, open the following URL:

```
http://<tomcat_host>:8080/webconsole/InChargeApplet.html
```

When attached, the web console displays inside the web browser window. If the user opens additional consoles using the File > Open menu options, each new console is displayed in its own window. When the user exits the web browser, all web console are closed, as is the connection to the Global Manager.

Using Java Web Start

You can also use Java Web Start for caching the applet locally for improved performance; the necessary applet files are not reloaded each time the applet is launched. To do so, point your browser at the following URL:

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

When you launch the cached applet, you are prompted to optionally add a Java Web Start icon on your desktop and a link in your Start menu.

If Java Web Start is not installed, you are prompted to use an automatic installer or directed to a Sun Microsystems web page, depending upon your operating system or browser.

Examples of Dashboard viewlets

6

About the sample viewlets

Each viewlet type is specified in an HTML page, with slight variations specified in separate HTML pages. Each sample combines several viewlets, which are displayed in a single HTML page through the use of HTML frames.

To ensure that the HTML pages that specify a viewlet display properly on all supported platforms and web browsers, VMware, Inc. has processed the HTML pages using Java Plug-in HTML Converter. The Java Plug-in Converter converts an HTML page that specifies Java applet to a form that can be processed by a web browser using the Java Plug-in.

Because the converted HTML files are not easily read, both the original HTML file and the converted files are included. For example, the original file for the Notification Log viewlets is named NotificationsAppOrig.html. The converted files, which are used to display the viewlet, have names such as NotificationsApp1.html.

To view the samples in a web browser, the VMware Smart Assurance Servlet Engine must be installed and running. You can open the following URL in your web browser, appending the filename specified for each sample. For example, if the VMware Smart Assurance Servlet Engine is running on host1, you can view sample 1, by opening the following URL:

```
http://<tomcat_host>:<port>/templates/sample1.html
```

In addition, you also can find all the sample viewlets in *BASEDIR*\smarts\tomcat\webapps\templates\

Index to the sample Dashboard viewlets

The best way to view the samples is to open the following HTML page in your web browser:

```
        BASEDIR\smarts\tomcat\webapps\templates\DashboardIndex.html
```

This page provides an index listing of all the sample viewlets and a ReadMe.html with hyperlinks to each.

Sample 1

You can view the first sample by opening sample1.html in your web browser. This sample includes the viewlets described in Viewlets of Sample1.html. The layout of this sample includes a Notification Log in the top frame row and a Summary and Status Table in the bottom frame row.

Table 6-1. Viewlets of Sample1.html

Frame page	Viewlet pages	Viewlet	Additional parameters
sample1.html	NotificationsApp1.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.collectionEntity has a a value of true. This indicates that the Notification Log viewlet will display notifications related to elements selected in the Status Table. Context sharing for viewlets provides additional information about context sharing.
	SummaryApp1.html	Summary View	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information about properties files.
	StatusTableAppAll.html	Status Table	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information on properties files.

Sample 2

You can view the second sample by opening sample2.html in your web browser. This sample includes the viewlets described in Viewlets of Sample2.html. The layout of this sample includes a Status Table and Map in the top frame row and a Notification Log in the bottom frame row.

Table 6-2. Viewlets of Sample2.html

Frame page	Viewlet pages	Viewlet	Additional parameters
sample2.html	StatusTableAppAll.html	Status Table	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.
	MapApp2a.html	Map	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.collectionEntity has a a value of true. This indicates that the Map viewlet will display the elements selected in the Status Table. Context sharing for viewlets provides additional information.
	NotificationsApp2a.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.collectionEntity has a a value of true. This indicates that the Notification Log viewlet will display notifications for elements selected in the Status Table. Context sharing for viewlets provides additional information. ■ com.smarts.context.instance has a a value of true. This indicates that the Notification Log viewlet will display notifications for elements selected from the Map. Context sharing for viewlets provides additional information.

Sample 3

You can view the third sample by opening sample3.html in your web browser. This sample includes the viewlets described in Viewlets of Sample3.html. The layout of this sample includes a Map in the top frame row and a Notifications Log and Containment viewlet in the bottom frame row.

Table 6-3. Viewlets of Sample3.html

Frame page	Viewlet pages	Viewlet	Additional parameters
sample3.html	MapApp3a.html	Map	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.notification has a a value of true. This indicates that the Map viewlet will display the elements related to notifications selected in the Notification Log. Context sharing for viewlets provides additional information.
	NotificationsAppAll.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.
	ContainmentAppAll.html	Containment	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.

Sample 4

You can view the fourth sample by opening sample4.html in your web browser. This sample includes the viewlets described in Viewlets in sample4.html. The layout of this sample includes a Notifications Log in the top frame row, a Summary and Notification Properties viewlets in the middle frame row, and a map in the bottom frame row.

Table 6-4. Viewlets in sample4.html

Frame page	Viewlet pages	Viewlet	Additional parameters
sample4.html	NotifcationsApp4.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file.Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.summary has a a value of true. This indicates that the Notification Log viewlet will display the notifications graphed by the selected Summary. Context sharing for viewlets provides additional information.
	SummaryAppAll.html	Summary	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.

Table 6-4. Viewlets in sample4.html (Continued)

Frame page	Viewlet pages	Viewlet	Additional parameters
	NotificationPropertiesAppAll.html	Notification Properties	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.
	MapApp4a.html	Map	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.notification has a a value of true. This indicates that the Map viewlet will display the elements related to notifications selected in the Notification Log. Context sharing for viewlets provides additional information.

Sample 5

You can view the fifth sample by opening sample5.html in your web browser. This sample includes the viewlets described in Viewlets in Sample5.html. The layout of this sample includes a Notification Log in the top frame row, a Summary and a map in the middle frame row, and a second Notification Log in the bottom frame row.

Table 6-5. Viewlets in Sample5.html

Frame page	Viewlet pages	Viewlet	Additional parameters
sample5.html	NotificationsAppAll.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.
	SummaryAppAll.html	Summary	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.

Table 6-5. Viewlets in Sample5.html (Continued)

Frame page	Viewlet pages	Viewlet	Additional parameters
	MapApp5a.html	Map	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.notification has a a value of true. This indicates that the Map viewlet will display the elements related to notifications selected in the Notification Log. Context sharing for viewlets provides additional information.
	NotificationsApp5.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.summary has a a value of true. This indicates that the Notification Log viewlet will display the notifications graphed by the selected Summary. Context sharing for viewlets provides additional information.

Sample 6

You can view the sixth sample by opening sample6.html in your web browser. This sample includes the viewlets described in Viewlets of Sample6.html. The layout of this sample includes a Status Table and Summary in the first row, a Notification Log in the second row, and a second Summary in the bottom row.

One notable difference between this sample and the previous samples is that sample6.html uses HTML TABLE tags in conjunction with IFRAME tags to position viewlets. An advantage of IFRAME tags is that the viewlets within the tags remain at their specified size. As the web browser window is resized, the viewlets within the table do not resize. When the browser window is made smaller than the specified size of a viewlet, scroll bars are displayed within the IFRAME.

Table 6-6. Viewlets of Sample6.html

IFrame page	Viewlet pages	Viewlet	Additional parameters
sample6.html	StatusTableAppAll.html	Status Table	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.
	SummaryApp6.html	Summary	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.collectionEntity has a a value of true. This indicates that the Summary viewlet will display a summary based on selections in the Status Table. Context sharing for viewlets provides additional information.

Table 6-6. Viewlets of Sample6.html (Continued)

IFrame page	Viewlet pages	Viewlet	Additional parameters
	NotificationsApp6.html	Notification Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information. ■ com.smarts.context.collectionEntity has a a value of true. This indicates that the Notification Log viewlet will display notifications for elements selected in the Status Table. Context sharing for viewlets provides additional information.
	SummaryAppAll.html	Summary Log	■ com.smarts.propertiesFile includes the parameters specified in the dashboard.properties file. Using a properties file to specify viewlet parameters provides additional information.

Example of URL-addressable viewlet

A URL-addressable viewlet is a viewlet that is displayed when a web client loads a particular URL. This enables you to dynamically generate a URL that displays the Map, Notification Properties, or Containment viewlets.

The base URL for displaying a viewlet is: http://<*tomcat_host*>:<*port*>/templates/context.html

This sample uses context.html as the base URL to load the JAR files necessary to display one of these three viewlets. At a minimum you must specify the type of viewlet to display.

Note Opening this page without specifying a viewlet displays a blank page.

Additional viewlet properties describes parameters and separators you can add to the base URL. *URL-addressable viewlets on page 48* provides additional information on the syntax of URL-addressable viewlets.

The following URL is an example of a URL-addressable viewlet that loads a map viewlet with a specific element class and instance as the focus. It does not specify a map type:

```
thttp://<tomcat_host>:8080/templates/context.html?
class=com.smarts.webapps.SmWebMapApp&
com.smarts.propertiesFile=dashboard.properties&
com.smarts.launch.class=Router&
com.smarts.launch.instance=lab-gw.example.com s
```

The following URL, which is similar to the preceding, provides an example of a URL-addressable viewlet that loads a Map viewlet with a specific element class and instance as the focus, and specifies a map type:

```
thttp://<tomcat
         host>:8080/templates/context.html?
class=com.smarts.webapps.SmWebMapApp&
com.smarts.launch.mapType=IP&
com.smarts.launch.class=Host&
com.smarts.launch.instance=qa-fancy s
```

The following URL provides an example of a URL-addressable viewlet that loads a Notification Properties viewlet. In the example, the class (optional) is ICS_Notification, and the instance is the name of the notification.

```
thttp://<tomcat
         host>:8080/templates/context.html?
class=com.smarts.webapps.SmWebNotificationPropertiesApp&
com.smarts.launch.class=ICS_Notification&
com.smarts.launch.instance=NOTIFICATION-Router_qa-mplspe4.example.com_Unresponsive s
```

The following URL provides an example of a URL-addressable viewlet that loads a Containment viewlet. The parameters for the Containment viewlet are the same as for a map: the class of the element and the instance name of the element.

```
thttp://<tomcat
         host>:8080/templates/context.html?
class=com.smarts.webapps.SmWebContainmentApp&
com.smarts.launch.class=Router&
com.smarts.launch.instance=qa-mplspe4.example.com s
```

Troubleshooting

7

This chapter includes the following topics:

- Troubleshooting overview
- Debugging the web console
- Sun Java Console (Plug-in message log)
- Log file location

Troubleshooting overview

Note The information contained in this chapter should be checked against your installation *before* problems arise. Contact VMware Smart Assurance support if it becomes necessary to clarify or resolve any questions or issues.

When troubleshooting reasons why the web console stops responding, the following files and parameters are reviewed:

- Required files

  When troubleshooting issues, VMware always asks for the following files:

  - rps files
  - Service Assurance Manager server log files
  - tomcat log files
  - sm_monitor log files
  - Console log files

- ics.conf files

  The console log files include the VMware Smart Assurance Java Message log file, the Sun Java Plug-in Console log file, and the VMware Smart Assurance server messages received log file. By reviewing the console log files, VMware can clearly determine if the console stopped responding because the system ran out of memory, or if the issue was the result of some other error. Using the server log file, VMware can review what occurred at the time the console stopped responding; for example, was the server busy or was there some other exception.

  From prior experience, a web console or global console stops responding as a result of the machine running out of memory.

- Debugging parameters.

  To have debugging information entered in the log files, you must have the console debugging parameters set by default, before you use the console and the console stops responding. You need to verify that the debugging information is being entered into the log files before there is a problem. Having the debugging parameters set ensures that no critical information is lost.

- Primary Java log file.

  The primary Java log file is the key tool used in diagnosing issues. This file gets overwritten at each startup time and is the file that saves the contents of the Console Java Log. Enable this file by typing the following command:

```
–Dcom.smarts.java.log=<
filename
>
```

- Asynchronous message log file.

  This log file is controlled by versioning. This file can get very large if the server is processing numerous notifications. Enable the asynchronous message log file by typing the following command:

```
Dcom.smarts.server.logMessages=true
```

Debugging the web console

Web console debugging parameters defines the parameters used to capture information when a web console stops responding.

Table 7-1. Web console debugging parameters

Parameter	Description
-Xmx350m -Dsun.net.inetaddr.ttl=0	This parameter gets around the broker failover cache problem.
-Dcom.smarts.server.logMessages=true	This parameter logs all the asynchronous messages that the server sends to the console, primarily, notification adds/changes/deletes.

Table 7-1. Web console debugging parameters (Continued)

Parameter	Description
-Dcom.smarts.java.log=<name>-webconsole.log	This parameter saves the VMware Smart Assurance Java Message Log content. This content is lost when the console stops responding. This log file must be copied and saved before restarting the web console. Content included in this log file could provide information on why the console stopped responding. **Note** Restarting the web console overwrites this file.
-Dcom.smarts.log.level=TRACE	This parameter results in maximum information being entered in the Java Message Log.
Heap size	VMware has tested a heap size of 350 M

Note The parameters are configured in the Java Plug-in Control Panel.

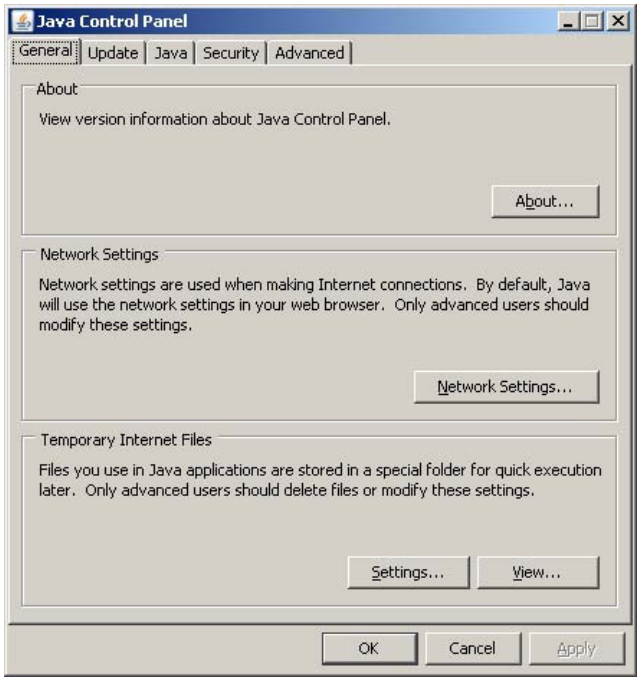
Configuring the log file directories

You can set up the log file directories for the com.smarts.server.logMessages and com.smarts.java.log files when the VMware Smart Assurance graphical user interface (GUI) is running as a web console in Internet Explorer.

The parameters can be enabled if they are set in the Java Control Panel in the workstations. To display the **Java Control Panel**, click **Settings** > **Control Panel** > **Java**.

The **Java Control Panel** appears as illustrated in Java Control Panel.

Figure 7-1. Java Control Panel



Internet Explorer defaults the console's com.smarts.writeable parameter to the desktop location where log files are written. You can verify this in the Sun Java Console window (not the VMware Smart Assurance Java Log file).

To verify that the com.smarts.writeable parameter is configured correctly:

From Internet Explorer, click **Tools** > *Sun Java Console*.

The Java Console appears. An example of the type of information available from the Sun Java Console is:

```
Java Plug-in 1.6.0_15
Using JRE version 1.6.0_15 Java HotSpot(TM) Client VM
User home directory = C:\Documents and Settings\bm1.SMARTS
----------------------------------------------------
c: clear console window
f: finalize objects on finalization queue
g: garbage collect
h: display this help message
l: dump classloader list
m: print memory usage
o: trigger logging
p: reload proxy configuration
q: hide console
r: reload policy configuration
s: dump system and deployment properties
t: dump thread list
v: dump thread stack
x: clear classloader cache
0-5: set trace level to <n>
----------------------------------------------------
Properties: com.smarts.fixed, com.smarts.sitemod, com.smarts.writeable not found.
Defaulting com.smarts.fixed, com.smarts.sitemod, com.smarts.writeable paths to: C:\Documents and
Settings\bm1.SMARTS\Desktop
```

The Properties line defines the location of the com.smarts.writable path. In this example, the path has been defaulted to the desktop and the user ID has been defined by default as the user ID for the specific workstation (in the example, the user ID is: bm1.SMARTS).

Configuring the log files

To retrieve the log files when running the web console:

1 Verify that the log files have been configured.

Note Configuring the log file directories provides additional information.

2 Create a new folder labeled "logs" on the workstation's desktop.

Use a command prompt to go to the directory specified in the Java Console message and create the new folder. For example, the path for the folder could be C:\Documents and Settings\bm1.SMARTS\Desktop\logs.

3 Click **Settings** > **Control Panel** > **Java**.

The **Java Control Panel** appears.

4 Click **Advanced** and add a new Java Runtime Parameter. (For Java 1.5 or later, click the Java tab and then click the View button to display the Java Runtime Settings window.) An example of the settings is:

-Dcom.smarts.server.logMessages=true

-Dcom.smarts.java.log=ZZZZ

-Dcom.smarts.log.level=TRACE

5 Edit the -Dcom.smarts.java.log=ZZZZ to the name of your java log file.

Note This log file does not get appended with versions. It needs to be saved before restarting the console. If the file is not saved before starting the console, the contents of the file are overwritten.

After configuring the java log file location, when you run Internet Explorer, the log of server message log file and the java log file are written into the desktop's logs directory. The java log file lists trace level messages.

Because the parameter was updated in the client's Java Plug-in, you do not need to restart the tomcat application.

You need to configure the log files location *before* a console stops responding so that information about the console not responding can be retrieved. VMware suggests verifying that the two debugging files (webconsole.log and server_Mesg_<*userid*>.log) get written to the logs directory.

Sun Java Console (Plug-in message log)

When the VMware Smart Assurance GUI is run as a web console, important debugging information is written to the Sun Java Plug-in Console. When the VMware Smart Assurance GUI stops responding but Internet Explorer continues to run (for example, if the web browser window goes blank), it is helpful to capture the contents of the Sun Java Plug-in console.

To enable the Sun Java Plug-in Console

To enable the Sun Java Plug-in Console:

1 From Settings, click **Control Panel**.

The **Control Panel** appears.

2 Double-click **Java**.

The **Java Control Panel** appears.

3 Click **Advanced**.

4 Expand the **Java console** option.

5 Click the **Show Console** option.

By default, the Sun Java Plug-in Console displays as a window (typically in the upper-left corner of the screen) in Internet Explorer. You can also display the Sun Java Console by clicking the Internet Explorer's Tools menu and selecting the Sun Java Console.

Capturing the Sun Java Plug-in Console content

To capture the Sun Java Plug-in Console content:

1 From the Java Console, click **Copy**.

The content displayed in the Sun Java Console is copied to the clipboard.

2 Open a new file in a text editor and paste the clipboard contents into the new file.

3 Save the file and send the file to VMware along with other debugging information.

Note If you are reporting web console problems that did not cause the web console to stop responding, you should also copy, save, and send this log file.

Log file location

Log file location defines the log file locations.

Table 7-2. Log file location

The	Is located in the
rps file	<*BASEDIR*>/smarts/local/repos/icf directory. **Note** Run dmctl -s NA-SAM-PRES and invoke the ICF_PersistenceManager::ICF-PersistenceManager. Save this information before sending the rps file to ensure all notifications are saved to the rps.
SAM server log	<*BASEDIR*>/smarts/local/logs directory.
sm_monitor logs	<*BASEDIR*>/smarts/local/logs/SM_Monitor directory.
Tomcat logs	<*BASEDIR*>/smarts/jakarta-tomcat-5.0.16/logs directory from where tomcat started.
ics.conf file	<*BASEDIR*>/smarts/local/conf/ics directory.
Console logs	C:\Documents and Settings\<*username*>.SMARTS\Desktop\logs directory.