# Contents

What’s New in Middle-Tier Administration for the SAS 9.4 Intelligence Platform .................................................. vii
Accessibility ......................................................................................................................................................... xv

## PART 1  Middle-Tier Overview  1

**Chapter 1 / Working in the Middle-Tier Environment** ................................................................. 3
- Understanding the Middle-Tier Environment ......................................................................................... 3
- Middle-Tier Software Components ........................................................................................................ 5
- SAS Web Infrastructure Platform ........................................................................................................... 6
- SAS Content Server ........................................................................................................................................ 11
- SAS Web Applications ............................................................................................................................. 11
- Starting the Web Applications ................................................................................................................... 13
- Middle-Tier Log Locations ....................................................................................................................... 14

**Chapter 2 / Interacting with the Server Tier** ...................................................................................... 15
- Configuration Shared between the Middle Tier and the Server Tier ................................................... 15
- SMTP Mail Server ....................................................................................................................................... 16
- SAS Web Infrastructure Platform Data Server ...................................................................................... 17
- JDBC Data Sources .................................................................................................................................... 21
- Job Execution Service ................................................................................................................................. 22

## PART 2  Middle-Tier Components  27

**Chapter 3 / Administering SAS Web Server** ..................................................................................... 29
- About SAS Web Server ............................................................................................................................... 29
- Installing SAS Web Server ....................................................................................................................... 30
- Understanding the SAS Web Server Configuration ............................................................................... 31
- Understanding SAS Web Server Management ....................................................................................... 32
- Monitoring SAS Web Server ................................................................................................................... 33

**Chapter 4 / Administering SAS Web Application Server** ............................................................... 35
- About SAS Web Application Server ........................................................................................................ 35
- Installing SAS Web Application Server ................................................................................................... 36
- Understanding SAS Web Application Server Configuration ............................................................... 37
- Deploying Web Applications ..................................................................................................................... 38
- Understanding SAS Web Application Server Management ................................................................. 38
- Monitoring SAS Web Application Server ............................................................................................... 40
- Checking Prerequisite Servers ................................................................................................................. 40

**Chapter 5 / Administering Cache Locator** ....................................................................................... 43
- About the Cache Locator ........................................................................................................................... 43
- Installing Cache Locator ............................................................................................................................ 43
### Contents

Understanding the Cache Locator Configuration ........................................... 44
Setting the Bind Address ............................................................................. 44
Modifying the Configuration to Accommodate a Firewall .......................... 45

**Chapter 6 / Administering JMS Broker** ................................................. 49
About JMS Broker ...................................................................................... 49
Installing JMS Broker ................................................................................ 49
Understanding the JMS Broker Configuration .......................................... 50
Monitoring JMS Broker .............................................................................. 50

**PART 3  Middle-Tier Applications** .......................................................... 51

**Chapter 7 / Administering the SAS Web Infrastructure Platform** .......... 53
About SAS Web Infrastructure Platform ...................................................... 54
Using Configuration Manager .................................................................... 56
Setting Global Properties for SAS Applications ....................................... 59
Specifying Connection Properties ................................................................ 64
Configuring Auditing for SAS Web Applications ....................................... 67
Using the SAS Web Administration Console ............................................. 73

**Chapter 8 / Administering SAS Web Applications** ............................... 85
About SAS Deployment Manager ............................................................... 85
Rebuilding the SAS Web Applications ....................................................... 86
Redeploying the SAS Web Applications ..................................................... 91
Reconfiguring the Web Application Server ................................................. 93
Administering Logging for SAS Web Applications .................................... 93

**Chapter 9 / Administering SAS Logon Manager** .................................. 101
About SAS Logon Manager ...................................................................... 101
Customizing Sign-in, Sign-out, and Time-out Messages ............................ 102
Displaying a Warning Message for Inactive User Sessions ........................ 103
Configuring the HTTP Session Time-out Interval ...................................... 105
Customizing the Sign-out URL ................................................................... 108
Changing the Banner Title ....................................................................... 109
Changing the Appearance of the Sign-in Page ........................................... 109
Configuring the Global Single Sign-On Time-out Interval ....................... 110
Configuring Guest Access ......................................................................... 111
Configuring Middle-Tier Security Policies ................................................. 113
Disabling Concurrent Sign-in Sessions ..................................................... 115
Disabling the SAS Trusted User Account ................................................ 115
Disabling Cross-Frame Scripting ............................................................... 116

**Chapter 10 / Administering the SAS Content Server** ............................. 117
About the SAS Content Server .................................................................. 118
SAS Content Server Storage ...................................................................... 120
Moving Content or Backing Up the SAS Content Server .......................... 120
Preventing File Types from Being Uploaded ............................................. 120
Deploying Content Manually to the SAS Content Server .......................... 121
Using the SAS Content Server Administration Console ....................... 125
Enabling the Data Store ............................................................................ 131
Implementing Authorization for the SAS Content Server .......................... 133
Manual Configuration Tasks ...................................................................... 135
Chapter 11 / Administering the SAS BI Web Services ........................................ 139
  Overview of SAS BI Web Services ......................................................... 139
  Managing Generated Web Services ..................................................... 140
  Configuring SAS BI Web Services for Java ......................................... 140
  Overview of Security for Web Services ................................................ 144
  Securing SAS BI Web Services for Java .............................................. 145

Chapter 12 / Administering SAS Web Application Themes .......................... 147
  Overview .............................................................................................. 147
  Steps for Defining and Deploying a New Theme .................................. 150
  Deleting a Custom Theme from the Metadata ..................................... 160
  Migrating Custom Themes .................................................................. 160

Chapter 13 / Administering SAS Flex Application Themes ....................... 163
  Introduction to SAS Flex Application Themes .................................. 163
  Benefits of SAS Flex Application Themes ......................................... 163
  Location of SAS Flex Application Themes ......................................... 164

Chapter 14 / Administering the Search Facility ......................................... 165
  About the Search Facility .................................................................... 165
  About the Search Index Providers ....................................................... 166
  Specifying Configuration Properties for the Search Interface to SAS .... 167

PART 4  Advanced Topics 169

Chapter 15 / Managing Devices ................................................................. 171
  Viewing SAS Web Report Studio Reports on Mobile Devices .............. 172
  Managing Mobile Devices ................................................................... 173
  Capabilities for SAS Mobile BI ........................................................... 176
  Accessing and Using SAS Mobile BI .................................................. 178
  Adjusting the Logging Configuration .................................................. 180
  Configuration Properties: SAS Mobile BI .......................................... 180
  Supported OLAP Functionality ............................................................ 183
  Troubleshooting: SAS Mobile BI ......................................................... 183
  Advanced Properties for SAS Mobile BI ............................................ 184

Chapter 16 / Best Practices for Configuring Your Middle Tier .................. 187
  Sample Middle-Tier Deployment Scenarios ........................................ 187
  Adding a Vertical Cluster Member ....................................................... 192
  Adding a Horizontal Cluster Member .................................................. 193
  Updating a Horizontal Cluster Member .............................................. 195
  Tuning the Web Application Server .................................................... 195
  Configuring HTTP Sessions in Environments with Proxy Configurations 195

Chapter 17 / High-Availability Features in the Middle Tier ....................... 197
  Overview of High-Availability Features .............................................. 197
  SAS Web Application Server ............................................................... 198
  SAS Web Server ................................................................................. 200
  JMS Broker ......................................................................................... 201
  Cache Locator ..................................................................................... 202
  SAS Environment Manager ................................................................. 205
What's New in Middle-Tier Administration for the SAS 9.4 Intelligence Platform

Overview

SAS is currently shipping the third maintenance release for SAS 9.4.

The SAS 9.4 middle-tier software includes changes to several SAS clients and infrastructure components. This book provides details associated with these capabilities. The capabilities that are introduced or enhanced since the initial SAS 9.4 release are highlighted below.

- "SAS Web Server and SAS Web Application Server"
- "Enhancements to Support SAS Web Application Server Clustering"
- "SAS Environment Manager"
- "SAS Web Infrastructure Platform Data Server"
- "Enhancements to SAS Logon Manager"
- "Enhancements for SAS Content Server"
- "Support for Web Application Archive Files"

The first maintenance release for SAS 9.4 has the following changes and enhancements:

- "Support for Customizing Web Application Content"
- "Enhancements for Managing Devices"
- "Support for TLS with Client Certificate Authentication"
- "Enhancements to SAS Logon Manager"

The second maintenance release for SAS 9.4 has the following improvements and changes:

- "Enhancements for SAS Content Server"
- "Enhancements for Managing Devices"

The October 2014 release for SAS 9.4 has updates to SAS Information Retrieval Studio. See "Updates to SAS Information Retrieval Studio for TLS".
Starting in May 2015, guest access support is available through SAS Logon Manager. See “Support for Guest Access”.

The third maintenance release for SAS 9.4 provides significant enhancements that support security configuration and management. Some of the configuration changes are required for existing deployments. Support has been added for the following enhancements and updates:

- “Support for Enabling Auditing of Internal Accounts”
- “Support for Forcing Users to Log Off”
- “Support for Enabling Audit Profiles”
- “Support for Management of the Trusted CA Bundle”
- “Security Support for SAS Web Applications”
- “Reduction in the Start-Up Time of SAS Web Application Server”

SAS Web Server and SAS Web Application Server

The initial SAS 9.4 middle-tier software includes SAS Web Server for use as an HTTP server and SAS Web Application Server. SAS Web Application Server is a lightweight server that provides enterprise-class features for running SAS web applications. Both products can be installed and configured automatically with the SAS Deployment Wizard.


For more information about SAS Web Application Server, see Chapter 4, “Administering SAS Web Application Server,” on page 35.

Enhancements to Support SAS Web Application Server Clustering

The initial SAS 9.4 release includes enhancements to the SAS Deployment Wizard to simplify SAS Web Application Server clustering. In previous releases, the following steps required manual configuration, but are performed automatically in this release:

- install a Java environment and web application server software
- create repository configuration files for each instance of SAS Content Server
- configure a load-balancing HTTP server

With the enhancements, you can easily configure vertical cluster members (additional server instances on the same machine) and horizontal cluster members (install and configure servers on additional machines).

Combining vertical and horizontal clustering is also supported and can be configured easily.
Note: There are SAS applications that do not support middle-tier clustering. As a result, those applications run on the master node in the cluster. If the master node is not available, then the application is not available (even if other SAS applications are available on other nodes in the cluster). For a list of SAS applications that do not support clustering, see Exceptions to the Middle-Tier Clustering Support in the SAS Guide to Software Updates.

SAS Environment Manager

SAS Environment Manager provides a number of systems and application management features for managing the SAS servers in your deployment. An agent is installed on each machine in the deployment. The agent collects metrics from the server processes and operating system running on the machine and sends them to the SAS Environment Manager server.

Both the agents and the server can be installed and configured automatically with the SAS Deployment Wizard.

SAS Web Infrastructure Platform Data Server

Starting with the initial SAS 9.4 release, SAS Web Infrastructure Platform Data Server is included, which replaces the SAS Framework Data Server that was used in SAS 9.3. The data server provides a transactional store for SAS middle-tier software.

The server can be installed and configured automatically with the SAS Deployment Wizard. The server is based on PostgreSQL 9.1.9. SAS configures a single server instance and SAS Web Application Server instances are configured with JDBC data sources that access the server. SAS Environment Manager also stores transactional information in the server.

For more information, see “SAS Web Infrastructure Platform Data Server” on page 17.

Enhancements to SAS Logon Manager

In previous releases, the SAS Logon Manager enabled administrators to deny concurrent logons. Starting with the initial SAS 9.4 release, this feature is enhanced to offer the ability to log off from the existing session. This setting enables users to access the applications that they need, and administrators are assured that only one session is active at a time.

For SAS 9.4, SAS Logon Manager uses the Central Authentication Service (CAS) that is available from Jasig. This change enables single sign-on so that users can access multiple SAS web applications seamlessly.
In the first maintenance release for SAS 9.4, SAS Logon Manager enables you to customize the behavior of the Sign Out button, in order to integrate with various security scenarios.

For more information, see Chapter 9, “Administering SAS Logon Manager,” on page 101.

---

**Enhancements for SAS Content Server**

SAS Content Server is a web application that provides WebDAV features for your SAS deployment. Starting with the initial SAS 9.4 release is an update for SAS Content Server to provide JCR 2.0 features.

By default, the SAS Content Server is also enhanced to use the SAS Web Infrastructure Platform Data Server for storage. In previous releases, this was an option during the installation process. Using the database for storage simplifies using SAS Content Server in a web application server cluster because there is no longer any need for repository reconfiguration.

Beginning in the second maintenance release for SAS 9.4, SAS Content Server enables you to prevent certain file extensions and MIME types from being uploaded by specifying the extensions and types in the config.xml file. By default, any file type can be uploaded to the SAS Content Server. By disallowing certain attachment types from being uploaded, you can ensure that a file extension matches its contents and provide file scanning capabilities.

Also, starting in the second maintenance release for SAS 9.4, you can manually configure a file or database data store for SAS Content Server. The data store enables you to store large files or databases. The benefits of using data stores over traditional storage methods include elimination of redundant files and reduced temporary file overhead.

For more information, see Chapter 10, “Administering the SAS Content Server,” on page 117.

---

**Support for Web Application Archive Files**

Starting with the initial SAS 9.4 release, the web applications are managed as EAR files, but they are deployed as web application archive (WAR) files. In previous SAS releases, the SAS web applications were managed and deployed as enterprise web application archive (EAR) files.

---

**Support for Customizing Web Application Content**

Beginning in the first maintenance release for SAS 9.4 is the ability to add custom content to a SAS web applications.
Enhancements for Managing Devices

Starting with the first maintenance release for SAS 9.4 is added functionality for managing mobile devices that use SAS Mobile BI. Devices are managed either by inclusion or exclusion.

Starting with the second maintenance release for SAS 9.4, SAS Mobile BI 7.1 has a new user interface. The new interface does not include a banner. For this reason, support for the configuration properties that customize the banner in the native mobile viewers is discontinued in SAS Mobile BI 7.1.

For more information, see Chapter 15, “Managing Devices,” on page 171.

Support for TLS with Client Certificate Authentication

Beginning in the first maintenance release for SAS 9.4, Transport Layer Security (TLS) configuration allows clients to authenticate with the SAS middle tier using a client certificate that is installed in their web browser. When a client certificate is used for authentication and installed in a web browser, you are not required to provide a user name and password to log on. There are two possible configurations: TLS for SAS Web Server and SAS Web Application Server and TLS for a stand-alone SAS Web Application Server.

For more information, see “Support for TLS with Client Certificate Authentication” on page 242.

Updates to SAS Information Retrieval Studio for TLS

Starting with the October 2014 release for SAS 9.4, TLS is supported for search by SAS Information Retrieval Studio. To configure TLS for previous releases, manual updates must be made to the \SAS-configuration-directory\Lev
Web\Applications\SearchInterfacetoSASContent\url_list.txt file.

For more information, see “Configuring for TLS” on page 166.

Support for Guest Access

Beginning in the May 2015 release for SAS 9.4, guest access is available through SAS Logon Manager for software clients that specifically allow guest logons. An optional feature, guest access provides anonymous access to a subset of resources and functionality in some SAS web applications.
Support for Enabling Auditing of Internal Accounts

Beginning in the third maintenance release for SAS 9.4, you can enable auditing support for the following accounts by updating the `-Dspring.profiles.active` JVM option:

- creating internal accounts
- updating internal account settings
- deleting internal accounts
- setting passwords for internal accounts
- changing passwords for internal accounts

For more information, see “Enable Auditing for Internal Accounts” on page 70.

Support for Forcing Users to Log Off

Starting with the third maintenance release for SAS 9.4, an administrator can close a session, effectively causing a user's logoff from a SAS web application, using the SAS Web Administration Console.

For more information, see “Force Users to Log Off” on page 75.

Support for Enabling Audit Profiles

Beginning in the third maintenance release for SAS 9.4, you can enable Spring audit profiles by updating the `-Dspring.profiles.active` JVM option.

For more information, see “Enable Audit Profiles” on page 70.

Support for Management of the Trusted CA Bundle

Starting with the third maintenance release for SAS 9.4 are security improvements that provide additional controls and setup for TLS encryption, and simplify TLS support. The SAS Deployment Manager can be used to automate the process of updating the list of trusted CA certificates, known as the trusted CA bundle. At installation, a list of trusted CA certificates that are distributed by Mozilla is installed and SAS products are automatically configured to use this.
You can then use the SAS Deployment Manager to add your own trusted certificates to this list.

For more information, see “Configuring Middle-Tier Services for the Third Maintenance Release for SAS 9.4” on page 243.

**Security Support for SAS Web Applications**

Beginning in the third maintenance release for SAS 9.4 is added security for SAS web applications. In scenarios where applications are using the SAS middle tier as a proxy for accessing external URLs, additional security has been added through a whitelist, or security filter, of allowed sites. You can also whitelist certain HTTP request methods.

SAS web applications that require external access to URLs must also have a whitelist of URLs that can be accessed.

For additional information, see:
- “Whitelist of Websites and Methods Allowed to Link to SAS Web Applications” on page 266
- “Configuring the Cross Domain Proxy Servlet through a Whitelist” on page 268
- “Enabling Support for Forward Proxy Authentication” on page 269

**Reduction in the Start-Up Time of SAS Web Application Server**

Starting with the third maintenance release for SAS 9.4, SAS has made changes that are expected to result in a 40% to 50% improvement (decrease) in start-up time for SAS Web Application Server. No configuration changes are needed after applying the maintenance release. You should automatically see the improvements when you restart the application server. For more information, see [http://support.sas.com/resources/papers/proceedings15/SAS1904-2015.pdf](http://support.sas.com/resources/papers/proceedings15/SAS1904-2015.pdf).

**Note:** The improvement in start-up time will vary based on the specific hosting environment, including but not limited to the operating system and hardware of the server where SAS Web Application Server is installed.
Accessibility

For information about the accessibility of any of the products mentioned in this document, see the usage documentation for that product.
Part 1

Middle-Tier Overview

Chapter 1
  Working in the Middle-Tier Environment ........................................ 3

Chapter 2
  Interacting with the Server Tier ......................................................... 15
Understanding the Middle-Tier Environment

The middle tier of the SAS Intelligence Platform enables users to access intelligence data and functionality with a web browser. This tier provides web-based interfaces for report creation and information distribution, while passing analysis and processing requests to the SAS servers.

The middle tier of the SAS Intelligence Platform provides an environment for running applications such as SAS Web Report Studio and SAS Information Delivery Portal. These applications run in a web application server and have a
graphical user interface that users navigate with a web browser. These applications rely on servers on the SAS server tier to perform SAS processing, including data query and analysis.

The following figure shows how the middle tier interacts with the other tiers of the SAS Intelligence Platform. For a description of these components, see SAS Intelligence Platform: Overview.

Figure 1.1  Architecture of the SAS Intelligence Platform

The middle tier includes the following software elements:

- SAS Web Server and SAS Web Application Server.
- a Java Runtime Environment (JRE).
- SAS web applications, which can include SAS Web Report Studio, the SAS Information Delivery Portal, the SAS BI Dashboard, and other SAS products and solutions.
- the SAS Web Infrastructure Platform, which includes the SAS Content Server and other infrastructure applications and services.
- the JMS Broker, which is used to provide distributed communication with Java Messaging Services. Some SAS web applications use queues and topics for business logic.
- the Cache Locator, which is used by SAS web applications to locate and connect to a distributed cache. The SAS web applications use the cache to maintain awareness of user sessions and to share application data.
- SAS Environment Manager, which is used to monitor and manage the server tier and middle tier of the SAS deployment.

The SAS Intelligence Platform architecture provides the flexibility to distribute these components according to your organization’s requirements. For small implementations, the middle-tier software, SAS Metadata Server, and other SAS servers, such as the SAS Workspace Server and SAS Stored Process Server,
can all run on the same machine. In contrast, a large enterprise might have multiple servers and a metadata repository that are distributed across multiple platforms. The middle tier in such an enterprise might distribute the web applications to many web application server instances on multiple machines.

The following figure illustrates the middle-tier components:

**Figure 1.2 Middle-Tier Components**

Middle-Tier Software Components

**SAS Web Server**

SAS Web Server is included with SAS 9.4 software. It is an HTTP server that is configured as a single connection point for SAS web applications. When the SAS middle tier is clustered, SAS Web Server is automatically configured to perform load balancing.

HTTPS is also supported and can be configured during initial installation and configuration of SAS Web Server. Alternatively, SAS Web Server can be reconfigured after the initial deployment to support HTTPS.

**SAS Web Application Server**

SAS Web Application Server is provided with SAS 9.4 software. It provides the execution environment for the SAS web applications. The SAS Deployment
Wizard can automatically configure the web application server, or you can configure it manually.

The following applications and services run in the web application server environment:
- applications and services that are part of the SAS Web Infrastructure Platform
- the SAS Web Report Studio, SAS Information Delivery Portal, SAS BI Dashboard, and SAS Help Viewer for the Web applications

Depending on which products and solutions you have purchased, your site might have additional web applications.

**Java Runtime Environment**

The SAS middle-tier environment includes a Java Runtime Environment that is included with SAS 9.4 software. You do not need to install a separate Java environment for the middle-tier environment.

**JMS Broker**

A JMS Broker instance is configured as a server on the machine that is used for the SAS middle tier. This software fully implements the Java Message Service 1.1 specification and acts as a message broker. It provides advanced features such as clustering, multiple message stores, and the ability to use file systems, and databases as a JMS persistence provider.

**Cache Locator**

SAS Web Application Server uses the distributed data cache that is available with VMware vFabric GemFire. SAS uses the cache as a peer-to-peer cache. In order for the instances of SAS Web Application Server to join as members of the cache, the Cache Locator is used. The locator provides the mechanism for peer discovery. The locator is used by instances of SAS Web Application Server and the SAS Web Infrastructure Platform Scheduling Services.

**SAS Environment Manager**

The SAS middle-tier environment includes SAS Environment Manager. This software includes an agent process that is installed on each server-tier and middle-tier machine in the deployment. Each agent gathers performance metrics and transfers the data to a server process that runs on a middle-tier machine. The server process includes a web application server that provides a web-based administrative interface. Administrators use a web browser to monitor and manage numerous components in the SAS environment.

**SAS Web Infrastructure Platform**

The SAS Web Infrastructure Platform is a collection of services and applications that provide common infrastructure and integration features for the SAS web applications.
Services and Applications in the SAS Web Infrastructure Platform

Services and applications in the Web Infrastructure Platform provide the following benefits:

- consistent installation, configuration, and administration tasks for web applications
- consistent user interactions with web applications, such as logon
- integration among web applications as a result of sharing common resources

The following services and applications are included in the SAS Web Infrastructure Platform:

<table>
<thead>
<tr>
<th>Application or Service</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Authorization Service</td>
<td>This service is used by some SAS web applications that manage authorization through web services.</td>
</tr>
<tr>
<td>SAS BI Web Services for Java</td>
<td>Can be used to enable your custom applications to invoke and obtain metadata about SAS Stored Processes. Web services enable distributed applications that are written in different programming languages and that run on different operating systems to communicate using standard web-based protocols. Simple Object Access Protocol (SOAP) is a common protocol. SAS includes support for JSON and REST as well. The SAS BI Web Services for Java interface is based on the XML For Analysis (XMLA) Version 1.1 specification.</td>
</tr>
<tr>
<td>SAS Content Server</td>
<td>Stores digital content (such as documents, reports, and images) that can be created and used by the SAS web applications.</td>
</tr>
<tr>
<td>SAS Deployment Backup and Recovery Tool</td>
<td>Enables deployment-wide backup and recovery services. For more information, see SAS Intelligence Platform: System Administration Guide.</td>
</tr>
<tr>
<td>SAS Identity Services</td>
<td>Provides SAS web applications with access to user identity information.</td>
</tr>
<tr>
<td>SAS Logon Manager</td>
<td>Provides a common user authentication mechanism for SAS web applications. It displays a dialog box for user ID and password entry, authenticates the user, and launches the requested application. SAS Logon Manager supports a single sign-on authentication model. When this model is enabled, it provides access to a variety of computing resources (including servers and web pages) during the application session without repeatedly prompting the user for credentials. You can configure SAS Logon Manager to display custom messages and to specify whether a logon dialog box is displayed when users log off.</td>
</tr>
<tr>
<td>Application or Service</td>
<td>Features</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAS Preferences Manager</td>
<td>Provides a common mechanism for managing preferences for SAS web applications. The application enables administrators to set default preferences for locale, theme, alert notification, time, date, and currency. In the SAS Information Delivery Portal, users can view the default settings and update their individual preferences.</td>
</tr>
<tr>
<td>SAS Principal Services</td>
<td>Enables access to core platform web services for SAS applications.</td>
</tr>
<tr>
<td>SAS Shared Web Assets</td>
<td>Contains graph applet JAR files that are shared across SAS web applications. They display graphs in stored processes and in the SAS Stored Process Web Application.</td>
</tr>
</tbody>
</table>
| SAS Stored Process Web Application     | Provides a mechanism for web clients to run SAS Stored Processes and return the results to a web browser. The SAS Stored Process Web Application is similar to the SAS/IntrNet Application Broker, and has similar syntax and debug options. Web applications can be implemented using the SAS Stored Process Web Application, the Stored Process Service API, or a combination of both. Here is how the SAS Stored Process Web Application processes a request:  
1 A user enters information in an HTML form using a web browser and then submits it. The information is sent to a web server, which invokes the first component, the SAS Stored Process Web Application.  
2 The Stored Process Web Application accepts data from the web server, and contacts the SAS Metadata Server for retrieval of stored process information.  
3 The stored process data is then sent by the Stored Process Web Application to a stored process server via the object spawner.  
4 The stored process server invokes a SAS program that processes the information.  
5 The results of the SAS program are sent back through the web application and web server to the web browser. |
<p>| SAS Notification Template Editor        | Enables administrators to create and edit messages that are sent as notifications to end users of SAS applications.                        |</p>
<table>
<thead>
<tr>
<th>Application or Service</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Web Administration Console</td>
<td>Provides features for monitoring and administering middle-tier components. This browser-based interface enables administrators to perform the following tasks:</td>
</tr>
<tr>
<td></td>
<td>1. Monitor users who are logged on to SAS web applications, and send email to them.</td>
</tr>
<tr>
<td></td>
<td>2. View user-level audit information such as the number of users, successful logons, unsuccessful logons, and find the time of a user’s last logon.</td>
</tr>
<tr>
<td></td>
<td>3. Manage permissions for folders and documents that are managed by SAS Content Services.</td>
</tr>
<tr>
<td></td>
<td>4. Manage templates and letterheads that are used as part of messages that are sent as notifications to end users of SAS applications.</td>
</tr>
<tr>
<td></td>
<td>5. View configuration information for each middle-tier component.</td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform</td>
<td>Enables administrators to set web-layer permissions on folders and documents for SAS applications that use SAS Content Services for access to digital content. You can access the permissions manager with the SAS Web Administration Console.</td>
</tr>
<tr>
<td>Platform Services</td>
<td>Provides a common infrastructure for SAS web applications. The infrastructure supports activities such as auditing, authentication, configuration, status and monitoring, email, theme management, and data sharing across SAS web applications.</td>
</tr>
<tr>
<td>SAS Workflow</td>
<td>Provides the web services that implement workflow management. The SAS Workflow services are used by SAS applications and solutions for tightly integrated workflow management.</td>
</tr>
</tbody>
</table>

In the middle tier, the SAS Web Infrastructure Platform plays a critical role with a collection of middle-tier services and applications that provide basic integration services.

In the web application server, two sets of services are available to all SAS web applications:

- SAS Foundation Services
- SAS Web Infrastructure Platform Services

**SAS Foundation Services**

The SAS Foundation Services is a set of core infrastructure services that enables Java programmers to write distributed applications that are integrated with the SAS platform. This suite of Java application programming interfaces provides core middleware infrastructure services. These services include the following:

- client connections to SAS Application Servers
- dynamic service discovery
- user authentication
■ profile management
■ session management
■ activity logging
■ metadata and content repository access
■ connection management
■ WebDAV service

Extension services for information publishing, event management, and SAS Stored Process execution are also provided. All of the SAS web applications that are described in this document use the SAS Java Platform Services. If you have correctly installed and configured the web applications, the platform services are defined in your SAS metadata repository.

You can verify this metadata in the SAS Management Console. Depending on the web applications that were installed, the SAS Portal Local Services (used by the SAS Information Delivery Portal) are displayed in the SAS Management Console.

In addition, other applications and portlets might have deployment of their own local services.

### SAS Web Infrastructure Platform Services

The SAS Web Infrastructure Platform Services provide common infrastructure and integration features that can be shared by any SAS application. Here is a description of the features:

■ Audit provides a single, common auditing capability.

■ Authentication is a common method for authenticating middle-tier applications. A corresponding web service provides connectivity based on WS security standards for web service clients.

■ Configuration is a standard way to define, store, and retrieve configuration information for SAS applications.

■ Directives provide application integration so that SAS applications can share intelligence and data. Applications can link to one another without requiring specific information about a particular deployment location.

■ Mail is a single, common mechanism for Simple Mail Transfer Protocol (SMTP)-based mail.

■ Status and monitoring is a collective set of services providing information about the configured or functioning system.

■ Comment service enables users to add comments, with or without an attachment. This feature enables the capture of human intelligence and supports collaborative decision making related to business data.

■ Alerts service enables users to register to receive time-sensitive, action-oriented messages when a specified combination of events and conditions occurs. Alerts can be sent to the user's email address or displayed in the SAS Information Delivery Portal.

■ Themes provide access to theme definitions for presentation assets used in web applications.
- SAS Workflow Services enable applications to interact with business processes that run in the SAS Workflow Engine.
- Registry provides access to services for desktop clients; a client needs to know only a single endpoint to determine other required locations.

**SAS Workflow**

SAS Workflow provides services that work together to model, automate, integrate, and streamline business processes. It provides a platform for more efficient and productive business solutions. SAS Workflow is used by SAS solutions that benefit from business process management.

SAS Workflow Studio is a desktop client application that is used to design and deploy workflows. The SAS middle tier hosts the workflow engine and the workflow services.

**SAS Content Server**

The SAS Content Server is part of the SAS Web Infrastructure Platform. This server stores digital content (such as documents, reports, and images) that is created and used by SAS web applications. For example, the SAS Content Server stores report definitions that are created by users of SAS Web Report Studio, as well as images and other elements that are used in reports. A process called content mapping ensures that report content is stored using the same folder names, folder hierarchy, and permissions that the SAS Metadata Server uses to store corresponding report metadata.

In addition, the SAS Content Server stores documents and other files that are to be displayed in the SAS Information Delivery Portal or in SAS solutions.

To interact with the SAS Content Server, client applications use Web Distributed Authoring and Versioning (WebDAV) based protocols for access, versioning, collaboration, security, and searching. Administrative users can use the browser-based SAS Web Administration Console to create, delete, and manage permissions for folders on the SAS Content Server. Administrative users can also search the SAS Content Server by using industry-standard query syntax, including XML Path Language (XPath) and DAV Searching and Locating (DASL).

**SAS Web Applications**

The SAS web applications described in this section have user interfaces that are used by people other than administrators. These applications require a web browser on each client machine and run in an instance of SAS Web Application Server that is installed on a middle-tier machine. These applications communicate with the user by sending data to and receiving data from the user's web browser. For example, these applications display a user interface by sending HTML that includes HTML forms, Java Applets, or Adobe Flash content. The user can interact and submit input to the application by sending an HTTP response, usually by clicking a link or submitting an HTML form.
SAS Web Report Studio

SAS Web Report Studio is a web application that anyone can use to view, interact with, create, and distribute public and private reports. Reports can be scheduled to run unattended on a recurring basis and then distributed using email. SAS Web Report Studio requires the SAS BI Report Services (which includes the report output generation tool) and the SAS BI Report Services Configuration (which creates libraries used by the SAS Web Report Studio).

SAS Information Delivery Portal

The SAS Information Delivery Portal is a web application that enables you to aggregate data from a variety of sources and present the data in a web browser. The web browser content might include the output of SAS Stored Processes, links to web addresses, documents, syndicated content from information providers, SAS Information Maps, SAS reports, and web applications. The portal also provides a secure environment for sharing information with users.

Using the portal, you can distribute different types of content and applications as appropriate to internal users, external customers, vendors, and partners. You can use the portal along with the Publishing Framework to perform the following tasks:

- Publish content to SAS publication channels or WebDAV repositories
- Subscribe to publication channels
- View packages published to channels

The portal’s personalization features enable users to organize information about their desktops in a way that makes sense to them.

For more information, see the SAS Information Delivery Portal Help, which is available from within the product.

SAS BI Dashboard

SAS BI Dashboard enables users to create, maintain, and view dashboards to monitor key performance indicators that convey how well an organization is performing. SAS BI Dashboard includes an easy-to-use, drag and drop interface for creating dashboards that include graphics, text, colors, and hyperlinks. The application leverages Flash in the Rich Internet Application (RIA) architecture.

The Dashboard Viewer enables users to complete the following tasks:

- Interact with data through interactive highlighting
- Quickly get to a subset of data through prompts and filters

Dashboards can link to the following:

- SAS reports and analytical results
- Scorecards and objects associated with solutions such as SAS Strategy Management
- Stored Processes
- Indicators
- Virtually any item that is addressable by a Uniform Resource Identifier (URI)
With the ability to save favorite dashboards and add comments, users can collaborate and easily access dashboards with customized information. All content is displayed in a role-based, secure, customizable, and extensible environment.

**SAS BI Portlets**

The SAS BI Portlets are based on JSR 168 and are available with SAS Enterprise Business Intelligence Server. These portlets are seamlessly integrated into the SAS Information Delivery Portal. SAS BI Portlets enable users to access, view, or work with content items that reside in either the SAS Metadata Server or the SAS Content Server.

**SAS Help Viewer for the Web**

Your installation can include the SAS Help Viewer for the Web. This application enables users to view and navigate SAS online Help in the various SAS web applications. This application combines the Help viewer with the Help content for various SAS web applications and creates a WAR file that is deployed on the web application server. Users access the Help contents for each application through the Help menu that is provided with each SAS web application.

The application also provides an administrative interface that is used to view the status of the documentation products. Administrators can use this interface to determine whether the documentation products were installed correctly, or whether there was a configuration problem. The administration interface is available from http://hostname.example.com/SASWebDoc.

**Starting the Web Applications**

To start the web applications, follow these steps:

1. Start the SAS servers and services in the correct order. For more information about the sequence, see “Overview of Server Operation” in SAS Intelligence Platform: System Administration Guide.

2. Start a browser session and point the browser to the web application that you want to access. For the correct URL, see the Instructions.html document, which resides in the Documents subdirectory of your configuration directory. The exact URL varies depending on the host name and port number that was defined for your environment.
## Middle-Tier Log Locations

The following table lists the log locations for the middle-tier web applications and servers:

**Table 1.2 Log Locations for Applications and Servers**

<table>
<thead>
<tr>
<th>Application or Server</th>
<th>Log Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Locator</td>
<td><code>SAS-configuration-directory\Lev\Web\gemfire\instances\ins_port-number\gemfire.log</code> file</td>
</tr>
<tr>
<td>JMS Broker</td>
<td><code>SAS-configuration-directory\Lev\Web\activemq\data\activemq.log</code> file</td>
</tr>
<tr>
<td>SAS Environment Manager Agent</td>
<td><code>SAS-configuration-directory\Lev\Web\SASEnvironmentManager\agent-version-EE\log</code> directory</td>
</tr>
<tr>
<td>SAS Environment Manager Server</td>
<td><code>SAS-configuration-directory\Lev\Web\SASEnvironmentManager\server-version-EE\logs</code> directory</td>
</tr>
<tr>
<td>SAS Web Application Server</td>
<td><code>SAS-configuration-directory\Lev\Web\WebAppServer\SASServernm\logs</code> directory</td>
</tr>
<tr>
<td>SAS web applications</td>
<td><code>SAS-configuration-directory\Lev\Web\Logs\SASServernm\directory</code></td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform Data Server</td>
<td><code>SAS-configuration-directory\Lev\Web\InfrastructurePlatformDataServer\Logs</code> directory</td>
</tr>
<tr>
<td>Note: In a multi-machine deployment, the default log location is on the server tier.</td>
<td></td>
</tr>
<tr>
<td>SAS Web Server</td>
<td><code>SAS-configuration-directory\Lev\Web\WebServer\logs</code> directory</td>
</tr>
</tbody>
</table>

For more information about SAS server logging, see “Administering Logging for SAS Servers” in *SAS Intelligence Platform: System Administration Guide*.

For additional information about specific web application logs, see *SAS Intelligence Platform: Web Application Administration Guide*.
# Interacting with the Server Tier

## Configuration Shared between the Middle Tier and the Server Tier

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTP Mail Server</td>
<td>16</td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform Data Server</td>
<td>17</td>
</tr>
<tr>
<td>- About the Server</td>
<td>17</td>
</tr>
<tr>
<td>- Installation Directory</td>
<td>17</td>
</tr>
<tr>
<td>- Databases</td>
<td>18</td>
</tr>
<tr>
<td>- Network Access</td>
<td>18</td>
</tr>
<tr>
<td>- Password Policy</td>
<td>19</td>
</tr>
<tr>
<td>- Administering Logging for the Server</td>
<td>19</td>
</tr>
<tr>
<td>- pgAdmin III Tool</td>
<td>19</td>
</tr>
<tr>
<td>- Creating a New Database</td>
<td>20</td>
</tr>
<tr>
<td>- Database Roles</td>
<td>20</td>
</tr>
<tr>
<td>- Deleting a Database</td>
<td>20</td>
</tr>
<tr>
<td>- Backing Up or Restoring a Database</td>
<td>20</td>
</tr>
<tr>
<td>JDBC Data Sources</td>
<td>21</td>
</tr>
<tr>
<td>- About the Data Sources Used by the Middle Tier</td>
<td>21</td>
</tr>
<tr>
<td>- Connection Information for the JDBC Data Source</td>
<td>21</td>
</tr>
<tr>
<td>Job Execution Service</td>
<td>22</td>
</tr>
</tbody>
</table>

### Configuration Shared between the Middle Tier and the Server Tier

The web applications and services that form the SAS middle tier require specific connections to servers that are associated with the server tier. You might want to modify the connections and settings in the following ways:

- Change the connection to an SMTP mail server.
- Understand the use of the SAS Web Infrastructure Platform Data Server.
- Modify the JDBC data source that provides a connection to a relational database.
- Modify the Job Execution Services settings.
SMTP Mail Server

The SAS Web Infrastructure Platform includes a SAS Mail Service that is used by SAS web applications and services to send email messages such as alert notifications and administrative status updates. The SAS Mail Service relies on a Java Mail Session that is defined in SAS Web Application Server. The Java Mail Session provides the single point of configuration to an external SMTP mail server that your site designates to use for application email. Because the SAS Mail Service relies on this single configuration location, if the SMTP mail server changes, you can modify the appropriate settings in a single place.

The Java Mail Session depends on configuration information that defines the mail transport capabilities. The SAS Mail Service requires that the following minimum set of mail properties be specified:

- `mail.transport.protocol`: This property must be set to `smtp`.
- `mail.smtp.host`: This property must be set to the host name of the SMTP mail server.
- `mail.smtp.port`: This property must be set to the corresponding port (typically 25 for SMTP servers).
- `mail.debug`: This property is set to `false`. You can set the value to `true` for assistance with debugging mail transactions.

In a standard installation of SAS middle-tier components, the configuration of the Java Mail Session is typically automated using prompted values that are provided by the installer. To modify the settings for the Java Mail Session (for example, if the host name of the SMTP mail server changes), edit the `server.xml` file. If you have more than one server instance, edit the `server.xml` file for each server. Change the following line:

```xml
<Resource auth="Container"
    mail.smtp.host="smtp.example.com"
    mail.smtp.port="25"
    name="sas/mail/Session"
    type="javax.mail.Session"/>
```

You can configure SMTP authentication by adding the following properties to the Resource definition that is shown above:

```
<Resource auth="Container"
    mail.smtp.host="smtp.example.com"
    mail.smtp.port="25"
    name="sas/mail/Session"
    type="javax.mail.Session"/>
```

You can configure SMTP authentication by adding the following properties to the `server.xml` file:

- `mail.smtp.auth`: true
- `mail.smtp.user`: username

### Table 2.1 SMTP Authentication Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.smtp.auth</td>
<td>true</td>
</tr>
<tr>
<td>mail.smtp.user</td>
<td>username</td>
</tr>
</tbody>
</table>
In addition, you can configure SSL by adding `mail.smtp.ssl.enable="true"` to the Resource definition.

If the mail server information, such as host name or port number, is changed, then it must be changed in SAS metadata as well. To set the new values, follow these steps:

1. Log on to SAS Management Console and select **Application Management ▶ Configuration Manager**.
2. Right-click **SAS Application Infrastructure** and select **Properties**.
3. Click **Advanced**, and then set the new values for **Email.Host** or **Email.Port**.

---

### SAS Web Infrastructure Platform Data Server

#### About the Server

SAS Web Infrastructure Platform Data Server is included in your deployment for use as transactional storage by SAS middle-tier software and some SAS solutions software. The server is based on PostgreSQL 9.1.9. The server is configured specifically to support SAS software. Some of the settings are provided in the next section.

The server is automatically configured by the SAS Deployment Wizard during installation and configuration. By default, the SAS installer account is used to start the server.

The databases that are managed by the server are backed up and restored with the Backup and Recovery Deployment Tool. For information about the tool, see *SAS Intelligence Platform: System Administration Guide*.

#### Installation Directory

The SAS Deployment Wizard installs and configures a server instance in the `SAS-configuration-directory \Lev1\WebInfrastructurePlatformDataServer` directory. This path includes the following script and directories:

- `webinfdsrvc.bat`
  This script is used to start, stop, and determine the running status for the server. It specifies the network port number and the path to the data directory. For UNIX deployments, the script is named `webinfdsrvc.sh` and is configured to start the server as the SAS installer account.
This directory contains server configuration files and the data files for the databases that are managed by the server. SAS configures the server to store data in the UTF-8 character encoding. Do not modify the files in this directory without direction from SAS Technical Support.

SAS configures the server to generate log files in this directory. Log files are rotated automatically after they reach 10 MB.

The _webinfdfsrvc_console.log file is generated during start-up. Look at this log first if you have trouble starting the server.

**Databases**

In a SAS 9.4 Enterprise Business Intelligence deployment, the server is configured to manage the following databases:

**Administration**

This database contains configuration information for the modules that SAS develops to extend the features of SAS Environment Manager.

**EVManager**

This database is used by SAS Environment Manager. The database contains configuration and metric information for the machines and servers that SAS Environment Manager manages in your deployment.

**SharedServices**

This database is used by the SAS web applications and middle-tier software. For example, comments that are added through various web applications are stored in this database. Digital content that is stored with SAS Content Server is also stored in this database.

**Note:** You can choose to use a third-party vendor database server for this database when you install and configure software with the SAS Deployment Wizard. This database is identified as the SAS Web Infrastructure Platform Database on the pages in the wizard.

**transportsvcs_db**

This database is used by SAS Visual Analytics Transport Service. The database stores mobile logon history information, as well as the device’s blacklist and whitelist data that is maintained through SAS Visual Analytics Administrator. It is also used to support caching within the Transport Service application.

If your deployment includes SAS solutions software that supports SAS Web Infrastructure Platform Data Server, then more databases might be configured on the server.

**Network Access**

The server is configured to accept connections on all network interfaces and requires password authentication. By default, SAS configures the server to use network port number 9432. This network port number avoids conflicts with the default port (5432) that other PostgreSQL servers might use.

SAS Web Application Server instances are configured with JDBC Data Sources that reference the SharedServices database and the Administration database.
SAS Environment Manager is configured for access to the EVManger and to the Administration database.

**Password Policy**

The user name and password for the SAS Web Infrastructure Platform Data Server administrator are specified during installation, using the SAS Deployment Wizard. The password can be updated using the SAS Deployment Manager. Passwords for the SAS Web Infrastructure Platform Data Server are subject to the following guidelines:

1. The password must be at least six characters long.
2. The password can contain a mix of alphanumeric and most special characters, except a single quotation mark or the @ symbol.
3. The password cannot include any trailing spaces.
   
   Note: Leading spaces, and blank spaces among the characters, are allowed.

There are no restrictions for including numbers or mixed case characters.

**Administering Logging for the Server**

To administer logging for SAS Web Infrastructure Platform Data Server, follow these steps:

1. Stop SAS Web Infrastructure Platform Data Server.
2. Edit the `SAS-configuration-directory\Lev1\WebInfrastructurePlatformDataServer\data\postgresql.conf` file to set or change logging parameters. For more information about PostgreSQL logging, see [http://www.postgresql.org/docs/manuals/](http://www.postgresql.org/docs/manuals/).
   
   Note: If more than one instance is defined in SAS Web Infrastructure Platform Data Server, you must change the logging parameters for each instance. Each instance has a separate `postgresql.conf` file.

For information about stopping and restarting the server, depending on your operating system, see “Methods for Operating Servers” in SAS Intelligence Platform: System Administration Guide.

**pgAdmin III Tool**

The pgAdmin III tool is a PostgreSQL database design and management system tool. The pgAdmin III tool provides a graphical user interface that is available on Windows systems and enables you to administer the SAS Web Infrastructure Platform Data Server.

For UNIX deployments, you can access the pgAdmin III tool from a Windows client machine. You must download and install the tool into your Windows client. You can download the pgAdmin III tool from [http://www.pgadmin.org/download/](http://www.pgadmin.org/download/).
Creating a New Database

When creating a SAS Web Infrastructure Platform Data Server database, you receive an UTF-8 encoded copy of the template database with the large object extension enabled. When you create the administrator user, the user will have all of the privileges that are granted to the database.

To create a database, run the following command:

```bash
createdb name
```

For more information about how to create a database using a shell program, see http://www.postgresql.org/docs/manuals/. Navigate to PostgreSQL Client Applications ➔ createdb.

Database Roles

There are usually two roles that are created when a database is created. The first is a login role (the administrator user), which is usually specified during installation, using the SAS Deployment Wizard. When the database is deleted, this role should also be deleted. The second role that is created is a group role named `database name_admin`. This role should also be deleted when the database is deleted.

To delete a database role, run the following command:

```bash
dropuser name
```

For more information about how to delete a role using a shell program, see http://www.postgresql.org/docs/manuals/. Navigate to PostgreSQL Client Applications ➔ dropuser.

Deleting a Database

You should delete a database when the following conditions occur:

- You are instructed to do so during configuration.
- The database is no longer needed after removing a SAS product's configuration.

To delete a database, run the following command:

```bash
dropdb name
```

For more information about how to delete a database using a shell program, see http://www.postgresql.org/docs/manuals/. Navigate to PostgreSQL Client Applications ➔ dropdb.

Backing Up or Restoring a Database

You can use the Deployment Backup and Recovery Tool to back up and restore your SAS Web Infrastructure Platform Data Server database. For more
JDBC Data Sources

About the Data Sources Used by the Middle Tier

The SAS Web Infrastructure Platform and some solutions provide a set of features that rely on a relational database to store service data. These relational tables differ from the data that is analyzed, modeled, or otherwise processed by SAS applications, which typically is derived from a site's enterprise or legacy sources. Instead, the relational tables in the SAS Web Infrastructure Platform database are intrinsic to or used primarily for the operations of a particular application, product, or service.

SAS web applications and services access data from the SAS Web Infrastructure Platform database through JDBC. SAS Web Infrastructure Platform provides support for the following third-party vendor databases:

- Oracle Database
- IBM DB2
- Microsoft SQL Server
- MySQL
- PostgreSQL

If you have not already done so, make sure that you review “Configuring an Alternate Database for SAS Web Infrastructure Platform Services” in SAS Intelligence Platform: Installation and Configuration Guide.

Your site can choose to use the database that you are familiar with. However, some SAS solutions have requirements for specific databases. Consider these requirements when you select a database to use as the data source for the SAS Web Infrastructure Platform. As a default option, the SAS Web Infrastructure Platform Data Server can be configured as the data source for SAS Web Infrastructure Platform.

Connection Information for the JDBC Data Source

The database used by the SAS Web Infrastructure Platform must be configured in SAS Web Application Server as a JDBC data source. The JDBC data source is configured with the JDBC driver and connection information for the selected database. These settings are provided to the SAS Deployment Wizard during installation and configuration. You need to know the JDBC connection parameters if you make changes later, such as changing the connection to access a database on another machine. JDBC connection settings typically require a user ID and password for access to the data source.
The default database server for SAS Web Infrastructure Platform is the SAS Web Infrastructure Platform Data Server. The JDBC connection parameters for the server are provided in the following table:

<table>
<thead>
<tr>
<th>Connection Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI name:</td>
<td>sas/jdbc/SharedServices</td>
</tr>
<tr>
<td>JDBC URL:</td>
<td>jdbc:postgresql://serverName:port/SharedServices</td>
</tr>
<tr>
<td></td>
<td>In the URL, substitute the server name and port number of the SAS Web Infrastructure Platform Data Server at your site. The default port is 9432.</td>
</tr>
<tr>
<td>JDBC driver class:</td>
<td>org.postgresql.Driver</td>
</tr>
</tbody>
</table>

These settings are configured during initial deployment. However, you need to know the connection information if you make changes later, such as moving the server to another host system.

**Note:** You must specify the user name and password values as required to access the data source.

These settings are represented in SAS Web Application Server in the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer1_1\conf\server.xml` file:

```xml
```

The `postgresql.jar` JAR file provides the `org.postgresql.Driver` class. SAS provides the JAR file in the `SASHOME\SASWebInfrastructureDataBaseJDBCDrivers\9.4\Driver` directory.

### Job Execution Service

The service provides a common, standardized way for applications to create, submit, store, retrieve, and queue jobs for SAS servers. The service can be configured with the Configuration Manager plug-in to SAS Management
Console. The settings define the job thread pool and the execution thread pools for all logical servers that the service uses for delegating work.

**Figure 2.1  Job Execution Service Settings**

![Job Execution Service Settings](image)

**Table 2.3  Job Execution Service Settings Descriptions**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Queue Minimum Threads</td>
<td>5</td>
<td>Minimum number of job queue threads to create for incoming job requests.</td>
</tr>
<tr>
<td>Job Queue Maximum Threads</td>
<td>30</td>
<td>Maximum number of job queue threads to create if the demand requires additional resources.</td>
</tr>
<tr>
<td>Enable role-based security</td>
<td>Disabled</td>
<td>If enabled, then the Job Execution Service checks the identity and the job characteristics to make sure the identity making the request meets the assigned permissions. For more information, see Table 2.4 on page 25.</td>
</tr>
<tr>
<td>Setting</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Enable job persistence              | Enabled       | Jobs are kept in memory only if persistence is disabled. If persistence is disabled and the SAS Web Infrastructure Platform Services application or the web application server is stopped, then no records are written to the SAS Web Infrastructure Platform database about any jobs that were submitted. When persistence is enabled, the job execution services can restart any jobs that were submitted, queued, or running. For jobs that are complete, clients can fetch the results after a restart, when persistence is enabled.  
Note: Persistence must be enabled when SAS Web Application Server is clustered. |
| Enable Distributed-IP Scheduler job runner | Disabled     | If enabled, then the distributed in-process scheduler is used for running scheduled jobs. Disable this setting if Platform Suite for SAS is available and the preferred scheduling method. |
| Available Server Contexts           | SASApp        | Use the controls to select the server context to configure.                                                                                   |
| Enable for interactive execution    | Disabled      | If enabled, then the servers in the associated server context perform interactive workspace tasks and interactive stored process tasks only. If disabled, then the servers can perform batch and interactive job execution. |
| Server Minimum Threads              | 1             | Minimum number of task threads to create for incoming job requests.                                                                            |
| Server Maximum Threads              | varies        | Maximum number of task threads to create if the demand requires additional resources.                                                          |
| Server Resources                    |               | You can associate resources with servers and then a job can specify that it requires a resource. For example, you can associate a printer name with SASApp. When a client submits a job, and specifies that it requires the printer resource, the job execution service makes sure that the job runs on that server even when other servers are available. |

The default settings are designed to provide good performance in a variety of operating environments. Before modifying the settings, consider enabling the auditing features of the job execution services to review the performance with the default settings. For information about enabling auditing, see “Configuring Auditing for SAS Web Applications” on page 67.

To modify any of these settings, follow these steps:

1. Log on to SAS Management Console as an administrator.
2 On the Plug-ins tab, navigate to Application Management ➤ Configuration Manager ➤ SAS Application Infrastructure ➤ Web Infra Platform Services 9.4.

3 Right-click JobExecutionService and select Properties.

4 Click the Settings tab.

5 Modify the settings and then click OK.

When a new server context is configured for use by the Job Execution Service, the Configuration Manager notifies the Job Execution Service instances to reload their configurations to add the new server context. The following settings are updated at run time by the Job Execution Service:

- A new logical server that is configured to be used by the Job Execution Service
- The following job execution queues:
  - minimum thread pool size
  - maximum thread pool size
  - algorithm

All other settings are not applied and made active automatically. They are activated as follows:

- When you restart the SAS Web Infrastructure Platform Services or SAS Web Application Server
- When you can set the state of some properties at run time through the JMX bean (MBean) for the service with a JMX console
- When you click the Reconfigure button in SAS Web Administration Console.

For more information, see "Updating the Job Execution Service Configuration" on page 78.

The default configuration for the job execution services does not check role-based permissions. If role-based security is enabled, then the job execution service checks that the identity submitting the request has sufficient permission.

Table 2.4  Job Execution Service Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Execution: Job Administrator</td>
<td>Can submit jobs of high, normal, and low priority and perform all job-related operations.</td>
</tr>
<tr>
<td>Job Execution: Job Designer</td>
<td>Can add, update, or remove jobs and tasks from metadata.</td>
</tr>
<tr>
<td>Job Execution: Job Scheduler</td>
<td>Can schedule jobs.</td>
</tr>
<tr>
<td>Job Execution: Job Submitter</td>
<td>Can submit normal priority jobs for execution.</td>
</tr>
</tbody>
</table>
The following figure shows the default capabilities associated with the job administrator role.

**Figure 2.2  Job Administrator Capabilities**
Part 2

Middle-Tier Components

Chapter 3
Administering SAS Web Server .................................................. 29

Chapter 4
Administering SAS Web Application Server ................................. 35

Chapter 5
Administering Cache Locator ....................................................... 43

Chapter 6
Administering JMS Broker .......................................................... 49
Administering SAS Web Server

About SAS Web Server

SAS Web Server is an HTTP server. The server is based on Pivotal Web Server. SAS configures the server with the following features:

- automatically configured as a load-balancing HTTP server when SAS Web Application Server is clustered.
- automatically updated to route web sessions (round robin) to SAS Web Application Server instances when clustered.
- can be configured for HTTPS automatically. You must supply a signed certificate and a private key. You can follow manual steps to change a configuration that used HTTP to HTTPS.
- automatically configured to cache static web content like JavaScript files, cascading style sheets, and graphics files.

The following advanced configurations are possible, but require manual configuration that is not automatically updated:

- adding instances of SAS Web Server to form a cluster
- interacting with customer-supplied load-balancing hardware or software
Installing SAS Web Server

Automatic Configuration

SAS Web Server is installed with SAS Deployment Wizard. The wizard can also automatically configure the server. By default, the server is installed on the same machine as SAS Web Application Server. However, because the topology is defined in a plan file that the wizard uses, the server can be deployed to a different machine if the topology is defined that way in the plan file.

To use this feature, select the **Configure SAS Web Server automatically** check box on the SAS Web Server: Automated or Manual Configuration Option page of SAS Deployment Wizard.

Manual Configuration

If you prefer to configure SAS Web Server manually, make sure the **Configure SAS Web Server automatically** check box is not selected when you use SAS Deployment Wizard. Once the wizard completes, the Instructions.html file provides step-by-step instructions that describe how to configure the server manually. The instructions are customized for your deployment, including the correct host names and file system paths.

If you choose to configure the server manually, you must also configure SAS Web Application Server manually.

Using HTTPS

If you plan to use HTTPS, then it is best to enable the feature during the installation and configuration time frame with SAS Deployment Wizard. SAS Deployment Wizard prompts for a CA-signed certificate and private key. Both must be in PEM encoded format.

If you have a CA-signed certificate, SAS Deployment Wizard prompts for the path to the certificate and the path to the RSA private key that is not protected with a pass phrase. An RSA private key file that is not protected with a pass phrase begins as follows:

**Example Code 3.1 RSA Private Key without a Pass phrase**

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQC4vFOQyvIKvj1ERVNfa34iVxeauzcUa8zc2xHR1J43uAvvWuL
63yeG18QQoT55yqhAWhs62i241E34t21tuhCm0QYbUIKiy9B9Pyf0k3/2E7Y7o1T
-----END RSA PRIVATE KEY-----
```

Do not use an encrypted private key. An encrypted RSA private key file begins as follows:

**Example Code 3.2 Encrypted RSA Private Key**

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,FB353F3E4F1719EB
LigQnszN4ji024OnLHCGBd44L1La6uMEqdxh11PX804o+PBY5cEQQJBBciRlEmf
Io5V/YUu+uGG82ULsAUy3zWTHP+CjxpTV/3gjLwbbD3+JMsD0jFLgenfP5hld
-----END RSA PRIVATE KEY-----
```
SAS Deployment Wizard also prompts for the certificate. A certificate file from a certificate authority typically begins as follows:

**Example Code 3.3 Certificate Authority-Signed Certificate**

```plaintext
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption

...-----BEGIN CERTIFICATE-----
MIIDhDCCAu2gAwIBAgIBATANBgkqhkiG9w0BAQQFADB+MQswCQYDVQQGEwJVUzEL
MAkGA1UECBMCTkMxDTALBgNVBAcTBENhcnkxDDAKBgNVBAoTA1NBUzENMAsGA1UE
...-----END CERTIFICATE-----
```

---

**Understanding the SAS Web Server Configuration**

The default location for SAS Web Server is `SAS-configuration-directory\Lev\WebServer`. Key files and directories are as follows:

**bin**

This directory includes a command for starting and stopping the server. For more information, see "Using the httpdctl Command".

**conf**

SAS software manages the configuration files in this directory. If you modify a file, your customizations are overwritten the next time SAS software configures the server.

Do not modify configuration files manually. Many settings, such as network port number, are managed in SAS metadata as well. Use SAS Deployment Manager and SAS Deployment Wizard for configuring SAS Web Server.

**ssl**

If you enabled HTTPS during installation and configuration with SAS Deployment Wizard, then this directory is used to store the certificate and private key for the server. If you supplied a CA-signed certificate and private key to the wizard, both files are copied to this directory. The files are also renamed to include the host name, as follows:

`hostname.crt`

`hostname.key`

**TIP** If you need to replace a certificate—for example, to avoid having a certificate expire—then replace the file in this directory.
Understanding SAS Web Server Management

Using the httpdctl Command

The server is configured with a \httpdctl.ps1 command in the bin directory. On UNIX, the command is httpdctl.

**UNIX Specifics:** If you configured SAS Web Server to use network port numbers below 1024, then you must run the \httpdctl command with super user privileges, such as sudo.

```bash
sudo ./httpdctl restart
```

**Windows Specifics:** The \httpdctl.ps1 is a Windows PowerShell script. You might need to set the execution policy with `powershell set-executionpolicy remotesigned`.

```bash
powershell .\httpdctl.ps1 restart
```

Using the appsrvconfig Command

A configuration scripting tool for SAS Web Server is located in the `SAS-configuration-directory\Lev\Web\Scripts\WebServer` directory. The `appsrvconfig.cmd` command can be used for starting, stopping, and restarting SAS Web Server.

```cmd
appsrvconfig.cmd start
appsrvconfig.cmd stop
appsrvconfig.cmd restart
```

The actual task is identified in a command task file that is located in the `SAS-configuration-directory\Lev\Web\Scripts\WebServer\props`. The file is generated and then executed. The file does not exist until the `appsrvconfig.cmd` command is used.

Information about using the `appsrvconfig.cmd` command for configuration tasks is provided in SAS Configuration Scripting Tools on page 287.

Using Windows Services

For deployments that use the Windows operating environment, the default action for SAS Deployment Wizard is to register each server instance as a service. The naming convention is similar to the following example:

```
SAS [Config-Lev1] httpd - WebServer
```

Using SAS Environment Manager

SAS Environment Manager provides an interface that you can access with a web browser. You can start and stop SAS Web Server with the web interface.
Monitoring SAS Web Server

Viewing Performance With SAS Environment Manager

The primary user interface for monitoring the server is SAS Environment Manager. Numerous metrics are collected from the server.

In SAS Environment Manager, SAS Web Server is represented as vFabric Web Server version Virtual Host.

For administrators that are familiar with monitoring Apache HTTP Server, the metrics that are collected for vFabric Web Server version Virtual Host are related to mod_bmx.

See Also
SAS Environment Manager: User's Guide

Viewing Load-Balancing Statistics

SAS Web Server is configured to load-balance requests, even if only one SAS Web Application Server instance is configured. You can access the information by opening a web browser from the machine that is hosting SAS Web Server and accessing the following URL:

http://localhost/balancer-manager

The web page provides information about each load balancer. Some of the information is identified in the following list:

- routes (each instance of SAS Web Application Server is identified as a route)
- route status
- the amount of network traffic to and from each route
- stop requests being sent to a specific cluster member
About SAS Web Application Server

SAS Web Application Server is a lightweight server that provides enterprise-class features for running SAS web applications. The server is based on Pivotal tc Server. By packaging the server and software that can automate server configuration tasks, SAS simplifies the demands for managing a web application server.

Though the server is based on a commercially available third-party software product, the server is deployed and configured specifically to provide an environment for the SAS web application and the middle-tier environment. The configuration tools that are packaged with the software ease the administration of the server in a SAS environment because the tools are designed to interact
with the SAS Metadata Server and other SAS software products to maintain reliability and reduce administration in the SAS deployment.

The following list identifies some enhancements that are implemented in SAS Web Application Server:

- automatically connects to Cache Locator on server start-up for distributed communication.
- accesses the JMS resources provided by JMS Broker.
- automatic directory scanning for changes to files is disabled. This change conserves computing resources.
- JAR file scanning is optimized to reduce start-up times.

### Installing SAS Web Application Server

#### Automatic Configuration

By default, SAS Web Application Server is installed by the SAS Deployment Wizard when you install SAS software for your deployment. The SAS Deployment Wizard can automatically configure a server instance, deploy the web applications, and also automatically configure related middle-tier components such as SAS Web Server, JMS Broker, and Cache Locator.

To use this feature, select the **Configure the web application server automatically** check box on the Web Application Server: Automatic Configuration page of the SAS Deployment Wizard.

#### Manual Configuration

If you prefer to configure SAS Web Application Server manually, make sure the **Configure the web application server automatically** check box is not selected when you use the SAS Deployment Wizard. Once the wizard completes, the Instructions.html file provides step-by-step instructions for how to configure the server manually. The instructions are customized for your deployment, including the correct host names and file system paths.

The generated Instructions.html file also includes information about installing and configuring the related middle-tier components: SAS Web Server, JMS Broker, and Cache Locator.

#### Multiple Machine Installation

You can install and configure SAS Web Application Server on multiple machines to provide better performance, scalability, and high availability. This is called horizontal clustering.

You can have the SAS Deployment Wizard automatically configure the additional instances, or configure them manually. For more information, see "Adding a Horizontal Cluster Member" on page 193.
Understanding SAS Web Application Server Configuration

Server Naming

The default name for the first server instance is SASServer1_1.

The server name and instance is broken down as follows:

SASServer1
   This portion identifies the server name.

   _1
   This portion identifies the first instance of the server. Additional instances of
   this server (for vertical clustering) increment the number as in _2, _3, and so
   on.

Your deployment might include additional managed servers. If your deployment
includes a SAS solution, the web applications related to the solution might be
deployed to managed servers with names like SASServer8_1 or
SASServer12_1.

Your deployment might include SASServer2_1. This server instance is created
when the SAS Deployment Wizard is used at the custom prompting level and
enabling the multiple managed server option. This option is useful for distributing
some of the web applications to the SASServer2_1 instance.

If you have configured multiple instances of a managed server, such as
SASServer1_1 and SASServer1_2, then the web applications that support
clustering are deployed identically to each instance. Each of these instances is a
vertical cluster member. For applications that do not support clustering, only one
instance is configured on the first server instance.

See Also
“Adding a Vertical Cluster Member” on page 192

Server Directories

Configured instances of SAS Web Application Server are stored in the SAS-
configuration-directory\Lev\Web\WebAppServer directory and
subdirectories.

SAS-configuration-directory\Lev\Web\WebAppServer\SASServer1_1
   This directory represents an instance of SAS Web Application Server.
   Information about some of the subdirectories is as follows:

    bin
    This directory includes a command for starting and stopping the server.
    More information about controlling the server is described in
    “Understanding SAS Web Application Server Management”.

    conf
    SAS software manages the configuration files in this directory. If you
    modify a file, your customizations are overwritten the next time SAS
    software configures the server.
This directory is used for the SAS web applications. SAS software manages the addition and removal of web applications from the directory.

Specifying JVM Options

For some advanced configuration procedures, you might need to change JVM options for the server.

For Windows deployments, the JVM options are specified in the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\conf\wrapper.conf` file and the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\bin\setenv.bat` file. If you have multiple instances of SAS Web Application Server, make the same changes in each of the files.

For UNIX deployments, JVM options are specified in the `SAS-configuration-directory/Lev/\Web/WebAppServer/SASServer_m/bin/setenv.sh` file. If you have multiple server instances, make the changes in each setenv.sh file.

Deploying Web Applications

During the installation and configuration that is performed with the SAS Deployment Wizard, the SAS web applications are automatically deployed if SAS Web Application Server is automatically configured.

See Also

For information about redeploying, see “Redeploy Web Applications” on page 91.

Understanding SAS Web Application Server Management

Using the tcruntime-ctl Command

Each server instance provides a `tcruntime-ctl.cmd` command in the `bin` directory. The command is `tcruntime-ctl.sh` on UNIX.

If you use this command to start, stop, or restart a server instance, be aware that it affects only the single server instance. The command does not start or stop any middle-tier components that the server depends on. The command syntax is as follows:

```
tcruntime-ctl.cmd start
  tcruntime-ctl.cmd stop
  tcruntime-ctl.cmd restart
```

Note: On Windows, the `status` option does not indicate whether the server is running or stopped.
Using the appsrvconfig Command

Each machine that is used to run SAS Web Application Server for the SAS middle-tier includes the SAS Configuration Scripting Tools in the `SAS-configuration-directory\Lev\Web\Scripts\AppServer` directory. The `appsrvconfig.cmd` command can be used for starting, stopping, and restarting all SAS Web Application Server instances on the machine as well as any middle-tier components that the server depends on.

For example, the command `appsrvconfig.cmd restart` automatically performs the following tasks:

1. Stops all SAS Web Application Server instances
2. Stops JMS Broker
3. Stops Cache Locator
4. Starts Cache Locator
5. Starts JMS Broker
6. Starts all SAS Web Application Server instances

The actual tasks are identified in a command task file that is located in the `SAS-configuration-directory\Lev\Web\Scripts\AppServer\props`. The file is generated and then executed. The file does not exist until the `appsrvconfig.cmd` command is used.

Information about using the `appsrvconfig.cmd` command for configuration tasks is provided in `SAS Configuration Scripting Tools on page 287`.

Using Windows Services

For deployments that use the Windows operating environment, the default action for the SAS Deployment Wizard is to register each server instance as a service. The naming convention is similar to the following example:

```
SAS [Config-Lev1] WebAppServer SASServer1_1
```

The Windows service has the advantage of providing the server status (started or stopped), which is not available with the `tcruntime-ctl.bat` command line tool. In addition, the Windows service manages the service dependencies.

Using SAS Environment Manager

SAS Environment Manager provides an interface that you can access with a web browser. You can start and stop SAS Environment Manager with the web interface. When you start a server instance with SAS Environment Manager, the application indicates that the server started successfully before the server actually completes starting.

The command-line interface (`tcsadmin`) that is available with SAS Environment Manager can be used for inventory and control operations. Do not use it for application management or configuring instances and groups because you can create inconsistencies with the deployment software developed by SAS.

**See Also**

`SAS Environment Manager: User's Guide`
Monitoring SAS Web Application Server

The SAS 9.4 release introduces SAS Environment Manager. A SAS Environment Manager Agent is installed on the same machine as SAS Web Application Server and reports metrics to SAS Environment Manager.

You can access SAS Environment Manager from a URL that is similar to the following example:

http://hostname.example.com:7080

Note: The server portion of SAS Environment Manager runs in its own instance of a web application server. However, SAS Environment Manager is configured to use SAS Logon Manager for authentication, and this requires that SAS Web Application Server is running before you can access SAS Environment Manager.

See Also

SAS Environment Manager: User's Guide

Checking Prerequisite Servers

About Checking Prerequisite Servers

Beginning with the second maintenance release for SAS 9.4, a LifeCycle Listener is provided with SAS Web Application Server that can force the server to wait for prerequisite servers to start and begin listening on their service ports. In order to start properly, many SAS web applications must connect to other servers during their initialization process. If these prerequisite servers are not running, failures might occur during application initialization. The prerequisite servers include the following:

- SAS Web Server, if it is configured
- database servers, including SAS Web Infrastructure Platform Data Server and third-party data servers
- SAS Metadata Server
- SAS Cache Locator, only for the server where SAS Web Infrastructure Platform is deployed
- SAS JMS Broker

When one or more of the SAS servers are restarted at the same time, it is recommended that you let these prerequisite servers start before SAS Web Application Server instances. When SAS servers are configured to automatically start as Windows Services, this is the recommended process.

By default, the LifeCycle Listener is not enabled in the current release. In order to use this feature, you must configure it manually.
Enabling the Prerequisite Checker

To enable the LifeCycle Listener feature, edit the `SAS-configuration-directory\Levr\Web\WebAppServer\SASServern_m\conf\server.xml` file for each instance of SAS Web Application Server. Locate the `Server` element and add the highlighted line to the top of the file, along with the other `Listener` directives, for example:

```xml
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>
<Listener className="com.sas.vfabriccsvr.atomikos.AtomikosLifecycleListener"/>
<Listener className="com.sas.vfabriccsvr.startup.PrerequisiteServerListener"/>
```

For information about configuring the LifeCycle Listener for clustered servers, see “Configuring the Prerequisite Checker for Clustered Servers” on page 198.
About the Cache Locator

The Cache Locator is based on VMware vFabric GemFire. The software is used by applications on server-tier and middle-tier machines to locate other members and form a data cache. When SAS Web Application Server starts, it contacts one of the locators that is specified in the `sas.cache.locators` JVM option to initialize communication with the distributed cache. With that information, SAS Web Application Server instances form the cache that is needed to share run-time information.

A locator is also configured on the server tier to provide access to the data cache for stand-alone client applications like the SAS Web Infrastructure Platform Scheduling Services (wipschedbatch.bat).

Installing Cache Locator

Single Machine Deployments

In a single-machine deployment where the middle tier and the server tier are on the same machine, only one locator is installed by the SAS Deployment Wizard.

SAS Web Application Server uses the locator. If more than one instance of SAS Web Application Server is configured, each instance uses the locator to learn about the other server instances to form the cache.
Multiple Machine Deployments

A locator is installed on the first middle-tier machine by the SAS Deployment Wizard. A locator is also installed on each server-tier machine that includes SAS Web Infrastructure Platform Scheduling Services.

Understanding the Cache Locator Configuration

The default location for the cache locator is `SAS-configuration-directory\Lev\Web\gemfire\instances\ins_port-number`. Key files and directories are as follows:

- **gemfire-locator.sh**
  This script exists on UNIX deployments only. It can be used with one of the following arguments: `start`, `stop`, or `status`.

- **gemfire-start-locator-sas.sh**
  This script exists on UNIX deployments only. Use this file to specify JVM options for UNIX deployments.

- **gemfire-locator-zos.jcl**
  This script exists on z/OS deployments only when the locator is installed on the server tier. Use this file to specify JVM options for z/OS deployments.

- **gemfire.log**
  This is the log file for the cache locator. Be aware that it is different from the log file with the same name that is written to `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer1_1\logs`.

- **wrapper.conf**
  This file exists on Windows deployments only. It is used when you operate the SAS [Config-Lev1] Cache Locator service. Use this file to specify JVM options for Windows deployments.

Setting the Bind Address

When the locator is deployed on a machine that has more than one network interface, one network interface is used by default. In some cases, the network interface that is selected as the default is not the network interface that you want the locator to use.

You specify the network bind address to use for network traffic, add the `-Dgemfire.bind-address=preferred-ip-address` JVM option. For information about how to specify the options, see “Understanding the Cache Locator Configuration” on page 44.

If SAS Web Application Server is deployed on the same machine, specify the same `-Dgemfire.bind-address=preferred-ip-address` JVM option. For more information, see Specifying SAS Web Application Server JVM Options on page 38.
Modifying the Configuration to Accommodate a Firewall

About Cache Locator Port Requirements

During installation, one HTTP port number is reserved for each instance of the Cache Locator. In addition, TCP and UDP ports are needed to support peer-to-peer communication among the Cache Locator and each member component that uses the cache. The Cache Locator and cache members dynamically allocate these ephemeral ports as needed from ports that are available in the environment.

If a firewall exists among any of your server tier and middle tier machines, issues could occur in the peer-to-peer communication. To prevent these issues, you can modify the firewall configuration to permit traffic to Java applications. Alternatively, you can manually update the SAS configuration to ensure that ports in the appropriate range are available through the firewall.

Updating the JVM Options to Accommodate a Firewall

In each member’s configuration, update the JVM options to specify a set of port numbers that are available on the machine. From the specified port numbers, unique ephemeral ports will be dynamically allocated to members as needed. The following properties must be specified:

- **Dgemfire.membership-port-range**
  - specifies the range of ports that are available for UDP and TCP communication. For each member, the Cache Locator randomly chooses one unique integer from the range for UDP unicast messaging and another unique integer for TCP failure detection messaging. The combined host IP address and UDP port number uniquely identifies the member. Be sure to allocate a large enough range to accommodate your deployment. See “Determining the Number of Required Ports” on page 47. The default range is 1024-65535.

- **Dgemfire.tcp-port**
  - specifies a value between 0 and 65535 that represents the TCP listening port for a member’s cache communications. If the value is zero, the operating system selects an available port. Each process on a machine must have its own TCP port.

  **Note:** Some operating systems restrict the range of ports that can be used by non-privileged users, and using restricted port numbers can cause Cache Locator start-up errors.

Specify the preceding properties in each installed instance of the member components, as described in the following tables. Replace the highlighted values with appropriate values for your deployment. Except where indicated, all paths are within `SAS-configuration-directory/Levnn`.
### Table 5.1 Windows Updates to Accommodate a Firewall

<table>
<thead>
<tr>
<th>Component</th>
<th>File Location and Property Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Locator</td>
<td>In \Web\gemfire\instances\ins_port\wrapper.conf, add the following lines: wrapper.java.additional.7=-Dgemfire.membership-port-range=40000-50000 wrapper.java.additional.8=-Dgemfire.tcp-port=45500</td>
</tr>
<tr>
<td>SAS Web Application Server</td>
<td>In \Web\WebAppServer\SASServern_m\conf\wrapper.conf, add the following lines: wrapper.java.additional.48=-Dgemfire.membership-port-range=40000-50000 wrapper.java.additional.49=-Dgemfire.tcp-port=45600</td>
</tr>
<tr>
<td></td>
<td>In \Web\WebAppServer\SASServern_m\bin\setenv.bat, add JVM options as follows: set JVM_OPTS=&quot;... -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45600&quot;</td>
</tr>
<tr>
<td>SAS Distributed In-Process Scheduler Job Runner</td>
<td>In \Web\Applications\SASWIPSchedulingServices9.4\dip\wrapper.conf, add JVM options as follows: wrapper.java.additional.5=-Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45700</td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform Scheduling Services</td>
<td>In \Web\Applications\SASWIPSchedulingServices9.4\serverTrigger.ini, add the following lines: JavaArgs_16=-Dgemfire.membership-port-range=40000-50000 JavaArgs_17=-Dgemfire.tcp-port=45800</td>
</tr>
<tr>
<td>Report Output Generation Tool (for SAS Web Report Studio)</td>
<td>In \SAS-installation-directory\SASBIReportServices\4.4\outputgen.ini, add the following lines: JavaArgs_16=-Dgemfire.membership-port-range=40000-50000 JavaArgs_17=-Dgemfire.tcp-port=45900</td>
</tr>
</tbody>
</table>

### Table 5.2 UNIX Updates to Accommodate a Firewall

<table>
<thead>
<tr>
<th>Component</th>
<th>File Location and Property Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Locator</td>
<td>In /Web/gemfire/instances/ins_port/gemfire-start-locator-sas.sh, update the Java arguments as follows: JAVA_ARGS=&quot;... -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45500&quot;</td>
</tr>
<tr>
<td>SAS Web Application Server</td>
<td>In /Web/WebAppServer/SASServern_m/bin/setenv.sh, update the Java arguments as follows: JVM_OPTS=&quot;... -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45600&quot;</td>
</tr>
<tr>
<td>SAS Distributed In-Process Scheduler Job Runner</td>
<td>In /Web/Applications/SASWIPSchedulingServices9.4/dip/DIPJobRunner.sh, add the following line in the section JVM DIP Jobrunner specific VM properties: PROPS=&quot;$PROPS -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45700&quot;</td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform Scheduling Services</td>
<td>In /Web/Applications/SASWIPSchedulingServices9.4/serviceTrigger.ini, add the following line: JavaArgs_16=-Dgemfire.membership-port-range=40000-50000 JavaArgs_17=-Dgemfire.tcp-port=45800</td>
</tr>
</tbody>
</table>

---

46 Chapter 5 / Administering Cache Locator
In **SAS-installation-directory/SASBIReportServices/4.4/outputgen.ini**, add the following lines:

```
JavaArgs_16=-Dgemfire.membership-port-range=40000-50000
JavaArgs_17=-Dgemfire.tcp-port=45900
```

## Determining the Number of Required Ports

An adequate number of ports must be allocated to accommodate peer-to-peer communication among all of the cache members. To calculate the minimum number of ports to allocate per member, multiply the total number of members in the deployment by 4. To calculate the number of ports that are needed on each machine, multiply the number of installed members by the number of ports per member, as shown in the following table:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Members in the Deployment</td>
<td></td>
<td>Ports per Member (Column A * 4)</td>
<td>Member(s) per Machine</td>
<td></td>
<td>Total Ports (Column C * Column D)</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---</td>
<td>-------------</td>
<td>-----------</td>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td>1 Web application server instance</td>
<td>5</td>
<td>Server tier</td>
<td>20</td>
<td>3</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle tier</td>
<td>20</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>3 Web application server instances</td>
<td>7</td>
<td>Server tier</td>
<td>28</td>
<td>3</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle tier</td>
<td>28</td>
<td>4</td>
<td>112</td>
</tr>
<tr>
<td>5 Web application server instances</td>
<td>9</td>
<td>Server tier</td>
<td>36</td>
<td>3</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle tier</td>
<td>36</td>
<td>6</td>
<td>216</td>
</tr>
</tbody>
</table>

* In these scenarios, the middle tier machine contains one instance of the Cache Locator and one or more instances of SAS Web Application Server. The server tier machine contains one instance each of the Cache Locator, SAS Distributed In-Process Scheduler Job Runner, and SAS Web Infrastructure Platform Scheduling Services.
About JMS Broker

JMS Broker is based on Apache ActiveMQ. The broker is installed and configured with the SAS Deployment Wizard. By default, the broker listens on network port number 61616.

SAS middle-tier software uses the broker for Java Messaging Services (JMS). Some SAS web applications use JMS connection factories, queues, and topics for implementing business logic. These resources are configured in SAS Web Application Server for use by the SAS web applications.

Installing JMS Broker

The broker is installed and configured with the SAS Deployment Wizard. If you perform an automatic configuration of SAS Web Application Server, then the broker is automatically installed and configured. If you prefer to perform a manual configuration of SAS Web Application Server, then you must install and configure the broker. The step-by-step instructions are provided in the Instructions.html file that is generated by the SAS Deployment Wizard.

An instance of the broker is installed on the first machine that is used for the SAS middle tier. If you use the SAS Deployment Wizard to configure an additional middle-tier node on another machine, then those server instances are configured with connection information for the broker.
Understanding the JMS Broker

Configuration

The default location for the broker is \SAS-configuration-directory\Levn
\Web\activemq. Key files and directories are as follows:

- **bin**
  - On UNIX deployments, the activemq command is included in this directory. You can use the `start`, `stop`, `restart`, or `status` options with the command.
  - On Windows deployments, use the service that is registered with Windows to manage the broker. The `activemq.bat` command is not configured for use with SAS software.

- **data**
  - The activemq.log file is written in this directory.

Monitoring JMS Broker

The primary user interface for monitoring the server is SAS Environment Manager. Numerous metrics are collected from the broker.

In SAS Environment Manager, the broker is represented as `host-name ActiveMQ 5.7`. Statistics for the broker itself as well as the queues and topics are also gathered.
Part 3

Middle-Tier Applications

Chapter 7
  Administering the SAS Web Infrastructure Platform ............................................. 53

Chapter 8
  Administering SAS Web Applications ................................................................. 85

Chapter 9
  Administering SAS Logon Manager ................................................................. 101

Chapter 10
  Administering the SAS Content Server .......................................................... 117

Chapter 11
  Administering the SAS BI Web Services ......................................................... 139

Chapter 12
  Administering SAS Web Application Themes .................................................. 147

Chapter 13
  Administering SAS Flex Application Themes ................................................... 163

Chapter 14
  Administering the Search Facility ................................................................. 165
## Administering the SAS Web Infrastructure Platform

### About SAS Web Infrastructure Platform
- Purpose of the SAS Web Infrastructure Platform .................. 54
- SAS Preferences Manager .............................................. 54
- SAS Comment Manager ................................................. 55

### Using Configuration Manager
- Overview of Configuration Manager .................................. 56
- Summary of Steps for Using Configuration Manager ............... 57
- Example: Configure a Property for SAS Web Report Studio .... 57

### Setting Global Properties for SAS Applications
- Purpose of the SAS Application Infrastructure Properties ....... 59
- Changing a SAS Application Infrastructure Property ............. 60
- SAS Application Infrastructure Property Descriptions .......... 60
- Using the SMS Alert Notification Type .............................. 62

### Specifying Connection Properties
- About Internal and External Connections .......................... 64
- Changing Connection Properties .................................... 65
- Changing External Connection Properties .......................... 66

### Configuring Auditing for SAS Web Applications
- Overview of Auditing .................................................. 67
- Audit Record Storage .................................................. 68
- Guidelines for Auditing the SAS Middle Tier ....................... 68
- Enable Auditing for Additional Services ............................ 69
- Enable Auditing for Internal Accounts ............................. 70
- Enable Audit Profiles .................................................. 70
- Archive Process for Audit Records .................................. 71
- Purging Audit Records .................................................. 73

### Using the SAS Web Administration Console
- About the SAS Web Administration Console ...................... 73
- Access the SAS Web Administration Console ...................... 74
- Monitor Users ............................................................ 75
- Viewing Audit Reports .................................................. 76
- Performing Server Maintenance ....................................... 77
- Updating the Job Execution Service Configuration ................ 78
- Managing Notification Templates and Letterheads ............... 78
- Managing Web Infrastructure Platform Privileges and Roles .... 80
- Managing Web-layer Permissions ..................................... 82
About SAS Web Infrastructure Platform

Purpose of the SAS Web Infrastructure Platform

The SAS Web Infrastructure Platform is a collection of services and applications that provide common infrastructure and integration features to be used by SAS web applications. These services and applications provide the following benefits:

- consistency in installation, configuration, and administration tasks for web applications
- greater consistency in users’ interactions with web applications
- integration among web applications as a result of the ability to share common resources

For a description of the SAS Web Infrastructure Platform services and applications, see “SAS Web Infrastructure Platform” on page 6.

SAS Preferences Manager

The SAS Preferences Manager is a web application that provides a central facility for users to manage their preferences and settings.

You can invoke the application by using the following URL:

http://server:port/SASPreferences

Users of SAS Information Delivery Portal can invoke the SAS Preferences Manager from within the portal. For instructions, see the product Help.

The following figure shows a generic preferences application. The actual preferences that are available vary depending on the software that is installed. The SAS Preferences Manager at your site might have additional settings.

Figure 7.1 SAS Preferences Manager Console

Here are the generic settings:

General

Specify a theme for the applications. A theme includes settings for colors, fonts, and graphics.

Users can also specify the format for notifications that are generated by SAS applications and solutions.
Language
Select the locale (language and country) that you prefer.

Format
Select the preferred format for dates, time, and currency.

Portal
Specify the position of the portal navigation bar in the SAS Information Delivery Portal. You can also specify the sort order for packages that are published in the portal. You can sort packages in descending order (newest packages are at the top) or in ascending order (oldest packages are at the top).

SAS Comment Manager

The SAS Comment Manager can be used by SAS web applications to capture user comments. For example, in SAS Web Report Studio, the File ▶ Comments menu item enables users to add comments to reports and graphs.

By default, all users who can log on to an application that uses the SAS Comment Manager can view and create comments. As an administrator, you might also want to edit and delete comments. Editing and deleting comments are considered administrative functions.

To edit and delete comments, you must belong to the predefined role, Comments: Administrator. This role includes the capabilities in the following list. Users that have a need to edit or delete comments should be assigned to this role.

Note: Due to possible conflicts that can occur when multiple users delete comments in the same comment thread, the best practice is to limit the number of users to just a few.

To edit or delete a comment, follow these steps:

1. Select the comment in the left pane of SAS Comment Manager.

2. To edit the comment, in the right pane, click Edit. An Edit Comment page opens in which you can make changes. When you are finished, click Save.

3. To delete the comment, in the right pane, click Delete. You are prompted to confirm the deletion.
The following figure shows an example of SAS Comment Manager with a comment displayed.

**Figure 7.2** SAS Comment Manager

![SAS Comment Manager](image)

### Using Configuration Manager

#### Overview of Configuration Manager

Configuration Manager is a plug-in available in SAS Management Console. Using the Configuration Manager, you can perform various administrative tasks such as configuring properties and values and specifying settings for the SAS web applications.

Configuration Manager offers a consistent interface to set properties for all SAS web applications. Each application has its own properties window with tabs. For example, the following display shows the **Settings** tab of the Web Report Studio 4.4 Properties dialog box.

Here is a brief description of the five tabs available in the properties dialog box associated with a SAS application:

**Note:** For more information about using these tabs, see the online Help for the Configuration Manager plug-in in SAS Management Console.

- The **General** tab provides basic information about the application.
The **Connection** tab enables you to modify the parameters for connections to SAS web applications. For more information, see “Specifying Connection Properties” on page 64.

The **Settings** tab offers default values for settings that can be modified. For modifying values in the **Settings** tab, and to understand how the lock and unlock icons function, see “Setting Global Properties for SAS Applications” on page 59.

The **Advanced** tab includes a limited number of default property names and values. You can modify existing properties and their values, or add custom properties and values for SAS web applications.

The **Authorization** tab enables you to specify permissions for users and groups and apply Access Control Templates.

Although certain XML configuration files (for example, `SASWebReportStudioProperties.xml` file for SAS Web Report Studio) are available and supported for SAS web applications, it is recommended that you use the Configuration Manager to configure and set properties.

### Summary of Steps for Using Configuration Manager

Here are the main steps for using Configuration Manager:

1. **To access Configuration Manager,** in SAS Management Console, navigate to **Plug-ins >> Application Management >> Configuration Manager >> SAS Application Infrastructure.**

2. **To access the properties for an application,** right-click the application's node and select **Properties.**

3. **Add or modify properties as needed.** You might need to unlock particular properties before you can change them. See “Setting Global Properties for SAS Applications” on page 59.

4. **Changes to properties do not take effect immediately on the run-time system.** To apply these changes, you must perform one of the following tasks:
   - Stop and then restart the web applications whose properties you changed.
   - Use the application’s JMX management bean to reload the configuration (if the application supports JMX beans). For more information about JMX, see “Using JMX Tools to Manage SAS Resources” on page 281.
   - Alternatively, stop and then restart SAS Web Application Server.

### Example: Configure a Property for SAS Web Report Studio

Suppose that you want to add the property, `wrs.ReportViewPrefs.LeftPanelOpenState` for SAS Web Report Studio 4.4, and specify the value for this property. To configure this property and its value, follow these steps:

1. **Log on to SAS Management Console.**
2 In SAS Management Console, navigate to Plug-ins ➤ Application Management ➤ Configuration Manager ➤ SAS Application Infrastructure ➤ Web Report Studio 4.4. Right-click and select Properties to display the Web Report Studio 4.4 Properties dialog box.

3 Click the Advanced tab.

4 Click Add to display the Define New Property dialog box.

5 Enter the property name as shown and specify the property value:

   **Property Name:** `wrs.ReportViewPrefs.LeftPanelOpenState`
   **Property Value:** `user`

6 Click OK to exit the Define New Property dialog box.

7 Click OK to exit the Web Report Studio 4.4 Properties dialog box.

Changes to properties do not take effect immediately on the run-time system. For details, see “Summary of Steps for Using Configuration Manager” on page 57.

The following display shows the property name, `wrs.ReportViewPrefs.LeftPanelOpenState`, and its property value specified on the Advanced tab.

*Figure 7.3  Advanced Tab for SAS Web Report Studio 4.4 Properties*

The dimmed fields indicate that the values are inherited from the SAS Application Infrastructure, and these values are shared with other web applications. The values in the dimmed fields can be changed only in the SAS Application Infrastructure properties.
Setting Global Properties for SAS Applications

Purpose of the SAS Application Infrastructure Properties

The Configuration Manager plug-in within SAS Management Console enables you to configure properties that apply to all SAS applications that inherit their settings from SAS Application Infrastructure. Most SAS Application Infrastructure settings are locked, and the lock prevents individual SAS applications from overriding the settings. When you unlock a SAS Application Infrastructure setting, the setting can be overridden by individual applications. When you lock a SAS Application Infrastructure setting again, all applications inherit that setting from the SAS Application Infrastructure.

The following display shows the settings that can be set for SAS Application Infrastructure.

Figure 7.4  Settings Tab for SAS Application Infrastructure Properties

The locked icon indicates that a field is locked. When a field has a locked icon, the value or setting for that particular field cannot be overridden on the Settings tab for other SAS applications that inherit the setting. By default, all fields on the Settings tab of the SAS Application Infrastructure Properties dialog box are locked.
Changing a SAS Application Infrastructure Property

1. Log on to SAS Management Console as an administrator.
2. On the Plug-ins tab, navigate to Application Management ➤ Configuration Manager ➤ SAS Application Infrastructure.
3. Right-click SAS Application Infrastructure and select Properties.
4. Click the Settings tab.
5. Select the property to change from the left panel. Use the menus or text fields to set the property.
6. Click OK.

Settings are not applied and activated automatically. You must restart the SAS Web Infrastructure Platform Services and the applications that use the changed property. If unsure, restart the web application server.

SAS Application Infrastructure Property Descriptions

The following table identifies the settings that are available for the SAS Application Infrastructure.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application &gt; User Interface</td>
<td></td>
<td>This setting controls the default theme that is used by the SAS web applications. For information about creating an alternative theme, see Chapter 12, “Administering SAS Web Application Themes,” on page 147.</td>
</tr>
<tr>
<td>Default theme</td>
<td>SAS Default</td>
<td></td>
</tr>
<tr>
<td>Display Quick Help Tips</td>
<td>Off</td>
<td></td>
</tr>
</tbody>
</table>
## Setting Global Properties for SAS Applications

### Setting | Default Value | Description
--- | --- | ---
Default Logon Target | none | Use the menu to select the application to which default URL requests are directed upon successful authentication. In this way, a site can be configured to direct users to SAS Web Report Studio, SAS Information Delivery Portal, or some solution, as a default target depending on requirements. The typical choices are identified in the following list:
- AdminHome — SAS Web Administration Console
- WRSLogon — SAS Web Report Studio
- PortalLogon — SAS Information Delivery Portal
- DisplayDashboard — SAS BI Dashboard
- MobileAdmin — SAS BI Dashboard Mobile Device Administration

### Application > Regional Settings

- **Default locale** | varies | Use the menu to select the default locale.

### Application > Pooling

- **Use client-side pooling of SAS Servers where supported** | No | For information about the advantages and disadvantages, see “Choices in Workspace Server Pooling” in SAS Intelligence Platform: Security Administration Guide. For information about configuring client-side pooling, see “Configuring Client-side Pooling” in SAS Intelligence Platform: Application Server Administration Guide.

### Notifications > General Configuration

- **Alert notifications type** | Portal | Use the menu to select the default notification types. For information about using the SMS text message setting, see “Using the SMS Alert Notification Type” on page 62.
- **Character set for e-mail messages** | UTF-8 |
- **Allow multi-part e-mail messages** | Yes |
- **Alert prefix type** | Default |
- **Alert prefix** | SAS Alert:
### Setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail digest frequency</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Notifications > Administrative and Error Messages**

<table>
<thead>
<tr>
<th>Sender of messages</th>
<th>noreply@smt pserver</th>
<th>Used as the sender email address for administrative messages.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipient of administrative messages</td>
<td>varies</td>
<td>Administrative and error messages are sent to all email addresses in the list.</td>
</tr>
</tbody>
</table>

**Formats > Formats**

<table>
<thead>
<tr>
<th>Short date format</th>
<th>varies</th>
<th>Use the menu to set the default format for date, time, and datetime values.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long date format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time/Date format</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Formats > Currency Formats**

<table>
<thead>
<tr>
<th>Currency display format</th>
<th>varies</th>
<th>Use the menu to set the default format for currency values.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currency number format</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Policies**

For information about policies, see “Configuring Middle-Tier Security Policies” on page 113.

---

**Using the SMS Alert Notification Type**

The alert notification service can send alerts through Short Message Service (SMS) text messages, in addition to sending alert notifications through email and displaying them in a portal. In order to use the SMS text message setting, the users that are to receive the messages must have an email address that is specifically for the SMS messages. The following display shows an example of the User Manager plug-in to SAS Management Console. In the display, a user’s
email address type is set to sms, and the address is provided in an SMS text message format.

Figure 7.5 SMS E-mail Address

Make sure that you know the SMS E-mail gateway for the provider. Some SMS E-mail gateways for providers in the North American market are as follows:

- Verizon: phonenumber@vtext.com
- AT&T: phonenumber@txt.att.net
- Sprint: phonenumber@messaging.sprintpcs.com
- T-Mobile: phonenumber@tmomail.net

In addition to making sure that recipients of the SMS messages have a SMS-style email address, you might need to set two properties related to SMS.

Table 7.2 Advanced Properties for SMS Messages

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notifications.SMSMessageLength</td>
<td>120 characters</td>
<td>Modify this value as needed to increase or decrease the size of SMS messages that SAS software sends to the mail server.</td>
</tr>
<tr>
<td>Policy.EnforceSMSMessageLength</td>
<td>false</td>
<td>If set to true, then messages are truncated to the length of the previous property.</td>
</tr>
</tbody>
</table>
Specifying Connection Properties

About Internal and External Connections

The connection information for each application is stored in metadata. This information is as follows:

- Communication Protocol
- Host Name
- Port Number
- Service

This information is used to construct a URL (for example, http://hostname.example.com/SASBIDashboard). This information is also used by SAS applications that need to communicate with another application. In this case, the requesting application can look up the information from metadata.

By default, the information in the previous list is identified on the Internal Connection tab for each application. (In previous releases, this was the Connection tab.) In many network topologies, end users and SAS web applications can send requests to the same URL. In these cases, the External Connection tab has the Use internal connection information check box selected, and all communication is sent to the internal connection.

Some network topologies can prevent communication between SAS web applications. The following figure shows a sample topology that prevents applications in the SAS middle tier from accessing each other through the proxy.

Figure 7.6  Network Topology with a Firewall
In these topologies, the **External Connection** tab can be used to specify different connection information. This might be necessary in the following scenarios:

- A firewall denies access to the SAS Web Server machine that originates from the machine where SAS Web Application Server is installed.
- A third-party product such as IBM Tivoli Access Manager WebSEAL or CA SiteMinder is used to protect or rewrite URLs.

The previous two items are examples of a topology or software product that interacts with SAS Logon Manager. Any change that affects access to SAS Logon Manager can require you to specify external connection information because the change can affect the call backs that occur between the applications and SAS Logon Manager.

In any network topology that prevents access to the front-end processor (identified as the proxy in the previous figure) from the SAS middle tier, you can specify different settings for the external connection. When a SAS web application accesses another application, it uses the internal connection. When a user is redirected to a URL (for example, SAS Logon Manager redirecting to SAS BI Dashboard), then the external connection information is used.

### Changing Connection Properties

The **Internal Connection** tab in the properties dialog box for SAS applications enables you to modify the parameters for connecting to a SAS web application. The selections that are displayed on the tab determine the URL that is used to access the application’s resources or services.

To change connection properties, follow these steps:

1. Log on to SAS Management Console.
2. On the **Plug-ins** tab, select **Application Management ▶ Configuration Manager ▶ SAS Application Infrastructure**.
3. Right-click the SAS Web application name, and select **Properties**.
4. Click the **Internal Connection** tab.

The following display shows the internal connection information for SAS BI Dashboard properties.
If a SAS web application is moved to a different machine (and you are not using SAS Web Server), you must modify the connection information. If you configured SAS Web Server manually for HTTPS, you must change the protocol.

Changing the values for the **Host Name**, **Port**, or **Service** fields on the tab enables the SAS Web Application Infrastructure Platform to redirect clients to the proper locations in a custom environment. For the host name, you can supply an IP address. If you enter an IP version 6 address, you must enclose the address in brackets.

For example: `[FE80::202:B3FF:FE1E:8329]`

### Changing External Connection Properties

If your site changes its configuration after initial deployment, you might need to edit the external connection information parameters. One example is adding a third-party product to the network, such as IBM Tivoli Access Manager WebSEAL or CA SiteMinder. In this case, you must route connections through the proxy. These changes must be made on the **External Connection** tab.
Clear the **Use internal connection information** check box and then enter the connection information for the proxy.

In any environment where the internal and external connection information must differ due to different access rules, you must specify the following JVM option for SAS Web Application Server:

```
-Dsas.retry.internal.url=true
```

**See Also**
“Specifying JVM Options” on page 38

---

**Configuring Auditing for SAS Web Applications**

**Overview of Auditing**

SAS web applications and other SAS middle-tier services provide auditing features. Depending on the application and its configuration, these auditing features can record all actions performed both by the direct users of the system and by the system itself. Some applications might provide a more complete audit, detailing not only the actions that are performed but also the states of the objects that are affected by those actions.

Log on, log off, and unsuccessful log on attempts create audit records for all deployments. Additional actions that can be audited for SAS Web Infrastructure Platform are described in this section. If a SAS solution is installed, see the
solution documentation for information about additional actions that can be audited.

Audit Record Storage

Audit records are stored in the SAS Web Infrastructure Platform database. These audit records are stored in two relational tables, SAS_AUDIT and SAS_AUDIT_ENTRY. Two additional tables, SAS_AUDIT_ARCHIVE and SAS_AUDIT_ENTRY_ARCHIVE, provide archival audit data.

Do not access the tables directly for audit reporting. The SAS Web Administration Console provides an interface for viewing log on, log off, unsuccessful log on attempts, and last user logon information.

Depending on the auditing configuration of the deployed SAS applications, audit records can contain different types of audit information. However, all audit records contain the following information:

- user ID that performed the audited action.
- action that occurred. This is stored as an action code.
- data and time that the audited action occurred.

Guidelines for Auditing the SAS Middle Tier

The auditing process in the SAS middle tier is designed to be efficient for both processing time and storage. However, you might want to limit the number of audited events to minimize any effect on performance and minimize the size of the audit trail. The SAS middle tier auditing features provide the tools to help you balance the need to gather sufficient security or historical records with the ability to store and process it.

Consider these guidelines to make efficient use of the SAS middle tier auditing features:

- Evaluate the purpose of auditing an action. Make sure that records for an audited action can be used to serve a business purpose.
- When auditing for security, audit generally and then audit specifically. Analyze the records from general audit options to provide the basis for targeting specific audited actions.
- When auditing for historical information, audit for actions that are important to your business only. Avoid cluttering valuable audit records with less relevant audited actions. Narrowing the focus to valuable actions also reduces the amount of audit trail administration.
- Align the audit requirements to the most strictly regulated application. If your SAS deployment includes a number of SAS applications, the applications might have varying requirements. Make sure that the audited actions match the most strictly regulated application.

When auditing is enabled and audit records are generated, the audit trail size increases according to two factors:

- the number actions that are enabled for auditing
- how frequently the audited actions are performed
If the SAS Web Infrastructure Platform database becomes completely full and audit records cannot be inserted, the audited actions cannot be successfully executed until the audit trail is purged. The system administrator must control the rate of increase and size of the audit trail. To control the size of the audit trail, consider the following strategies:

- Be selective about which actions are enabled for auditing. If the number of audited actions is reduced, then unnecessary and useless audit records are not generated and are not stored in the audit trail.
- Design archive rules to move important, but not critically important, information out of the audit trail. This process archives the audit records of interest and removes them from the main audit table. For information about archiving, see “Archive Process for Audit Records” on page 71.
- Purge the audit archive tables as needed.

### Enable Auditing for Additional Services

All SAS products that include the SAS Web Infrastructure Platform provide audit records for logon, log off, and unsuccessful log on attempts. Other standard services can also be audited:

- mail service
- content service
- job execution service
- workspace service
- scheduling service
- impersonation service

To enable auditing for any of these services, follow these steps:

1. Edit the `SASHOME\SASWebInfrastructurePlatform\9.4\Static\wars\sas.wip.services\WEB-INF\spring-config\aop-config.xml` file.

2. Review the comments to locate the service that you want to audit. Each of the services is commented out in the initial deployment. The following example shows the job execution service:

   ```xml
   <!-- Job Execution Service auditing
   <bean class="com.sas.svcs.aop.auditing.jes.SuccessfulSubmitJobAuditAdvice">
   <property name="auditRecorder" ref="auditService" />
   </bean>
   
   Note: Subsequent upgrade activities can overwrite this file. For example, if you later install a maintenance release that includes aop-config.xml, then you must repeat this procedure.

   3. Add closing comment markup and then remove the original closing comment markup (``` ) from the bottom of the code block. Save your changes.

   4. Rebuild the SAS Web Infrastructure Platform with the SAS Deployment Manager.

   5. Redeploy the SAS Web Infrastructure Platform Services web application (sas.wip.services9.4.ear).
Enabling auditing for other SAS applications requires editing different files, but the steps are similar to the previous procedure. For example, auditing for SAS Workflow is controlled with the `SASHOME\SASWebInfrastructurePlatform\9.4\Static\wars\sas.workflow\WEB-INF\spring-config\aop-config.xml` file.

### Enable Auditing for Internal Accounts

Starting with the third maintenance release of SAS 9.4, auditing support for internal accounts is enabled. This includes creating, updating, and deleting internal accounts. It also includes setting and changing passwords for internal accounts. To enable this auditing support, append the `sas.audit.svcs.identity` value to the current definition for the `-Dspring.profiles.active` JVM option. Here is an example:

```
-Dspring.profiles.active="locators,sas.audit.svcs.identity"
```

You need to specify this option for the instances of SAS Web Application Server that are used for running SAS Logon Manager only.

### See Also

“Specifying JVM Options” on page 38

### Enable Audit Profiles

Starting with the third maintenance release of SAS 9.4, audit advice beans, which are disabled by default, can be enabled. To enable this auditing support, append one or more of the desired spring profile names to the current definition for the `-Dspring.profiles.active` JVM option. Here is an example:

```
-Dspring.profiles.active="locators,spring_profile_name"
```

You need to specify this option for the instances of SAS Web Application Server that are used for running SAS Logon Manager only.

The following table lists the audit profiles that can be enabled:

<table>
<thead>
<tr>
<th>Spring Profile Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas.audit</td>
<td>Enables all audit profiles.</td>
</tr>
<tr>
<td>sas.audit.svcs.content</td>
<td>Enables all Content Server audit profiles.</td>
</tr>
<tr>
<td>sas.audit.svcs.content.service</td>
<td>Enables Content Server audits that deal with services.</td>
</tr>
<tr>
<td>sas.audit.svcs.jes</td>
<td>Enables all Job Execution Service audits (definition, execution, and retrieval).</td>
</tr>
<tr>
<td>sas.audit.svcs.jes.definition</td>
<td>Enables Job Execution Service audits that deal with job definition.</td>
</tr>
<tr>
<td>sas.audit.svcs.jes.execution</td>
<td>Enables Job Execution Service audits that deal with job execution.</td>
</tr>
<tr>
<td>sas.audit.svcs.jes.retrieval</td>
<td>Enables Job Execution Service audits that deal with job retrieval.</td>
</tr>
</tbody>
</table>
### Spring Profile Name

<table>
<thead>
<tr>
<th>Spring Profile Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas.audit svcs.mail</td>
<td>Enables all Mail Service audits.</td>
</tr>
<tr>
<td>sas.audit svcs.scheduling</td>
<td>Enables all Scheduling audits.</td>
</tr>
<tr>
<td>sas.audit svcs.impersonation</td>
<td>Enables all Impersonation audits.</td>
</tr>
<tr>
<td>sas.audit svcs wss</td>
<td>Enables all Workspace Data audits (data, data set, file, fileref, format, language, libref, utilities, and workspace).</td>
</tr>
<tr>
<td>sas.audit svcs wss data</td>
<td>Enables all Workspace Data audits that deal with data.</td>
</tr>
<tr>
<td>sas.audit svcs wss dataset</td>
<td>Enables all Workspace Data audits that deal with data set.</td>
</tr>
<tr>
<td>sas.audit svcs wss file</td>
<td>Enables all Workspace Data audits that deal with file.</td>
</tr>
<tr>
<td>sas.audit svcs wss fileref</td>
<td>Enables all Workspace Data audits that deal with fileref.</td>
</tr>
<tr>
<td>sas.audit svcs wss format</td>
<td>Enables all Workspace Data audits that deal with format.</td>
</tr>
<tr>
<td>sas.audit svcs wss language</td>
<td>Enables all Workspace Data audits that deal with language.</td>
</tr>
<tr>
<td>sas.audit svcs wss libref</td>
<td>Enables all Workspace Data audits that deal with libref.</td>
</tr>
<tr>
<td>sas.audit svcs wss utilities</td>
<td>Enables all Workspace Data audits that deal with utilities.</td>
</tr>
<tr>
<td>sas.audit svcs wss workspace</td>
<td>Enables all Workspace Data audits that deal with workspace.</td>
</tr>
</tbody>
</table>

### See Also

“Specifying JVM Options” on page 38

### Archive Process for Audit Records

Once the audit features are enabled, records are added to the SAS_AUDIT and SAS_AUDIT_ENTRY tables. The records can be archived to the SAS_AUDIT_ARCHIVE and SAS_AUDIT_ENTRY_ARCHIVE tables. An archive job is used to control which records to archive. The archive job reads the archive rules in the SAS_AUDIT_ARCHIVE_RULE table. The archive job always starts when SAS Web Infrastructure Platform Services starts. In addition, the default archive job is scheduled to start every Monday at the start of day, but the archive job schedule can be configured.
The following table describes the columns in table SAS_AUDIT_ARCHIVE_RULE. Rows must be added to this table to identify the objects, actions, and age for the archive job to process.

Table 7.3 SAS_AUDIT_ARCHIVE_RULE Column Description

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT_TYPE_ID</td>
<td>Object type. Each object type is assigned an ID in table SAS_TYPE_OBJECT.</td>
</tr>
<tr>
<td>ACTION_TYPE_ID</td>
<td>Type of change. Each action type is assigned an ID in table SAS_TYPE_ACTION.</td>
</tr>
<tr>
<td>FREQUENCY_NO</td>
<td>A numeric value in milliseconds. Records that meet the criteria for OBJECT_TYPE_ID and ACTION_TYPE_ID, and are also older than this value, are archived.</td>
</tr>
</tbody>
</table>

To control the archive job schedule, you can add a JVM option to SAS Web Application Server. The -Dsas.audit.archive.cron JVM option can be used to specify the schedule. The schedule is set with a syntax that is similar to cron:

-Dsas.audit.archive.cron="second minute hour day_of_month month day_of_week"

The following example schedules the archive job to run each day at midnight on Windows:

-Dsas.audit.archive.cron="0 0 0 * * *"

The following example schedules the archive job to run each day at midnight on UNIX:

-Dsas.audit.archive.cron="0 0 0 \* \* \*\"

Note: On UNIX systems, the quotation marks and asterisks must be preceded by a backslash.

You can confirm the archive job runs and reads the archive rules by adding a logging context to com.sas.svcs.audit at the INFO level.

The following table identifies the common object types and actions that you might want to include in the SAS_AUDIT_ARCHIVE_RULE table:

Table 7.4 Common Audit Object Types and Actions

<table>
<thead>
<tr>
<th>Audit Action</th>
<th>Object Type ID Value</th>
<th>Action Type ID Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User log on</td>
<td>-1</td>
<td>8</td>
</tr>
<tr>
<td>User log off</td>
<td>-1</td>
<td>9</td>
</tr>
<tr>
<td>Sent E-mail</td>
<td>-1</td>
<td>44</td>
</tr>
<tr>
<td>Add job</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>Submit job</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Retrieve job</td>
<td>11</td>
<td>45</td>
</tr>
<tr>
<td>Audit Action</td>
<td>Object Type ID</td>
<td>Action Type ID</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Cancel job</td>
<td>10</td>
<td>47</td>
</tr>
<tr>
<td>Release job</td>
<td>10</td>
<td>48</td>
</tr>
<tr>
<td>Update job</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>Remove job</td>
<td>11</td>
<td>37</td>
</tr>
<tr>
<td>Start scheduled job</td>
<td>86</td>
<td>3</td>
</tr>
<tr>
<td>Remove scheduled job</td>
<td>86</td>
<td>37</td>
</tr>
</tbody>
</table>

### Purging Audit Records

After auditing has been enabled for some time and the audit archive process runs, you might want to delete records from the SAS_AUDIT_ARCHIVE and SAS_AUDIT_ENTRY_ARCHIVE tables. Purging records that are no longer needed recovers some archival space and facilitates better audit trail management.

To delete records from the audit archive when using the PostgreSQL database server, connect to the database using a database client and issue the following SQL statements:

```sql
DELETE FROM sas_audit_entry_archive;

DELETE FROM sas_audit_archive;
```

To delete records for a specific time frame, issue the following SQL statements:

```sql
DELETE FROM sas_audit_entry_archive where audit_id in (select audit_id
                              from sas_audit_archive where timestamp_dttm < 'year-month-day:time');

DELETE FROM sas_audit_archive where timestamp_dttm < 'year-month-day:time';
```

For information about deleting records from other database vendors, see the documentation for that database.

### Using the SAS Web Administration Console

#### About the SAS Web Administration Console

The SAS Web Administration Console provides a central location for the following activities:

- monitoring information about users who are currently logged on to SAS web applications
viewing audit reports that show user logon and logoff activity and failed logon attempts

performing server maintenance, as a part of system maintenance

updating the Job Execution Service configuration on page 78

managing notification templates and letterheads

managing authorization, including Web Infrastructure Platform roles and privileges and web-layer permissions

viewing the current configuration for web applications that have been deployed at your site

SAS Web Administration Console also enables you to access the SAS Content Server Administration Console, which you can use to manage folders and permissions for the SAS Content Server. For details, see “Using the SAS Content Server Administration Console " on page 125.

Here is the main page of SAS Web Administration Console with the navigation pane expanded:

Figure 7.9 Main Page in SAS Web Administration Console

Note: Depending on the software that is licensed at your site, your SAS Web Administration Console might include additional functionality. For more information about the console at your site, see the administration guides for your applications.

Access the SAS Web Administration Console

To access the SAS Web Administration Console, enter the following URL in your web browser and substitute the host name and port number of your web application server:

http(s)://server:port/SASAdmin
To use this application, you must log on as someone who is a member of the SAS Administrators group (for example, sasadm@saspw).

**Note:** The SAS Content Server Administration Console has its own logon requirements. For more information, see “Using the SAS Content Server Administration Console” on page 125.

**Monitor Users**

**About the Users That Appear on the Users Page**

The Users page in the SAS Web Administration Console lists the following types of users:

- **Authenticated users**
  - are users who are currently authenticated on the system.

- **System users**
  - are system-level users who are required to perform particular tasks, such as running a stored process or accessing metadata.

**Send Email to One or More Users**

You can send email to any of the authenticated users who are currently logged on to SAS web applications. This feature is useful if you want to notify users of an impending system operation or a system outage.

To send email to users, follow these steps:

1. Select **Environment Management ➔ Users** in the navigation pane.
2. In the Users pane, select the check box in the last column of the row that contains the name of an authenticated user.
   - You can select multiple check boxes in order to send email to several users.
   - To select all of the check boxes, select the check box in the heading of the last column.
3. Click the action menu [ ] in the heading of the last column, and select **Send E-mail**.
4. If necessary, enter the email address of the recipient. If you enter more than one address, separate the addresses with a semicolon.
   - The email addresses are already listed for users whose addresses are defined in SAS metadata.
5. Enter the subject and text of the message.
6. If you have more than one recipient, specify whether you want to send a single message to all recipients or to send a separate message to each recipient.
7. Click **Send**.

**Force Users to Log Off**

Starting with the third maintenance release for SAS 9.4, you can force users to log off from a SAS web application. In some cases, users might not be actively
working with a SAS web application, and yet their sessions remain active in the system. You can force the termination of these user sessions by using the SAS Web Administration Console.

To force users to log off, follow these steps:

1. Select **Environment Management > Users** in the navigation pane.

2. In the Users pane, select the check box next to an authenticated user's name.

   You can select multiple check boxes in order to force off several users. To select all of the check boxes, select the check box in the heading of the last column.

3. Click the **Action menu** in the heading of the last column and select **Force Log Off**.

   A confirmation page displays the user ID, email address, and last logon time for the selected user. Review this information to ensure that you want to continue with the logoff operation.

4. Click **OK** to force the logoff.

### Viewing Audit Reports

The Audit page enables you to review user logon and logoff activity and the number of failed logon attempts. You can also search by user ID for a user’s last logon time.

![Audit Reports Page](Figure 7.10)

To search for a user’s last logon time, follow these steps:

1. Select **Environment Management > Audit** in the navigation pane.
2. In the Audit Reports pane, enter an authenticated user's ID in the text field, and click Submit Query.

Performing Server Maintenance

Overview

Tasks such as making changes to the metadata, restarting a metadata server, restarting the object spawner, or restarting a web application can be performed safely only when users are not logged on to applications or when new users are prohibited from logging on to the applications.

You can use the console to enable session draining for a SAS Web Application Server instance. This prevents new sessions from being sent to the server instance. You can use this feature as one step in a sequence of other tasks to prepare the system for maintenance.

The SAS Web Administration Console cannot stop, pause, or start servers. For instructions about system maintenance tasks such as stopping, pausing, or starting servers, see the SAS Intelligence Platform: System Administration Guide.

Enable Session Draining

To enable session draining, follow these steps:

   
   **Note:** By default, the load balancer manager is accessible from the same host as SAS Web Server. Modify WebServer\conf\extra\httpd-info.conf to enable connections from other machines.

2. On the load balancer manager page, select the worker URL to drain, enable the Drain option, and click Submit.

3. In the SAS Web Administration Console, select Environment Management ➤ Server Maintenance in the navigation pane.

4. On the Server Maintenance page, select the check box for the server to drain sessions from.
   
   **Note:** If a server does not run an application that provides middle-tier services, then the server is not listed. This is because there is no reason to redirect connections away from that server.

5. Click the action menu in the heading of the last column, and select Drain Sessions.

Existing sessions on the server continue to work, but new sessions are not directed to the server. You can monitor the progress of session draining with SAS Environment Manager.

In SAS Environment Manager, monitor the hostname tc Runtime SASServern_m resource. Use the Views ➤ Application Management page to view the number of sessions. For more information, see the Help or SAS Environment Manager: User’s Guide.
Note: The sessions for the SAS BI Dashboard Event Generation application do not reach zero.
New sessions are accepted once you restart the server instance.

Updating the Job Execution Service Configuration

The Job Execution Service page in the SAS Web Administration Console enables you to dynamically update the Job Execution Service with new server contexts, instead of having to restart all instances of SAS Web Application Server.

Clicking the Reconfigure button on the Job Execution Service page reconfigures the running instance of the Job Execution Service with any newly added server contexts. The server contexts were added with the Configuration Manager plug-in to SAS Management Console.

Managing Notification Templates and Letterheads

About Notifications

Applications that are part of the SAS Web Infrastructure Platform can send event-driven notifications to users. When an event occurs, the application uses the notification template that is associated with that event to create an email message and send it to the appropriate users. SAS Workflow Studio is an example of an application that uses notifications.

SAS provides standard notification templates for the SAS Web Infrastructure Platform applications that you have licensed. You can use SAS Web Administration Console to do the following:

- customize the wording and format of the standard templates
- define customized letterheads to be incorporated into notifications
- create new templates and delete existing ones
- activate a previous version of a notification template or letterhead

Beginning with SAS 9.4, notifications are managed by SAS Content Services.

Create, Edit, Test, or Delete a Notification Template

To create, edit, or test a notification template, follow these steps:

1. Select Environment Management ➤ Notifications ➤ Templates in the navigation pane.
2. On the Notification Templates page, select the locale in which you want to work.
3. If you want to create a new template, click the plus icon (+) above the table. In the New Template window, enter a name and an optional description. Click Save.
4. On the Notification Templates page, click the name of the new template (or click the name of an existing template that you want to edit or test).
5. On the Edit page, you can do the following:
Activate a previous version of the template. See “Activate a Previous Version of a Notification Template or Letterhead” on page 80.

Edit the subject line (for HTML and text formats only).

Edit the template body in the HTML, text, and Short Message Service (SMS) formats, as needed.

Specify a letterhead to be incorporated into the notification (for HTML and text formats only).

Click Preview to verify that the notification appears as it is expected.

Click Send Test Notification to send a test notification. If the template includes merge variables (substitution variables), they are listed in the Send Test E-mail dialog box. To test the appearance of these variables, you can enter sample values in the Placeholder Value column.

When you click Send in the dialog box, the email is sent to the account that you used to log on to SAS Web Administration Console. (If your account is not associated with an email address, you can specify the address by using User Manager in SAS Management Console.) If the template includes content in both HTML and text format, you will receive two messages.

6 Click Save on the Edit page to save any changes that you have made. The version number is automatically updated, and the new version is automatically set as the active version.

If you need to delete a notification template, select the appropriate locale on the Notification Templates page. Then select the check box for the appropriate letterhead, and click the minus icon (−) above the table.

Note: You should not delete the templates that are provided by SAS.

Create, Edit, or Delete a Notification Letterhead

You can further customize your notifications by adding a letterhead. SAS provides one standard letterhead that you can modify, or you can create your own. To create or edit a letterhead, follow these steps:

1 Select Environment Management ➤ Notifications ➤ Letterheads in the navigation pane.

2 On the Notification Letterheads page, select the locale in which you want to work.

3 If you want to create a new letterhead, click the plus icon (+) above the table. In the New Letterhead window, enter a name and an optional description. Click Save.

4 On the Notification Letterheads page, click the name of the new letterhead (or click the name of an existing letterhead that you want to edit).

5 On the Edit page, you can do the following:

- Activate a previous version of the letterhead. See “Activate a Previous Version of a Notification Template or Letterhead” on page 80.

- Enter (or modify) the content for either or both of the available formats (HTML and text).

- Click Preview to verify that the letterhead content appears as expected.
6 Click Save when you are finished. If you edited an existing letterhead, the version number is updated and the new version is automatically set as the active version.

You can now associate the letterhead with a notification template and then preview or test the template to verify its appearance. See “Create, Edit, Test, or Delete a Notification Template” on page 78.

If you need to delete a notification letterhead, select the appropriate locale on the Notification Letterheads page. Then select the check box for the appropriate letterhead, and click the minus icon (-) above the table.

**Activate a Previous Version of a Notification Template or Letterhead**

To activate a previous version of a notification template or letterhead, follow these steps:

1 Open the template or letterhead for editing, as described in the preceding topics.

2 On the Edit page, use the drop-down box to select the version that you want to activate. Then click **Activate as new version**.

   The newly activated template or letterhead is saved with an updated version number.

3 Click **Cancel** to exit the Edit page.

### Managing Web Infrastructure Platform Privileges and Roles

**About Web Infrastructure Platform Privileges and Roles**

Some SAS applications (such as SAS Workflow Studio) use Web Infrastructure Platform privileges and roles to control the availability of features to users and groups.

A privilege represents a specific action in an application. Privileges can affect the visibility of certain application features (such as menu items, tabs, and buttons) to users. A role is a collection of privileges. Administrators grant privileges to users or groups by making them members of roles.

There is no order of precedence for privileges. A user has a privilege if he or she is a member of any role that provides that privilege.

Several predefined roles are provided in a new deployment. For example, the ADMIN role makes the authorization tasks visible in SAS Web Administration Console. The SAS Administrative User is the only initial member of the ADMIN role. Other predefined roles are provided for specific applications. For information about those roles, see the application’s administration documentation.

**Note:** The Web Infrastructure Platform roles and privileges are separate and distinct from the metadata-layer roles and capabilities that are administered in SAS Management Console.
Assign One or More Roles to a User or Group

To assign one or more roles to a user or a group, follow these steps:


2. On the Principal Type page, select Users or Groups. Click Next.

3. On the Choose Principal page, select the user or group to which you want to assign roles. The drop-down list displays users and groups that are registered in SAS metadata. After making a selection, click Next.

4. On the Choose Roles page, select the check box for each role that you want to assign to the user or group. To remove a role assignment, clear the check box.

5. Click Finish to save your changes.

Use Bulk Assign to Assign a Role to Multiple Users or Groups

You can use the bulk assign feature to assign a single role to multiple users or groups. Follow these steps:

1. Select Environment Management ➤ Authorization ➤ Bulk Assign a Role in the navigation pane.

2. On the Choose Role page, select the role that you want to assign, and click Next.

3. On the Choose Identities page, select the check box for each user and group to which the role is to be assigned. To remove the role assignment from a user or group, clear the check box.

   **TIP** You can select the Groups link at the top of the page to move quickly to the list of groups.

4. Click Finish to save your changes.

Edit a Role's Privileges

To change the privileges that are assigned to a role, follow these steps:

1. Select Environment Management ➤ Authorization ➤ Edit a Role's Privileges in the navigation pane.

2. On the Choose Role page, select the role whose privileges you want to edit, and click Next.

3. On the Choose Privileges page, select the check box for each privilege that is to be assigned to the role. To remove a privilege, clear the check box.

4. Click Finish to save your changes.
Managing Web-layer Permissions

About Web-layer Permissions

Some SAS applications (such as SAS Workflow Studio) use SAS Content Services to manage content. Web-layer permissions control users’ access to the folders and documents that make up this content. Five permissions are supported: Read, Write, Create, Delete, and Administer. Not all permissions are applicable to all objects. For information about how a particular application uses these permissions, see the administration documentation for the application.

In general, permissions for these folders and documents are managed by the SAS applications that use them. SAS Web Administration Console enables administrators to review the permissions and, as necessary, to update them. You should use SAS Web Administration Console to update web-layer permissions only when directed to do so by SAS Technical Support.

Note: Some SAS applications use the SAS Content Server (instead of SAS Content Services) to manage content. To manage SAS Content Server permissions, see Chapter 10, “Administering the SAS Content Server,” on page 117.

Precedence in Web-layer Permissions

Authorization decisions are based on where web-layer permissions are set and to whom they are assigned. The precedence principles are as follows:

- A permission that is set directly on an object has precedence over a permission that is inherited from a parent object.
- At any particular level in the object hierarchy, a permission that is assigned to a user has precedence over a permission that is assigned to a group.
- If a user has a grant from one group and a denial from another group, the outcome is a denial.

Applications use the following process to make authorization decisions:

1. Examine any direct access controls on the target object.
   - If the requesting user has a direct grant or denial, that determines the outcome.
   - If a group to which the requesting user belongs has a direct denial, the outcome is a denial.
   - If a group to which the requesting user belongs has a direct grant (and no relevant group denial is found), the outcome is a grant.

2. Examine any direct access controls on the object’s immediate parent, following the same process as in step 1.

3. Continue moving up the inheritance hierarchy, parent-by-parent, until a relevant direct access control is found.

4. If the top of the hierarchy is reached and no relevant access control is found, the outcome is a denial.
Reviewing and Setting Web-layer Permissions

You can use SAS Web Administration Console to review and update permissions for folders and documents that are managed by SAS Content Services.

**CAUTION!** In general, permissions for these folders and documents are managed by the SAS applications that use them. You should use SAS Web Administration Console to update permissions only when directed to do so by SAS Technical Support.

To review or update permissions on a folder or document that is managed by SAS Content Services, follow these steps:

1. Select **Environment Management** → **Authorization** → **Permissions** in the navigation pane.

2. The **Web Authorization: Access Controls** page displays content folders and objects in a tree format. Click the plus icons to expand the nodes, and use the scroll bars as needed to view the expanded tree.

3. Click the folder or object of interest to select it. The Properties section displays the path, object type, and owner information for the selected folder or object, and the Direct Access Controls section displays the current permission settings.

4. In the Direct Access Controls section, select the check box to select or clear the option **Child objects can inherit these settings**.

5. For each user or group that is displayed, use the drop-down boxes as needed to modify the permission settings.

6. To specify permissions for additional users or groups, follow these steps:
   a. In the first column of the last row of direct access controls, select the appropriate principal type (**User** or **Group**). From the second drop-down box, select the user or group for which you want to assign permissions. (The drop-down list displays users and groups that are registered in SAS metadata.) Use the drop-down boxes in columns three through seven to assign settings for each permission.
   b. To specify permissions for another user or group, click the plus icon (+) at the end of the last row. In the new row, select the principal type, the user or group, and the appropriate permission settings. To specify permissions for more users and groups, repeat this step as needed.

7. When you are finished, click **Save**.

Viewing Information about Web Applications

The SAS Web Administration Console provides configuration information about the SAS web applications that are installed and configured at your site. This information is also available in SAS Management Console. However, SAS Web Administration Console enables you to view the information from any machine with a web browser, without the need to install SAS Management Console on the machine.
To display a list of configured web applications, expand the **Application Management** node in the navigation pane. When you click the name of an application, the right pane displays information under the following headings:

**Application Settings**
- displays settings that are currently configured for the application. For example, SAS Information Delivery Portal settings include the locale that is in use, the location where portlets are deployed, the email host, and default settings for various user preferences.

You cannot change any of the application settings here. To change settings, use the **Application Management » Configuration Manager** plug-in in SAS Management Console.

**Directives**
- provides the internal direction to the application’s URL. This information is used internally to route applications. You might use this information to troubleshoot applications under the guidance of SAS Technical Support.

**Logging**
- displays a form that is used to configure logging for applications that are instrumented for dynamic logging control.
About SAS Deployment Manager

The SAS Deployment Manager enables a SAS administrator to perform the following tasks that are typical for the middle tier:

- **Rebuild web applications.** You can rebuild web applications that have previously been configured but whose configuration has changed. This option rebuilds the web application based on the current configuration. See "Rebuilding the SAS Web Applications" on page 86.

- **Redeploy web applications.** You can redeploy web applications that have been rebuilt. See "Redeploying the SAS Web Applications" on page 91.

- **Remove the existing configuration.** You can remove the product configuration for one or more products in the deployment. This option enables you to remove the product configuration for an application that you are no longer using or that you are moving to another machine. You can then use the SAS Deployment Wizard to reinstall or reconfigure the application. For details, see "Removing a SAS Configuration" in the SAS Intelligence Platform: Installation and Configuration Guide.

Note the following about removing a configuration:

- Installed products are not removed.
If you remove the configuration for the SAS Information Delivery Portal, do not select the Remove all User Content option unless you have made a backup copy of the content repository. If you choose this option, you must re-create the content later from your backup. When you choose to remove portal content, all pages, portlets, and other items created by the users are removed.

If you remove the configuration for the Web Infrastructure Platform, the contents of the SAS Content Server repository (located in the SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository directory) are not deleted. If you do not need the contents of this directory, you should manually delete the contents before rebuilding the Web Infrastructure Platform with the SAS Deployment Manager.

Access the SAS Deployment Manager by running the SAS-installation-directory\SASDeploymentManager\9.4\sasdm.exe command. On UNIX operating environments, the command is sasdm.sh.

Rebuilding the SAS Web Applications

When to Rebuild the SAS Web Applications

The Rebuild Web Applications option of the SAS Deployment Manager provides an automated way to rebuild the web applications that are deployed in your environment. You should rebuild the web applications in the following situations:

- You might need to rebuild applications that you have reconfigured. For example, if you change the HTTP time-out interval for an application, then you should rebuild the application.

  Note: This administration guide informs you when an application must be rebuilt after reconfiguration.

- Rebuild an application after you change the Java security configuration for the application.

- If a custom theme is created for your organization, then rebuild the SAS Web Application Themes.

- If custom content is created, then add files to the WAR directory and rebuild the application to which the custom content applies. For example, to create custom forms for SAS Stored Process, place the file for the EAR or the WAR in the SAS-configuration-directory\Lev1\Web\Common\SASServer1\SASStoredProcess9.4\CustomContent\ears\sas.storedprocess\input directory. Then, use the SAS Deployment Manager to rebuild the SAS Stored Process application. For more information, see “Web Application Custom Content” on page 90.

- If custom portal content is created, such as a custom portlet, then rebuild the SAS Information Delivery Portal. For more information, see “Rebuild Web Applications” on page 87.

- Rebuild SAS Help Viewer for Midtier Applications after your initial deployment if you install or upgrade a SAS web application that offers online
Help. (SAS Help Viewer for Midtier Applications combines SAS Help Viewer for the Web software with various help content into its EAR file.)

The following web applications use SAS Help Viewer for Midtier Applications:
- SAS Information Delivery Portal Help
- SAS Web Report Studio Help
- SAS Web Report Viewer Help
- SAS BI Dashboard Help
- SAS Comment Manager Help

After installing a maintenance release or hot fixes, rebuild the web applications that were updated at your site. Follow the instructions in the maintenance documentation or the hot fix instructions. Because the web applications are rebuilt, you might lose any customizations that you added after the initial deployment.

**Rebuild Web Applications**

The **Rebuild Web Applications** option in the SAS Deployment Manager enables you to rebuild one or more web applications. The rebuild process updates two directories for each rebuilt web application:

- `SAS-configuration-directory\Lev1\Web\Staging`. An EAR or WAR file for each rebuilt application is placed in this directory.
  - The approximate size of the collection of applications for SAS Enterprise Business Intelligence is 4 GB.

- `SAS-configuration-directory\Lev1\Web\Staging\exploded`. An exploded version of each rebuilt application is placed in this directory.
  - **Note:** You can delete any unwanted directories in the `exploded` directory to save disk space.

To rebuild one or more web applications, follow these steps:

1. The web application server can be running or stopped.
2. Make sure that the SAS Metadata Server is running.
3. Start the SAS Deployment Manager.
4. Select **Rebuild Web Applications** and click **Next**.
5. Specify the configuration directory and the level (for example, Lev1) on the Select Configuration Directory/Level page. Click **Next**.
6. Enter the user ID and password for an unrestricted administrative user (for example, sasadm@saspw) on the Specify Connection Information page. Click **Next**.
7. Select the check boxes for the web applications that you want to rebuild and click **Next**.
8. Review the Summary page and click **Start**. The SAS Deployment Manager builds the files for the selected applications. For the names and location of the files, see “Web Application Names and EAR and WAR Files” on page 88.
If you are rebuilding theme content, you might need to stop and restart the web application server as follows.

If SAS Web Application Themes is deployed to the web application server, then the first time a custom theme is deployed, the web application server must be stopped and restarted. Any subsequent modifications to the custom theme do not require a restart of the web application server unless the theme descriptors have been changed.

After rebuilding the web applications, the next action is typically to redeploy them. See “Redeploying the SAS Web Applications” on page 91.

### Web Application Names and EAR and WAR Files

The files for the SAS web applications are stored in the following directories:

- `SAS-configuration-directory\Lev1\Web\Staging`
- `SAS-configuration-directory\Lev1\Web\Staging\exploded`

When the SAS Deployment Manager is used to rebuild a web application, the files for the web application in the previous directories are overwritten. The following table identifies the product configuration name that is used in the SAS Deployment Manager for the web applications that are part of the SAS Enterprise Business Intelligence Server. Use this table to understand which web applications and files are updated when a product configuration is selected in the SAS Deployment Manager.

<table>
<thead>
<tr>
<th>Product Configuration</th>
<th>Application</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>BI Dashboard version</td>
<td>SAS BI Dashboard</td>
<td><code>sas.bidashboardversion.ear</code></td>
</tr>
<tr>
<td>BI Portlets version</td>
<td>SAS BI Portlets</td>
<td><code>sas.biportletsversion.ear</code></td>
</tr>
<tr>
<td>Environment Manager version</td>
<td>SAS Environment Manager</td>
<td><code>sas.environmentmanagerversion.ear</code></td>
</tr>
<tr>
<td>Middle Tier version</td>
<td>SAS Environment Manager</td>
<td></td>
</tr>
<tr>
<td>Flex Application Themes version</td>
<td>SAS Flex Application Themes</td>
<td><code>sas.flexthemesversion.ear</code></td>
</tr>
<tr>
<td>SAS Theme Designer for Flex</td>
<td>SAS Theme Designer for Flex</td>
<td><code>sas.themedesignerversion.ear</code></td>
</tr>
<tr>
<td>Help Viewer for Midtier App version</td>
<td>SAS Help Viewer for Midtier Applications</td>
<td><code>sas.webdocmdversion.ear</code></td>
</tr>
<tr>
<td>Information Delivery Portal version</td>
<td>SAS Information Delivery Portal</td>
<td><code>sas.portalversion.ear</code></td>
</tr>
<tr>
<td>SAS Themes</td>
<td>SAS Web Application Themes</td>
<td><code>sas.themes.ear</code></td>
</tr>
<tr>
<td>Product Configuration</td>
<td>Application</td>
<td>Filename</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>Visual Analytics version</strong></td>
<td>SAS Visual Analytics Administrator</td>
<td>sas.visualanalyticsadministratorversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Visual Analytics Designer</td>
<td>sas.visualanalyticsdesignerversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Visual Analytics Explorer</td>
<td>sas.visualanalyticsexplorerversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Visual Analytics Graph Builder</td>
<td>sas.visualanalyticsgraphbuilderversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Visual Analytics Hub</td>
<td>sas.visualanalyticshubversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Visual Analytics Services</td>
<td>sas.visualanalyticsservicesversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Visual Analytics Viewer</td>
<td>sas.visualanalyticsviewerversion.ear</td>
</tr>
<tr>
<td><strong>Web Infrastructure Platform version</strong></td>
<td>SAS Content Server</td>
<td>sas.wip.scsversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Stored Process</td>
<td>sas.storedprocessversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Web Administration Console</td>
<td>sas.wip.adminversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Web Infrastructure Platform Applications</td>
<td>sas.wip.appsversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Web Infrastructure Platform Resources</td>
<td>sas.wip.resourcesversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Web Infrastructure Platform Services</td>
<td>sas.wip.servicesversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Workflow</td>
<td>sas.workflowversion.ear</td>
</tr>
<tr>
<td></td>
<td>SAS Authorization Service</td>
<td>sas.authorization.services.war</td>
</tr>
<tr>
<td></td>
<td>SAS Identity Services</td>
<td>sas.identity.services.war</td>
</tr>
<tr>
<td></td>
<td>SAS Principal Services</td>
<td>sas.principal.services.war</td>
</tr>
<tr>
<td><strong>Web Report Studio version</strong></td>
<td>SAS Web Report Studio</td>
<td>sas.webreportstudioversion.ear</td>
</tr>
</tbody>
</table>
Web Application Custom Content

About Web Application Custom Content

You can add custom content to a SAS web application by creating and saving your custom content in the appropriate custom content directory structure, running the SAS Deployment Manager, and redeploying the web application.

The custom content root directory for a given web application is `SAS-configuration-directory\Levn\Web\Staging\Common\SASServern\ApplicationName\CustomContent`.

Note: Some SAS web applications do not support custom content.

The `CustomContent` directory contains subdirectories that correspond to the specific archive types. For example, the `ears` subdirectory contains EAR files. The `wars` subdirectory contains WAR files. The archive type directories contain subdirectories for each specific archive. These are the root directories for each archive within the application. Custom content should be placed in the archive's directory tree corresponding to where the content should appear within the archive.

Create and Use Custom Content

To create and use custom content, follow these steps:

1. Add the `ear_addon.xml` file to the `addons` directory in the `sas.webreportstudio` EAR file, create the `SAS-configuration-directory\Levn\Web\Common\SASServern\ApplicationName\CustomContent\ears\sas.webreportstudio\addons` directory and save the `ear_addon.xml` in the directory.

   The SAS process knows which WAR files are contained within EAR files, so if you want to add the `war_addon.jar` file to the `WEB-INF/lib` directory in the `sas.webreportstudio` WAR file in the `sas.webreportstudio` EAR file, create the `SAS-configuration-directory\Levn\Web\Common\SASServern\ApplicationName\CustomContent\wars\sas.webreportstudio\WEB-INF\lib` directory, and save the `war_addon.jar` file there.

2. To use your custom content, run the SAS Deployment Manager and choose to rebuild the web applications. Doing so rebuilds the web applications, inserting the custom content into the archives under the appropriate paths. For more information, see “Rebuild Web Applications” on page 87.

3. Redeploy the web applications. For more information, see “Redeploy Web Applications” on page 91.

Note: If custom content has the same path and name of content normally included in the archive, then the custom content takes precedence.
Redeploying the SAS Web Applications

About Redeploying Web Applications

When the SAS Deployment Manager rebuilds SAS web applications, the rebuilt EAR files are placed in the `SAS-configuration-directory\Lev1\Web \Staging` directory. All EAR files are placed in a single directory even if your deployment includes multiple web application servers (for example, SASServer1_1 and SASServer2_1).

If you have web application servers that were installed and configured by the SAS Deployment Wizard in your environment, make a note of the server names and the web applications that are installed on each server. For example, if six applications are located on SASServer1_1 and three applications are located on SASServer2_1, make a list of the applications that are installed on each of these servers. Alternatively, you can refer to your `Instructions.html` file, which specifies the following:

- the list of web applications to be deployed
- the location of the applications
- the web application server where each application should be deployed

When you redeploy the SAS web applications, you can refer to your list or the `Instructions.html` file, to ensure that you redeploy each application to the correct server.

Redeploy Web Applications

Steps to Perform with the SAS Deployment Manager

The SAS Deployment Manager manages the SAS web applications as EAR files but the applications are deployed as WAR files.

To redeploy one or more web applications, follow these steps:

1. The web application server must be running.
2. Make sure that the SAS Metadata Server is running.
3. Start the SAS Deployment Manager.
4. Select **Deploy Web Applications** and click **Next**.
5. Specify the configuration directory and the level (for example, Lev1) on the Select Configuration Directory/Level page. Click **Next**.
6. Enter the user ID and password for an unrestricted administrative user (for example, sasadm@saspw) on the Specify Connection Information page. Click **Next**.
7. The manager provides a warning that SAS Web Application Server will be stopped. Be aware that the web applications are not available while the...
server is stopped. Select the **Allow the application server to stop** check box and click **Next**.

8 Select the check boxes for the web applications that you want to redeploy and click **Next**.

   For the names, see "Web Application Names and EAR and WAR Files" on page 88.

9 Review the Summary page and click **Start**. The SAS Deployment Manager stops the server, deploys the web applications, and starts the server.

**Backups of Previous Web Application Versions**

Before the SAS Deployment Manager redeploys a web application, it creates backups of the existing version and the context file. The backups are as follows:

- Application backups are in the `SAS-configuration-directory\Lev\n\Web\WebAppServer\SASServer1_1\sas_webapps\Backup` directory.
- Context file backups are in the `SAS-configuration-directory\Lev\n\Web\WebAppServer\SASServer1_1\conf\Catalina\localhost\Backup` directory.

A timestamp is appended to the web application directory and context file to indicate when the backup was performed. If you frequently redeploy web applications, you can consume disk space. You can delete backup files that are no longer needed.

**Additional Steps for Horizontal Clusters**

To redeploy web applications on additional machines in a horizontal cluster, follow these steps:

1 Stop the instances of SAS Web Application Server on the additional machines.

2 Copy the updated application EAR and WAR files from the primary machine to the staging directory on the additional machines.

   The staging directory is typically `SAS-configuration-directory\Lev\n\Web\Staging`.

3 Use the `appsrvconfig` command on each additional machine to undeploy and redeploy the web applications:

   appsrvconfig.cmd -e run undeploy application application-name server SASServer1

   appsrvconfig.cmd -e run deploy application application-name server SASServer1

   The command is located in the `SAS-configuration-directory\Lev\n\Web\Scripts\AppServer` directory. On UNIX deployments, the command is `appsrvconfig.sh`. For more information, see Chapter 21, “SAS Configuration Scripting Tools,” on page 287.

4 Start the instances of SAS Web Application Server on the additional machines.

5 Restart SAS Web Server.
Reconfiguring the Web Application Server

Reconfigure your web application server when any of the following conditions apply:

- A new SAS web application is added to your deployment.
- A web application is unconfigured and reconfigured.
- A software bundle is added to an existing configuration.

It is important to reconfigure your web application server in the same manner that it was initially configured. If you manually configured SAS Web Application Server when you initially deployed, then configure it manually again. If the SAS Deployment Wizard automatically configured your web application server, then choose the automatic configuration option again.

If the environment was initially configured with the **Web Application Server: Multiple Servers** option in the SAS Deployment Wizard, reconfigure SAS Web Application Server by using the Custom path in the SAS Deployment Wizard and selecting the **Web Application Server: Multiple Servers** again. Reconfiguring SAS Web Application Server can cause the loss of some customizations, and they need to be reapplied.

For more information, see “Managing Your SAS Deployment” in the SAS Intelligence Platform: Installation and Configuration Guide.

Administering Logging for SAS Web Applications

Logging for SAS Web Applications

The SAS web applications use log4j to perform logging. As each web application begins running, the log4j configuration file for the application is read from `SAS-configuration-directory\Level1\Web\Common\LogConfig`. After the log4j configuration file is read, the applications that permit dynamic logging changes check for modifications that were set with the SAS Web Administration Console.

**Note:** Dynamic logging does not work in a clustered environment.

The following table identifies if customizations can be performed by editing the log4j configuration file, using dynamic logging changes, or both:

<table>
<thead>
<tr>
<th>Task</th>
<th>Log4j Configuration File</th>
<th>Dynamic Logging Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the logging levels.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Add a logging category.</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Changes persist after web application server restarts.

Add or change an appender to log to console, file, socket, or ARM.

Change a log filename or location.

Change the layout pattern for the log message.

Track user logons. You can monitor usage patterns by logging activity for SAS web application logons.


Logging categories use the fully qualified class name of the class where the logging message originates. Categories for the following classes are common to all SAS web applications:

- com.sas
- com.sas.services
- com.sas.services.deployment
- com.sas.services.discovery
- com.sas.services.util

Change the Logging Levels

Logging Level Descriptions

Log4j files offer many levels of logging detail. Enabling a level also enables the less detailed levels above the selected level. The default level is set to WARN, which means that WARN, ERROR, and FATAL messages are recorded. In large-scale deployments, the size of the log file can grow rapidly when INFO messages are enabled. However, you might want to enable the INFO messages during the development and testing phases.

**CAUTION!** Excessive logging can degrade performance. Therefore, you should not use the DEBUG level unless you are directed to do so by SAS Technical Support.

If you need to debug a problem, it is recommended that you dynamically change the log output temporarily.

Here is a brief description of each level:

- **ALL**
  
  enables all logging.
TRACE
displays finer-grained informational events then DEBUG.

DEBUG
displays the informational events that are most useful for debugging an application.

INFO
displays informational messages that highlight the progress of the application.

WARN
displays potentially harmful situations.

ERROR
displays error events that might allow the application to continue to run.

FATAL
displays very severe error events that might cause the application to end abnormally.

OFF
disables all logging.

**Using log4j Files**

To modify the logging level by editing the log4j files, follow these steps:

1. Change directory to `SAS-configuration-directory\Lev1\Web\Common\LogConfig` and edit the log4j file for the application to modify.

2. Locate the category for the class that you want to modify and modify the value of the priority parameter:

   ```xml
   <category
       additivity="false"
       name="com.sas.workflow">
       <priority
           value="WARN"/>
       <appender-ref
           ref="SAS_CONSOLE"/>
       <appender-ref
           ref="SAS_FILE"/>
   </category>
   ```

3. Restart the web application so that it uses the new configuration.

**Applications That Support Dynamic Logging**

The following applications support dynamic logging changes. The name in the left column can be found in the SAS Web Administration Console. The right
column shows the context root and path for the URL to the logging control console.

Table 8.2 Dynamic Logging

<table>
<thead>
<tr>
<th>Name in Web Administration Console</th>
<th>Context Root for Logging Control Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not listed*</td>
<td>SASAdmin/admin/Logging</td>
</tr>
<tr>
<td>BI Web Services for Java version</td>
<td>SASBIWS/admin/Logging</td>
</tr>
<tr>
<td>LASR Authorization Service version</td>
<td>SASLASRAuthorization/admin/Logging</td>
</tr>
<tr>
<td>Logon Manager version</td>
<td>SASLogon/admin/Logging</td>
</tr>
<tr>
<td>Notification Template Editor version</td>
<td>SASTemplateEditor/admin/Logging</td>
</tr>
<tr>
<td>Preferences Manager version</td>
<td>SASPreferences/admin/Logging</td>
</tr>
<tr>
<td>Risk Management for Banking version</td>
<td>SASRiskManagementForBanking/admin/Logging</td>
</tr>
<tr>
<td>SAS Deployment Backup and Recovery Tool version</td>
<td>SASDeploymentBackup/admin/Logging</td>
</tr>
<tr>
<td>SAS Studio Mid-Tier version</td>
<td>SASStudio/admin/Logging</td>
</tr>
<tr>
<td>Search Interface to SAS Content version</td>
<td>SASSearchService/admin/Logging</td>
</tr>
<tr>
<td>Shared Applications version</td>
<td>SASSharedApps/admin/Logging</td>
</tr>
<tr>
<td>Stored Process Web App version</td>
<td>SASStoredProcess/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Admin version</td>
<td>SASVisualAnalyticsAdministrator/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Designer version</td>
<td>SASVisualAnalyticsDesigner/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Explorer version</td>
<td>SASVisualAnalyticsExplorer/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Graph Builder version</td>
<td>SASVisualAnalyticsGraphBuilder/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Hub version</td>
<td>SASVisualAnalyticsHub/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Transport Service version</td>
<td>SASVisualAnalyticsTransport/admin/Logging</td>
</tr>
<tr>
<td>Visual Analytics Viewer version</td>
<td>SASVisualAnalyticsViewer/admin/Logging</td>
</tr>
<tr>
<td>Visual Data Builder version</td>
<td>SASVisualDataBuilder/admin/Logging</td>
</tr>
</tbody>
</table>
Using SAS Web Administration Console

You can use SAS Web Administration Console to change logging levels at runtime. This feature is useful if you want to temporarily change the levels. Once you restart SAS Web Server, the logging levels revert to the levels defined in the log4j file. The applications that support dynamic logging control from the console are listed in “Applications That Support Dynamic Logging”.

Initially, the unrestricted user is the only user that can change dynamic logging levels. You can grant other users and groups this access by assigning them to the ROLE_ADMIN role. For more information, see “Assign One or More Roles to a User or Group” on page 81.

To change the logging levels with SAS Web Administration Console, follow these steps:

1. Log on to SAS Web Administration Console.
2. Expand Application Management and then select the web application that you want to change.
3. Expand the Logging section.
   
   Note: The first time you expand this section, it might indicate that logging configuration management is not enabled for the application. The applications can require one minute to refresh and display the control console.

4. Select the radio button for the class and logging level that you want to change.
5. Click Submit Changes. The change takes effect immediately. You do not need to restart the web application.
Accessing the Logging Control Console

The logging control console that is displayed in the Logging subsection of the SAS Web Administration Console can also be accessed directly from the application.

Open a web browser and enter a URL that is similar to the following example:

http://hostname.example.com/SASBIWS/admin/Logging

The list of applications and the context that you need to specify are listed in “Applications That Support Dynamic Logging”.

Changing the Authorization Requirement for Changing Logging Levels

To accommodate changing logging levels for some of the applications that support dynamic logging control without restarting the middle tier, you can change a parameter that controls security. The parameter can be used to enable or disable the authorization requirement.

The default value is to require authorization.

To change the security setting, follow these steps:

1. Edit one or more of the following XML files:
   - SASServer1_1\sas_webapps\sas.authorization.services.war\WEB-INF\web.xml
   - SASServer1_1\sas_webapps\sas.wip.services.war\WEB-INF\web.xml
   - SASServer1_1\sas_webapps\sas.identity.services.war\WEB-INF\web.xml
   - SASServer1_1\sas_webapps\sas.svcs.logon.war\WEB-INF\web.xml
   - SASServer1_1\sas_webapps\sas.principal.services.war\WEB-INF\web.xml

2. Locate the logging servlet section and set the applySecurity parameter:

   <servlet>
     <servlet-name>logging</servlet-name>
     <servlet-class>
       com.sas.svcs.webapp.servlet.http.LoggingAdminServlet
     </servlet-class>
     <init-param>
       <param-name>applySecurity</param-name>
       <param-value>true</param-value>
     </init-param>
   </servlet>

If you made a change and want it to persist when applications are rebuilt and redeployed, then make the same change in the web.xml.orig file for the application. See the following list for the locations of the files.

- `SASHOME\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.authorization.services\WEB-INF\web.xml.orig`
- `SASHOME\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.services\WEB-INF\web.xml.orig`
- `SASHOME\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.principal.services\WEB-INF\web.xml.orig`
- `SASHOME\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.identity.services\WEB-INF\web.xml.orig`
- `SASHOME\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig`

### Changing the Location of the Log Files

To modify the location of a log file, follow these steps:

1. Change directory to `SAS-configuration-directory\Lev1\Web\Common\LogConfig` and edit the log4j file for the application to modify.

2. Locate the file appender and modify the value of the file parameter:

   ```xml
   <appender class="org.apache.log4j.FileAppender"
     name="SAS_FILE">
     <param name="append" value="true"/>
     <param name="file" value="C:/SAS/Config/Lev1/Web/Logs/SASLogon9.4.log"/>
     <layout class="com.sas.svcs.logging.CustomPatternLayout">
       <param name="ConversionPattern" value="%d [%t] %-5p [%u] %c - %m%n"/>
     </layout>
   </appender>
   ``

   **TIP** The CustomPatternLayout that is provided by SAS accepts the log4j conversion characters and two conversion characters that are added by SAS. The %u conversion character is used to report the client identity that is in the security context. The %s conversion character is used to report the session identifier that is in the security context. The log4j conversion characters are described at [http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html](http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html).

3. Restart the web application so that it uses the new configuration.
About SAS Logon Manager

The SAS Logon Manager is a web application that handles all authentication requests for SAS web applications. As a result, users see the same sign-in page when they access the SAS web applications.
The purpose of the SAS Logon Manager is to authenticate and direct a successful sign-in to the appropriate web application. The application also serves as the central point for handling changes to authentication mechanisms, such as the addition of Windows SSPI or third-party single sign-on products.

When a user successfully authenticates to SAS Logon Manager, the user receives a global single sign-on session. This is introduced in the SAS 9.4 release. The global single sign-on session enables the user to access all the SAS web applications that the user is authorized to use, without a credential challenge for each web application. The global single sign-on time-out is independent of the web application time-out interval. For more information, see the Log user off on timeout policy in Configuring Middle Tier Security Policies.

Customizing Sign-in, Sign-out, and Time-out Messages

Step 1: Customize the Message

You can configure a customized message that is displayed when users of SAS web applications sign in, sign out, or the session reaches the time-out interval. To enable the display of a custom message, follow these steps:

1. In the `SASHOME\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\view\jsp\default\ui` directory, edit the files that you want to change:
   - logon_custom.jsp
   - logoff_custom.jsp
   - timeout_custom.jsp

   Each file is included as part of an HTML page. Therefore, each should contain valid HTML code.

2. Save your changes.

Step 2: Configure SAS Application Infrastructure

1. Log on to SAS Management Console.

2. On the Plug-ins tab, select Application Management ▶ Configuration Manager, right-click SAS Application Infrastructure, and select Properties.

3. Click the Settings tab.

4. Select Policies in the left pane.

5. Set any or all of these properties to Yes:
   - Display custom logon message
   - Display custom logoff message
   - Display custom timeout message
Click **OK**.

6  Exit from SAS Management Console.

---

**Step 3: Rebuild and Redeploy SAS Web Infrastructure Platform**

1  Rebuild the SAS Web Infrastructure Platform with the SAS Deployment Manager.

2  Redeploy the SAS Web Infrastructure Platform with SAS Deployment Manager. (Stop SAS Web Application Server before performing the redeploy.)

3  Verify that the custom sign-out message is displayed when you sign in and sign out from the web application.

---

**Step 4: Back Up the Customized Files**

Back up the customized files from the `SASHOME\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\view\jsp\default\ui` directory.

If a maintenance release is applied to the system, those files are overwritten and your changes are lost. After applying a maintenance release, restore the customized files.

---

**Displaying a Warning Message for Inactive User Sessions**

**Understanding Inactive Users and Time-out Warnings**

Inactive users are directed to a time-out page when their sessions are inactive for 30 minutes or for the amount of time specified by the administrator in the `web.xml` files. (You can change this behavior to log users off instead by setting the **Log user off on timeout** policy.)

Before being directed to the time-out page, you can alert users about the impending time-out by displaying a warning message. When the warning message is displayed, users can click the **Continue** button to activate and extend their sessions. The following applications support the display of a warning message:

- SAS Web Report Studio
- SAS Information Delivery Portal
- SAS BI Dashboard
- SAS Package Viewer
- SAS Shared applications
If you want to specify a different session time-out interval for each SAS application, complete this task for each SAS application by defining the `App.SessionTimeoutWarningInterval` property and a custom value in minutes.

**Step 1: Configure the SAS Application Infrastructure**

To configure the SAS application infrastructure:

1. Log on to SAS Management Console.
2. On the **Plug-ins** tab, select **Application Management ➤ Configuration Manager**, right-click **SAS Application Infrastructure**, and select **Properties**.
3. In the **SAS Application Infrastructure Properties** dialog box, click the **Advanced** tab.

**Step 2: Set the Interval for the Inactive Session Warning**

This set of steps is optional. If you do not specify a value for the `App.SessionTimeoutWarningInterval`, a default value of 5 minutes is used.

To set the interval for the inactive session warning:

1. Click **Add** to define a new property.
2. Enter `App.SessionTimeoutWarningInterval` in the **Property Name** field.
3. Enter the number of minutes for the inactive session warning in the **Property Value** field and click **OK**.

**Step 3: Enable the Inactive Session Warning**

To enable the inactive session warning:

1. Click **Add** to define another new property.
2. Enter `Policy.DisplaySessionTimeoutWarning` in the **Property Name** field.
3. Set the value to `true` and click **OK**.

To enable these properties to take effect, restart the web application server.
Configuring the HTTP Session Time-out Interval

A session time-out interval logs off users’ inactive sessions after a specific period of time that is defined in the web application server configuration. The default value for a session time-out interval is 30 minutes. You can customize the session time-out interval for your environment by modifying one or more of the web.xml files, and specifying a different time-out interval.

Be aware that reaching the time-out limit for an application does not end the user’s global single sign-on session unless the Log user off on timeout policy is set to Yes. For more information, see Configuring Middle Tier Security Policies.

To specify a session time-out interval, follow these steps:

1. Use the table that follows this procedure to identify the files to modify.

2. Modify the following code in the appropriate files:

   `<session-config>
    <session-timeout>time-out-interval</session-timeout>
   </session-config>

   Replace time-out-interval with the time-out interval in minutes. As a recommendation, the number should be no smaller than 5.

   When you are finished, save and close the file.

3. Use the SAS Deployment Manager to rebuild the modified SAS web applications.

4. Use the SAS Deployment Manager to redeploy the modified SAS web applications.

The following table lists the file or files that should be modified to specify a time-out interval for each web application.

<table>
<thead>
<tr>
<th>Web Application</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Deployment Backup and Recovery Tool</td>
<td><code>SAS-installation-directory\SASDeploymentBackupandRecoveryTool\9.4\configurable\wars\sas.svcs.admin.backup\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Environment Manager Middle-Tier Configuration</td>
<td><code>SAS-installation-directory\SASEnvironmentManagerMidTier\9.4\Configurable\wars\sas.admapp.fldmod\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td></td>
<td><code>SAS-installation-directory\SASEnvironmentManagerMidTier\9.4\Configurable\wars\sas.admapp\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Help Viewer for Midtier Applications</td>
<td><code>SAS-installation-directory\Documentation\9.4\Static\wars\sas.webdoc\WEB-INF\web.xml</code></td>
</tr>
<tr>
<td></td>
<td><code>SAS-installation-directory\Documentation\9.4\Static\wars\sas.webdoc\WEB-INF\web.spring-enabled.xml</code></td>
</tr>
<tr>
<td>Web Application</td>
<td>File Location</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAS BI Dashboard</td>
<td>SAS-installation-directory\SASBIDashboard\4.4\Configurable\wars\sas.bidashboard\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>Event generation framework in SAS BI Dashboard</td>
<td>SAS-installation-directory\SASBIDashboard\4.4\Configurable\wars\sas.eventsgenerationframework\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS BI Portlets</td>
<td>SAS-installation-directory\SASBIPortlets\4.4\Configurable\wars\sas.biportlets\WEB-INF\web.xml-thirdparty.orig</td>
</tr>
<tr>
<td>JSR 168 for SAS BI Portlets</td>
<td>SAS-installation-directory\SASBIPortlets\4.4\Configurable\wars\sas.jsr168remoteportlet\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>Flex Themes for SAS</td>
<td>SAS-installation-directory\SASFlexApplicationThemes\3.51\Configurable\FlexThemes\wars\sas.flexthemes\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Theme Designer for Flex</td>
<td>SAS-installation-directory\SASFlexApplicationThemes\3.51\Configurable\ThemeDesigner\wars\sas.themedesigner\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Package Viewer</td>
<td>SAS-installation-directory\SASInformationDeliveryPortal\4.4\Configurable\wars\sas.packageviewer\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Information Delivery Portal</td>
<td>SAS-installation-directory\SASInformationDeliveryPortal\4.4\Configurable\wars\sas.portal\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS BI Web Services</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.biws\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Preferences</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.preferences\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Shared Applications</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.shared.apps\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Stored Process</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.storedprocess\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Logon Manager</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.permissions\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Content Server</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.scs\WEB-INF\web.xml.orig</td>
</tr>
<tr>
<td>SAS Authorization Services</td>
<td>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.authorization.services\WEB-INF\web.xml.orig</td>
</tr>
</tbody>
</table>
## Configuring the HTTP Session Time-out Interval

<table>
<thead>
<tr>
<th>Web Application</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Web Infrastructure Platform</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.access\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>Client Access</td>
<td></td>
</tr>
<tr>
<td>SAS Identity Services</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.identity.services\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Web Administration Console</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.admin\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Notification Template Editor</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.templateeditor\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Principal Services</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.principal.services\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Web Infrastructure Platform</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.services\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>Platform Services</td>
<td></td>
</tr>
<tr>
<td>SAS SOAP Services</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.soapservices\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Workflow Web Service</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.workflow.webservice\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Workflow</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.workflow\WEB-INF\web.xml.orig</code></td>
</tr>
<tr>
<td>SAS Shared Web Assets</td>
<td><code>SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Static\wars\sasweb\WEB-INF\web.xml</code></td>
</tr>
<tr>
<td>SAS Web Report Studio</td>
<td><code>SAS-installation-directory\SASWebReportStudio\4.4\Configurable\wars\sas.webreportstudio\WEB-INF\web.vfabricctcsvr.xml.orig</code></td>
</tr>
</tbody>
</table>
Web Application | File Location
---|---
SAS Visual Analytics | `$SAS-installation-directory\SASVisualAnalyticsServices\6.3\Configurable\wars\sas.lasr.authorization\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsServices\6.3\Configurable\wars\sas.va.linkservices\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsServices\6.3\Configurable\wars\sas.bitransportservices\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualDataBuilder\6.3\Configurable\wars\sas.visualdatabuilder\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsHub\6.3\Configurable\wars\sas.visualanalyticshub\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsAdministrator\6.3\Configurable\wars\sas.adminclient\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SearchInterfacetoSASContent\3.2\configurable\wars\sas.searchsas\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsExplorer\6.3\Configurable\wars\sas.visualanalyticsexplorer\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsExplorer\6.3\Configurable\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsReportViewer\6.3\Configurable\WEB-INF\web.xml.orig`<br>`$SAS-installation-directory\SASVisualAnalyticsDesigner\6.3\Configurable\WEB-INF\web.xml.orig`<br>

* The session-config element described in Step 2 must be added to the web.xml.orig file for this application.

---

**Customizing the Sign-out URL**

You can customize the behavior of the Sign Out button in the SAS Logon Manager in order to integrate with various security scenarios, such as the Butler Group's CA SiteMinder Web Access Manager. You can do this by adding a property for changing the URL so that users are redirected after they sign out from a SAS web application. The new property is added to the configuration metadata for the SAS Application Infrastructure.

To add the new property, follow these steps:

1. Log on to SAS Management Console.
2. On the Plug-ins tab, navigate to Application Management ➤ Configuration Manager.
3. Right-click SAS Application Infrastructure and select Properties.
4. Click the Advanced tab, and then click Add.
5. Enter `Logoff.Url` in the Property Name field.
Enter the sign-out URL to which users should be redirected in the Property Value field.

Click OK to close the Define New Property dialog box.

Click OK to close the SAS Application Infrastructure Properties dialog box.

To enable these properties to take effect, restart SAS Web Application Server.

Changing the Banner Title

You can customize the default banner title for the SAS Logon Manager. To change the banner title, follow these steps:

1 Determine how many property files to edit.

The property files are localized, so there is one property file for each language that is supported. Typically, you need to edit only those files that match the languages needed by your users. In the SAS-installation-directory\SASWebInfrastructurePlatform\9.4\Static\wars \sas.svcs.logon\WEB-INF\classes directory, the title values are stored in the messages.properties and messages_locale.properties files.

2 Edit the appropriate files with a plain text editor.

a Search for the lines that begin with sas.browser.title and sas.page.title.

b Replace the values in the files with your desired title. For example, to change the banner and browser title to Custom sign in manager title, specify the following values:

sas.browser.title=Custom sign in manager title
sas.page.title=Custom sign in manager title

c Rebuild and redeploy the Web Infrastructure Platform web applications. For more information, see “Step 3: Rebuild and Redeploy SAS Web Infrastructure Platform” on page 103.

Note: The properties files might be overwritten by updates such as hot fixes or maintenance releases. If so, you must repeat the preceding steps.

Changing the Appearance of the Sign-in Page

In SAS 9.4, the appearance of the SAS Logon Manager sign-in page is not affected by customized themes that you create using SAS Web Application Themes. Instead of using a customized theme, follow these steps to change the appearance of the page:

1 To change fonts and background colors, edit the sas.css and sas_ie.css style sheet that is located in the following path:
To change the logo or other images:

a. Edit or replace the images in the following directory:

```
SAS-installation-directory/
SASWebInfrastructurePlatform/9.4/Static/wars/sas.svcs.logon/themes/default/images
```

b. Update the following file to point to the new or updated image files:

```
SAS-installation-directory/
SASWebInfrastructurePlatform/9.4/Static/wars/sas.svcs.logon/WEB-INF/classes/default-theme.properties
```

3. Use the SAS Deployment Manager to rebuild the SAS Web Infrastructure Platform web application. See “Rebuilding the SAS Web Applications” on page 86.

4. Use the SAS Deployment Manager to redeploy the SAS Web Infrastructure Platform web application. See “Redeploying the SAS Web Applications” on page 91.

5. Back up each new or changed image file, and keep a list of the changes that you made to the style sheet.

Note: Your changes might be overwritten by updates such as hot fixes or maintenance releases. If so, you must repeat the preceding steps. (Do not replace the updated sas.css with a backup copy of the modified style sheet, because the sas.css file might have been updated by the hot fix or maintenance release.)

---

### Configuring the Global Single Sign-On Time-out Interval

#### Understanding the Time-out Interval

The time-out interval for the global single sign-on is different from the HTTP session time-out interval that is set in the web.xml file for web applications. The default HTTP session time-out interval is 30 minutes. When it is met, the web application ends the HTTP session. However, the default value for the global single sign-on time-out interval is 12 hours. If the user accesses a timed-out web application within that interval, or any other SAS web application, a new HTTP session is created.

**TIP** This behavior can be changed so that reaching an HTTP session time-out causes the global single sign-on session to time-out as well. Set the Log user off on timeout policy is set to Yes. For more information, see Configuring Middle Tier Security Policies.

One area where the HTTP session time-out and global single sign-on time-out are similar is that they both are reset when a user accesses an application.
Considerations for Changing the Time-out Interval

The interval should be short enough to alleviate security concerns that the single sign-on session remains available for too long.

The interval must be long enough that users do not reach the time-out interval while they are using the application. If a user reaches the global single sign-on time-out interval, the user must provide credentials and reauthenticate.

Specifying a Different Time-out Interval

If you choose to use a different value than the default, 12 hours, then specify the number of milliseconds in the \-Dsas.tgt.expiration.period=\text{interval-in-milliseconds} JVM option.

You need to specify this option for the instances of SAS Web Application Server that are used for running SAS Logon Manager only.

**Windows Specifics:** Add the JVM option to the SASServer1_1\conf\wrapper.conf file.

**UNIX Specifics:** Add the JVM option to the SASServer1_1/bin/setenv.sh file.

Configuring Guest Access

About Guest Access

Starting in the May 2015 release of SAS 9.4, the guest access feature is available. Guest access is an optional feature that provides anonymous access to a subset of resources and functionality in some SAS web applications. For details about whether a particular SAS web application supports guest access, see the administration guide for that solution.

In guest access, there is no individualized authentication of the requesting user, so there are no requirements for individual user accounts or metadata identities. Instead, all guest users are authenticated as the same service account (the SAS Anonymous Web User). That service account functions as the single surrogate identity for all guest users.

Here are some key points:

- To disable guest access system-wide, use the Configuration Manager to set the Policy.DisallowGuestAccess property to true for SAS Logon Manager. For information about how to set the property, see “Summary of Steps for Using Configuration Manager” on page 57.

- In a deployment in which guest access is enabled, the Guest option is available on the SAS Logon Manager sign-in screen for applications that support guest access. Therefore, users can choose guest access when they sign in.

  **Note:** For SAS Visual Analytics, users can also explicitly specify a guest access URL. For example: http://hostname/SASVisualAnalyticsHub/guest.jsp.
Because all guest users connect as the same shared, surrogate identity (the SAS Anonymous Web User), all guest users see the same features and resources. All guest users can see what the SAS Anonymous Web User can see and perform the same functions as the SAS Anonymous Web User.

**TIP** To ensure that the resources and functionality that are available to guests remain as intended, periodically access SAS web applications as a guest.

To provide guest access within an intranet only, place the applications behind a firewall. For more information, see Chapter 16, “Best Practices for Configuring Your Middle Tier,” on page 187.

In a deployment that uses web authentication, additional middle-tier configuration is required to support guest access. For more information, see “Fallback to SAS Form-based Authentication” on page 240.

**Limiting Content for Guest Access**

Any content that the SAS Anonymous Web User can access is available to all guest users.

**CAUTION!** Grants to the SASUSERS and PUBLIC groups can introduce additional content. If your deployment supports guest access, it is important to review access that is granted to the SASUSERS and PUBLIC groups. The SAS Anonymous Web User is an implicit member of those groups, so any content that you make available to those groups is potentially available to the guest user.

Here are some guidelines for managing access:

- Do not expect user or group-based access distinctions (such as row-level security) for guests. Guest access provides only generic, lowest-common-denominator access to content.

- Review the metadata-layer permissions that are granted to the SASUSERS and PUBLIC groups. You can exclude content from guest access where access is granted to SASUSERS or PUBLIC and add denials for the SAS Anonymous Web User.

  For information about permissions, see "Metadata Authorization Model" in *SAS Intelligence Platform: Security Administration Guide*.

- Do not revoke the SAS Anonymous Web User’s ReadMetadata access to the `/System` folder.

  **Note:** When the SAS Anonymous Web User is defined, anyone logged on as that user will be allowed to transition only between applications that explicitly support guest logins (even if the user logged on as the SAS Anonymous Web User outside of the guest login functionality).

**Enabling or Disabling Guest Access**

**How to Enable Guest Access**

The preferred method for configuring guest access is to make the following choices during installation:

- Create a SAS Anonymous Web User (webanon).
Enable guest access for the SAS web application.

For more information about the webanon account, and creating the account after the initial installation has finished, see "Using the SAS Anonymous Web User with SAS Authentication" on page 254.

**How to Disable Guest Access System Wide**

On the SAS Logon Manager application, set the `App.DisallowGuestAccess` property to `true`.

**How to Disable Guest Access for a Single Application**

For an application that supports guest access, you can disable guest access by setting the `App.AllowGuest` property to `false`.

*Note:* If the `App.AllowGuest` property is not set, guest access is disabled.

---

### Configuring Middle-Tier Security Policies

The policies identified in the following table are configured with SAS Management Console. For more information, see “Setting Global Properties for SAS Applications” on page 59.

**Table 9.2  Middle-Tier Security Policies**

<table>
<thead>
<tr>
<th>Policy Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for metadata updates</td>
<td>Check on navigation</td>
<td>This is a deprecated property. Do not change the value unless you are directed to by SAS Technical Support.</td>
</tr>
<tr>
<td>Profile refresh interval</td>
<td>600000</td>
<td>This is a deprecated property. Do not change the value unless you are directed to by SAS Technical Support.</td>
</tr>
<tr>
<td>Allow client password storage</td>
<td>Yes</td>
<td>Indicates whether the site permits remote SAS clients to store user password credentials locally on the client. Many sites prohibit end-user clients from caching or persisting passwords for use in distributed applications.</td>
</tr>
<tr>
<td>Log user off on time-out</td>
<td>No</td>
<td>Determines how a time-out in one SAS web application affects a user’s global single sign-on session. When this value is set to <code>no</code>, a user can reach a time-out limit in one web application but still have a valid global single sign-on session and be able to use other web applications. When this value is set to <code>yes</code>, whenever any web application reaches a time-out limit, the global single sign-on session is ended and the user must reauthenticate to use a web application. Setting this value to <code>yes</code> reproduces the behavior provided in SAS 9.3 and earlier releases.</td>
</tr>
<tr>
<td>Policy Name</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allow user sign-in from web sign-out page</td>
<td>Yes</td>
<td>Determines whether to display a <strong>Sign In</strong> button on the sign-out successful page. Some sites, especially those that deploy walk-up kiosks, might want to ensure that their application users close the browser for added security.</td>
</tr>
<tr>
<td>Allow user sign-in from web time-out page</td>
<td>Yes</td>
<td>Determines whether to display a <strong>Sign In</strong> button on the session timed out page. Some sites, especially those that deploy walk-up kiosks, might want to ensure that their application users close the browser for added security.</td>
</tr>
<tr>
<td>Display custom sign-in message</td>
<td>No</td>
<td>Determines whether to display a custom message or custom page on the standard sign-in page.</td>
</tr>
<tr>
<td>Display custom sign-out message</td>
<td>No</td>
<td>Determines whether to display a custom message or custom page on the standard sign-out successful page.</td>
</tr>
<tr>
<td>Display custom time-out message</td>
<td>No</td>
<td>Determines whether to display a custom message or custom page on the standard session timed out page.</td>
</tr>
<tr>
<td>Display sign-out security message</td>
<td>Yes</td>
<td>Determines whether to display a security message on the sign-out successful page. Some sites, especially those that deploy walk-up kiosks, might want to ensure that their application users close the browser for added security. For more information about time-out values, see “Configuring the HTTP Session Time-out Interval” on page 105.</td>
</tr>
<tr>
<td>Display time-out security message</td>
<td>Yes</td>
<td>Determines whether to display a security message on the session timed out page. Some sites, especially those that deploy walk-up kiosks, might want to ensure that their application users close the browser for added security. For more information about time-out values, see “Configuring the HTTP Session Time-out Interval” on page 105.</td>
</tr>
<tr>
<td>Display failed sign-in hints</td>
<td>No</td>
<td>Determines whether to display detailed messages on the failed sign-in page (for example, to indicate that the password was invalid). If this policy is set to <strong>No</strong>, the system-generated exceptions and errors are still displayed, such as if the system is quiesced or if the SAS Metadata Server is paused. If the value is <strong>No</strong>, the only message that is displayed for any user input failure is the invalid credentials message.</td>
</tr>
</tbody>
</table>
Disabling Concurrent Sign-in Sessions

The default behavior for the SAS Logon Manager and the other SAS web applications is to permit multiple sign-in sessions. However, it is possible to configure an advanced middle-tier security policy to prevent multiple sign-in sessions. When this policy is active, users can sign in to one SAS web application at a time. When users use the Sign Out link that is provided in the application banner, the sign-in session is destroyed, and users can sign in to a SAS web application again.

You must specify the concurrent sign-in session behavior:

- **deny**: When you specify deny, the user receives a message from SAS Logon Manager that a session is already active. The user cannot sign in until the existing session expires or an administrator uses the SAS Web Administration Console to Force Sign Out the user.

- **logoff**: When you specify logoff, the existing session is logged off and the user is logged on to the requested web application.

To disable concurrent sign-in sessions, follow these steps:

1. Log on to SAS Management Console.
2. On the Plug-ins tab, select Application Management ➤ Configuration Manager, right-click SAS Application Infrastructure, and select Properties.
3. In the SAS Application Infrastructure Properties dialog box, click the Advanced tab.
4. Click Add to define a new property.
5. Enter Policy.ConcurrentUserLogins in the Property Name. Enter either deny or logoff in the Property Value field.
6. Click OK.

Settings are not applied and made active automatically. You must restart the SAS Web Infrastructure Platform Services or the web application server.

Disabling the SAS Trusted User Account

By default, the SAS Trusted User (sastrust) is allowed to sign in to SAS Logon Manager. If someone using the sastrust account exceeds the number of allowed password attempts, the account will be locked out and SAS products might become unstable. To disable the SAS Trusted User, specify the -Dsas.logon.disable.system.logins=true JVM option.

You need to specify this option for the instances of SAS Web Application Server that are used for running SAS Logon Manager only.
Disabling Cross-Frame Scripting

To disable the embedding of SAS Logon Manager in an iframe, complete the following steps:

1. Edit the `SAS-configuration-directory\Levnn\Web\WebAppServer\SASServern_m\sas_webapps\sas.svcs.logon.war\WEB-INF\spring-configuration\filters.xml` file and locate the following block:

```xml
<property name="headersToSet">
  <props>
    <prop key="X-UA-Compatible"><![CDATA[IE=EmulateIE8]]></prop>
  </props>
</property>
```

Replace the previous block with the following:

```xml
<property name="headersToSet">
  <props>
    <prop key="X-UA-Compatible"><![CDATA[IE=EmulateIE8]]></prop>
    <!-- For greater security the logon page can be blocked from being embedded in an iframe. -->
    <!-- This can be done by uncommenting one of the X-Frame-Options lines below -->
    <!-- See https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options for details-->
    <!-- <prop key="X-Frame-Options">DENY</prop> -->
    <!-- <prop key="X-Frame-Options">SAMEORIGIN</prop> -->
  </props>
</property>
```

2. From a web browser, navigate to `https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options` and decide which of the following two options is appropriate for your environment: DENY or SAMEORIGIN.

3. Uncomment the line that represents the option that you choose.

4. To enable your changes to take effect, restart SAS Web Application Server.
About the SAS Content Server

The SAS Content Server is a content repository that stores digital content (such as documents, reports, and images) created and used by SAS client applications. Examples of such content include reports and documents created by users of SAS Web Report Studio and the SAS Information Delivery Portal.

The Web Distributed Authoring and Versioning (WebDAV) protocol is currently the main method used to access the SAS Content Server. In addition to the basic features of HTTP, the WebDAV protocol is an extension to HTTP and provides Write access, version control, search, and other features.

The SAS Content Server is a web application and starts when the web application server is started.

The JVM options in the following table are related to the SAS Content Server deployment. In the event that the deployment of SAS Content Server changes, the JVM options can be used to set the new values.

<table>
<thead>
<tr>
<th>JVM Option</th>
<th>Description</th>
<th>Applies To</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dsas.scs.scheme</td>
<td>Specifies http or https.</td>
<td>Web application server</td>
</tr>
<tr>
<td>-Dsas.scs.host</td>
<td>Specifies the host name of the web application server.</td>
<td>Web application server</td>
</tr>
<tr>
<td>-Dsas.scs.port</td>
<td>Specifies the port number of the web application server instance.</td>
<td>Web application server</td>
</tr>
<tr>
<td>-Dsas.scs.repository.dir</td>
<td>Specifies the path to the repository. The specified directory is used for indexes and metadata. The digital content is stored in a relational database.</td>
<td>Web application server</td>
</tr>
<tr>
<td>-Dsas.scs.svc.internal.uri</td>
<td>Specifies the internal URI of the SAS Content Server.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td>-Dsas.scs.cas.scheme</td>
<td>Specifies http or https.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td>-Dsas.scs.cas.host</td>
<td>Specifies the host name of the Central Authentication Service (CAS) server.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td>JVM Option</td>
<td>Description</td>
<td>Applies To</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td><code>-Dsas.scs.cas.port</code></td>
<td>Specifies the port number of the CAS server.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This value should be specified as a string. For the first</td>
<td></td>
</tr>
<tr>
<td></td>
<td>maintenance release for SAS 9.4, if a value other than the default port</td>
<td></td>
</tr>
<tr>
<td></td>
<td>is being used, the value must be preceded by a colon. For example, the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>default port for HTTP is 80 and you specify it as: <code>-Dsas.scs.cas.port=&quot;&quot;</code>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specify a non-default port as: <code>-Dsas.scs.cas.port=&quot;:8125&quot;</code>. Beginning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>with the second maintenance release for SAS 9.4, both the default and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>non-default port should be specified as: <code>Dsas.scs.cas.port=port_number</code>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>-Dsas.scs.svc.scheme</code></td>
<td>Specifies http or https.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td><code>-Dsas.scs.svc.host</code></td>
<td>Specifies the host name of the web server.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td><code>-Dsas.scs.svc.port</code></td>
<td>Specifies the port number of the web server.</td>
<td>Central Authentication Service</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This value should be specified as a string. For the first</td>
<td></td>
</tr>
<tr>
<td></td>
<td>maintenance release for SAS 9.4, if a value other than the default port</td>
<td></td>
</tr>
<tr>
<td></td>
<td>is being used, the value must be proceeded by a colon. For example, the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>default port for HTTPS is 443 and you specify it as: `-Dsas.scs.cas.port=&quot;&quot;.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specify a non-default port as: <code>-Dsas.scs.cas.port=&quot;:9230&quot;</code>. Starting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>with the second maintenance release for SAS 9.4, both the default and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>non-default port should be specified as: <code>Dsas.scs.cas.port=port_number</code>.</td>
<td></td>
</tr>
</tbody>
</table>

For information about how to update the Central Authentication Service (CAS) options, see "Configuring the SAS Content Server to Use an Existing Customer Reverse Proxy " on page 137.
SAS Content Server Storage

The SAS Content Server uses a database for storage. SAS Content Server uses the same database that is used by the SAS Web Infrastructure Platform. The default configuration for the SAS Web Infrastructure Platform is to use the SharedServices database instance on the SAS Web Infrastructure Platform Data Server. However, the SAS Web Infrastructure Platform can be configured to use a third-party vendor database such as Oracle, MySQL, PostgreSQL, DB2, or SQL Server.

When a third-party vendor database is used, make sure that the database is configured to accept large binary objects such as documents and images. For example, on MySQL, the max_allowed_packet variable must be set at least as large as the largest binary object in the SAS Content Server repository.

Moving Content or Backing Up the SAS Content Server

The SAS Content Server should be backed up whenever the metadata server is backed up. For instructions about how to back up the SAS Content Server, see “Best Practices for Backing Up Your SAS System” in the SAS Intelligence Platform: System Administration Guide.

Use the WebDAVDump and WebDAVRestore utilities to perform the following tasks:

- Back up specific locations such as a subset of the WebDAV content.
- Create a backup for input to a system other than the SAS Content Server.
- Move content from one SAS Content Server to another one.
- Share content that is available in the SAS Content Server.

For instructions about using the WebDAVDump and WebDAVRestore utilities, see SAS Note 38667.

Preventing File Types from Being Uploaded

By default, any file type can be uploaded to the SAS Content Server. Starting with the second maintenance release for SAS 9.4, you can now prevent certain file extensions and MIME types from being uploaded to the SAS Content Server via the WebDAV utility. By disallowing certain attachment types from being uploaded, you can ensure that a file extension matches its contents and provide file scanning capabilities.

**CAUTION!** The parser might change the specified MIME type. Although application/bat might be specified, the parser might incorrectly determine that the file is text/plain. If this happens, the filter is not aware of the change in MIME type and allows the file to be uploaded.
To prevent certain file extension from being uploaded, follow these steps:

1. Edit the `SAS-configuration-directory\Leon\Web\WebAppServer\SAServer 1_1\sas_webapps\sas.svcs.scs.war\WEB-INF\config.xml` file.

2. Locate the following block:
   ```xml
   <iohandler>
   <class name="com.sas.contentserver.io.SASDefaultHandler" />
   <!-- param name="disallowedMimeTypes" value="application/exe, application/bat"/ -->
   <!-- param name="disallowedFileExtensions" value="exe,bat"/ -->
   </iohandler>
   
   Note: The highlighted types are provided as examples.
   
3. Remove the comment that encloses the parm name and specify the file extensions and MIME types that are applicable for your environment.

   Note: Separate the list of file extensions and MIME types with commas.

To enable these settings to take effect, restart SAS Web Application Server.

---

Deploying Content Manually to the SAS Content Server

Overview

SAS web applications such as the SAS Information Delivery Portal and SAS Web Report Studio require the availability of content for its users. The SAS Content Server provides a WebDAV content repository that stores digital content (such as documents, reports, and images) that is created and used by SAS client applications.

To enable the availability of the content in the SAS Content Server, you can load content, update existing content, and adjust web applications that store SBIP URLs. These tasks can be automated or they can be performed manually.

The following table shows the choices available in the SAS Deployment Wizard, and the results or manual tasks that follow these choices.

<table>
<thead>
<tr>
<th>Options Selected in SAS Deployment Wizard</th>
<th>Results and Instructions for Manual Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Web Server: Automated or Manual Configuration Option</td>
<td>SAS Web Server and SAS Web Application Server are configured automatically. SAS web applications are deployed automatically, and content is loaded to the SAS Content Server. If applicable, web applications that store SBIP URLs are adjusted automatically.</td>
</tr>
<tr>
<td>Web Applications: Automatic Deployment Deploy web applications automatically is selected</td>
<td></td>
</tr>
</tbody>
</table>
The following table shows when you can load or update content (and adjust URLs) either automatically or manually.

Table 10.3 Criteria for Deploying Content to the SAS Content Server

<table>
<thead>
<tr>
<th>Configuration of Web Application Server</th>
<th>Deployment of Web Applications</th>
<th>Load Content</th>
<th>Update Content</th>
<th>Adjust URLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>Automatic</td>
<td>Automatic</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

The following table shows the files associated with loading content to the SAS Content Server or updating content. The filename for the batch or script file includes the order number.

Security Considerations for SAS Content Server Scripts

The scripts that are described in this section for loading content, updating content, and adjusting URLs use the SAS Administrator and SAS Trusted User credentials. For deployments that performed a manual deployment of the SAS web applications, these scripts include the user IDs and an encoded form of the password. For deployments that performed an automatic deployment of the SAS web applications, the scripts include the user IDs, but do not include the passwords in any form.

Passwords in these files, whether added by the SAS Deployment Wizard, or by a SAS administrator, are not updated with the Update passwords feature of the...
SAS Deployment Manager. Running the scripts with an expired password, or no password, provides a log result like the following example:

**Output 10.1 Log File Example for Invalid Credentials**

```plaintext
config.init:
    [echo] ant.version=Apache Ant version 1.7.0 compiled on December 13 2006
    [echo] ant.file=/opt/SASHome/SASWebInfrastructurePlatform/9.4/Config/webinfpltfm_config.xml
    [echo] file.encoding=ISO646-US
    [echo] about to read property file because config.init.set=${config.init.set}
    [GetObjectProperties]    Host: hostname.example.com
    [GetObjectProperties]    Port: 8561
    [GetObjectProperties]    User: sasadm@saspw
    [GetObjectProperties]    m_mdFactory: com.sas.metadata.remote.MdFactoryImpl@74db2c
BUILD FAILED
/opt/SASHome/SASDeploymentManager/9.4/products/
/cfgwizard__nnnnn__prt__xx__sp0__1/Utilities/configuration_targets.xml:95: null
```

If you need to update or add a password, use the PWENCODE procedure. The following code example shows how to generate the encoded form of the password `changeit`. Copy and paste the result into the scripts.

**Example Code 10.1 PWENCODE Procedure Example**

```sas
proc pwencode in="changeit" method=sas002; run;
```

The SAS log shows the value to copy and paste into the script:

```
{SAS002}4DE4CF4F130AC6BE4A6934E0596C8222
```

After you run the scripts, remove the encoded form of the passwords from the scripts as an additional security measure.

**Load Content Manually to the SAS Content Server**

If you deploy SAS web applications manually, you need to load content manually to the SAS Content Server. For information about how to load content manually for SAS web applications, see your `Instructions.html` file.

Use the following batch file or shell script to load content manually:

- On Windows:
  ```plaintext
  SAS-configuration-directory\Lev1\Web\Utilities\manualLoadContent-OrderNumber.bat
  ```
- On UNIX:
  ```plaintext
  SAS-configuration-directory/Lev1/Web/Utilities/manualLoadContent.sh-OrderNumber.sh
  ```

If web applications were deployed manually, this script contains the credentials for the SAS Administrator, as well as the SAS Trusted User. The password is always encrypted in the file. After loading content successfully, remove credentials for the SAS Administrator and the SAS Trusted User.

If web applications were deployed automatically, the script does not contain the required credentials. You must manually enter the required credentials in this script file.
Update Content Manually for the SAS Content Server

If you deploy updated SAS web applications manually, you must manually update the DAV content in the SAS Content Server. For more information, see your `UpdateInstructions.html` file, which is located in the `SAS-configuration-directory/Lev1/Documents` directory.

You must update content manually before portal content is promoted to SAS Information Delivery Portal 4.4. In this case, data explorations must be converted to reports, and directive URLs should be adjusted manually. For more information, see “Promotion Exceptions and Variances” in SAS Intelligence Platform: Web Application Administration Guide.

Use the following batch file or shell script to update the DAV content manually:

- On Windows:
  ```
  SAS-configuration-directory\Lev1\Web\Utilities\manualUpdateContent-OrderNumber.bat
  ```

- On UNIX:
  ```
  SAS-configuration-directory/Lev1/Web/Utilities/manualUpdateContent-OrderNumber.sh
  ```

If web applications were deployed manually, this script contains the credentials for the SAS Administrator, as well as the SAS Trusted User. The password is always encrypted in the file. After loading content successfully, remove credentials for the SAS Administrator and the SAS Trusted User.

If web applications were deployed automatically, the script does not contain the required credentials. You must manually enter the required credentials in this script file.

Adjust Directive URLs Manually

Directive URLs are updated either during the migration of a product from one version to another version, or when a product's content is modified and updates are required. When the script is run to adjust URLs, it updates references to metadata that has moved either during migration or an upgrade. These references are stored as SBIP URLs.

You must update content manually before portal content is promoted to SAS Information Delivery Portal 4.4. In this case, data explorations must be converted to reports and directive URLs should be adjusted manually. For more information, see “Promote the Entire Portal Application Tree” in SAS Intelligence Platform: Web Application Administration Guide.

Here are some examples of instances that require adjusting URLs manually:

- When a migration is performed, some reports might be moved to a user’s home folder. If there were references to the data in those reports (in the form of SBIP URLs), then those references are updated by the script.

- During a migration or an upgrade, data explorations are converted to reports. If there were references to the data explorations (in the form of SBIP URLs), then those references are updated by the script.
After updating content manually for the SAS Content Server, adjust directive URLs manually by running the appropriate script or batch file:

- On Windows:
  
  `SAS-configuration-directory\Lev1\Web\Utilities\manualAdjustURLs-OrderNumber.bat`

- On UNIX:
  
  `SAS-configuration-directory/Lev1/Web/Utilities/manualAdjustURLs-OrderNumber.sh`

The instructions for running the script or batch file are provided in the `Instructions.html` migration or the `UpdateInstructions.html` file during an upgrade. The script contains the credentials for the SAS Administrator, as well as the SAS Trusted User. The password is always encrypted. When you have successfully loaded the content, remove the credentials for the SAS Administrator and the SAS Trusted User.

Log Files Generated by the Scripts

When any of the scripts in the previous sections are run, log files are produced for each SAS web application that is affected. Log messages are written to a file called `product-name_script-name_date-and-time.log`. For UNIX machines, the log filename always includes the date and timestamp. For Windows machines, the log filename includes the date and timestamp for machines that use an English locale only.

These log files are located in the following directories:

- On Windows:
  
  `SAS-configuration-directory\Lev1\Logs\Configure`

- On UNIX:
  
  `SAS-configuration-directory/Lev1/Logs/Configure`

Using the SAS Content Server Administration Console

About the SAS Content Server Administration Console

The SAS Content Server Administration Console enables you to manage files and WebDAV folders in the SAS Content Server. Using the console, you can perform the following management tasks:

- view folders
- control access to WebDAV folders and files by setting permissions
- create folders
- delete folders
Access the SAS Content Server Administration Console

To access the console, enter the following URL in your web browser and substitute the server name and port number of your SAS Content Server:

http://server:port/SASContentServer/dircontents.jsp

Note: This console is also part of the SAS Web Administration Console. You can administer the SAS Content Server by using either interface. For more information about accessing the SAS Web Administration Console, see “Using the SAS Web Administration Console” on page 73.

Log on to the console with an unrestricted user ID (for example, sasadm@saspw). In order to use the console, you must be logged on as an unrestricted user. This provides full administrator rights to use the console.

As a security precaution, make sure that you log off when you are finished using the console. If you go to another URL or close the tabbed page in your browser without logging off, your console logon remains in effect. This means that the console can be accessed again without re-entering a user name and password.

A Brief Tour of the Console Interface

The following display shows an example SAS Content Server Administration Console as it appears in a browser window:

Figure 10.1  SAS Content Server Administration Console

Objects in the console are either folders or files. By default, the initial view of the console displays the following folders:

sascontent
contains content that has been added to SAS Content Server by SAS applications. You see a folder only if the folder contains content.

sasdav
contains content that has been added to the SAS Content Server. By default, sasdav contains the following folders:

- sasdav/users contains personal repository folders for users. A user’s folder is created automatically when the user logs on to a SAS web application. Users have full rights to their own folders.
- sasdav/Templates contains templates that are used for email notification in SAS solutions.
sasfolders contains content that has been defined in the SAS Folders tree in the SAS Metadata Server. You see a folder only if the folder contains content.

**CAUTION! Administrators should not manage folders and content here.**
The content within this folder and subfolders is mapped to SAS Folders in the SAS Metadata Server. It is recommended that you use the SAS Management Console to add and manage folders.

Depending on the software that is installed at your site, your console might contain additional folders.

To navigate in the console, follow these steps:

1. Click an item in the list to display information about that item.
2. Use the breadcrumb trail above the list to return to a parent folder. For example, in the breadcrumb trail, click `sasav` to return to the sasav folder.

The console displays the following information for each item listed:

- **Item name**
  - displays the name of the folder or file.

- **Primary type**
  - is an internal value that designates the type of object in the repository.

- **Date created**
  - is the date on which the object was created.

- **Date modified**
  - is the date on which the object was modified.

- **Delete**
  - when the delete button is clicked, the selected objects are deleted.

- **Permissions**
  - when the permissions icon is clicked, opens a page where permissions can be modified for the object.

### Modify Permissions for WebDAV Folders and Files

The sasfolders directory should be accessed only by trusted or unrestricted users. These users are recognized as unrestricted administrators for the SAS Content Server, and do not require the Access Control List (ACL) to grant them access to this directory. If other types of users attempt to access this location, their permissions are verified before they are granted any access.

The sasdav directory can be accessed by regular users, and ACLs can be used to grant access to specific users and groups.

Principals can be granted permissions for folders and files. In the SAS Content Server, a principal is either a user or a group of users defined in the SAS Metadata Server. Principals can be given permissions that allow them to perform specific tasks such as reading an object, writing to an object, deleting an object, and so on.

You set permissions for an object by specifying which principals have which types of access. To modify permissions for an object, follow these steps:
1 Click the permission icon next to the item that you want to modify. A permissions page appears.

2 For each principal listed, modify the permissions by changing each permission to Yes or No.

Note: You might see a principal named jcr:authenticated. This principal refers to any user who can log on to a SAS web application. By default, authenticated users have Read and Inherit Read permissions only.

3 To add more principals to the page, do one of the following:
   - If you know the principal's name, enter it in the field and click Save changes.
   - Click Search for Principals to search for a name. When you find the principal that you want to add, select the check box next to the principal's name and then click Return.

After the principal's name appears on the permission page, you can set permissions for the principal.

The following display shows a portion of the console with permissions for a folder:

Figure 10.2 Folder Permissions in the SAS Content Server

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Allows the principal to read the object. For folders, this permission allows the principal to see the members of the folder.</td>
</tr>
<tr>
<td>Write</td>
<td>Allows the principal to write an object. For folders, this permission allows the principal to create new objects in a folder.</td>
</tr>
<tr>
<td>Delete</td>
<td>Allows the principal to delete the object.</td>
</tr>
</tbody>
</table>
Permissions | Purpose
--- | ---
Admin | Allows the principal to change the permissions on an object.
Inherit Read | Objects created in this folder inherit this setting for their Read permission (and Inherit Read permission for subfolders).
Inherit Write | Objects created in this folder inherit this setting for their Write permission (and Inherit Write permission for subfolders).
Inherit Delete | Objects created in this folder inherit this setting for their Delete permission (and Inherit Delete permission for subfolders).
Inherit Admin | Objects created in this folder inherit this setting for their Admin permission (and Inherit Admin permission for subfolders).

Note: Inherited permissions are assigned when objects are created. Each object has its own set of permissions. Inherited permissions are static; dynamic inheritance does not occur.

If you are applying permissions to folders, then the following options are available:

*Table 10.5  Results of Applying Permissions to Folders*

<table>
<thead>
<tr>
<th>Permissions for Folders</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subfolders and files</td>
<td>Changed permissions are applied to subfolders and files that exist below the current folder.</td>
</tr>
<tr>
<td>This folder only</td>
<td>Changed permissions are applied to subfolders and files that exist in the current folder.</td>
</tr>
<tr>
<td>Overwrite permissions for all</td>
<td>Changed permissions are applied to all folders and files.</td>
</tr>
</tbody>
</table>

**Create a New Folder**

To add a folder below the current folder, enter the name of the new folder in the field and click **Add Folder**.

*Note:* Although you can add a folder to the `sasfolders` location, the folder that you add is not added to the SAS Metadata Server. The best practice is to add folders to metadata using SAS Management Console.
Add Files to the SAS Content Server

You cannot use the SAS Content Server Administration Console to add files to folders. To add files, you can use one of the following methods:

- Use Microsoft web folders to add content to the appropriate folder. You must use a browser on a Windows client machine in order to use this method.
  
  For example, the sasdemo user might open the following location as a web folder:

  http://myServer/SASContentServer/repository/default/sasdav/Users/sasdemo/

  Then, copy and paste content into the folder.

- Use the SAS DAVTree utility to drag and drop folders or files into console folders.
  
  To use this utility, run the following command:

  SAS-configuration-directory\Left\Web\Utilities\DAVTree.bat

  On UNIX, the utility command is DAVTree.sh.

  For more information about using DAVTree, see “Using the DAVTree Utility to Manage WebDAV Content” on page 273.

- Use the SAS Publishing Framework to publish files to the WebDAV repository.
  
  Portal users can publish portal content to the WebDAV repository by using the portal’s publish and subscribe tools.

- Programmatically publish content to WebDAV.

  Usage of these tools and techniques is beyond the scope of this documentation (with the exception of the DAVTree utility).

Delete Folders or Files

Delete a single or multiple folders when you are sure that the folders and their contents are not required.

**CAUTION!** Exercise caution when deleting items from the SAS Content Server.

When deleting folders, the following rules apply:

- Do not delete the sasdav or sasfolders directories.

- If you delete an item in the sasfolders tree, then applications that rely on the content mapping between the SAS Content Server and the SAS Metadata Server might not be able to access the content. To add and delete SAS metadata objects, use SAS Management Console.

  For information about the best practices to follow for managing SAS folders in SAS Management Console, see “Working With SAS Folders” in the SAS Intelligence Platform: System Administration Guide.

- When you delete a folder, all objects within that folder are also deleted.

  To delete a folder or file, select the check box for the folder or file from the **Delete** column. Click the **Delete** button. The item is deleted. You are not
prompted to confirm the deletion. To delete multiple items, select multiple check boxes from the Delete column.

Enabling the Data Store

About the Data Store

Beginning in the second maintenance release for SAS 9.4, SAS Content Server supports the data store, an append-only storage mechanism for large files. The data store offers a number of advantages over traditional persistence manager storage, including:

- Space saving; only one copy per unique object is kept
- Fast copy; only the identifier is copied
- Storing and reading does not block others
- Objects in the data store are immutable
- Supports larger file storage

One drawback of the data store is that garbage collection must be periodically run to purge unused objects. In addition, all cluster nodes use the same data store, so a shared network location must be available to all cluster nodes.

Configuring the Data Store

The data store is not configured by default. To use the data store, manual configuration is required after the SAS configuration process completes. To manually configure the data store, complete the following steps:

1. Shut down any running web application servers and open the following file on each cluster node: 
   `SAS-configuration-directory\Lev\n\AppData\SASContentServer\SASServer1_m\Repository\repository.xml`

   Note: The SASServer1_m directory might not be present in your configuration.

2. At the end of the file, locate the following element and uncomment the element:

   ```
   <!-- DataStore class="com.sas.contentserver.core.data.TenantFileDataStore"> 
   <param name="path" value="${rep.home}/data/datastore"/>
   <param name="minRecordLength" value="1024"/>
   </DataStore -->
   ```

3. Replace the value of the path parameter to the location where you want the data store to store the data. If you are running in a clustered environment, a shared network location must be provided. To verify whether the SAS Content Server is configured to run in a cluster, complete the following steps:

   a. Depending on your operating system, open the following file:

      - On Windows: Open the `SAS-configuration-directory\Lev\n\WebAppServer\SASServer1_1\conf\wrapper.conf` file.
a On UNIX: Open the `SAS-configuration-directory/Lev/n/Web/WebAppServer/SASServer1_1/bin/setEnv.sh` file.

b Search for the `com.sas.server.isclustered` property.

c If the property is set to true, the system is running as a cluster node.

**CAUTION!** A shared network location must be supplied for cluster environments. Failure to supply a shared network location on each cluster node in the environment can result in an inconsistent repository that does not function properly.

4 Restart all web application servers. The SAS Content Server will now store all files larger than the specified size (default is 1024 bytes) in the data store.

**Using the Garbage Collection Utility**

Once the data store is configured, it will be necessary to periodically run the Garbage Collection utility to remove any files that are no longer referenced. To initiate the garbage collection process, issue an HTTP POST request to the following URL: `http(s)://webserver-host:webserver-port/SASContentServer/admin/collectDataStoreGarbage`.

When the garbage collection process is finished, the response will contain the status code, either 200(OK) or 500(Internal Server Error). In addition, the body of the response will contain a message indicating how many files were removed.

You can use the Postman utility (or a similar HTTP request generator tool) to send POST requests and the necessary redirects for logging in. For more information, see [https://www.getpostman.com/](https://www.getpostman.com/). To automate the process, you can use the Curl utility or a similar utility to send a POST request to the URL above.

The POST request URL is protected by the Central Authentication Server (CAS). To access the POST request, you must first obtain a ticket from CAS with administrative credentials and append it to the URL with the `ticket` parameter name. Complete the following steps:

1. Get a ticket granting ticket (TGT) by sending a POST request to `http(s)://host:port/SASLogon/v1/tickets` with the Content-Type header set to `text/plain` and a body containing the following:
   ```
   username=userid&password=password.
   ```
   Each of these values should be individually URL encoded before sending. Do not encode the entire string:
   ```
curl -H Content-Type:text/plain -d 'username=userid&password=password' -X POST http(s)://host:port/SASLogon/v1/tickets -D-
   ```
   Upon successful completion, the following headers are returned:
   ```
   201 Created
   Location: https://host:port/SASLogon/v1/tickets/
   TGT-18-umUeNL4yUkWHE62VdtKxZ15mFzatga43KNNe3niguULWuUXl1A-K-cas
   ```

2. After obtaining the TGT location, get a service ticket for the Garbage Collection utility by POSTing to the location returned in the previous step:
   ```
curl -H Content-Type:text/plain -d 'service=http%3A%2F%2Fhost%3Aport%2FSASContentServer%2Fadmin%2FcollectDataStoreGarbage'
   ```
Perform a POST to the Garbage Collection utility with the ticket obtained in the previous step, appended to the end of the URL with the ticket=\textit{ticket} parameter:

\texttt{curl -X POST http(s)://webserver-host:webserver-port/SASContentServer/admin/collectDataStoreGarbage?ticket=ticket}

### Implementing Authorization for the SAS Content Server

#### Overview of SAS Content Server Authorization

SAS users and groups are defined in a SAS Metadata Repository. The SAS Web Administration Console enables you to specify which users or groups are authorized to access specific folders in the SAS Content Server repository. In addition, you can specify what type of access permissions they have for the folders.

Use the SAS Web Administration Console to create folders and associate access controls with the folders.

\textbf{Note:} This topic does not describe authentication for the SAS Content Server. By default, SAS Content Server users are authenticated by using SAS token authentication.

Before you can associate access controls with a folder, you must complete these tasks:

1. Use the SAS Web Administration Console to create the folder on the SAS Content Server.

2. Ensure that the appropriate user and group definitions exist on the SAS Metadata Server for the SAS Content Server users and groups for whom you want to control access to the folder.

After you have created the WebDAV folders and have ensured that the appropriate user and group definitions are created on the SAS Metadata Server, use SAS Web Administration Console to associate access controls with the folders.

#### Example Scenario: SAS Content Server Authorization

Within your portal implementation, you might use the publish and subscribe capabilities to publish (write) and subscribe to (read) group folders on a WebDAV publication channel.

The following scenario shows the application's publish and subscribe setup for sales and executive teams that need different access to read (subscribe to) and write (publish) information that is stored in three different directories on the SAS
Content Server. On the SAS Metadata Server, these teams are represented by two groups, Americas Sales and Sales Executives.

This publish and subscribe scenario has a requirement for three different content areas, or group folders, on the SAS Content Server:

- **Catalog Sales**: The `/sasdav/Catalog Sales` directory contains catalog sales information. The Americas Sales and Sales Executives groups can both read (subscribe to) and write (publish) information.

- **Field Sales**: The `/sasdav/Field Sales` directory contains direct sales information. The Americas Sales and Sales Executives groups can both read, but only the Sales Executives group can write information.

- **Sales Execs**: The `/sasdav/Sales Execs` directory contains executive-level sales information. Only the Sales Executives group can read and write information.

The following table summarizes this scenario's group-based folders on the SAS Content Server, and the permissions for each group:

<table>
<thead>
<tr>
<th>Folder</th>
<th>Americas Sales</th>
<th>Sales Executives</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/sasdav/Catalog Sales</code></td>
<td>Read, Write</td>
<td>Read, Write</td>
</tr>
<tr>
<td><code>/sasdav/Field Sales</code></td>
<td>Read</td>
<td>Read, Write</td>
</tr>
<tr>
<td><code>/sasdav/Sales Execs</code></td>
<td>(none)</td>
<td>Read, Write</td>
</tr>
</tbody>
</table>

To create this sample configuration, follow these steps:

1. In SAS Management Console, define the users, groups, and login credentials that need to access the SAS Content Server. When you define login credentials, you must specify the same authentication domain name that you specified for the SAS Content server during installation.

   For this example, the following users, groups, and logins are defined:

<table>
<thead>
<tr>
<th>Group Metadata Identities</th>
<th>User Metadata Identities</th>
<th>User ID</th>
<th>Authentication Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>America Sales</td>
<td>salesusr1</td>
<td>salesusr1</td>
<td>DefaultAuth</td>
</tr>
<tr>
<td>Sales Executives</td>
<td>execusr1</td>
<td>execusr1</td>
<td>DefaultAuth</td>
</tr>
<tr>
<td>SAS Trusted User</td>
<td>sastrust</td>
<td>sastrust</td>
<td>DefaultAuth</td>
</tr>
</tbody>
</table>

   For example, the America Sales group contains a user named salesusr1 as a member, and salesusr1 has an associated login with a user ID of salesusr1 and an authentication domain of DefaultAuth. The America Sales group might include other members as well.
2 In the SAS Web Administration Console, create your new directory under the sasdav directory. For this example, navigate to the sasdav directory, and then create these three subdirectories: Catalog Sales, Field Sales, and Sales Execs.

3 In the SAS Web Administration Console, configure the access permissions for the folders that you created. For this example, set the access permissions for each subdirectory, using the following tables as guides:

Table 10.8  WebDAV Permissions for /sasdav/Catalog Sales

<table>
<thead>
<tr>
<th>Group</th>
<th>Read</th>
<th>Write</th>
<th>Delete</th>
<th>Inherit Read</th>
<th>Inherit Write</th>
<th>Inherit Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas Sales</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sales Executives</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 10.9  WebDAV Permissions for /sasdav/Field Sales

<table>
<thead>
<tr>
<th>Group</th>
<th>Read</th>
<th>Write</th>
<th>Delete</th>
<th>Inherit Read</th>
<th>Inherit Write</th>
<th>Inherit Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas Sales</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sales Executives</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 10.10  WebDAV Permissions for /sasdav/Sales Execs

<table>
<thead>
<tr>
<th>Group</th>
<th>Read</th>
<th>Write</th>
<th>Delete</th>
<th>Inherit Read</th>
<th>Inherit Write</th>
<th>Inherit Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Americas Sales</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sales Executives</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Manual Configuration Tasks

When Do I Need to Perform These Tasks?

Whenever there is a change that affects how applications access SAS Content Server, the connection information related to the server might need to be updated. Two common changes that require that you update the information are as follows:
changing from HTTP to HTTPS manually. (When HTTPS is configured with the SAS Deployment Wizard, the wizard sets all the connections automatically.)

adding a proxy to the network, such as Apache HTTP Server, IBM Tivoli Access Manager WebSEAL, or CA SiteMinder.

Reconfiguring the WebDAV Repository URL

In a SAS Enterprise Business Intelligence deployment, the following applications use an information service to retrieve the repository connection information from metadata:

- Platform Local Services
- Remote Services
- SASBIPortletsversion Local Services
- SASJSR168RemotePortletversion Local Services
- SASLASRAuthorizationversion Local Services
- SASPackageViewerversion Local Services
- SASPortalversion Local Services
- SASStoredProcessversion Local Services
- SASStudioMidTierversion Local Services
- SASVisualAnalyticsTransportversion Local Services

Note: Your deployment can include additional applications that need to be reconfigured.

You need to perform this task after the following changes:

- You reconfigured SAS Web Server from HTTP to HTTPS manually.
- You altered the network topology for high availability by adding a load balancer or reverse proxy.

To reconfigure the WebDAV URL for the applications, perform the following steps in SAS Management Console:

1. Select Environment Management » Foundation Services Manager.
2. Select the application and then select Core » Information Service.
3. Right-click Information Service and select Properties.
4. In the Information Service Properties dialog box, click the Service Configuration tab and then click Configuration.
5. In the Information Service Configuration dialog box, click the Repositories tab.
6. Select WebDAV and then click Edit.
7. Change the connection information. See the following list for common changes:
   - If you added a proxy or load balancer to the network to provide high availability, specify the connection information for the proxy.
If you configured SAS Web Server manually for HTTPS, enter the HTTPS port and select the **Secure** check box.

8. Click **OK** to close the Information Service Configuration dialog box.

9. Click **OK** to close the Information Service Properties dialog box.

### Reconfiguring the Server Connection

Use the Server Manager plug-in in SAS Management Console to reconfigure the connection information for SAS Content Server.

You need to perform this task after the following changes:

- You reconfigured SAS Web Server from HTTP to HTTPS manually.
- You altered the network topology by adding a load balancer or existing customer-supplied reverse proxy.

To reconfigure the server connection, perform the following steps in SAS Management Console:

1. Select **Environment Management** ➤ **Server Manager** ➤ **SAS Content Server**

2. In the right-hand pane, right-click the connection icon, and select **Properties**.

3. Click **Options**, modify the connection parameters, and click **OK**.

4. Select the **Folders** tab.

5. Select **SAS Folders**, right-click, and select **Properties**.

6. Click the **Content Mapping** tab and select **SAS Content Server** from the **Server** menu. Click **OK**.

7. Click **Yes** to confirm that you want to change the content mapping options.

### Configuring the SAS Content Server to Use an Existing Customer Reverse Proxy

For a network topology or protocol change, the SAS Content Server web application must also be updated with information about the connection point that is accessed with a web browser. For both changes, you need to configure the SAS Content Server JVM options to override the values in the configuration files.

Here are example JVM options for a standard SAS installation with SAS Web Server on port 7980 and an existing customer reverse proxy on port 443:

- `-Dsas.scs.cas.scheme=https`
- `-Dsas.scs.cas.host=proxy.example.com`
- `-Dsas.scs.cas.port=443`
- `-Dsas.scs.svc.scheme=https`
- `-Dsas.scs.svc.host=proxy.example.com`
- `-Dsas.scs.svc.port=443`
- `-Dsas.scs.svc.internal.url=http://example.org:7980`
- `-Dsas.retry.internal.url=true`
**Note:** The option `-Dsas.retry.internal.url=true` is needed to support authentication against both the external and internal URIs.

**Note:** For installations that are not current with the second maintenance release for SAS 9.4 or later, you must use the format `":port-number"` for non-default ports (for example, `-Dsas.scs.cas.port=":8443"`) or an empty string for default ports (for example, `-Dsas.scs.cas.port=""`).

For more information, see “Specifying JVM Options” on page 38 and the list of options in “About the SAS Content Server” on page 118.
Overview of SAS BI Web Services

A web service is an interface that enables communication between distributed applications. Web services enable cross-platform integration by enabling applications that are written in various programming languages to communicate by using a standard web-based protocol, typically the Simple Object Access Protocol (SOAP) or Representational State Transfer (REST). This functionality makes it possible for businesses to bridge the gaps between different applications and systems.

The following list identifies key changes that were introduced with SAS 9.3 and are still applicable for SAS 9.4:

- SAS BI Web Services is supported only in a Java application server deployment. Previously, in SAS 9.2, there were two implementations of SAS BI Web Services: one written in Java that requires a servlet container, and another written in C# that uses the .NET framework.
- Artifacts are not required to be generated in SAS 9.3. Only the metadata that is associated with the generated web service is published.
- All stored processes are presented as web services without the need for any additional processing. If the metadata for a web service is not required to be published to the SAS Metadata Server, the additional step to generate the metadata is no longer required.
Managing Generated Web Services

You can select a set of stored processes in SAS Management Console and use the Web Service Maker to deploy them as web services. The Web Service Maker generates a new web service that contains one operation for each stored process that you selected. For information about developing web services, see the *SAS BI Web Services: Developer's Guide*. For information about using the Deploy as Web Service wizard in SAS Management Console, see the product Help.

When you generate a web service, the Web Service Maker publishes metadata about the new web service to the SAS Metadata Server. The metadata includes information such as the URL of the web service, keywords, and the stored processes are used by the web service. You can view and update some of this information by using SAS Management Console and the Configuration Manager plug-in in. To import or export a generated web service, use the SAS Management Console folder view.

To delete a web service that was generated by the Web Service Maker, use SAS Management Console. Navigate to **Application Management** ▶ **Configuration Manager** ▶ **SAS Application Infrastructure** ▶ **BI Web Services for Java 9.4** ▶ **WebServiceMaker**. Expand the node, right-click the web service, and select **Delete**. Deleting a web service removes the metadata that is associated with the service. This action cannot be reversed.

**Note:** You must grant permissions on the `/System/Services` folder to users who want to create SAS BI Web Services. You can also delete a web service directly from the `/System/Services` folder. Users need ReadMetadata and WriteMemberMetadata to create and delete web services. By default, a group named **BI Web Services Users** has these permissions. You can add users to this group to enable them to create and delete web services, or use your own groups and permission settings.

Configuring SAS BI Web Services for Java

SAS BI Web Services for Java is initially configured during installation using the SAS Deployment Wizard. To modify this initial configuration, use the Configuration Manager plug-in for SAS Management Console.

To modify common configuration properties that apply to XMLA, WebServiceMaker, and generated web services, go to SAS Management Console. Navigate to **Application Management** ▶ **Configuration Manager** ▶ **SAS Application Infrastructure** ▶ **BI Web Services for Java 9.4**. Right-click to select **Properties** and click the **Settings** tab.
In the **Application » General Configuration** section, you can modify the following configuration properties:

**Acceptable SYSCC List**
When a web service operation is invoked, it in turn calls the appropriate SAS Stored Process running on the server tier. SAS execution always returns the SYSCC macro variable upon completion. By default, if this completion code is not 0, a SOAP fault is generated and returned to the invoking client. Alternatively, a comma-separated list of acceptable SAS completion codes can be specified to alter this behavior. Also, a hyphen separating two values can be used to specify a range of acceptable completion codes. In this case, the acceptable list of completion codes are treated as warnings rather than errors and do not cause a SOAP fault.

Note that SYSCC can be set directly by SAS code developers. Likewise, some SAS procedures set this value. See the appropriate SAS documentation to determine possible values that might be returned and whether these values are errors or warnings. For example, if a SAS procedure states that a SYSCC value less than 4 is a warning and you are willing to accept those values, set this property as follows: 0-4. Therefore, if the SAS stored process returns a value of 4 or less, it is considered successful as far as the web service is concerned and the client receives an appropriate response rather than a fault.

**Enable dynamic prompts validation**
When invoking web service operations for stored processes that have been configured with dynamic prompt data parameters, you can turn off validation to obtain better throughput if you are certain that these stored processes have been written in a robust manner to handle any possible data passed by clients. Dynamic prompt validation is enabled by default so that the middle-tier web service validates the client data against data providers to ensure that incoming data meets the specified criteria before calling the appropriate stored process on the server.

**SAS Stored Process timeout**
Set this property if you want to limit the amount of time that a stored process is allowed to run. If the stored process fails to execute in the specified time, it is canceled and a SOAP fault is returned to the invoking client. A value of zero indicates no time-out period.

**Enable allowing anonymous execution**
Specify whether you want to enable or disable anonymous execution.

To modify configuration properties that are specific to the Web Service Maker, navigate to the **WebServiceMaker** folder. Then, navigate to the **Settings** tab within the Properties dialog box.

**Base namespace**
This property is the base namespace that is concatenated with the service name to create a target namespace to uniquely identify generated web services. For example, if the base namespace is set to `http://tempuri.org`, and a client creates a new service named `test` without specifying an overriding namespace for this new service, then the target namespace for the web service becomes `http://tempuri.org/test`.

**Attachment conformance**
Specifies the attachment conformance that should be enabled for generated web services. There are two options: Message Transmission Optimization Mechanism (MTOM) and SOAP Messages with Attachments (SwA). The default is MTOM.
Validate Request With Schema
Setting this property to True causes the incoming request to be validated against the service’s schema. The default is false because this operation can be CPU intensive.

Validate Response With Schema
Setting this property to True causes the resulting output created by the service execution to be validated against the service’s schema. The default is false because this operation can be CPU intensive.

Attachment Optimized Threshold
The default value is 2048 bytes. This attachment threshold is the number of bytes contained in the attachment that causes the data to be included as an out-of-band XOP/Include MTOM attachment. An attachment containing fewer bytes is transferred inline as base64 encoding for optimization.

To modify configuration properties that are specific to a web service, navigate to the folder for that service. Then navigate to the Advanced tab within the Properties dialog box. Specify the name of each configuration property and its value in the Define New Property dialog box.

The following advanced configuration properties are available:

AcceptSysccList
See “Acceptable SYSCC List” on page 141. This property overrides its analogous common configuration property.

DynamicPromptsSupport
See “Enable dynamic prompts validation” on page 141. This property overrides its analogous common configuration property.

MaxSTPExecTime
See “SAS Stored Process timeout” on page 141. This property overrides its analogous common configuration property.

AnonymousExecution
Enabled by default. This property requires the SAS Anonymous Web User or Webanon account to have been created previously.

BaseNameSpace
This property is the base namespace that is concatenated with the service name to create a target namespace to uniquely identify web services. For example, if the base namespace is set to http://tempuri.org, and a client creates a new service named test without specifying an overriding namespace for this new service, then the target namespace for this web service becomes http://tempuri.org/test.

AttachmentConformance
This property specifies the attachment conformance that should be enabled for generated web services. There are two options: Message Transmission Optimization Mechanism (MTOM) and SOAP Messages with Attachments (SwA). The default is MTOM.

ValidateRequestWithSchema
Setting this property to true causes the incoming request to be validated against the service’s schema. The default is false, because this operation can be CPU intensive.

ValidateResponseWithSchema
Setting this property to true causes the resulting output that is created by the service execution to be validated against the service’s schema. The default is false because this operation can be CPU intensive.
AttachmentOptimizedThreshold

The default is 2048 bytes. This attachment threshold is the number of bytes contained in the attachment that causes the data to be included as an out-of-band XOP/Include MTOM attachment. An attachment containing fewer bytes is used as base 64 encoding for optimization.

Changes to properties do not take effect immediately. To apply these changes, perform one of the following tasks:

- Either stop and restart SAS Web Application Server, or stop and restart the SAS BI Web Services for Java Web application (sas.wip.services9.4.ear).
- Use a Java Management Extensions (JMX) console to communicate with the com.sas.svcs:service=biws,type=ConfigMBean management bean.

The following image shows the use of the JMX console bundled with the JDK to reload the configuration metadata into a running SAS BI Web Services for Java application:
Overview of Security for Web Services

A default installation of SAS BI Web Services for Java is not highly secure. The default security mechanism for SAS web applications is SAS authentication. All requests and responses are sent as clear text. If users want to authenticate as a specific user, then they can send a user name and password as clear text as part of the WS-Security headers. If you use a RESTful request, send the user name and password in a base64 encoded Authorization HTTP header. Authentication is performed by authenticating client credentials at the SAS Metadata Server. Whenever user names and passwords must be sent as clear text or base64 encoded, Transport Layer Security (TLS) should be enabled to provide transport layer security.

If you want to use HTTPS to secure the transmission of credentials with the web services, and you also want to use the Deploy as Web Service wizard in SAS Management Console, then you need to import the server certificate to SAS Management Console. To import the server certificate to SAS Management Console, follow these steps:

1. Create a Java keystore on the local machine and import the server certificate of the server that you want to communicate with. For more information about how to perform this step, see http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/keytool.html.

2. Pass the keystore location and password into SAS Management Console using JVM options. The options that need to be set are:

   javax.net.ssl.trustStore= "fully qualified path to keystore created with keytool from step 1"
   javax.net.ssl.trustStorePassword= "trust store password"

   To complete this step, add the following JavaArgs arguments to the sasmc.ini file, which is found at C:\Program Files\SASHome\SASManagementConsole\9.4:

   JavaArgs_14=-Djavax.net.ssl.trustStore = "fully qualified path to keystore created with keytool from step 1"
   JavaArgs_15=-Djavax.net.ssl.trustStorePassword = "trust store password"

If you are using XMLA web services or generated web services, an anonymous user can be configured. The anonymous web user is configured during SAS Deployment Wizard configuration. Anonymous users cannot use the Web Service Maker; credentials must always be provided to use the Web Service Maker. If you are using XMLA web services, you can pass user credentials as XMLA properties in the payload.

SAS BI Web Services can also be secured by configuring web authentication. This provides a way for SAS BI Web Services to identify the calling user with basic web authentication that uses HTTP transport layer security.

Note: Web authentication can be used with both XMLA web services and generated web services. Web authentication cannot be used with theWebServiceMaker web service when SAS clients are used because these clients authenticate by using one-time passwords.
Securing SAS BI Web Services for Java

SAS Authentication

When SAS authentication is used, SAS Web Application Server does not perform any authentication on behalf of the application. Instead, SAS BI Web Services for Java authenticates client credentials against the SAS Metadata Server. Client credentials are obtained by one of the following ways (in this order):

1. Use credentials that are passed in the UsernameToken WS-Security SOAP header. For RESTful invocation, use the credentials passed in the Authorization HTTP header.
2. Use credentials that are passed in the payload as properties (XMLA only).
3. Use anonymous credentials that are configured with the Webanon SAS metadata login account (XMLA and generated web services).

Typically, the WebServiceMaker service is invoked via the Deploy As Web Service wizard in SAS Management Console. Therefore, this service must be able to process SAS one-time passwords. For this reason the WebServiceMaker service functions only in SAS authentication mode.

Web Authentication

As an alternative to SAS authentication, SAS Web Application Server can be configured to perform the authentication on behalf of the SAS BI Web Services for Java application. This is known as web authentication. Beginning with SAS 9.3, web authentication can also be used with RESTful web services.

Editing the web.xml File for Third-Party Authentication

If you configure third-party authentication with products such as CA SiteMinder, and use the JavaScript Objects Notation (JSON) and REST web services, edit the deployment descriptor file. This file is located in the SASHOME\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.biws\WEB-INF directory. Change the configuration section in the web.xml.orig file as follows:

```xml
<!-- comment out or remove this filter
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-class>
        org.springframework.web.filter.DelegatingFilterProxy
    </filter-class>
</filter>
-->
```
Transport Layer Security

HTTP transport layer security can be used instead of message-level security. The following security constraints should be applied to the web.xml.orig deployment descriptor. (See the previous section for the location.) Change the file by adding the security constraints as follows:

```xml
<security-constraint>
  <web-resource-collection>
    <web-resource-name>All-resources</web-resource-name>
    <url-pattern>/services/XMLA/*</url-pattern>
    <url-pattern>/services/dynamicServicePath/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>*</role-name>
  </auth-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>
</login-config>
```
Overview

Introduction to SAS Web Application Themes

SAS Web Application Themes provide a way to define a consistent look and feel across SAS web applications. You can use themes to apply uniform visual customizations and company branding to all SAS web applications that support the theme infrastructure. A typical custom theme might include a banner with a standard corporate color scheme and company logo, a navigation bar with colors that coordinate with the banner, and new colors for borders and title bars.
Note: Custom themes do not affect the appearance of the SAS Logon Manager sign-in page. See “Changing the Appearance of the Sign-in Page” on page 109.

Theme Components

A theme is a collection of resources that control the appearance of a SAS web application. The following figure shows the components of a theme:

**Figure 12.1  Components of a Theme**

Here is an explanation of each theme component:

- **theme templates** are HTML fragments that render specific portions of pages in SAS web applications. The templates contain dynamic substitution variables of the form `%VARIABLE-NAME` that are replaced by application-specific values when the templates are used in SAS web applications.

- **cascading style sheets** determine the colors, fonts, backgrounds, alignment, and spacing for page elements in SAS web applications. A cascading style sheet (CSS) is a standard mechanism for defining consistent and reusable presentation for web-based content.

- **theme descriptors** are XML files that describe the style sheets, templates, and images that make up a theme.

- **images** include graphics for icons, a company logo, and banner and page backgrounds. You can incorporate your own customized graphics files as part of a new theme. Images can be in any format supported in the browser, including GIF, PNG, and JPEG.

**Note:** The application title that appears in the banner of the SAS web application is not part of the theme. You also cannot use themes to change the application name that appears in the title bar of the browser window.

The SAS Default Theme

The initial theme that is installed with the theme infrastructure is named Default. This theme is typically used as the basis for creating new themes, so you should understand its structure before you attempt to create a custom theme.
Specifications for the Default theme are provided in \SAS-configuration\directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\index.html.

How Custom Themes Are Created and Deployed

The SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions directory contains the scripts and resources needed to create a new theme:

- The NewTheme script creates a directory structure for your new theme, and populates it with configuration files that are modified to create a new theme definition. The new theme is based on the SAS default theme that is shipped with the software.
- The specs directory provides documentation for the general color palette and color and image guidelines that are specific to each user interface component. This document is useful when you are designing and defining your custom theme.

Developing a custom theme involves creating CSS files, image files, theme template files, and theme descriptor files. It is possible to create a new theme by authoring these files from scratch, but the task is laborious and requires a thorough understanding of web page design. The theme infrastructure provides a templating mechanism to simplify the process.

Instead of editing CSS and theme descriptor files directly, template files (extension .vtl) are provided that contain key and value pairs that isolate the elements of the theme that you are likely to want to customize. In addition, context files (extension .vctxt) enable you to create a centralized set of definitions for key values that you can use in place of explicit values to simplify the process of maintaining the template files. When you use the SAS Deployment Manager to rebuild the SAS Web Application Themes, the context files are merged into the template files to create a complete set of shared and product-specific style sheets and theme descriptors. The build process also packages your new theme into a WAR file that is deployed to make the themes available in your production environment.

Once the theme archive is deployed, users can use the Preferences page in their SAS web application to apply the new theme (or any other deployed theme). You can also specify the custom theme as the default for all SAS web applications. This means that the theme is applied automatically for users who do not make a selection on the Preferences page.

Steps for Defining and Deploying a New Theme

Overview

SAS provides a default theme for your use. You also have the choice of designing and deploying a custom theme for your environment.

To develop and deploy a new theme, follow these steps:

1. "Step 1: Design the Theme" (See page 150.)
2. "Step 2: Create a Work Area for the Theme" (See page 151.)
3. "Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates" (See page 155.)
4. "Step 4: Rebuild SAS Web Application Themes" (See page 158.)
5. "Step 5: Deploy SAS Web Application Themes in Your Test Environment" (See page 158.)
6. "Step 6: Test the New Theme" (See page 158.)
7. "Step 7: Move the New Theme from Test to Production Environment" (See page 158.)
8. "Step 8: Assign the Default Theme" (See page 159.)

Note: You might choose to perform steps 3 through 6 iteratively, making limited changes to the theme during each iteration, so that you can more readily determine the effects of each set of changes to the theme. To deploy multiple themes in your environment, follow steps 1 to 6 to design and create your themes. Then follow step 7 to move each theme from test to production environment.

You can deploy multiple themes in your corporate environment. Before deploying the new theme in a production environment, you should first test it in a test environment to ensure that SAS web applications function as expected with the new theme applied.

Step 1: Design the Theme

Overview

The first step in creating a custom theme is to plan the visual elements. Usually, the new theme is based on an existing design, your organization's intranet standards, another in-house written application, or a purchased application or solution. Some organizations have a standard color palette with color specifications.

Review the specifications for the Default theme at \SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\index.html, and identify the component keys and image keys for the visual
elements that you want to change in the new theme. Establish a set of colors that are compatible with your organization, and choose the images (for example, logos, banner images) you want to use in the new theme.

Generally, you can make the largest impact by updating the background colors, border colors, and text attributes for web application pages and SAS Information Delivery Portal portlets. In addition, you might want to replace the SAS logo in the banner with our own organization's logo. If you select a different color palette, consider that you might need to adjust the colors in images to match the new palette.

The Color Palette page at `SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\html\colorPalette.html` lists all 55 color keys of the default theme and specifies the default hexadecimal color value for each color key. It also provides links to documentation on each user interface element where the color is applied.

**Options in Designing the Theme**

When you create a new theme, there are three ways to define your theme:

- Use the Color Palette and replace the 55 default SAS colors with your organization's palette. The colors are applied automatically across the user interface.

- Specify the color to be used for each interface component. You must specify the color for each context key of the user interface component. This approach takes more time, but it provides maximum flexibility and control.

- Start with the Color Palette, and make individual changes to selected user interface components. This approach overrides how the color palette is applied in some cases.

If you choose to set colors for the context key of each user interface component, the web pages at `SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\index.html` provide tools and resources to assist you with this process.

**Step 2: Create a Work Area for the Theme**

To create a work area that contains a copy of the Default theme as a basis for your new theme, use one of the following scripts provided in the `SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions` directory:

- for Windows: `NewTheme.bat theme-name true`

- for UNIX: `NewTheme.sh theme-name true`

To use the Color Palette option, the `true` parameter is required in the command.

Note: The theme name must not contain spaces.

The following figure shows the `theme-name` directory, which is the root directory for theme resources. The `\theme-name\MetadataTools` directory contains SAS programs for managing the theme. The `Velocity` directory contains several subdirectories with files.
The following figure shows the subdirectory structure that is created under the SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\themes\theme-name\themes\theme-name directory.
Figure 12.3  Subdirectories for Images, Styles, and Templates

Here is an explanation of the folders and their contents:

\texttt{\textbackslash theme-name\textbackslash themes\textbackslash theme-name\textbackslash images}  
contains the standard collection of images for SAS web applications that use the theme infrastructure. The images are divided into the following subdirectories by category:

- **Common**  
  contains images that are commonly used in SAS web applications.

- **Components**  
  contains images for the collection of components (widgets) that are shared by SAS web applications.

- **WRS**  
  contains images for SAS Web Report Studio.

\texttt{\textbackslash theme-name\textbackslash themes\textbackslash theme-name\textbackslash styles}  
contains a cascading style sheet file named \texttt{custom.css} that can be used to define additional style elements for the theme. This file is empty when the work area is created.

\texttt{\textbackslash theme-name\textbackslash themes\textbackslash theme-name\textbackslash templates}  
contains theme templates, which are HTML fragments that render specific portions of pages in SAS web applications. The template files are divided into the following subdirectories by category:
Common
contains theme templates for page elements that are commonly used in SAS web applications.

Components
contains theme templates for the collection of components that are shared by SAS web applications.

WRS

The following figure shows the subdirectories below the SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\themes\theme-name\Velocity directory.

Figure 12.4 Subdirectories within the Velocity Directory

Here is an explanation of the contents of the directories:

\theme-name\Velocity\Stylesheets\_shared\contexts\themes
contains a context file named theme-name.vctxt that defines context values for font families and standard colors that can be used in CSS templates.

\theme-name\Velocity\Stylesheets\Common\contexts\themes\theme-name
contains CSS template files that are used to build style sheets for page elements that are commonly used in SAS web applications, including portal.theme-name.vtl, sasStyle.theme-name.vtl, and sasScorecard.theme-name.vtl.
Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates

Changing Colors

To make style changes to specific page features, you must first identify the component key associated with that feature and then locate the CSS template file that sets the value for that key.

For example, suppose your new theme design calls for changing the color for the title text in the banner at the top of SAS web applications. The Banner specifications at the Themes website show that the context key for the title text is Banner_Title_Text_Color and it displays its context value.

Each Themes web page displays the context keys and context values.
You can specify a new color explicitly, as follows:

Banner_Title_Text_Color=#e69b00

Because components.theme-name.vtl is a CSS template file, another option is to use the generic color values that are defined in the theme-name.vctxt file in the \Velocity\Stylesheets\_shared\contexts\themes subdirectory of the work area for the new theme. For example, you might specify the following value instead of an explicit value:

Banner_Title_Text_Color=${Color53}

The corresponding color value is substituted in the resulting CSS when the new theme is built.

The general form for using a context value in a template file is ${context-value-name}. Using context values instead of explicit values can make it easier to maintain the theme because you can change all component keys that use a given value by making one change to the context file.

Changing Graphics

Image files are located in three subdirectories located in the SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default folder. These subfolders are: Common, Components, and WRS. The properties of each image are defined in the Theme Descriptors files.

The process for customizing images is similar to that for customizing styles. For example, suppose your new theme design calls for changing the background image for the banner at the top of SAS web applications. A review of the Banner specifications at SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\index.html shows that the image key for the banner background is banner_background. A search for that string in the work area for the new theme shows the following IMAGE element in the ComponentsThemes.vtl file in the Velocity\ThemeDescriptors\custom\theme-name subdirectory of the work area:

<Image name="banner_background" ...  file="BannerBackground.gif"/>

You can change the image used for the banner background image in either of the following ways:

- by replacing the existing BannerBackground.gif file in the themes\theme-name\images\Components subdirectory of the work area with a revised image with the same name. Make sure that the new image has the following criteria:

  □ The filename of the new graphic is identical to the filename of the graphic being replaced.

  □ The new graphic is in the same format as the original image (for example, .jpg or .gif).

  □ The dimensions of the new graphic and its pixels are same as the graphic being replaced.

If you need to change the size, filename, or the image format of the graphic, modify the theme descriptor. For example, if you replace the logo.gif file with a new file called myLogo.jpg that has a width of 300 pixels and height of 70 pixels, modify the ComponentsThemes.vtl file as follows:

<Image name="logo" description="My Logo" altTextKey="desktop.logo.text" appliesTo="ALL" width="300" height="70" file="myLogo.jpg"/>
by changing the FILE= attribute in the IMAGE element in the ComponentsThemes.vtl context file to point to a different image file.

Note: You should not change the value of the NAME= attribute in the IMAGE element. SAS web applications depend on the NAME= attributes remaining constant.

Another common image change is to replace the SAS logo in the standard banner with your organization's logo. You can change the graphic used for the banner logo either by replacing the existing logo.gif file in the themes \theme-name\images\Components subdirectory of the work area with a copy of your logo with that filename or by changing the target of the FILE= attribute for the IMAGE element in the ComponentsThemes.vtl context file for which the NAME= attribute has the value logo.

When customizing images, you should ensure that the replacement graphics have approximately the same dimensions as the original graphics. Otherwise, the images might disrupt the appearance of the applications in which they are used.

**Changing Theme Templates**

You should make changes to theme templates only in situations where you want to change the layout of a page element (for example, to change the logo's placement in the banner or to adjust the padding between rows in a menu). If you decide to alter a theme template, proceed with caution. SAS web applications rely on the template structure being consistent with the versions that are shipped with the software. Improper changes to theme templates might prevent SAS web applications from functioning properly. In particular, do not change the dynamic substitution variables in theme templates because SAS web applications expect the existing values.

Dynamic substitution variables should not be changed in theme templates because SAS web applications expect the existing values. However, if you need to change a dynamic substitution variable, here is an example where %BANNER_TITLE is the dynamic substitution variable:

```html
<td nowrap id="bantitle" class="banner_title">%BANNER_TITLE</td>
```

Note: When a new release of themes is installed at your site or an upgrade is performed, the existing theme template files are replaced by the new theme template files. If you have customized theme template files and want to retain them for future use, copy them to a different location before the installation or upgrade.

**Additional Considerations**

Another change that you might want to make when creating your new theme is to update the theme_displayName= element in the theme-name.themeDescriptor.vctxt file in the Velocity\ThemeDescriptors \contexts subdirectory of the work area. Provide a descriptive name for the new theme. The name is used in the selection list of available themes in the Preferences page in SAS web applications.
Step 4: Rebuild SAS Web Application Themes

To rebuild SAS Web Application Themes and register your themes in metadata, follow the steps provided in “Rebuild Web Applications” on page 87.

The rebuilt SAS Web Application Themes archive file (sas.themes.ear) can be found in the SAS-configuration-directory\Lev1\Web\Staging directory. It should contain a new web archive (WAR) file for the new theme named sas.theme.theme-name.war.

Step 5: Deploy SAS Web Application Themes in Your Test Environment

To deploy the rebuilt SAS Web Application Themes to your web application server in a test environment, see “Redeploying the SAS Web Applications” on page 91.

If you chose to configure the web application server manually or deployed the SAS web applications manually, see your Instructions.html generated by the SAS Deployment Wizard.

Step 6: Test the New Theme

After you have completed the deployment procedures, follow these steps to test the new theme:

1. Navigate to the portal in the production environment.
2. Log on and select Options ➤ Preferences. The new theme should appear as a selection on the Preferences page.
3. Select the new theme and observe the effect of the changes that you made in “Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates” on page 155. To view the new theme, log off from the portal. Then log on to the portal to view the new theme that was applied.
4. Repeat the procedures outlined in “Steps for Defining and Deploying a New Theme ” on page 150 until you are satisfied with the display of the new theme.

If you test the new theme several times, log off from the portal and log on again to view the updated theme each time.

Step 7: Move the New Theme from Test to Production Environment

To move a theme from a test to a production environment, follow these steps:

- Copy the entire contents of the SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions directory to the same directory path on the production machine.
- Run SAS Deployment Manager, and use the Rebuild Web Applications option to register the theme in the metadata. See “Step 4: Rebuild SAS Web Application Themes” on page 158.
Deploy SAS Web Application Themes to your web application server. See “Step 5: Deploy SAS Web Application Themes in Your Test Environment” on page 158.

Step 8: Assign the Default Theme

Overview

If you want your new or custom theme to be the default theme for all users who have not selected a theme for themselves in their application’s Preferences, then you should set the new theme as the default.

There are two ways to modify the theme metadata:

- Use SAS Management Console. See “Assign the Default Theme from SAS Management Console” on page 159.
- Use the `UpdateDefaultTheme.sas` program. See “Assign the Default Theme with the UpdateDefaultTheme.sas Program” on page 159.

Assign the Default Theme from SAS Management Console

To assign a new theme as the default theme by using the SAS Management Console, follow these steps:

1. Deploy SAS Web Application Themes using the SAS Deployment Manager.
2. In SAS Management Console, on the Plug-ins tab, navigate to Application Management ▶ Configuration Manager ▶ SAS Application Infrastructure and right-click to display the SAS Application Infrastructure Properties dialog box.
3. Click the Settings tab.
4. In the Default Theme field, enter the name of your theme.
5. Click OK to exit the SAS Application Infrastructure Properties window.
6. To enable the new theme to go into effect, restart the SAS Web Infrastructure Platform application in the web application server.

Assign the Default Theme with the UpdateDefaultTheme.sas Program

To assign a theme as the default theme, use the `UpdateDefaultTheme.sas` program located in the `SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\themes\theme-name\MetadataTools` directory. After the `UpdateDefaultTheme.sas` program has been run, the new theme will be in effect for users who have not selected a different theme on their Preferences page.

If SAS is not installed on the middle-tier machine, copy the `UpdateDefaultTheme.sas` program to the metadata server, and submit the SAS program on that machine.
Deleting a Custom Theme from the Metadata

To delete a custom-developed theme from the deployment for the SAS Information Delivery Portal, use the `DeleteTheme.sas` program located in the `SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\themes\theme-name\MetadataTools` directory.

If SAS software is not installed on the middle-tier machine, copy the `DeleteTheme.sas` program to the metadata server, and submit the program on that system machine.

Migrating Custom Themes

Overview

To apply a custom theme that you developed for an earlier release, follow these steps:

1. Create a new theme structure. For information about creating a work area in which to construct the new version of your existing theme, see “Step 2: Create a Work Area for the Theme” on page 151.

2. Migrate the cascading style sheets used in your theme.

3. Migrate the images used in your theme.

4. Migrate the theme templates.

5. Migrate the descriptors used in your theme.

Migrating Cascading Style Sheets

Before attempting to move any CSS files from an existing theme to the `themes\theme-name\styles` subdirectory of the work area for the new theme, you should first review the specifications for the Default theme at `SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\index.html`. For any feature for which a component key has been defined, you should update the corresponding component key values in the CSS template (`.vtl`) files in the `\Velocity\Stylesheets\Common\contexts\themes\theme-name, \Velocity\Stylesheets\Components\contexts\themes\theme-name, and \Velocity\Stylesheets\WRS\contexts\themes\theme-name` subdirectories of the work area to achieve a compatible look and feel.

Custom style sheet files are required only if you need to provide theme support to features that are not covered by the CSS templates. For each style sheet file that you add, you must ensure that a corresponding `STYLESHEET` element is added to in the appropriate theme descriptor template (`.vtl`) file in the
\Velocity\ThemeDescriptors\contexts\custom\theme-name
subdirectory of the work area for the new theme. The STYLESHEET element
must specify the value all for its PRODUCT= attribute.

**Migrating Images**

Before attempting to move any image files from an existing theme to the
\themes\theme-name\images subdirectory of the work area for the new
theme, see the image specifications for the Default theme at SAS-
configuration-directory\Lev1\Web\Utilities\SASThemeExtensions
\specs\Default\index.html. If the image from the existing theme replaces
one of the images in the new theme, then you should ensure that the image from
the existing theme is saved over the default image in the proper directory under
the \themes\theme-name\images Subdirectory. If the image from the existing
theme does not replace an image in new theme, save it in the \themes\theme-
name\images\Common subdirectory.

For each image file that you update or add, you must ensure that a
corresponding IMAGE element is present in the appropriate theme descriptor
template (.vtl) file in the \Velocity\ThemeDescriptors\contexts\custom
\theme-name subdirectory of the work area for the new theme.

**Migrating Theme Templates**

Before attempting to move any theme template files from an existing theme to
the \themes\theme-name\templates subdirectory of the work area for the
new theme, you should consider carefully whether they are compatible with the
SAS web applications. SAS web applications rely on the theme template
structure being consistent with the versions that are shipped with the software.
Theme templates must have the expected set of dynamic substitution variables
in order for the applications to function properly.

**Migrating Theme Descriptors**

The theme descriptor template (.vtl) files in the \Velocity
\ThemeDescriptors\contexts\custom\theme-name subdirectory of the
work area for the new theme should represent the structure of the migrated
theme resources. Review the files to ensure the following:

- If you add cascading style sheet files to provide theme support for features
  that are not covered by CSS templates, ensure that you add corresponding
  new STYLESHEET elements to the STYLES section.
- For each image file that you update or add, ensure that you update or add a
  corresponding IMAGE element in the IMAGES sections.
- If you migrate existing theme template files, ensure that you update or add a
  corresponding TEMPLATE element in the TEMPLATES sections to reflect
  the change.
Administering SAS Flex Application Themes

Introduction to SAS Flex Application Themes

Some SAS web applications, such as SAS BI Dashboard and SAS BI Portlets, are displayed with the Flex interface that is provided by SAS Flex Application Themes. At start-up time, Flex applications load Flex themes automatically. A theme consists of ShockWave Flash (SWF) files that include cascading style sheets (CSS) files. The theme content is downloaded to the client and is cached by the user's web browser. As a result, subsequent uses of the web application result in quicker loading of theme content than it is at initial loading. The SAS Corporate theme is the default theme for all Flex applications.

Themes can be created with the SAS Theme Designer for Flex. For information about custom themes for Flex applications, see SAS Theme Designer for Flex User’s Guide.

Benefits of SAS Flex Application Themes

SAS Flex Application Themes are required for Flex applications, and they are downloaded as SWF files to the client's web browser. Flex theme content runs within the Adobe Flash player and offers the following benefits:

- SAS Flex Application Themes coexist with SAS Web Application Themes. For example, SAS Information Delivery Portal uses the default web theme, but it displays SAS BI Portlets with SAS Flex Application Themes.
- Applications that use SAS Flex Application Themes offer more visual impact, interactivity, and responsiveness.
- Improved visual impact and perceived depth are achieved through the use of skins. Skins are graphics that are applied to common user interface...
components that change their appearance. For example, the Corporate theme provides skins with a color palette that reflects the SAS visual identity. Skins also include some stylized graphics in the user interface.

Location of SAS Flex Application Themes

SAS Flex Application Theme files are located in the `SAS-configuration-directory\Lev1\Web\Staging\sas.flexthemes4.1.ear` file.
About the Search Facility

Solutions such as SAS Visual Analytics rely on a search facility that is installed on the middle tier. The search facility indexes and searches SAS content that is registered in metadata and provides results subject to the requesting user’s roles and permissions.

The search facility consists of the following components:

- the Search Interface to SAS Content web application, which performs searches against a generated index and provides results to the requesting client
- an indexing provider (either SAS Information Retrieval Studio or Apache Lucene) to create the index

The indexing process is run using the SAS Administrator account because it requires unrestricted access to metadata and assignment to ROLE_ADMIN in the middle tier. When the search facility is first initialized, all available content is fully indexed. After that, incremental changes to content are indexed periodically. If there is a failure in generating or loading the index, an email is sent to the address that is designated for administrative messages for your deployment.

For detailed configuration information, see Integrating Search Interface to SAS Content at support.sas.com.
About the Search Index Providers

SAS Information Retrieval Studio

About SAS Information Retrieval Studio
In the standard configuration, SAS Information Retrieval Studio creates the index.

Windows Specifics: The server runs as a local service (for example, SAS [Config-Lev1] Information Retrieval Studio).

UNIX Specifics: The IRStudio.sh script is in the SAS configuration directory under /Applications/SASInformationRetrievalStudioforSAS. You can use the following commands to operate the server:

```
IRStudio.sh start | stop | status | restart
```

Generated log files are in the /logs subdirectory.

Configuring for TLS
Beginning with the October 2014 release for SAS 9.4, Transport Layer Security (TLS) is supported for search by SAS Information Retrieval Studio. In previous releases, by default, TLS will not work. To update your configuration for TLS, follow these steps:

1. Edit the `SAS-configuration-directory\Lev\Web\Applications\SearchInterfacetoSASContent\url_list.txt` file.

2. Locate the HTTP URL. The file contents will look similar to the following example:

   ```
   # This will feed all supported SAS content to index server
   http://hostname.example.com:80/SASSearchService/rest/searchAdmin/searchIndex?
   userName=sassearch@saspw&password={sas002}3CD4BA1E35CA49324A0C4D63
   
   Note: The previous line must be entered on one line. It is shown on more than one line for display purposes only.
   ```

3. Change `http` to `https`.

4. Save the file.

Apache Lucene
If the configuration property searchsas.irstudio.is_available is set to `false`, Apache Lucene creates the index.

Here are some tips and details:

- Use Apache Lucene only in an IPv6 environment (where SAS Information Retrieval Studio is not supported).
Apache Lucene is a library (not a server or service), so it does not have a script.

To verify that content is being indexed, navigate to your equivalent of Web/Applications/SearchInterfaceToSASContent/Index/Default/ (in the SAS configuration directory). If the folder is empty, content is not being indexed.

Specifying Configuration Properties for the Search Interface to SAS

To use SAS Management Console to view or change the configuration properties for the Search Interface to SAS, follows these steps:

1. Log on to SAS Management Console.
2. On the Plug-ins tab, select Application Management ➤ Configuration Manager, right-click Search Interface to SAS Content version, and select Properties.
3. Click the Advanced tab. On this tab, you can specify values for the following properties:
   
   - `searchsas.feeder.scheduler.interval.minutes` specifies how frequently the index is generated and loaded. The interval is set during configuration. In the standard configuration, the interval is 15 minutes. Shorter intervals provide more current search data at the price of additional consumption of system resources (because shorter intervals require the more frequent polling for updated data).
   
   - `searchsas.feeder.scheduler.is_enabled` specifies whether index generation and loading occurs. The default is `true`. To disable indexing, specify `false`.
   
   - `searchsas.irstudio.is_available` specifies which provider is used. If the value is `true`, or the property is not set, SAS Information Retrieval Studio is used. If the value is `false`, Apache Lucene is used.
   
   - `searchsas.notification.email.is_active` controls whether notifications are sent. To disable notifications, set this property to `false`.
   
   - `searchsas.notification.email.sender.address` specifies the sender’s email address.
   
   - `searchsas.notification.email.to.address` specifies the recipient’s email address. To assign multiple recipients, provide a comma-separated list of addresses.
4. Click OK.
Part 4

Advanced Topics

Chapter 15
  Managing Devices .................................................. 171

Chapter 16
  Best Practices for Configuring Your Middle Tier .................. 187

Chapter 17
  High-Availability Features in the Middle Tier ..................... 197

Chapter 18
  Enterprise Integration ............................................ 207

Chapter 19
  Middle-Tier Security ............................................ 253
Managing Devices

Viewing SAS Web Report Studio Reports on Mobile Devices ........................................... 172
  Getting Started .................................................................................................................. 172
  Supported Reports .......................................................................................................... 172
  How Mobile Content Is Protected ..................................................................................... 173

Managing Mobile Devices .................................................................................................... 173
  About Managing Mobile Devices ...................................................................................... 173
  How to Blacklist a Device ................................................................................................. 174
  How to Whitelist a Device ............................................................................................... 174
  How to Determine Which List Is Enforced ......................................................................... 174
  How to Determine When a Device Was Blacklisted ............................................................ 175
  How to Change How Devices Are Managed ....................................................................... 175
  About the Mobile Devices Tab .......................................................................................... 175

Capabilities for SAS Mobile BI .......................................................................................... 176
  Predefined Roles ............................................................................................................... 176
  Capabilities for End-Users ............................................................................................... 177
  Capabilities for Administrators ........................................................................................ 178

Accessing and Using SAS Mobile BI .................................................................................. 178
  About the SAS Demo Server ............................................................................................. 178
  iPad: How to Add a Connection ....................................................................................... 178
  Android: How to Add a Connection .................................................................................. 179
  About Searching for Reports ............................................................................................ 179
  User Assistance Resources for Visual Analytics Access .................................................. 180

Adjusting the Logging Configuration .................................................................................. 180

Configuration Properties: SAS Mobile BI .......................................................................... 180
  How to Set Advanced Properties for SAS Mobile BI ....................................................... 180
  Reference for Selected Properties ................................................................................... 181

Supported OLAP Functionality ........................................................................................... 183
  Feature Set of Graphs and Crosstabs with Non-Relational Data ...................................... 183
  Stored Processes with Prompts ......................................................................................... 183

Troubleshooting: SAS Mobile BI ....................................................................................... 183

Advanced Properties for SAS Mobile BI ............................................................................. 184
  Configuring the Display of Banner-Related Items for SAS Mobile BI ............................... 184
  Modifying the Value Used for Resizing Images in the Middle Tier for SAS Mobile BI ...... 184
Viewing SAS Web Report Studio Reports on Mobile Devices

Getting Started

Mobile reporting for SAS Web Report Studio enables users to view certain types of relational reports on supported devices (iPad and Android). Supported reports are displayed in the native format of the device.

You do not have to perform any post-installation tasks to enable mobile reporting. Here are the key points:

- In the initial configuration, all registered users can view supported reports on supported devices. For more information, see "Accessing and Using SAS Mobile BI" on page 178.
- In the initial configuration, only an unrestricted user (such as sasadm@saspw) can blacklist devices. For more information, see "Managing Mobile Devices" on page 173.
- The SAS Mobile BI for iPad app is available in the iTunes App Store. The Android app is available from Google Play.

Supported Reports

Not all reports that are created in SAS Web Report Studio can be viewed on mobile devices. In the current release, the following types of reports can be viewed on mobile devices:

- relational reports that provide all SAS Web Report Studio graph elements with the exception of maps
- relational reports that use list tables
- relational reports that provide tile charts
- relational reports that provide text objects
- relational reports that provide stored processes with default prompts
- relational reports that provide images
- relational reports that provide group breaks
- relational reports that provide applied filters
- relational reports that provide a display of stored processes
- relational reports that provide conditional highlighting in a list
- relational reports that provide a relational crosstabulation table
- relational reports that provide row and column totals in relational crosstabulation table
How Mobile Content Is Protected

Protections for mobile content include the following:

- Users must authenticate in order to establish a connection.
  
  **Note:** By default, authentication is against the metadata server’s authentication provider. If web authentication is configured, accounts are validated against the web application server’s authentication provider. For more information, see “Web Authentication” on page 210.

- SAS metadata security is enforced on all reports.

- You can manage device eligibility by exclusion or inclusion. For more information, see “About Managing Mobile Devices” on page 173.

- To minimize persistence of mobile data, assign users or groups to a role that has the Purge Mobile Report Data capability.

- To require knowledge of an application passcode, assign users or groups to a role that has the Require Passcode on Mobile Devices capability.

- Content on a mobile device is encrypted by the device’s operating system.

- You can encrypt connections between mobile devices and SAS servers using Transport Layer Security (TLS). For more information, see “Support for TLS with Client Certificate Authentication” on page 242.

Managing Mobile Devices

About Managing Mobile Devices

Here are the key points:

- Use SAS Management Console to assign the Visual Analytics: Administration role to administrators. This role grants the privileges that are needed to manage mobile devices.

- To manage devices that use SAS Mobile BI, select Tools ▶ Manage Devices from the main menu in SAS Visual Analytics Administrator. The administrator interface is available from http://hostname.example.com/SASVisualAnalyticsAdministrator. You can manage devices either by exclusion or by inclusion.

  - If you manage by exclusion, all devices can use SAS Mobile BI, except those that are on the blacklist. A blacklist is a list of mobile devices that are not authorized to use SAS Mobile BI.

  - If you manage by inclusion, only devices that are on the whitelist can use SAS Mobile BI. A whitelist is a list of mobile devices that are authorized to use SAS Mobile BI.

- A deployment enforces only one list (either the blacklist or the whitelist). In a new deployment the blacklist is enforced, so there are no device-level barriers to participation.

- You can modify both lists. Making changes to a list that is not currently enforced can help accommodate a future change.
These lists affect devices, not users. To manage what a particular user can see or do in SAS Mobile BI, use permissions and capabilities.

How to Blacklist a Device

Note: These instructions have an effect only if the blacklist is enforced.

To prevent a mobile device from using SAS Mobile BI, follow these steps:

1. Select Tools > Manage Devices.
2. On the Mobile Devices tab, select the Blacklist tab.
3. At the right edge of the tab, click +.
4. In the Add Device To Blacklist window, enter the ID of the device that you want to exclude from using SAS Mobile BI. (Or, to add multiple device IDs, click Add List.) Click OK.

Note: The information that you supply is not validated by the software.

TIP For a device that has already connected (or attempted to connect), you can initiate this task from the Logon History tab. Right-click on the device, and select Add to Blacklist.

To remove a device from the blacklist, select it on the Blacklist tab, right-click, and select Move to Whitelist.

How to Whitelist a Device

Note: These instructions have an effect only if the whitelist is enforced.

To enable a mobile device to use SAS Mobile BI, follow these steps:

1. Select Tools > Manage Devices.
2. On the Mobile Devices tab, select the Whitelist tab.
3. At the right edge of the tab, click +.
4. In the Add Device To Whitelist window, enter the ID of the device that you want to enable to use SAS Mobile BI. (Or, to add multiple device IDs, click Add List.) Click OK.

Note: The information that you supply is not validated by the software.

TIP For a device that has already connected (or attempted to connect), you can initiate this task from the Logon History tab. Right-click on the device, and select Add to Whitelist.

To remove a device from the whitelist, select it on the Whitelist tab, right-click, and select Move to Blacklist.

How to Determine Which List Is Enforced

In the toolbar at the top of the Mobile Devices tab, the Enforced drop-down list indicates which list is enforced.
In addition, text at the top of either the **Blacklist** tab or the **Whitelist** tab indicates the list that is not currently enforced.

**TIP** You can also verify the current configuration in SAS Management Console. The blacklist is enforced unless the `viewerservices.enable.whitelist.support` property is set to `true`. For more information, see "Configuration Properties: SAS Mobile BI" on page 180.

### How to Determine When a Device Was Blacklisted

To determine when a device was blacklisted, follow these steps:

1. On the **Blacklist** tab, right-click on the device, and select **Copy Device ID**.
2. On the **Management History** tab, select **Device ID** from the **Filter** dropdown list.
3. Click in the text field, and enter Ctrl-V from the keyboard. (You cannot perform the paste action from the right-click menu.)
4. Click **Apply**.

**TIP** You can also copy a device ID from the **Whitelist** tab. You can also paste a device ID into the **Device ID** filter on the **Logon History** tab.

### How to Change How Devices Are Managed

**CAUTION!** These are deployment-level instructions that affect all access to SAS Mobile BI.

To switch from enforcing one list to enforcing the other, follow these steps:

1. Select **Tools** > **Manage Devices**.
2. Verify that the list that you intend to enforce is appropriately populated.
   - If you enforce the whitelist, the whitelist should contain all eligible devices. The blacklist is ignored.
   - If you enforce the blacklist, the blacklist should contain all excluded devices. The whitelist is ignored.
3. In the toolbar at the top of the **Mobile Devices** tab, make a selection from the **Enforced** drop-down list. In the confirmation window, click **Yes**.

### About the Mobile Devices Tab

Here are some details about the **Mobile Devices** tab:

- On the **History** tabs, you can filter by selecting an item from a **Filter** dropdown list, specifying a value, and clicking **Apply**.
- The **Logon History** tab displays logon events. By default, only one logon event for each device is displayed. To view previous logon events, select the
Include device history check box. The following occurrences are logon events:

- a connection attempt that comes from a new source (a unique combination of device ID and user ID)
- a connection attempt that is accompanied by a device change (such as a new operating system version or application version)

- On the Logon History tab, the Status column provides information about a logon event. The Status column does not indicate the current status of a device connection.

- When you right-click on a device on the Logon History tab, remember that only one list is in use. Adding a device to the list that is not in use has no immediate effect. For example, if your deployment uses the blacklist, adding a device to the whitelist has no immediate effect.

- On the Blacklist and Whitelist tabs, each cell in the User ID column contains the user ID that connected (or attempted to connect) to SAS Mobile BI from the associated device. The user ID is provided for the purpose of helping you identify a device. If no user has attempted to connect from a particular device, no user ID is listed for that device. If multiple users have attempted to connect from a particular device, all of those user IDs are listed.

- On the Manage tabs, you can right-click on a device ID, and select Copy Device ID. On the History tabs, you can paste a device ID into the text field next to the Filter drop-down list. (To paste, enter Ctrl-V from your keyboard.)

  Note: A device ID is a unique identifier (usually a hardware device number) that is determined and communicated by the connecting mobile application.

- The Management History tab displays device management events, such as adding a device to a list or removing a device from a list. The Admin ID column provides the user ID of the administrator who performed each action.

- When you right-click on a device in the blacklist or whitelist, you can choose either a move action or a remove action. In terms of immediate effect, there is no difference between these two actions.

Capabilities for SAS Mobile BI

Predefined Roles

The following predefined roles are relevant in mobile reporting for SAS Web Report Studio:

**Visual Analytics: Report Viewing**

provides the ability to view reports on mobile devices. The initial member is the SASUSERS group, which includes all registered users. In general, it is not necessary to make any changes to this role.

Note: As of SAS Visual Analytics 7.2, SAS Mobile BI does not support anonymous or PUBLIC-only access. In previous versions of SAS Visual Analytics, SAS Mobile BI does not support anonymous, guest, or PUBLIC-only access.
Visual Analytics: Administration
provides the ability to manage mobile devices. You can either make
administrators members of this role, or add the relevant capability to the Web
Report Studio: Advanced role.

Capabilities for End-Users

The following capabilities are relevant in mobile reporting for SAS Web Report Studio:

**Visual Analytics**

_view report and stored process_

enables users to view reports.

**Add and View Comments**
enables users to add comments, view comments, and edit their own comments.

*Note:* The capabilities that are listed under **SAS Application Infrastructure > Comments** enable you to delete comments and edit other user’s comments. You can add these capabilities to an administrative role. Or you can make any users that need these capabilities members of the **Comments: Administrator** role.

**Email**
enables users to send a link to a report via email.

**Visual Analytics Transport Service**

_purge mobile report data_

causes cached data on mobile devices to be purged when reports are closed. For users who do not have this capability, cached data is retained locally on the mobile device for use in offline mode.

*Note:* Because unrestricted users always have all capabilities, their mobile data is always purged when they close reports. In general, you should not use an unrestricted identity (for example, sasadm@saspw) to view reports.

**Require Passcode On Mobile Devices**

requires users to enter an application passcode on their devices each time they use SAS Mobile BI. For users who do not have this capability, an application passcode is not required.

*Note:* Because unrestricted users always have all capabilities, they are always subject to the application passcode requirement. In general, you should not use an unrestricted identity (for example, sasadm@saspw) to view reports.

See [viewerservices.passcode.attempts](#) on page 182 and [viewerservices.passcode.timeout](#) on page 182.

**Tip** It is not necessary to make any changes to the predefined roles and capabilities in order to support mobile report viewing for all registered users.
Capabilities for Administrators

The following capabilities affect the availability of mobile device management functionality:

**Visual Analytics version: Advanced: Manage Mobile Devices**
provides back-end support for managing mobile devices.

*Note:* In a deployment that includes the entire suite of SAS Visual Analytics applications, the Manage Environment capability is also required.

**TIP** Consider adding this capability to the Web Report Studio: Administration role.

**Visual Analytics version: Advanced: Manage Environment**
provides access for the SAS Visual Analytics Administrator. This capability is enforced only in deployments that include the entire SAS Visual Analytics suite of applications.

*Note:* For instructions for working with roles and capabilities, see the SAS Management Console: Guide to Users and Permissions.

---

Accessing and Using SAS Mobile BI

About the SAS Demo Server

When you open a SAS Mobile BI app for the first time, the only connection that is available is to the SAS Demo Server. The sample reports on the SAS Demo Server demonstrate features that are provided by reports that are created in SAS Visual Analytics. Not all of the features in the sample reports are supported for reports that are created in SAS Web Report Studio.

**iPad: How to Add a Connection**

To access your corporate library server, you must add a connection to that server. If your company has more than one server, you can add additional connections in the same way.

*Note:* Contact your system administrator for your user ID, password, and server information.

To connect to a server, follow these steps:

1. If you are in My Portfolio, tap **Library**.
2. Tap **Connections** in the library.
3. Tap **Add Connections**.
4. Tap the **Server** field, and enter the address of the new server.

*Note:* If the server requires a secure (TLS) connection, type `https://` at the beginning of the server address.
5 Tap the User ID field, and enter your user ID.

6 Tap the Password field, and enter your password.

7 Tap the Description field, and enter a description for the new connection.

8 Tap Next to verify the connection.

9 Tap Save. The connection is saved, and the Connections window closes.

Note: If the connection fails, a message is displayed to help you correct the problem.

Information about available reports is automatically downloaded and appears in the library.

Android: How to Add a Connection

To access your corporate library server, you must add a connection to that server. If your company has more than one server, you can add additional connections in the same way.

Note: Contact your system administrator for your user ID, password, and server information.

Here are the steps to connect to a server:

1 If you are in My Portfolio, tap Library.

2 Tap the Connections icon in the library.

3 Tap Add Connections.

4 Tap the Server field, and enter the address of the new server. If the port is a nonstandard port, then you should enter the port number.

   Note: If the server requires a secure (TLS) connection, type https:// at the beginning of the server address.

5 Tap the User ID field, and enter your user ID.

6 Tap the Password field, and enter your password.

7 Tap the Description field, and enter a description for the new connection.

8 On the keyboard, tap Done to verify the connection.

9 Tap OK. The connection is saved, the Connections window closes, and the library appears.

Note: If the connection fails, the Extended Connection Properties window displays a message to help you correct the problem.

About Searching for Reports

In the current release of SAS Mobile BI, the search feature within the SAS Mobile BI apps does not discover any reports unless Search SAS is installed on the server. However, if the SAS Visual Analytics Suite is installed, Search SAS is included in that installation and is available for use.
User Assistance Resources for Visual Analytics Access

All of the SAS Mobile BI apps provide online Help. The online Help also provides access to tutorial videos about how to use the app.

For resources that help users interact with mobile reports, see the SAS Mobile BI product documentation page on the SAS support site.

Adjusting the Logging Configuration

To increase logging of actions and events for SAS Mobile BI, follow these steps:

1. In the SASVisualAnalyticsTransport-log4j file, change logging levels, modify paths, and remove comments, as applicable.
2. Restart SAS Web Application Server.

Note: The viewservices.validate.schema.* properties enable you to increase logging for rendering of reports on mobile devices.

Configuration Properties: SAS Mobile BI

How to Set Advanced Properties for SAS Mobile BI

To set advanced properties, follow these steps:

3. On the Advanced tab of the Properties dialog box, add or set values.

   **TIP** The lock icons indicate which settings can be changed in child components. The lock icons do not indicate which changes you can make to the current component.

Reference for Selected Properties

**Printing.Timeout**
sets a maximum wait time that affects printing reports from applications such as the designer and the web viewer. The default is 900000 milliseconds (15 minutes). To disable this property, set its value to 0.

**Note:** This setting does not affect the first phase of a print request, which generates a report package. This setting affects only the second phase of a print request, which uses a stored process call to execute the print routine.

**viewerservices.company.banner.logoUrl**
provides the URL for an alternate logo in the SAS Mobile BI banner. The URL must be accessible to the mobile device. Support for this property is discontinued in the 7.1 release of SAS Mobile BI.

**viewerservices.company.banner.message**
provides a custom message for the SAS Mobile BI banner. Support for this property is discontinued in the 7.1 release of SAS Mobile BI.

**viewerservices.company.banner.title**
provides a custom title for the SAS Mobile BI banner. Support for this property is discontinued in the 7.1 release of SAS Mobile BI.

**viewerservices.data.default.interactive.drill.depth**
determines how much data is sent to a mobile device for offline drilling. This property is applicable to visualizations that reference a hierarchy. The default is 3 (users can drill down three levels). If certain reports require users to have the ability to drill down more than three levels into a hierarchy, modify the value.

**viewerservices.default.max.cells.produced**
sets the maximum number of data cells that can be delivered to a mobile device for a single data query. The default is 250,000 data cells, which is sufficient for most environments and does not cause the web application server to crash. In very rare scenarios, you might need to modify the value.

**Note:** If the number of data cells in a query exceeds the value specified for this property, the data that is returned to SAS Mobile BI is truncated, and an incomplete report is presented. Data used for that report is not complete.

**CAUTION!** Modifying the limits on the number of cells can cause the device to become unstable

**viewerservices.enable.whitelist.support**
controls which approach is used to manage mobile devices. Valid values are:

- **false** causes the blacklist to be enforced and the whitelist to be ignored. With this setting, all mobile devices can use SAS Mobile BI except for those devices that are on the blacklist. This is the default.

- **true** causes the whitelist to be enforced and the blacklist to be ignored. With this setting, only mobile devices that are on the whitelist can use SAS Mobile BI.

**CAUTION!** Enabling the whitelist can disrupt existing users. Make sure that all valid devices are on the whitelist before you make the change.
TIP As an alternative to setting this property explicitly, you can set it from within SAS Visual Analytics Administrator. For more information, see “How to Whitelist a Device” on page 174.

viewerservices.image.default.max.bytes
sets the maximum size of images (PNG, BMP, JPEG, or GIF) that can be delivered to a mobile device. Larger images are resized on the server side before delivery. The default is 300 KB, which is sufficient for most environments. In very rare scenarios when you want to change this constraint, consider modifying the value. To entirely disable resizing of images in the middle tier, set the value to 0. However, to ensure faster download times and smaller memory footprints on the mobile device, do not increase the value of this property or set the value to 0.

Note: Users can customize image resizing on their devices by setting the Scale type option (under Insert ➤ Other ➤ Image). If the option is set to None, the user’s device is exempt from middle-tier resizing.

viewerservices.lasr.socketTimeout.milliseconds.interactions
sets the maximum wait time for when SAS Mobile BI attempts to contact SAS LASR Analytic Server. This property is applicable to live requests from a mobile device for tasks such as filtering, brushing, and drilling. The default is 30000 milliseconds (30 seconds), which is sufficient for most environments. If sessions between SAS Mobile BI and SAS LASR Analytic Server are timing out, consider modifying the value.

viewerservices.lasr.socketTimeout.milliseconds.subscribe
sets the maximum wait time for a response to a query in a subscribed report when SAS Mobile BI contacts the SAS LASR Analytic Server. The default is 300000 milliseconds (5 minutes), which is sufficient for most environments. If the queries within some reports take an excessive amount of time for completion, consider modifying the value.

viewerservices.passcode.attempts
limits the number of sequential failed attempts to enter a passcode. The default is 5. If a user reaches the limit, the user is locked out of the app for 15 minutes. After the lockout interval, the user can again attempt to enter his or her passcode. If the user reaches the limit again, all custom content (data, reports, settings, and connection information) is removed from the device.

Note: This property is applicable to only those users who are subject to the capability Require Passcode on Mobile Devices on page 177.

viewerservices.passcode.timeout
specifies, in minutes, how frequently a user must re-enter his or her passcode. The default is 15.

viewerservices.validate.schema.write
enables XML schema validation when reports are rendered in SAS Mobile BI. When this property is set to true, all actions that apply to the writing of reports are captured in the SASVisualAnalyticsTransport-log4j file. The default is false. Set this property only if SAS Technical Support instructs you to do so.

viewerservices.validate.schema.create
enables XML schema validation when reports are rendered in SAS Mobile BI. When this property is set to true, all actions that apply to the creation of reports are captured in the SASVisualAnalyticsTransport-log4j file. The
default is `false`. Set this property only if SAS Technical Support instructs you to do so.

`viewerservices.validate.schema.read` enables XML schema validation when reports are rendered in SAS Mobile BI. Also, this property checks for schema validation errors when reports are created in the mobile viewer. When this property is set to `true`, all actions that apply to opening and viewing reports are captured in the `SASVisualAnalyticsTransport-log4j` file. The default is `false`. Set this property only if SAS Technical Support instructs you to do so.

---

### Supported OLAP Functionality

#### Feature Set of Graphs and Crosstabs with Non-Relational Data

The following list shows supported OLAP features:

- In a fully expanded form or a drilled form, all visuals are supported with non-relational data source (cube and information map).
- In a drilled form, the graph visuals display a crumb trail at the top indicating the drill-down level. The presence of a crosstab visual in the report does not display the crumb trail.
- For a crosstab visual, it is important that the column axis lists categories ahead of measures in the ROM report.

#### Stored Processes with Prompts

Prompts with default values listed in the prompts definition are supported.

---

### Troubleshooting: SAS Mobile BI

**Issue: A user cannot open reports in an offline device.**

Resolution: Make sure that the user is not unrestricted and is not in any role that provides the capability that prevents this action. For more information, see “Purge Mobile Report Data” on page 177.

**Issue: A user is prompted for an application passcode.**

Resolution: Make sure that the user is not unrestricted and is not in any role that provides the capability that introduces this requirement. For more information, see “Require Passcode On Mobile Devices” on page 177.

**Issue: On the Mobile Devices tab, a message indicates that a list is not currently in use.**

Resolution: By design, only one list (either the blacklist or the whitelist) is in use.
Advanced Properties for SAS Mobile BI

Configuring the Display of Banner-Related Items for SAS Mobile BI

Deployment of the SAS Mobile BI application configures the display of banner-related items with default values. You can customize the display of your company logo, banner, and a title in SAS Mobile BI on mobile devices.

Note: Support for this feature is discontinued in the 7.1 release of SAS Mobile BI.

To customize the display of a custom logo, banner, and title in SAS Mobile BI, follow these steps:

1. Log on to SAS Management Console.
4. Click the Advanced tab and specify the following properties and required values:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>viewerservices.company.banner.logoUrl</td>
<td>URL for the logo</td>
</tr>
<tr>
<td></td>
<td>Note: The logo web address that you specify must be accessible by the mobile device.</td>
</tr>
<tr>
<td>viewerservices.company.banner.message</td>
<td>Message for the banner</td>
</tr>
<tr>
<td>viewerservices.company.banner.title</td>
<td>Title for the banner</td>
</tr>
</tbody>
</table>

5. Click OK to close the Visual Analytics Transport Service Version Properties window.

To enable these properties to take effect, restart SAS Web Application Server.

Modifying the Value Used for Resizing Images in the Middle Tier for SAS Mobile BI

The Default Maximum Bytes Property

The viewerservices.image.default.max.bytes property represents the number of bytes that are used to determine whether server-side resizing of images occurs...
before an image is delivered to the mobile device. The types of images that are
resized include PNG, BMP, JPEG, and GIF files.

By default, this property is set to 300 KB. The limit on the size of images that are
delivered ensures both faster download times and smaller memory footprints on
the mobile device. The default value is sufficient for most environments. If this
property is set to 0, images are not resized before they are delivered to the
device.

You should modify this value only when you want to increase or decrease the
number of image bytes that can be delivered to a mobile device. Modify the
value for this property cautiously because it impacts the download time and
memory in the mobile devices.

**Determine Whether Images Are Scaled**

If your SAS software can customize image resizing, select Insert ▶ Other ▶
Image to display the Image Selection window. In that window, if the Scale type
option is set to None, any images that are delivered to mobile devices are not
scaled down to a size below the value that is specified for the
viewerservices.image.default.max.bytes property.

For the Scale type option in the Image Selection window, if you select Stretch,
Fit All, Fit Width, or Fit Height, the value that is specified for the
viewerservices.image.default.max.bytes property is not impacted.

**Modify the Maximum Number of Bytes**

To modify the number of bytes that is specified for the
viewerservices.image.default.max.bytes property, follow these steps:

1 Log on to SAS Management Console.

2 On the Plug-ins tab, select Application Management ▶ Configuration
Manager ▶ SAS Application Infrastructure ▶ Visual Analytics version.

3 Right-click Visual Analytics Transport Service version and select
Properties.

4 Click the Advanced tab, and then click Add.

5 Enter viewerservices.image.default.max.bytes in the Property
Name field.

6 Enter the number of bytes in the Property Value field.

7 Click OK to close the Define New Property dialog box.

8 Click OK to close the Visual Analytics Transport Service Version Properties
window.

To enable these properties to take effect, restart SAS Web Application Server.
Best Practices for Configuring Your Middle Tier

Sample Middle-Tier Deployment Scenarios

Overview of Middle-Tier Deployment Scenarios ........................................... 187
Scenario 1: Web Applications Deployed in a Single Web Application Server . 187
Scenario 2: Web Applications Deployed across a Web Application Server Cluster .......................................................... 190
Adding a Vertical Cluster Member ............................................................... 192
Adding a Horizontal Cluster Member .......................................................... 193
Updating a Horizontal Cluster Member ....................................................... 195
Tuning the Web Application Server ............................................................. 195
Configuring HTTP Sessions in Environments with Proxy Configurations .... 195
Resolve HTTP Session Requests in a Secure Environment ....................... 195

Sample Middle-Tier Deployment Scenarios

Overview of Middle-Tier Deployment Scenarios

This section describes sample topologies for the middle-tier components. These sample topologies can help you design a middle-tier configuration that meets the needs of your organization with regard to performance, security, maintenance, and other factors.

As with all tiers in the SAS Intelligence Platform, deployment of the middle tier involves careful planning. When you design and plan the middle tier, you must balance performance requirements against a number of other criteria.

The topologies that are presented in the following sections range from simple to complex. Scenario 1 represents the deployment that results from using the SAS Deployment Wizard to configure all the middle-tier software automatically and deploy the SAS web applications. Scenario 2 provides advanced features, such as greater security and efficiency, but can require more effort to implement and to maintain.

All scenarios include the SAS server tier. The server tier consists of a SAS Metadata Server that resides on a dedicated machine. The server tier also includes additional systems that run various SAS Application Servers, including

Scenario 1: Web Applications Deployed in a Single Web Application Server

Overview

This scenario illustrates the most basic topology. All of the SAS middle-tier components are installed on a single system. All the SAS web applications run in a single SAS Web Application Server instance.

The following figure illustrates the topology for Scenario 1.

Figure 16.1  Scenario 1: Middle-Tier on a Single System
Here are the advantages and disadvantages of this topology:

### Table 16.1 Scenario 1 Advantages and Disadvantages

<table>
<thead>
<tr>
<th>Topic</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>SAS Web Server acts as a reverse proxy and provides a layer of security. Transport Layer Security (TLS) can be enabled on the client side of SAS Web Server without affecting the work load on SAS Web Application Server or the performance of the applications.</td>
<td>Adding firewalls to the network is a good next step.</td>
</tr>
<tr>
<td>Performance</td>
<td>SAS Web Server is automatically configured to cache static content.</td>
<td>This topology does not support hundreds of concurrent users.</td>
</tr>
<tr>
<td>Scalability</td>
<td>There are no advantages in this scenario, but the topology provides an upward path to clustering web application servers.</td>
<td>This topology does not support hundreds of concurrent users.</td>
</tr>
<tr>
<td>Availability</td>
<td>None</td>
<td>This topology has no provision for planned or unplanned down time.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>The SAS Deployment Wizard can automate the configuration and deployment. This topology is simple to maintain and is ideal for development environments where frequent changes might be required.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Further Considerations for Scenario 1**

As the maintainability advantages in the previous table indicates, scenario 1 is easy to implement. This middle-tier topology can be completely installed and configured by the SAS Deployment Wizard.

A variation of this scenario is to use the SAS Deployment Wizard to add web application server instances on the same middle-tier machine. This is vertical clustering and can be configured automatically by the SAS Deployment Wizard.

Similar to clustering, the applications can be distributed to different managed servers. Distributing the applications is similar to clustering in that additional web application server instances are used. It is different in that the managed server profiles are different—single instances of the applications are distributed to web application servers rather than redundant instances. Distributing the applications enables more memory availability for the applications deployed on each managed server and also increases the number of users that can be supported. Some SAS Solutions are configured with multiple servers by the SAS Deployment Wizard automatically. However, you can choose to configure
multiple managed servers by running the wizard with the custom prompting level and selecting this feature.

**Scenario 2: Web Applications Deployed across a Web Application Server Cluster**

**Overview**

The sample topology in this scenario includes a cluster of web application servers and deploys SAS Web Server on its own machine.

The following figure illustrates the sample topology. In most cases, the instances of SAS Web Application Server and applications are identically configured. Some applications, such as SAS BI Dashboard Event Generator, and some SAS solutions applications cannot be clustered. Those are examples of when the server instances and applications are not identically configured.

*Figure 16.2  Scenario 2: Clustered Web Application Servers*

The majority of the topology can be configured automatically with SAS software. Because SAS Web Server is deployed on its own machine, it can be configured
automatically with the SAS Deployment Wizard or configured manually. Here are the advantages and disadvantages of this topology:

**Table 16.2 Scenario 2 Advantages and Disadvantages**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Security      | The SAS web applications and the web application server cluster are protected by firewalls.  
                The web application server and SAS web applications can be configured to perform web authentication for single sign-on to the applications and other web resources in the network. | None          |
| Performance   | Response time is improved because static content is cached by SAS Web Server.  
                The greater computing capacity of the web application server cluster also improves performance. | None          |
| Scalability   | Once the cluster is established, additional server instances can be added to support larger numbers of concurrent users. | None          |
| Availability  | Clustering provides fault isolation that is not possible with a single web application server. If a machine in the cluster fails, then only the users with active sessions on that machine are affected.  
                You can plan downtime for maintenance by taking some servers offline. New requests are then directed to the applications deployed on the remaining servers while maintenance is performed. | SAS Web Server remains a single point of failure. Software and hardware high-availability options exist to mitigate this disadvantage. |
| Maintainability| Configuration and deployment of the cluster and the applications can still be automated with the SAS Deployment Wizard. | Some operations, such as redeploying web applications, can require more effort when more machines are used. |

**Understanding Clusters**

In order to provide greater scalability, availability, and robustness, SAS Web Application Server supports both vertical and horizontal clustering. With clustering, multiple server instances participate in a load-balancing scheme to handle client requests. Workload distribution is managed by SAS Web Server. SAS Web Server is configured as a load-balancing HTTP proxy.
The server instances in a cluster can coexist on the same machine (vertical clustering), or the server instances can run on a group of middle-tier server machines (horizontal clustering). The web applications can be deployed on both vertical and horizontal clusters.

**Requirement for Session Affinity**

For SAS web applications to be deployed into a clustered environment, SAS Web Server implements session affinity. *Session affinity* is an association between a web application server and a client that requests an HTTP session with that server. This association is known in the industry by several terms, including session affinity, server affinity, and sticky sessions. With session affinity, once a client has been assigned to a session with a web application server, the client remains with that server for the duration of the session. By default, session affinity is enabled.

**Understanding Demilitarized Zones**

Many organizations use a series of firewalls to create a demilitarized zone (DMZ) between their servers and the client applications. A DMZ provides a network barrier between the servers and the clients. A DMZ provides this protection whether the clients reside within the organization's computing infrastructure (intranet) or reside outside the organization on the Internet.

In the previous figure, the outer firewall that connects to the public network is called the domain firewall. Typically, only the HTTP (80) and HTTPS (443) network ports are open through this firewall. Servers that reside directly behind this firewall are exposed to a wide range of clients through these limited ports, and as a result the servers are not fully secure.

An additional firewall, the protocol firewall, is configured between the non-secure machines in the DMZ and the machines in the secure middle-tier network. The protocol firewall has additional network ports open. However, the range of IP addresses that are allowed to make connections is typically restricted to the IP addresses of the servers that reside in the DMZ.

The DMZ usually contains HTTP servers, reverse proxies, and load-balancing software and hardware. Do not deploy SAS Web Application Server or any SAS servers that handle important business logic, data, or metadata in the DMZ.

If your applications are accessed by clients through the Internet, then you should include a DMZ as part of your deployment in order to safeguard critical information. For deployments on a corporate intranet, you might want to implement a DMZ as an additional layer of security.

---

**Adding a Vertical Cluster Member**

Vertical clustering is the practice of deploying multiple identically configured web application server instances on a single machine. This can assist with improving performance so long as the hardware is sufficiently powerful to run additional server instances. It can also offer some improvement for availability. In the event that one web application server instance crashes (or an application on one server instance stops), the applications remain available on the other web application server instances.
To add a vertical cluster member:

1. Stop the web application server instance and other middle-tier servers.
   
   ```
   SAS-configuration-directory\Lev1\Web\Scripts\AppServer\appsrvconfig.cmd stop
   ```

2. Locate the SAS Software Depot on the machine and start the SAS Deployment Wizard.

3. When offered the choice to install and configure software, select the check box for configuring software, clear the check box for installing software, and click **Next**.

4. Specify your plan file or select the plan that you used from the list of standard plans, and click **Next**.

5. Select the deployment step.
   
   **Note:** The listed deployment step will depend on several factors, including your plan file and middle-tier configuration.

6. When you specify the configuration directory, the wizard provides a warning that the directory contains existing files. Click **Yes** to confirm the warning.

7. On the Select Products to Configure page, select the **Clear All** check box, select the check box for **SAS Web Application Server Configuration** only, and then click **Next**.

8. On the Web Application Server: Managed Server Ports page, use the **Cluster Member Multiplier** menu to specify the number of web application server instances to configure.

   For the pages before this one, and after it, specify the same values that were entered during the initial configuration.

9. Stop the middle-tier servers again (they were started when the SAS Deployment Wizard completed).
   
   ```
   SAS-configuration-directory\Lev1\Web\Scripts\AppServer\appsrvconfig.cmd stop
   ```

10. Configure the SAS web applications and resources, such JDBC data sources and JMS queues.

    ```
    SAS-configuration-directory\Lev1\Web\Scripts\AppServer\appsrvconfig.cmd -a
    ```

    The configuration scripting tool (appsrvconfig.cmd) starts the servers when it completes.

    **TIP** Log on to SAS Environment Manager and add the new servers to your inventory.

---

**Adding a Horizontal Cluster Member**

Horizontal clustering is the practice of deploying SAS Web Application Server instances on multiple machines. This can assist with improving performance and provide greater availability to guard against hardware failure. In the event that one machine or web application server instance crashes (or an application on
one server instance stops), the applications remain available on the other
machines.

The SAS Deployment Wizard is used to add an additional middle-tier node. When it runs, it performs the following tasks:

- installs and configures a SAS Web Application Server instance
- configures SAS Web Server to load-balance HTTP requests to the new server instance
- starts the server instance

To add a horizontal cluster member:

1. On the machine that hosts SAS Web Server, make sure the SAS Deployment Agent is running. The agent can be started from `SASHome\SASDeploymentAgent\9.4\agent.bat start`.

   If the first instance of SAS Web Application Server is not installed on the same machine as SAS Web Server, then start the deployment agent on that machine too.

2. Copy the SAS software depot to the machine to use, or make sure the depot is available from a network share.

3. Start the SAS Deployment Wizard on the new machine to use. On the deployment step page, select **Middle Tier Node**.

   ![](image)

   **Figure 16.3** Select Deployment Step and Products to Install Page

   Note: You can use the **Cluster Member Multiplier** menu on the Web Application Server: Managed Server Ports page to combine vertical clustering with horizontal clustering.

4. If you disabled clustering for SAS Content Server during the configuration, the `-Dcom.sas.server.isclustered` JVM option will be set to false. In this case, on the first web application server instance that was configured with the SAS Deployment Wizard, enable the JVM option when the SAS Deployment Wizard completes as follows:

   ```
   -Dcom.sas.server.isclustered=true
   ```

   After you make this change, restart the web application server instance.

   **TIP** Log on to SAS Environment Manager and add the new machine and servers to your inventory.
Updating a Horizontal Cluster Member

After adding software to the primary middle-tier machine, you must run the SAS Deployment Manager to update the Middle Tier Nodes (horizontal cluster nodes). For more information, see “Add SAS Products to a SAS Middle-Tier Horizontal Cluster” in SAS Intelligence Platform: Installation and Configuration Guide.

Tuning the Web Application Server

In addition to specifying Java Virtual Machine options, you can improve the performance of SAS web applications by configuring other aspects of the web application server’s behavior. For example, two obvious ways to improve the performance of any web application are:

- to limit the frequency with which servers check for updated JavaServer Pages and servlets
- to make sure that the server can create sufficient threads to service incoming requests

SAS provides a set of JVM option settings in the Instructions.html file that is generated by the SAS Deployment Wizard. Use those settings as a starting point for your tuning.

For more information, see Tuning SAS Web Application Server in the SAS Web Applications: Tuning for Performance and Scalability.

Configuring HTTP Sessions in Environments with Proxy Configurations

Resolve HTTP Session Requests in a Secure Environment

SAS Web Report Studio uses absolute URL addresses that must be associated with the correct HTTP session. The SAS Logon Manager knows only the address that is stored in metadata, and the SAS Logon Manager redirects requests to that location.

If that address differs from the URL specified by the user, then the user's session is not tracked correctly. (For example, suppose the user specifies the internal address http://shortname/application instead of the external address http://shortname.example.com/application.)

When SAS Web Report Studio receives an HTTP request, the request is redirected to the SAS Logon Manager. The SAS Logon Manager authenticates the request, and redirects it back to SAS Web Report Studio.
An exception applies to this process if your environment has any front-end processor (for example, Apache HTTP Server, web clustering, IBM Tivoli Access Manager WebSEAL, or CA SiteMinder) configured. In these scenarios, or if a reverse proxy is configured with WebSEAL, the HTTP session request comes via an internal address. For example, the request might come via http://host:port/application instead of an external address http://proxiedhost/application. This sequence of events triggers a redirection filter, which typically sends the request to a location in the metadata where the request format is expected in the form of shortname.example.com. However, the redirection filter is not required because the proxy sends the request to the same location, and the same address is always used.

To ensure successful resolution of HTTP session requests in a secure environment (any environment with a front-end processor), the redirection filter must be disabled for SAS Web Report Studio. In addition, it is highly recommended that you disable this filter for all SAS applications.

To disable the redirection filter for all SAS web applications, follow these steps:

1. In SAS Management Console, navigate to **Plug-ins ▶ Application Management ▶ Configuration Manager ▶ SAS Application Infrastructure Properties** and right-click to display the SAS Application Infrastructure Properties dialog box.
2. Click the **Advanced** tab.
3. Click **Add** to display the Define New Property Window.
4. Enter the property name as shown, and specify the property value:
   
   **Property Name:** App.RedirectionFilterDisabled
   **Property Value:** True

5. Click **OK** to exit the Define New Property window.
6. Click **OK** to exit the SAS Application Infrastructure Properties dialog box.
7. To enable this change to go into effect, restart SAS Web Application Server.
Overview of High-Availability Features

The SAS middle tier can be configured for high availability. Some components, like SAS Web Application Server, can be configured in a cluster automatically. Other components, like JMS Broker, require manual configuration to enable high availability.

The following sections provide information about strategies for enabling high availability for each component in the middle tier.
SAS Web Application Server

SAS Web Application Server Clustering

You can configure a cluster of SAS Web Application Server instances to provide high availability for the SAS web applications. SAS Web Server provides load balancing to direct requests to the web application server instances. You can use the SAS Deployment Wizard to configure a vertical or horizontal cluster automatically.

SAS Web Server uses both cookies and URL encoding for session stickiness. As a result, requests are proxied to the same SAS Web Application Server instance where the session was established. Session replication across the cluster is not supported. If an instance becomes unavailable, subsequent requests are sent to a different server instance, but the original session and any data in the session are lost. Users do not need to log on again because the browser maintains a ticket granting ticket cookie from the CAS servlet in SAS Logon Manager.

SAS Environment Manager and the SAS web applications rely on the SAS Logon Manager web application for authentication. In a clustered configuration, a failure of a web application server instance that hosts SAS Logon Manager causes a brief impact to users that do not already have a session. Once SAS Web Server detects that the web application server instance is unavailable, it directs subsequent requests to available instances. There is no impact for users who already have a session. Restarting a web application server instance that hosts SAS Logon Manager does not require a restart of any other web applications that rely on it for authentication.

See Also

• “Understanding Clusters” on page 191
• “Adding a Horizontal Cluster Member” on page 193

Configuring the Prerequisite Checker for Clustered Servers

If you enabled the LifeCycle Listener in “Enabling the Prerequisite Checker” on page 41, and you are clustering any of the prerequisite servers, you must add the clustered servers to the configuration file for the LifeCycle Listener. The listener must know about the cluster, and the listener requires only one member of the cluster to be available before it lets SAS Web Application Server start.

Note: Metadata Server clusters are automatically processed by the Metadata Server quorum rules. You do not need to perform these manual steps for Metadata Server clusters.

To add cluster members, follow these steps:

1. Edit the prerequisite server configuration file, SAS-configuration-directory\Lev
  \n\Web\WebAppServer\SASServern_m\conf
  \startup.prerequisites.
2 Locate the line specifying the original server that you are now clustering. For example, if you are clustering JMS Broker, locate the following line:

```
server.domain.com 61616 60 SAS JMS Broker
```

3 Make a copy of the line and modify the host and port to match the host and service port for the additional cluster member. For example, if you are clustering JMS Broker that listens on port 61617, the entries should be as follows:

```
server.domain.com 61616 60 SAS JMS Broker
server.domain.com 61617 60 SAS JMS Broker
```

Note: The description must be identical for all members of the cluster. The cluster can contain as many servers as desired and can include both active cluster members and host standby servers, as long as the host standby servers are not listening on their service port when they are not available.

4 Save the file.

The prerequisite check will be performed the next time you restart SAS Web Application Server.

### Update the Connection to the Relational Database

SAS Web Application Server uses SAS Web Infrastructure Platform Data Server (or a third-party vendor database). The web application server is configured to test the database when it provides a new connection from the connection pool. The checks occur, at most, every 30 seconds. As a result, the web application server can recover from a failover or restart of the database but can experience up to 30 seconds of trouble connecting to the database before it recovers.

### Update the Connection to JMS Broker

If you configure JMS Broker for high availability, then you need to update the connection information in SAS Web Application Server.

To configure SAS Web Application Server for a high-availability broker connection, edit `SAS-configuration-directory\Lev1\Web\WebAppServer\SASServer1_1\conf\server.xml`. Locate the `Resource` elements that use the `org.apache.activemq.ActiveMQXAConnectionFactory` class name and update the `xaProperties.brokerURL` attribute as follows:

```
      <Resource auth="Container"
        factory="com.sas.vfabriccsvr.atomikos.BeanFactory" maxPoolSize="20"
        name="sas/jms/TopicConnectionFactory"
        type="com.atomikos.jms.AtomikosConnectionFactoryBean"
        uniqueResourceName="sas/jms/TopicConnectionFactory"
        xaConnectionFactoryClassName="org.apache.activemq.ActiveMQXAConnectionFactory"
        xaProperties.brokerURL="failover:(tcp://primary.example.com:61616,
tcp://secondary.example.com:61616)"
      />
```

Note: The `xaProperties.brokerURL` attribute must be on one line. It is shown on more than one line in the preceding code sample for display purposes only.

Note: The highlighted text in the previous code example should appear on one line and do not add space after the comma.
SAS Web Server

About SAS Web Server

SAS Web Server is used as a load balancer for distributing HTTP requests to SAS Web Application Server instances. The web server is the unique access point for customer to access all SAS web applications. It detects when an application server in the cluster is down and routes requests to other nodes. However, it does not have the capability to monitor the availability of individual web applications, or to monitor the health of an application server that is running, but might be performing poorly.

A single instance of the web server can be installed with the SAS Deployment Wizard. Additional instances must be configured manually by copying an existing instance to the machines to use. From that point, there are several options to achieve high availability:

- **Hardware strategy** You can run multiple identical web server instances behind a hardware load balancer. Because the web server is stateless, the server instances can be cloned. There is no overhead for session management. There is no failover, but the next request after the failure is directed to a running web server instance. Session stickiness to the web application server is honored by any web server instance. Multiple hardware load balancers can be used in combination with round-robin DNS (the next strategy) if you require it.

- **Round-robin DNS strategy** Multiple identical web server instances can be run on different hosts, and a special DNS name is created to resolve to multiple IP addresses. When clients resolve the name with DNS, they receive a list of IP addresses to use. Typically, the first IP address in the list is selected and some clients might use the next IP address if the connection times out. The DNS server rotates the sequence of the IP addresses that it returns with each request. Some products can be configured to drop an IP address from the list if a heartbeat stops. Round-robin DNS has some limitations but is simple and widely used.

- **Operating system strategy** You can use high-availability features in the operating system to achieve failover for the web server. Configure the web server identically on the main machine and on the hot standby. The two machines maintain a heartbeat between them. If the main machine fails or runs into difficulty, the operating system on the hot standby machine assumes the network address of the main machine and starts to service requests. Operating system failover support is available with Windows Server 2008 failover clusters and Red Hat failover domains. For other operating systems, such as IBM AIX or Oracle Solaris, there are similar functions to support high availability for failover. See your vendor documentation for more information.

Installing Additional Web Server Instances

To install additional web server instances, you can use the Install Additional Software option for the SAS Deployment Wizard and install SAS Web Server only. After the software is installed, you can copy SAS-configuration-
directory\Lev1\Web\WebServer from the primary machine to the additional machine. You need to modify the httpd.conf file so that the ServerName property matches the host name. You might need to set additional configuration options to match your network topology or to match features that are enabled in your deployment, such as HTTPS.

For the hardware-based strategy and the round-robin DNS strategy, perform the following tasks:

1. Update the connection information for each web application. For more information, see “Specifying Connection Properties” on page 64.

2. Based on the network topology or protocol change, perform the tasks that apply from “Manual Configuration Tasks” on page 135.

3. Edit the SAS-configuration-directory\Lev1\Web\WebAppServer \SASServer1_1\conf\server.xml file and specify the new connection information in the proxyName attribute for the Connector.

4. Update the server for SAS Environment Manager with the new connection information. Edit the following files and specify the correct host name and port:

   * SAS-configuration-directory\Lev1\Web \SASEnvironmentManager\server-version-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\web.xml
   * SAS-configuration-directory\Lev1\Web \SASEnvironmentManager\server-version-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\spring\security-web-context.xml

Enabling the Prerequisite Checker for Clusters

If you enabled the LifeCycle Listener in “Enabling the Prerequisite Checker” on page 41, you should update its configuration to include the additional cluster members. For information about configuring the LifeCycle Listener for clustered servers, see “Configuring the Prerequisite Checker for Clustered Servers” on page 198.

JMS Broker

About JMS Broker

The broker is based on Apache ActiveMQ. The Apache documentation offers more than one strategy for enabling high availability. One method is to use the Shared File System Master Slave configuration.

Adding Additional JMS Broker Instances

The SAS Deployment Wizard does not install or configure an additional instance of the broker. You can add an instance by archiving the component directory and extracting the archive on an additional machine.
To configure high availability for the broker:

1. In the existing Active MQ directory, edit the `SAS-configuration-directory\Lev1\Web\activemq\conf\activemq.xml` file.

2. Change the directory that is specified in the `kahadb` element. It initially references `${activemq.data}/kahadb`. Specify a directory that is shared between the machines that you want to use:

   ```xml
   <kahadb directory="/shared/directory/kahadb"/>
   ```

3. Archive `SAS-configuration-directory\Lev1\Web\activemq` with a utility like `zip` or `tar` and then extract the files on the additional machine. Use an identical directory structure.

4. Edit the `SAS-configuration-directory\Lev1\Web\WebAppServer\SASServern_m/conf/server.xml` file. Change all the `brokerURL` attributes for the resources to resemble the following example:

   ```xml
   brokerURL="failover:(tcp://primary.example.com:61616,
   tcp://secondary.example.com:61616)?randomize=false"
   ```

   **Note:** In the previous example, the code should appear on one line and do not add space after the comma.

For more information about the broker implementation, see [http://activemq.apache.org/shared-file-system-master-slave.html](http://activemq.apache.org/shared-file-system-master-slave.html).

**See Also**

"Update the Connection to JMS Broker" on page 199

---

**Enabling the Prerequisite Checker for Clusters**

If you enabled the LifeCycle Listener in “Enabling the Prerequisite Checker” on page 41, you should update its configuration to include the additional cluster members. For information about configuring the LifeCycle Listener for clustered servers, see “Configuring the Prerequisite Checker for Clustered Servers” on page 198.

---

**Cache Locator**

**Number of Installed Cache Locators**

The locator is used to tell new, connecting members like SAS Web Application Server where running members are located and provides load balancing for server use. Whether one or two locators are installed depends on your deployment topology:

- In a single machine deployment, the SAS Deployment Wizard prompts for a cache locator port on the Web Application Server: Cache Locator Configuration and Scheduling Services Cache Locator pages. If you specify different port numbers, then two locators are configured.
In a multiple machine deployment, two locators are configured. One is configured on the primary middle-tier machine and one is configured on the server-tier machine.

The SAS Deployment Wizard does not install and configure more than two locators. The two locators are peers and when one is down, the other can do all the work. The two locators provide a failover support.

Configuration Steps for Windows

To configure an additional locator for Windows deployments, follow these steps:

1. On the primary middle-tier machine, the locator software is archived at $SAS-configuration-directory\Lev1\Web\Scripts\AppServer\src\Config\vfabriccsvr\gemfire663.zip

2. Extract the archive to the identical $SAS-configuration-directory\Lev1\Web\gemfire directory on the additional machine.

3. Create an instance directory that is identical to the primary machine, for example, $SAS-configuration-directory\Lev1\Web\gemfire\instances\ins_41415.

4. Copy the files from the instance directory on the primary machine to the additional machine.

5. Copy the $gemfire\bin\winx86_64\wrapper.conf to the instance directory ($gemfire\instances\ins_41415). Update the following lines in the wrapper.conf file:

   set.GEMFIRE_HOME=../..
   set.INSTANCE_NAME=ins_41415
   set.INSTANCE_PORT=41415
   set.JAVA_HOME=$globalResource.jreHome
   set.GEMFIRE_SERVICE_NAME=SAS [Config-Lev1] SAS Cache Locator 41415
   set.GEMFIRE_LOCATORS=primary.example.com[41415],secondary.example.com[41415]

   Specify a comma-separated list of all the locators in the GEMFIRE_LOCATORS property.

6. Install Windows service for SAS Cache Locator:

   C:\SAS\Config\Lev1\Web\gemfire\bin\winx86_64\installservice.bat

7. Start the locator with the Windows service name SAS [Config-Lev1] SAS Cache Locator 41415.

8. Using the same list of locators (primary.example.com[41415], secondary.example.com[41415]), to update the following items:

   - Update the wrapper.conf file for all the previously installed locators with the complete list of locators.
   - Update the -Dsas.cache.locators JVM option for all SAS Web Application Server instances with the complete list of locators.
   - Update the -Dsas.cache.locators JVM option for all instances of the SAS Web Infrastructure Platform Scheduling Services with the complete list of locators. The change is made in the $SAS-configuration-
Configuration Steps for UNIX

To configure an additional locator for UNIX deployments, follow these steps:

1. On the primary middle-tier machine, the locator software is archived at `SAS-configuration-directory\Lev1\Web\Scripts\AppServer\src\Config\vfabriccsvr\gemfire663.zip`. Extract the archive to the identical `SAS-configuration-directory\Lev1\Web\gemfire` directory on the additional machine.

2. Copy the `SAS-configuration-directory\Lev1\Web\Scripts\AppServer\src\Config\vfabriccsvr\gemfire-sas` file from the primary middle-tier machine to the `SAS-configuration-directory\Lev1\Web\gemfire\bin` directory on the additional machine.

3. Update the following files to be executable:
   ```
dos2unix SAS-configuration-directory\Lev1\Web\gemfire\bin\gemfire
chmod 755 SAS-configuration-directory\Lev1\Web\gemfire\bin\gemfire

dos2unix SAS-configuration-directory\Lev1\Web\gemfire\bin\agent
chmod 755 SAS-configuration-directory\Lev1\Web\gemfire\bin\agent

dos2unix SAS-configuration-directory\Lev1\Web\gemfire\bin\cacheserver
chmod 755 SAS-configuration-directory\Lev1\Web\gemfire\bin\cacheserver

dos2unix SAS-configuration-directory\Lev1\Web\gemfire\bin\gfsh
chmod 755 SAS-configuration-directory\Lev1\Web\gemfire\bin\gfsh

dos2unix SAS-configuration-directory\Lev1\Web\gemfire\bin\gemfire-sas
chmod 755 SAS-configuration-directory\Lev1\Web\gemfire\bin\gemfire-sas

dos2unix SAS-configuration-directory\Lev1\Web\gemfire\lib/*.so
```

4. Create an instance directory that is identical to the primary machine, for example, `SAS-configuration-directory\Lev1\Web\gemfire\instances\ins_41415`.

5. Copy the files from the instance directory on the primary machine to the additional machine.

   Note: Do not copy the `.locator` file.

6. Update the following lines in the `instances\ins_41415\gemfire-locator.sh` file:
   ```
   GF_JAVA=/SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/java
   export GF_JAVA
   LOCATOR_HOME=/SAS-configuration-directory/Lev1/Web/gemfire
   GEMFIRE_LICENCE_KEY=6M0C3-4VW9H-M8J40-0D52F-DTM0H
   LOCATOR_PORT=41415
   LOCATORS=primary.example.com[41415],secondary.example.com[41415]
   USE_IPV4_STACK=false
   USE_IPV6_ADDRESS=false
   ```

   Specify a comma-separated list of all the locators in the `LOCATORS` property.

7. Update the `gemfire-start-locator-sas.sh` file to be executable:
   ```
   ```
Start the locator with the `instances/ins_41415/gemfire-locator.sh start` command.

Using the same list of locators (primary.example.com[41415], secondary.example.com[41415]), to update the following items:

- Update the `gemfire-locator.sh` file for all the previously installed locators with the complete list of locators.
- Update the `-D-sas.cache.locators` JVM option for all SAS Web Application Server instances with the complete list of locators.
- Update the `-D-sas.cache.locators` JVM option for all instances of the SAS Web Infrastructure Platform Scheduling Services with the complete list of locators. The change is made in the `SAS-configuration-directory/Lev1/Web/Applications/SASWIPSchedulingServices9.4/servicetrigger.ini` file.

Enabling the Prerequisite Checker for Clusters

If you enabled the LifeCycle Listener in “Enabling the Prerequisite Checker” on page 41, you should update its configuration to include the additional cluster members. For information about configuring the LifeCycle Listener for clustered servers, see “Configuring the Prerequisite Checker for Clustered Servers” on page 198.

SAS Environment Manager


An operating system-level fail over cluster can be used to replace the hardware-based load balancer that is mentioned in the VMware documentation. SAS Environment Manager can be configured to use an external database, so an operating system-level fail over cluster does not require shared storage devices. Consider for SAS deployment that as VMware indicates, the cluster detection and cache peer detection relies on multicast. Make sure that your router does not block multicast packets. Otherwise, the cluster fails to initialize properly. It is also common for virtualization technologies like VMware and Xen to not enable multicast by default.

Because SAS installs and configures SAS Environment Manager, the information provided by VMware about installation and configuration of the initial server instance and additional instances does not apply. The initial instance is installed and configured with the SAS Deployment Wizard.

To install and configure additional instances, follow these steps:

1. Use the **Install Additional Software** option for the SAS Deployment Wizard to install SAS Foundation and SAS Environment Manager on the remaining cluster machines. Use the same SASHome path that was used for the initial server instance.
2 Copy the `SAS-configuration-directory\Lev1\Web\SASEnvironmentManager` files and directories from the first machine to the remaining cluster machines.

3 Edit the server and agent property files to change the host name to the appropriate value.

4 When you follow the rest of the steps from the VMware documentation, keep in mind these two changes:
   - The load balancer needs to route traffic for the HTTPS port 7443 in addition to port 7080.
   - Steps 5 and 6 in the VMware documentation should be reversed. The servers need to be running before you configure the agents with the `hq-agent` script.

SAS Environment Manager makes calls to applications that are deployed in SAS Web Application Server. High availability for those applications is enabled when you cluster SAS Web Application Server.
Enterprise Integration

Configuring the Middle Tier to Use an Existing Customer Reverse Proxy . . . 208

Web Authentication ......................................................... 210
  About Web Authentication ........................................... 210
  Configuring Web Authentication for the First Maintenance Release ....... 210
  Configuring Web Authentication Starting with the Second Maintenance Release for SAS 9.4 ................................................. 215

Support for IBM Tivoli Access Manager WebSEAL .......................... 218
  Configuring Web Authentication ...................................... 218
  Deploying IBM Tivoli Access Manager Adapter .......................... 219
  Changing the Proxy Information for the First Maintenance Release for SAS 9.4 and Previous Releases ........................................ 220
  Changing the Proxy Information for the Second Maintenance Release for SAS 9.4 and Subsequent Releases ................................. 220
  Configuring the WebSEAL Junction .................................... 220
  Updating the Connection Information for SAS Web Applications ...... 221

Support for CA SiteMinder .................................................. 222
  Overview .................................................................. 222
  Dependencies ............................................................. 222
  Configuring the Java Cryptography Extension .......................... 223
  Configuring the Web Agent ............................................ 224
  SAS Web Application Contexts ....................................... 226
  Configuring SAS Web Application Server ................................ 227
  Configuring the Policy Server ........................................... 230

Support for Integrated Windows Authentication ............................ 231
  Overview of Integrated Windows Authentication in the Middle Tier .... 231
  Dependencies ................................................................ 232
  Verifying Prerequisites .................................................... 232
  Configuring SAS Web Application Server ................................ 234
  Configuring Web Authentication ....................................... 235
  Configuring the Microsoft Internet Explorer to Use SPNEGO ........ 236
  Configuring the Mozilla Firefox to Use SPNEGO ................. 237
  Configuring the Google Chrome to Use SPNEGO ............... 237
  (Optional) Configuring User Delegation ................................ 238
  Fallback to SAS Form-based Authentication ............................ 240

Support for TLS with Client Certificate Authentication .................. 242
  Overview of TLS with Client Certificate Authentication in the Middle Tier .......................................................... 242
  Client Certificate ................................................................ 242
  Configuring Middle-Tier Services for the Second Maintenance Release for SAS 9.4 and Previous Releases ......................... 242
Some network topologies already have a web server that is used to proxy connections. In these deployments, you can reconfigure the SAS middle tier so that it interacts with the existing web server. In these network topologies, it is simplest to keep SAS Web Server in the deployment so that it can continue to load balance connections to a SAS Web Application Server cluster.

To use an existing web server proxy:

1. Edit the `SAS-configuration-directory\Lev\Web\WebAppServer\SAServern_m\conf\server.xml` file. Change the value for the `proxyName` in the `/Service/Connector` element. Check the values for `proxyPort` and `scheme`:

   ```xml
   <Connector acceptCount="100" bindOnInit="false" connectionTimeout="20000"
            executor="tomcatThreadPool" maxHttpHeaderSize="16384"
            maxKeepAliveRequests="15" port="${bio.http.port}"
            proxyName="proxy.example.com" proxyPort="443"
            redirectPort="${bio.https.port}" scheme="https"
            useBodyEncodingForURI="true" />
   ``

   If you have more than one SAS Web Application Server instance, make the change for each one.

   **Note:** If the existing reverse proxy uses HTTPS with a site-signed certificate, import the certificate to the `SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security\cacerts` file.

2. In order to determine which URLs to proxy, review the `SAS-configuration-directory\Lev\Web\WebServer\conf\sas.conf` file.

   Each application that is identified in a pair of ProxyPass and ProxyPassReverse directives must be proxied.

3. Use SAS Management Console to specify an external connection for each SAS web application. For more information, see “Specifying Connection Properties” on page 64.
Note: Some SAS users prefer to update these values using a SAS DATA step. This approach is beyond the scope of these instructions. If you choose to modify these connections using a sas script rather than SAS Management Console, the SAS_THEME table in SharedServices DB will not be modified. It is possible to manually update this database entry. However, the simplest solution is to use SAS Management Console to modify the Themes Connection, even if you use a script to configure the rest of these values.

4 Use SAS Management Console to update the WebDAV connection information. For more information, see “Manual Configuration Tasks” on page 135.

5 Restart SAS Web Application Server.

6 Edit the following files and locate all instances of the URLs that begin with http://server:port. Modify them to point to https://server:port:

   a Edit the `SAS-configuration-directory\Lev\Web \SASEnvironmentManager\server-version-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\spring\security-web-context.xml` file. Locate the following lines and enter the correct information for your environment:

      ```xml
      <property name="loginUrl" value="https://server:ssl-port/SASLogon"/>
      <property name="casUrl" value="https://server:ssl-port/SASLogon" />
      <constructor-arg value="https://server:ssl-port/SASLogon"/>
      ```

   b Edit the `SAS-configuration-directory\Lev\Web \SASEnvironmentManager\server-version-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\web.xml` file. Locate the following lines and enter the correct information for your environment:

      ```xml
      <param-value>https://server:ssl-port/SASWebDoc</param-value>
      <param-value>https://server:ssl-port/SASEnvironmentMgrMidTier</param-value>
      <param-value>https://server:ssl-port/SASLogon/TimedOut.do?
sas_svcs_logon_LogonUrl=http://server:ev-server-port/</param-value>
      ```

   c Restart SAS Environment Manager.

7 Update the SAS Content Server JVM options with the new HTTPS URI values that were specified in Step 6. Otherwise, users will not be able to access the SAS Content Server administration console.

The following JVM options must be updated:

- `-Dsas.scs.cas.host`
- `-Dsas.scs.cas.port`
- `-Dsas.scs.cas.scheme`
- `-Dsas.scs.svc.host`
- `-Dsas.scs.svc.port`
- `-Dsas.scs.svc.scheme`

For a description of each JVM option and more information, see Table 10.1 on page 118.
Web Authentication

About Web Authentication

By default, SAS web applications use the form-based authentication that is provided by the SAS Logon Manager application. When credentials are provided to SAS Logon Manager, the credentials are sent to the SAS Metadata Server for authentication. The metadata server then authenticates the credentials against its authentication provider. The default provider is the host operating system.

As an alternative, you can configure the SAS web applications to authenticate on the middle tier. When users log on to a SAS web application, SAS Web Application Server handles the initial authentication for container-managed security.

Performing web authentication facilitates single sign-on. Most likely, your organization has several applications behind a common set of reverse proxy and HTTP servers. By having a common server handle authentication, users do not need to re-authenticate for access to each application.

See Also
For more information, see "Authentication Mechanisms" in SAS Intelligence Platform: Security Administration Guide.

Configuring Web Authentication for the First Maintenance Release

If you have server instances on multiple machines for the first maintenance release for SAS 9.4, then perform these steps on each machine. If you use vertical clustering (multiple servers on a machine), then perform these steps only once on the machine. These instructions configure every instance of SAS Web Application Server on a machine. The following list identifies some considerations:

- Before you perform this procedure, make sure that you grant administrators access to SAS Environment Manager. Once web authentication is configured, internal accounts like sasadm@saspw are unlikely to exist in the authentication provider that you use for web authentication.

- If you have users in SAS metadata that do not have a user ID on the Accounts tab, then a SAS identity will not be found after authentication to the web application server container succeeds and authorization takes place. Use SAS Management Console to create an authentication domain named web. Add an account on the Accounts tab for each of those users in the web authentication domain.
To configure web authentication for the first maintenance release for SAS 9.4, follow these steps:

**Modifying SAS Logon Manager Installation Files**

1. Edit the `SASHome\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\cas-servlet.xml` file and add the following code above the closing `</beans>` tag:

```
<bean id="principalFromRemoteAction"
        class="org.jasig.cas.adaptors.trusted.web.flow.PrincipalFromRequestRemoteUserNonInteractiveCredentialsAction"
        p:centralAuthenticationService-ref="centralAuthenticationService" />
```

**Note:** The previous bean definition must be entered on one line. It is shown on more than one line for display purposes only.

**Note:** As an alternative to updating the `cas-servlet.xml` file, you can edit the deployed file, `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\m\sas_webapps\sas.svcs.logon.war\WEB-INF\cas-servlet.xml`. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

2. Edit the `SASHome\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\login-webflow.xml` file and locate the following block:

```
<action-state id="generateLoginTicket">
  <evaluate
    expression="generateLoginTicketAction.generate(flowRequestContext)" />
  <transition on="success" to="viewLoginForm" />
</action-state>
```

Replace the previous block with the following:

```
<action-state id="generateLoginTicket">
  <evaluate
    expression="generateLoginTicketAction.generate(flowRequestContext)" />
  <transition on="success" to="remoteAuthenticate" />
</action-state>

<action-state id="remoteAuthenticate">
  <evaluate expression="principalFromRemoteAction" />
  <transition on="success" to="sendTicketGrantingTicket" />
  <transition on="error" to="viewLoginForm" />
</action-state>
```

**Note:** As an alternative to updating the `login-webflow.xml` file, you can edit the deployed file, `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\m\sas_webapps\sas.svcs.logon.war\WEB-INF\login-webflow.xml`. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

3. Edit the `SASHome\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\deployerConfigContext.xml.orig` file and add the following bean definition. Add this within the `\beans\bean[id="authenticationManager"]\property[name="credentialsToPrincipalResolvers"]\list:`
4 In the same file, add the following bean definition within the `beans` section:[code]
<bean id="authenticationManager"
   class="org.jasig.cas.adaptors.trusted.authentication.handler.support.PrincipalBearingCredentialsAuthenticationHandler" />
</bean>
[endcode]

Note: The previous bean definition must be entered on one line. It is shown on more than one line for display purposes only.

Note: As an alternative to updating the `deployerConfigContext.xml.orig` file, you can edit the deployed file, `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\sas_webapps\sas.svcs.logon.war\WEB-INF\deployerConfigContext.xml`. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

5 Edit the `SASHome\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig` file and add the following code above the closing `</web-app>` tag:

```xml
<security-constraint>
  <web-resource-collection>
    <web-resource-name>
      HTMLHostManager and HostManager commands
    </web-resource-name>
    <url-pattern>/login</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>*</role-name>
  </auth-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Tomcat Host Manager Application</realm-name>
</login-config>
```

Note: As an alternative to updating the `web.xml.orig` file, you can edit the deployed file, `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\sas_webapps\sas.svcs.logon.war\WEB-INF\web.xml`. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

Modifying SAS Visual Analytics Transport Service Installation Files

The following steps apply to deployments that distribute reports for SAS Mobile BI users. If you choose to configure Transport Service for web authentication, make sure that you use BASIC for the `auth-method`. SAS Mobile BI supports
BASIC authentication only. Guest access is not compatible with web authentication.

6 Edit the SASHome\SASVisualAnalyticsServices\version
   \Configurable\wars\sas.bitransportservices\WEB-INF
   \web.xml.orig file and remove the comment that encloses the <security-constraint> tag. The file should look similar to the following:

   <!-- uncomment and configure for Basic Auth (realm-name and role-name may need to change) -->
   <!--
   <security-constraint>
   <web-resource-collection>
     <web-resource-name>TransportLogin</web-resource-name>
     <url-pattern>/onebi/logon</url-pattern>
     <url-pattern>/rest/session/</url-pattern>
     <http-method>POST</http-method>
   </web-resource-collection>
   <auth-constraint>
     <role-name>*</role-name>
   </auth-constraint>
   </security-constraint>
   </login-config>
   -->

   Note: As an alternative to updating the web.xml.orig file, you can edit the deployed file, SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_name\sas_webapps\sas.bitransportservices.war\WEB-INF\web.xml. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

7 (Optional) For consistency, you can also set the value of the <realm-name> tag to the same value that was used for SAS Logon Manager.

Rebuilding and Redeploying Web Applications

8 Use the SAS Deployment Manager to rebuild the SAS web applications.
   Rebuild SAS Web Infrastructure Platform.
   If you modified SAS Transport Service, also rebuild Visual Analytics Services.

9 Stop SAS Web Application Server and then use the SAS Deployment Manager to redeploy the SAS Web Infrastructure Platform and Visual Analytics Services (if it was modified).

Confirming that Users Have Accounts in SAS Metadata

10 Start SAS Management Console and access the User Manager plug-in.

11 Check that each user has an account on the Accounts tab. If any user that requires access to the web applications does not, then right-click the User Manager plug-in and select Authentication Domains. Click New and specify web as the name.
For each user that does not already have an account on the **Accounts** tab, add an account with the user ID in the web authentication domain.

**(Optional) Validating the Previous Steps**

You can validate the previous steps by using “file” validation at this point. This is possible because SAS configures a UserDatabaseRealm by default in the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\conf\server.xml` file. Edit the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\conf\tomcat-users.xml` file to be similar to the following example:

```xml
<?xml version="1.0"?>
<tomcat-users>
  <role rolename="ROLE_USER" />
  <user username="sasdemo" password="Password1" roles="ROLE_USER" />
</tomcat-users>
```

**Note:** If you have more than one web application server instance, you must copy the `tomcat-users.xml` file to each one.

**Note:** You can substitute a real user account that is in SAS metadata instead of `sasdemo`. Either way, the specified user must have an account on the **Accounts** tab in metadata.

Start SAS Web Application Server and then access an application such as SAS Web Report Studio. The previous steps are valid if the following occur:

- you are challenged for credentials
- the credentials in the `tomcat-users.xml` file are accepted
- you are able to access the web application

Remember to remove the user and role information when you complete this procedure.

**Configuring the Realm for SAS Web Application Server**

Edit the `server.xml` file and locate the existing `/Server/Service/Engine/Realm` definition.

**Note:** If you have more than one web application server instance, you must make the following changes to each one.

Modify the realm information so that it accesses the system that you want to use for identity management. The following is an example for accessing an LDAP server:

```xml
<Realm className="org.apache.catalina.realm.JNDIRealm"
     allRolesMode="authOnly"
     connectionName="cn=Directory Manager,dc=example,dc=com"
     connectionPassword="******"
     connectionURL="ldap://directory.example.com:389"
     roleBase="ou=groups,dc=example,dc=com"
     roleName="cn"
     roleSearch="(uniqueMember={0})"
     roleSubtree="false"
     userPattern="uid={0},ou=people,dc=example,dc=com"
/>
```
Configuring Web Authentication Starting with the Second Maintenance Release for SAS 9.4

If you have server instances on multiple machines for the second maintenance release for SAS 9.4 or later, then perform these steps on each machine. If you use vertical clustering (multiple servers on a machine), then perform these steps only once on the machine. These instructions configure every instance of SAS Web Application Server on a machine. The following list identifies some considerations:

- Before you perform this procedure, make sure that you grant administrators access to SAS Environment Manager. Once web authentication is configured, internal accounts like sasadm@saspw are unlikely to exist in the authentication provider that you use for web authentication.

- If you have users in SAS metadata that do not have a user ID on the Accounts tab, then a SAS identity will not be found after authentication to the web application server container succeeds and authorization takes place. Use SAS Management Console to create an authentication domain named web. Add an account on the Accounts tab for each of those users in the web authentication domain.

To configure web authentication beginning with the second maintenance release for SAS 9.4, follow these steps:

**Modifying SAS Logon Manager Installation Files**

1. Edit the `$SASHome\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig` file. Scroll to the bottom of the file and remove the comment that encloses the `<security-constraint>` and `<login-config>` tags to enable the desired authentication.

   The following example displays the section that should be uncommented for Integrated Windows Authentication using SPNEGO. Starting with the third maintenance release for SAS 9.4, a similar section is also provided for form-based login.

   ```xml
   <!-- Enable SPNEGO authentication -->
   <security-constraint>
   <web-resource-collection>
     <web-resource-name>
     HTMLHostManager and HostManager commands
   ```
Note: As an alternative to updating the web.xml.orig file, you can edit the deployed file, SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\_m\sas_webapps\sas.svcs.logon.war\WEB-INF\web.xml. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

2 For BASIC authentication, uncomment the section shown above and replace SPNEGO with BASIC in the <auth-method> tag.

Modifying SAS Visual Analytics Transport Service Installation Files

The following steps apply to deployments that distribute reports for SAS Mobile BI users. If you choose to configure Transport Service for web authentication, make sure that you use BASIC for the auth-method. SAS Mobile BI supports BASIC authentication only. Guest access is not compatible with web authentication.

3 Edit the SASHome\SASVisualAnalyticsServices\version\Configurable\wars\sas.bitransportservices\WEB-INF\web.xml.orig file and remove the comment that encloses the <security-constraint> tag. The file should look similar to the following:

```xml
<security-constraint>
  <web-resource-collection>
    <web-resource-name>TransportLogin</web-resource-name>
    <url-pattern>/onebi/logon</url-pattern>
    <url-pattern>/rest/session/</url-pattern>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>*</role-name>
  </auth-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Tomcat Host Manager Application</realm-name>
</login-config>
```

Note: As an alternative to updating the web.xml.orig file, you can edit the deployed file, SAS-configuration-directory\Lev\Web\WebAppServer
4 (Optional) For consistency, you can also set the value of the <realm-name> tag to the same value that was used for SAS Logon Manager.

**Rebuilding and Redeploying Web Applications**

5 Use the SAS Deployment Manager to rebuild the SAS web applications. Rebuild SAS Web Infrastructure Platform.

   If you modified SAS Transport Service, also rebuild Visual Analytics Services.

6 Stop SAS Web Application Server and then use the SAS Deployment Manager to redeploy the SAS Web Infrastructure Platform and Visual Analytics Services (if it was modified).

**Confirming that Users Have Accounts in SAS Metadata**

7 Start SAS Management Console and access the User Manager plug-in.

8 Check that each user has an account on the **Accounts** tab. If any user that requires access to the web applications does not, then right-click the User Manager plug-in and select **Authentication Domains**. Click **New** and specify **web** as the name.

9 For each user that does not already have an account on the **Accounts** tab, add an account with the user ID in the **web** authentication domain.

**(Optional) Validating the Previous Steps**

10 If you specified BASIC authentication in Step 2 on page 216, you can validate the previous steps by using "file" validation at this point. This is possible because SAS configures a **UserDatabaseRealm** by default in the **SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\conf\server.xml** file. Edit the **SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\conf\tomcat-users.xml** file to be similar to the following example:

   ```xml
   <?xml version="1.0"?>
   <tomcat-users>
     <role rolename="ROLE_USER" />
     <user username="sasdemo" password="Password1" roles="ROLE_USER" />
   </tomcat-users>
   ```

   **Note:** If you have more than one web application server instance, you must copy the tomcat-users.xml file to each one.

   **Note:** You can substitute a real user account that is in SAS metadata instead of **sasdemo**. Either way, the specified user must have an account on the **Accounts** tab in metadata.

11 Start SAS Web Application Server and then access an application such as SAS Web Report Studio. The previous steps are valid if the following occur:

- you are challenged for credentials
- the credentials in the tomcat-users.xml file are accepted
you are able to access the web application

Remember to remove the user and role information when you complete this procedure.

Configuring the Realm for SAS Web Application Server

Edit the server.xml file and locate the existing /Server/Service/Engine/Realm definition.

Note: If you have more than one web application server instance, you must make the following changes to each one.

Modify the realm information so that it accesses the system that you want to use for identity management. The following is an example for accessing an LDAP server:

```xml
<Realm allRolesMode="authOnly" className="org.apache.catalina.realm.LockOutRealm">
  <Realm className="org.apache.catalina.realm.JNDIRealm"
   connectionName="cn=Directory Manager,dc=example,dc=com"
   connectionPassword="******"
   connectionURL="ldap://directory.example.com:389"
   roleBase="ou=groups,dc=example,dc=com"
   roleName="cn"
   roleSearch="(uniqueMember={0})"
   roleSubtree="false"
   userPattern="uid={0},ou=people,dc=example,dc=com"/>
</Realm>
```

TIP This sample realm replaces the UserDatabaseRealm inside the LockoutRealm. For more information, see http://tomcat.apache.org/tomcat-4.0-doc/realm-howto.html.

TIP If you are unsure of the LDAP schema in use, a utility like ldapsearch or an LDAP browser can help you identify the values to use in your deployment.

Start SAS Web Application Server.

Make a copy of all the files that you changed in the first part of this procedure. These files can be overwritten when you apply a maintenance release.

Support for IBM Tivoli Access Manager WebSEAL

Configuring Web Authentication

Follow the steps in the "Web Authentication" procedure, with the following changes:
Specify `AMTomcatAuthenticated` for the role-name element in the `web.xml.orig` file.

Do not add users to `tomcat-users.xml` or configure a `Realm` in the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\conf\server.xml` file.

### Deploying IBM Tivoli Access Manager Adapter

This adapter provides integration between IBM Tivoli Access Manager and SAS Web Application Server to achieve single sign-on for applications hosted on SAS Web Application Server. To download and deploy the adapter, follow these steps:


2. Extract the `AMTomcatValue.jar` file from the archive and deploy it to `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\lib`.

   If your deployment includes additional server instances, deploy the JAR file to the `lib` directory for each server instance.

3. Edit the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\conf\server.xml` file and locate the existing `/Engine` definition. The following example shows the `/Engine` definition. Add the following valve definition:

   ```xml
   <Engine defaultHost="localhost"
   jvmRoute="288ec8cc0d087476e58c03d1a_SASServer1_1"
   name="Catalina">
   <Realm className="org.apache.catalina.realm.LockOutRealm">
   <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
   resourceName="UserDatabase"/>
   </Realm>
   <Host appBase="webapps" autoDeploy="false" deployOnStartup="true"
      deployXML="true" name="localhost" unpackWARs="true">
   <Valve className="org.apache.catalina.valves.AccessLogValve"
   directory="logs" pattern="%h %l %u %t "%r" %s %b"
   prefix="localhost_access_log." suffix=".txt"/>
   <Valve className="com.ibm.tivoli.integration.am.catalina.valves.AMTomcatValve" fallThrough="true"/>
   </Host>
   </Engine>
   
   Note: The `fallThrough` attribute must be set to `true`. If you have more than one server instance, you must make the change to each one.
Changing the Proxy Information for the First
Maintenance Release for SAS 9.4 and Previous
Releases

In the server.xml file, the proxyName and proxyPort values must be changed to
correspond to the WebSEAL host name and port number. For more information
about how to update these values, see Step 1 on page 208.

Changing the Proxy Information for the Second
Maintenance Release for SAS 9.4 and Subsequent
Releases

Edit the SAS-configuration-directory\Lev\Web\WebAppServer
\SASServern_m\conf\server.xml file. Add the following valve definition
immediately following the AMTomcatValve definition that was added in Step 3 on
page 219, specifying the WebSEAL host name, port number, and scheme (http
or https):

```xml
<Valve className="com.sas.vfabrictcsvr.valves.WebSEALRequestWrappingValve"
  host="hostname"
  port="port_number"
  scheme="http(s)" />
```

The valve also supports an option for specifying the name of the user header
processed by the Access Manager Adapter. This will default to "iv-user" but you
can specify the user header by adding the userHeader="user_header"
parameter to the valve definition shown above.

Configuring the WebSEAL Junction

For SAS 9.4, transparent junctions with no special consideration are supported.
Beginning with the second maintenance release for SAS 9.4, TCP junctions are
supported only with the use of a junction mapping table.

Create a TCP WebSEAL junction that uses the host name and port number on
which SAS Web Application Server is listening, with a command that is similar to
the following:

```
padmin> server task default-webseald-host_name create -t tcp -c iv-user,
  iv-groups -b ignore -h saswebserver.example.com -p 80 /junction_name -I
```

**Note:** Be sure to use the `-I` (capital i) argument to ensure unique Set-Cookie
name attributes.

Modify the Junction Mapping Table (JMT) to include the following entries:

```
/junction_name */FolderModule/*
/junction_name */SASAdmin/*
/junction_name */SASAuthorizationServices/*
/junction_name */SASBIDashboard/*
/junction_name */SASBIDashboardEventGen/*
/junction_name */SASBIPortlets/*
/junction_name */SASBIWS/*
/junction_name */SASContentServer/*
/junction_name */SASDeploymentBackup/*
/junction_name */SASEnvironmentMgrMidTier/*
```
These entries represent most of a SAS Enterprise Business Intelligence deployment. Look at the `SAS-configuration-directory\Lev\Web\WebServer\conf\sas.conf` file for the application context roots that are in your deployment.

Note: Do not include `/SASWIPClientAccess` in the junction mapping table. If you protect this web application, then desktop applications like SAS Management Console cannot authenticate. Also, do not include `/SASWIPSoapServices`. If you include `/SASBIWS`, make sure that custom applications can perform BASIC authentication.

Note: For those web applications that enable you to sign in again after exiting, you might need to add an additional entry in the junction mapping table to handle this redirection. The additional entry would be the same entry for the web application, but without the trailing slash. The following example shows the initial entry and the additional entry for a web application in the junction mapping table:

```
/junction_name */SASWebReportStudio/*
/junction_name */SASWebReportStudio*
```

Load the JMT with a command that is similar to the following:

```
padmin> server task default-webseald-host_name jmt load
```

See Also

*IBM Tivoli Access Manager for e-business WebSEAL Administration Guide*

**Updating the Connection Information for SAS Web Applications**

When users authenticate through WebSEAL, it adds headers to the request to indicate that the user has already authenticated. When SAS Web Application Server receives the request, the security module intercepts the request and determines that the user was authenticated. The security module sets a principal in the request with the user name that was authenticated and the role...
AMTomcatAuthenticated. In order for user’s requests to be directed back through the WebSEAL server, the external connection information for each SAS web application must reference the WebSEAL server.

Follow the instructions for configuring the External Connection at “Specifying Connection Properties” on page 64. Make sure that you also specify the -Dsas.retry.internal.url=true JVM option that is identified on that page.

Support for CA SiteMinder

Overview

SAS 9.4 support for CA SiteMinder requires configuring a Web Agent to communicate with SAS Web Server and a custom security module for SAS Web Application Server. SAS provides the custom security module. Successful authentication results in a security token (SMSESSION) being set in the user’s web browser cookies. The security module receives the security token in the request and communicates with the policy servers through an API to decode the user credentials from the security token. This works in conjunction with web authentication to integrate with existing CA SiteMinder single sign-on environments.

Dependencies

SAS 9.4 integration with CA SiteMinder depends on two software applications from CA:

- CA SiteMinder Web Agent (any version)
- CA SiteMinder SDK r12.x

The software applications are not included with SAS software. They can be downloaded from the CA support page. (Downloading the packages requires a CA support account and license.)

The application server security module has a run-time dependency on the SDK. For Java agents, CA provides two distinct implementations of the API. Either implementation can be used by including the API JAR file shown below in the classpath. However, the detailed instructions that follow describe how to use the Pure Java API (smagentapi.jar in the following table).

Table 18.1 API JAR Files and Dependencies

| smjavaagentapi.jar | smjavasdk2.jar | This JAR file requires setting the library path to the SDK and Web Agent native libraries in the Java process that runs SAS Web Application Server. You can share the SmHost.conf configuration file with the Web Agent. |
This JAR file requires the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files.

Create two host configurations. Configure one for the Web Agent to use with SAS Web Server and a separate one for the agent to use with SAS Web Application Server. The following table shows the sample values that are used in the following sections.

**Table 18.2 Sample Values for Agent Configurations**

<table>
<thead>
<tr>
<th><strong>Property Name</strong></th>
<th><strong>Value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy server</td>
<td>policyserver.example.com</td>
</tr>
<tr>
<td>Admin user name</td>
<td>siteminder</td>
</tr>
<tr>
<td>Admin password</td>
<td>Pass</td>
</tr>
<tr>
<td>Host configuration and host name</td>
<td>hostname_apache for the web server</td>
</tr>
<tr>
<td></td>
<td>hostname_tc for SAS Web Application Server</td>
</tr>
<tr>
<td>Agent name</td>
<td>sasagent</td>
</tr>
<tr>
<td>Agent configuration</td>
<td>sasagentconf</td>
</tr>
</tbody>
</table>

For information about configuring the agents and the policy servers, see the CA SiteMinder product documentation.

**Configuring the Java Cryptography Extension**

The CA SiteMinder Pure Java API requires that the Java environment used by SAS Web Application Server to be updated with the JCE.

To configure the extension on systems other than AIX, follow these steps:


2. Extract the archive. In the jce directory, extract all three files (local_policy.jar, README.txt, US_export_policy.jar) to \JAVA_HOME\jre\lib\security.

   Note: The default JRE is located in \SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre.

Configuring the Web Agent

Purpose
You can use this information to configure SAS Web Server with a web agent. This can be necessary if your site does not already have a web server that is configured with a web agent or the existing web agent is in a different top-level domain (company.com versus organization.com).

Note: If your site already has a web server that is configured with a web agent, you can skip to “SAS Web Application Contexts” on page 226.

The custom security module for CA SiteMinder relies on using SAS Web Server as a reverse proxy. The SAS Web Server can be configured with the Web Agent plug-in module for Apache HTTP Server. The following sections describe how to perform this configuration. The Web Agent software must already be installed.

Note: SiteMinder provides a configuration utility. However, on Windows, it does not recognize SAS Web Server, so manual configuration is necessary.

Registering the SAS Web Server Host
To register the machine with the CA SiteMinder policy server, follow these steps:

1. On Windows 64-bit platforms only, copy the ICE_JNIRegistry.dll from c:\Windows\System32 to c:\Windows\SysWOW64.

2. Run the smreghost.bat command in the bin directory under the Web Agent installation to register the host with the policy servers. On UNIX, make sure you source the ca_wa_env.sh script first.

   smreghost -i policyserver.example.com -u siteminder -p Pass -hc hostname_apache -hn hostname_apache -o -f ../config/SmHost.conf

   If successful, the command generates the SmHost.conf file.

Configuring SAS Web Server for the Web Agent

TIP You can try to use the CA SiteMinder Web Agent installer. If it does not detect SAS Web Server, then follow the manual steps in this section.

To configure the server manually, follow these steps:

1. Create a WebAgent.conf file in the SAS-configuration-directory\Lev\Web\WebServer\conf directory. Make sure that it specifies the path to the SmHost.conf file that was generated earlier and that the agent name is correct. See the following example:

   HostConfigFile="C:\Program Files (x86)\CA\webagent\config\SmHost.conf"
   AgentConfigObject="sasagentconf"
   EnableWebAgent="YES"
   ServerPath="c:\SAS\Config\Lev1\Web\WebServer\conf"
   LoadPlugin="C:\Program Files (x86)\CA\webagent\bin\HttpPlugin.dll"
   AgentIdFile="C:\SAS\Config\Lev1\Web\WebServer\conf\AgentId.dat"

For UNIX deployments, the library for the LoadPlugin property is named libHttpPlugin.so instead of HttpPlugin.dll.
2 Edit the `SAS-configuration-directory\Lev\Web\WebServer\conf\httpd.conf` file. Add lines that are similar to the following at the beginning of the LoadModule directives:

```plaintext
LoadModule sm_module "C:/Program Files (x86)/CA/webagent/bin/mod_sm22.dll"
SmInitFile "C:/SAS/Config/Lev1/Web/WebServer/conf/WebAgent.conf"
```

For UNIX deployments, the name of the library is `libmod_sm22.so` instead of `mod_sm22.dll`.

3 Add lines that are similar to the following in the Aliases section. Change the paths to match the location of the Web Agent software on your machine.

```plaintext
<IfModule alias_module>
  Alias /siteminderagent/nocert/[0-9]+/(.*) "C:/Program Files (x86)/CA/webagent/$1"
  <Directory "C:/Program Files (x86)/CA/webagent/$1">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>

  Alias /siteminderagent/pw cgi/ "C:/Program Files (x86)/CA/webagent/pw/"
  <Directory "C:/Program Files (x86)/CA/webagent/pw/">
    Options Indexes MultiViews ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>

  Alias /siteminderagent/pw/ "C:/Program Files (x86)/CA/webagent/pw/"
  <Directory "C:/Program Files (x86)/CA/webagent/pw/">
    Options Indexes MultiViews ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>

  Alias /siteminderagent/ "C:/Program Files (x86)/CA/webagent/samples/"
  <Directory "C:/Program Files (x86)/CA/webagent/samples/">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
</IfModule>
```

4 Restart SAS Web Server.

**Troubleshooting the Web Agent For SAS Web Server**

If SAS Web Server does not start or generates errors, use the following information to assist with troubleshooting.

1 Create a `WebAgentTrace.conf` file in `SAS-configuration-directory\Lev\Web\WebServer\conf`. Include the following lines:

```
  components:  AgentFramework, HTTPAgent, WebAgent
```
Use the CA SiteMinder Administrative UI to set the trace properties for the agent configuration. The following table provides sample values:

**Table 18.3 Sample Values for CA SiteMinder Web Agent Troubleshooting**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TraceAppend</td>
<td>Yes</td>
</tr>
<tr>
<td>TraceConfigFile</td>
<td>C:\SAS\Config\Lev\Web\WebServer\conf\WebAgentTrace.conf</td>
</tr>
<tr>
<td>TraceFile</td>
<td>Yes</td>
</tr>
<tr>
<td>TraceFileName</td>
<td>C:\SAS\Config\Lev\Web\WebServer\logs\webagent.trace</td>
</tr>
<tr>
<td>TraceFileSize</td>
<td>100</td>
</tr>
</tbody>
</table>

**SAS Web Application Contexts**

If you already have a reverse proxy that is configured, you must modify it to proxy the SAS web applications. You can use the `SAS-configuration-directory\Lev\Web\WebServer\conf\sas.conf` file as a starting point.

If you use the file, make a copy and make sure that you perform the following edits:

- Change all host name references from the machine where SAS Web Application Server is installed to the machine where SAS Web Server is installed, in the `ProxyPass` and `ProxyPassReverse` directives.
- Change the host name in the `BalancerMember` and `ProxySet` directives to use the SAS Web Server machine.

The following is a portion of the configuration file that shows the changes:

```
ProxyPass /SASLogon balancer://SAS_Web_Server_Cluster/SASLogon
ProxyPassReverse /SASLogon balancer://SAS_Web_Server_Cluster/SASLogon
...

<Proxy balancer://SAS_Web_Server_Cluster>
  BalancerMember http://<SAS_Web_Server:80 route=SAS_Web_Server_SASServerm
</Proxy>
```

This modified sas.conf file must be added to the httpd.conf file of reverse proxy server.

Here is an example:

```
<IfModule mod_proxy.c>
  Include conf/sas.conf
</IfModule>
```
Configuring SAS Web Application Server

Considerations for Multiple SAS Web Application Server Instances
If you have more than one instance of SAS Web Application Server, perform the steps in the following sections for each server instance.

Configuring Web Authentication
Follow the steps in the “Web Authentication” procedure, but specify SiteMinderAuthenticated for the role-name element in the web.xml.orig file.

Registering the Host for SAS Web Application Server
A host configuration object must be configured on the policy server for each host that runs SAS Web Application Server. Use a separate host configuration from the Web Agent used for SAS Web Server, even if the web server runs on the same host as SAS Web Application Server.

To register the machine with the CA SiteMinder policy server, follow these steps:

1. Check the smreghost.bat command in the bin directory where the CA SiteMinder SDK is installed. Check the values for the following variables:
   - JAVA_HOME
     Make sure this identifies an installation of Java. You can use SASHOME \SASPrivateJavaRuntimeEnvironment\9.4\jre.
   - SM_REGHOST_CLASSPATH
     Make sure this path includes the smagentapi.jar and cryptoj.jar files. They are located in the CA SiteMinder SDK java or java64 directories.

2. Run the script to register the host with the policy servers. On UNIX, make sure you source the ca_wa_env.sh script first.

   smreghost.bat -i policyserver.example.com -u siteminder -p Pass
   -hc hostname_tc -hn hostname_tc
   -o -f "C:\SAS\Config\Lev1\Web\WebAppServer\SASServer1_1\conf\SmHost.conf"

   If successful, the command generates the SmHost.conf file.

Configuring SAS Web Application Server for the Web Agent
To configure the server, follow these steps:

1. Edit the SAS-configuration-directory\Levn\Web\WebAppServer\SASServer\conf\server.xml file and locate the existing /Engine definition. Add the following valve definition:

   <Valve
       className="com.sas.svcs.security.vfabriccsvr.siteminder.SiteMinderValve"
       role="SiteMinderAuthenticated"
       agentName="sasagent"
       webagentConf="${catalina.base}/conf/WebAgent.conf" />

Support for CA SiteMinder 227
### Table 18.4 SiteMinder Security Module Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default Value</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
<td>SiteMinderAuthenticated</td>
<td>Specifies the name of the role to add to authenticated principals.</td>
<td>No</td>
</tr>
<tr>
<td>agentName</td>
<td>None</td>
<td>Specifies the name of the agent that was specified in the SiteMinder Administrator UI.</td>
<td>Yes</td>
</tr>
<tr>
<td>webagentConf</td>
<td>None</td>
<td>Specifies the path to the WebAgent.conf file.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2 Copy the sas.svcs.security.vfabriccsvr.siteminder.jar file from the SASHOME\SASWebApplicationServer\9.4\templates\sas\lib directory to SAS-configuration-directory\Lev1\Web\WebAppServer\SASServern_m\lib.

3 Create a WebAgent.conf file in the SAS-configuration-directory\Lev1\Web\WebAppServer\SASServern_m\conf directory. Make sure that it specifies the path to the SmHost.conf file that was generated earlier and that the agent config object is correct. See the following example:

   ```
   HostConfigFile="C:\SAS\Config\Lev1\Web\WebAppServer\SASServer1_1\conf\SmHost.conf"
   AgentConfigObject="sasagentconf"
   EnableWebAgent="YES"
   ```

   Note: If you are configuring CA SiteMinder for SAS Mobile BI, the value that you specify for the AgentConfigObject attribute must correspond to the agentName attribute that you specified in Step 1 on page 227.

4 In the same server.xml file, check the values for the proxyName and proxyPort in the existing /Connector definition. If you are using an external proxy, change the values so that they match the proxy instead of SAS Web Server.

5 Add the smagentapi.jar and cryptoj.jar files to the classpath using the following information, or copy the files to the lib directory for each server instance.

   For Windows deployments, edit the SASServern_m\conf\wrapper.conf file and make changes that are similar to the following example:

   ```
   wrapper.java.classpath.10=C:\Program Files (x86)\CA\sdk\java\smagentapi.jar
   wrapper.java.classpath.11=C:\Program Files (x86)\CA\sdk\java\cryptoj.jar
   ```
For UNIX deployments, edit the *SASServer_m\bin\setenv.sh* file and make changes that are similar to the following example:

```bash
CLASSPATH="/opt/CA/sdk/java/smagentapi.jar:/opt/CA/sdk/java/cryptoj.jar"
```


**Troubleshooting the Security Module for SAS Web Application Server**

If SAS Web Application Server does not start or generates errors, use the following information to assist with troubleshooting.

1. Edit the *SASServer_m\lib\log4j.xml* file and add the following lines:

   ```xml
   <category name="com.sas.svcs.security.vfabriccsvr">
     <priority value="DEBUG"/>
   </category>
   ```

2. Restart SAS Web Application Server and monitor the *SASServer_m\logs\server.log* file.

3. If the server is configured correctly to use the value, the log contains messages like the following example:

   ```
   yyyy-mm-dd 10:14:57,314 DEBUG (main) [SiteMinderValve] Valve starting...
   yyyy-mm-dd 10:14:59,243 DEBUG (main) [SiteMinderValve] AgentAPI getConfig successful
   yyyy-mm-dd 10:14:59,265 DEBUG (main) [SiteMinderValve] AgentAPI successfully initialized
   yyyy-mm-dd 10:14:59,270 DEBUG (main) [SiteMinderValve] AgentAPI doManagement successful
   yyyy-mm-dd 10:14:59,270 DEBUG (main) [SiteMinderValve] Valve initialization complete
   10:14:59,354 | INFO | [Catalina] | Server startup in 1422471 ms
   ```

   **Note:** The class name is shorted to SiteMinderValue for readability in the example.

4. After a successful logon attempt, the log contains messages like the following example:
Configuring the Policy Server

Configuring the Realm

In the CA SiteMinder Administrative UI, configure the realm used by the Web Agent for SAS Web Server, if you used it, and the Web Agent that is used for SAS Web Application Server.

If you used an existing reverse proxy instead of SAS Web Server, the SiteMinder domain, realm, rule, and policy should be configured from the SiteMinder Administrative UI. Use a resource filter that protects /SASLogon/login only. This is essential to internal web service calls between SAS web applications so that they are not blocked at the proxy by the Web Agent.

Configuring a single resource filter also keeps performance as high as possible. If you want to protect every SAS web application with CA SiteMinder, then you must create a separate realm and filter for each web application that is accessed with a web browser. (For example, /SASWebReportStudio, /SASAdmin, /SASPortal, and so on.).

Here are the high-level steps:

- Create a domain for the reverse proxy server.
- Add the user directory to the domain.
- Create a realm under the domain. Select the agent from the menu. Check that the resource filter is /SASLogon/login.
- Create a rule with the resource specified as *. When you view the rule that you generated, the attribute value for the Effective Resource should appear as follows:

  `agent_name/SASLogon/login*`

- Create a policy and add users from the user directory that you defined in the domain. Add the rule that you defined to the policy.
Repeat the preceding high-level steps for SAS Web Application Server.

If you plan to use CA SiteMinder authentication for the SAS applications that are listed in the following table, you also must create a realm and filter to protect the corresponding resources:

Table 18.5  Resources to Protect

<table>
<thead>
<tr>
<th>SAS Application</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS BI Web Services</td>
<td>/SASBIWS</td>
</tr>
<tr>
<td>SAS Mobile BI</td>
<td>/SASVisualAnalyticsTransport/onebi/logon</td>
</tr>
<tr>
<td></td>
<td>/SASVisualAnalyticsTransport/rest/session</td>
</tr>
</tbody>
</table>

**Special Considerations for Agent Configuration Parameters**

The following table identifies some agent configuration parameters that are known to cause problems in a SAS deployment:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadUrlChars</td>
<td>This parameter is used by the Web Agent to reject requests that have certain characters in them. This parameter interferes with the DAV requests that are used by SAS Content Server. You can remove the parameter or modify it to allow all the characters that are used in the DAV requests.</td>
</tr>
<tr>
<td>RequiredCookies</td>
<td>This parameter can interfere with clients that use SiteMinder authentication to SAS web services. Set this parameter to no if access to web services is affected.</td>
</tr>
</tbody>
</table>

**Support for Integrated Windows Authentication**

**Overview of Integrated Windows Authentication in the Middle Tier**

Integrated Windows Authentication (IWA) is a Microsoft technology that is used in an environment where users have Windows domain accounts. With IWA, the credentials (user name and password) are hashed before being sent across the network. The client browser proves its knowledge of the password through a cryptographic exchange with the web application server.

The key components of IWA in the middle tier are an Active Directory Controller machine (Windows 2000 Server or higher), a Kerberos Key Distribution Center (KDC) in a Domain Controller machine, a machine with a client browser, and SAS Web Application Server.
When IWA is used in conjunction with Kerberos, IWA enables the delegation of security credentials. Kerberos is an industry-standard authentication protocol that is used to verify user or host identity. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection.

When Active Directory is installed on a Domain Controller running Windows 2000 Server (or higher), and the client browser supports the Kerberos authentication protocol, Kerberos authentication is used. Use of the Kerberos protocol is determined by the following requirements:

- The client must have a direct connection to Active Directory.
- Both the client and the server must have a trusted connection to a Key Distribution Center (KDC) and be compatible with Active Directory.
- Service principal names (SPNs) are required for multiple worker processes.

### Dependencies

Review the following list of software requirements and required information:

- An Active Directory Domain Controller that is running Windows 2000 Server or higher is needed.
- The desktops for users must be Microsoft Windows 2000 (or higher) domain members and have a browser client that supports the SPNEGO authentication mechanism. Microsoft Internet Explorer Version 7.0 or later qualifies as the client.
- The clock on the desktop machines, the domain controller, and the machine for SAS Web Application Server should be synchronized to within five minutes.
- The machine that is used for SAS Web Application Server must have the service principal name (SPN) registered with Active Directory. If you request this from your information technology support group, also request the following:
  - keytab file
  - the user name that the principal is mapped to
- Understand the organization of users and groups in your Active Directory deployment if you plan to use organizational unit or group information for authorizing access to the SAS web applications.

### Verifying Prerequisites

#### Verifying the Kerberos Service Principal Name

Active Directory provides support for service principal names (SPN). SPNs are a key component in Kerberos authentication. SPNs are unique identifiers for services running on servers. Every service that uses Kerberos authentication needs to have an SPN set for it so that clients can identify the service on the network. An SPN usually matches the pattern of HTTP/hostname.example.com. You must confirm that an SPN for the machine used with SAS Web Application Server is registered in the Kerberos realm. If an SPN is not set for a service,
clients have no way of locating that service. Without correctly set SPNs, Kerberos authentication is not possible.

To verify that the SPN for the service is registered, follow these steps:

1. Verify that there is a mapping already configured:
   
   ```bash
   setspn -F -Q HTTP/hostname.example.com
   ```

   **Output 18.1  Sample SPN Query**
   
   ```
   CN=user-logon-name,OU=Service Accounts,OU=Domain Controllers,OU=Servers,DC=EXAMPLE,DC=com
   HTTP/hostname.example.com
   HTTP/HOSTNAME
   ```

   Existing SPN found!

   If an SPN is not found, then contact your information technology support group for assistance with registering the machine.

2. Verify that the service is linked to the service account:
   
   ```bash
   setspn -L user-logon-name
   ```

   **Output 18.2  Sample Account Query**
   
   Registered ServicePrincipalNames for CN=user-logon-name,OU=Service Accounts,OU=Servers,DC=EXAMPLE,DC=com:
   
   ```
   HTTP/hostname.example.com
   HTTP/hostame
   ```

   The value for `user-logon-name` is the same one identified in the CN from the previous command output, or as the sAMAccountName on the service account in Active Directory.

### Verifying the Kerberos Keytab File

A keytab is a file containing pairs of Kerberos principals and encrypted keys. The keys are derived from the Kerberos password. The keytab file contains the information for SAS Web Application Server to authenticate to the Key Distribution Center (KDC). You can get the keytab file from your information technology support group. The file is in the SAS private JRE. The file must be copied to the machine used for SAS Web Application Server. The file must be readable by the user account running SAS Web Application Server. The file should not be readable by other accounts.

The command for verifying a key tab depends on the operating environment:

**Windows Specifics:**

```bash
ktab.exe -l -k FILE:hostname.keytab
```

**UNIX Specifics:**

```bash
ktutil
```

```
rkt path-to/hostname.keytab
```
list -e

slot KVNO Principal
--- ---- -------------------------------------------------------------
1  3  HTTP/hostname@example.com (arcfour-hmac)

**TIP** The encryption type or types (arcfour-hmac) is used in the next section for configuring SAS Web Application Server.

Verify that Kerberos authentication succeeds. Use the `kinit` command that is provided in the \SASHOME\SASPrivateJavaRuntimeEnvironment\9.4\jre\bin\ directory.

```
kinit -k -t c:\path-to-hostname.keytab\hostname.keytab
 user-principal-name -J-Djava.security.krb5.conf=
 path-to-Kerberos-file.conf
```

Note: The kinit command is shown on more than one line in the preceding code sample for display purposes only.

Be sure the results are similar to the following:

```
New ticket is stored in cache file C:\path
```

For more information about the `ktab.exe` or `ktutil` commands, see the vendor documentation.

### Configuring SAS Web Application Server


Repeat the following steps for each SASServer\_m instance:

1. If the machine already has a Kerberos configuration file, such as either \c:\windows\krb5.ini or /etc/krb5.conf, you can use the existing file.

2. If you do not have an existing Kerberos configuration file, you can create a `krb5.ini` file with contents that are similar to the following example:

   ```
   [libdefaults]
   default_realm = EXAMPLE.COM
   forwardable=true
   
   [realms]
   EXAMPLE.COM = {
     kdc = domain-controller.com
   }
   
   [domain_realm]
   example.com = EXAMPLE.COM
   .example.com = EXAMPLE.COM
   ```

3. Set the following JVM option to point to your Kerberos configuration file:

   ```
   -Djava.security.krb5.conf=c:/path-to-krb5.ini
   ```

   Note: If your Kerberos configuration file is saved in the \SASconfiguration-directory\Lev1\Web\WebAppServer\SASServer\_m\conf directory, you do not have to set the JVM option.
Note: Java expects forward slashes in the keytab pathname, even on Windows systems.

Note: For Windows deployments, edit the setenv.cmd and SAS-configuration-directory\Lev1\Web\WebAppServer\SASServern_m\conf\wrapper.conf files. For UNIX deployments, edit the SAS-configuration-directory/Lev1/Web/WebAppServer/SASServern_m/conf/setenv.sh file.

4 If AES256 encryption ciphers are used, be aware that they require using the Java Unlimited Strength Cryptography Extension. For more information, see "Configuring the Java Cryptography Extension" on page 223.

5 Edit the SAS-configuration-directory\Lev1\Web\WebAppServer\SASServern_m\conf\jaas.config file. Add the following to the end of the file:

```java
com.sun.security.jgss.krb5.initiate {
    com.sun.security.auth.module.Krb5LoginModule required
doNotPrompt=true
    principal="HTTP/hostname.example.com@EXAMPLE.COM"
    useKeyTab=true
    keyTab="C:/path-to-hostname.keytab"
    storeKey=true;
};

com.sun.security.jgss.krb5.accept {
    com.sun.security.auth.module.Krb5LoginModule required
doNotPrompt=true
    principal="HTTP/hostname.example.com@EXAMPLE.COM"
    useKeyTab=true
    keyTab="C:/path-to-hostname.keytab"
    storeKey=true;
};

Note: Java expects forward slashes in the keytab pathname, even on Windows systems.
```

6 Edit the SAS-configuration-directory\Lev1\Web\WebAppServer\SASServern_m\conf\server.xml file and add the following:

```xml
<Realm className="com.sas.vfabriccsvr.realm.GSSContextEstablishedRealm"
    allRolesMode="authOnly"/>
```

Note: For deployments that include SAS Mobile BI, you must modify the realm so that it can also be used for BASIC authentication.

**Configuring Web Authentication**

Follow the steps in the "Web Authentication" on page 210 task, ensuring that you specify SPNEGO as the auth-method in the web.xml file for SAS Logon Manager.

**TIP** These changes should be made to the same section of web.xml that is required to implement web authentication. You can make the changes to the web.xml.orig file as described in that task.
Configuring the Microsoft Internet Explorer to Use SPNEGO

Configuring Security Settings
To configure the security settings, follow these steps:

2. Select Local intranet and then click Sites.
3. Configure the intranet domain settings:
   a. Verify that the check boxes for the following items are selected:
      - Include all local (Intranet) sites not listed in other zones
      - Include all sites that bypass the proxy server
   b. Click Advanced and add your domain name to the Websites list to ensure that Internet Explorer recognizes any site with your domain name as the intranet.
4. Configure intranet authentication:
   a. In the Security level for this zone area, click Custom level.
   b. Scroll to the User Authentication section, select Automatic Logon only in Intranet Zone and click OK.

Configuring Connection Settings
If your site uses a proxy server, follow these steps:

1. Select Tools ➔ Internet options ➔ Connections.
2. Click LAN settings.
3. Verify that the proxy server address and port number are correct.
4. Click Advanced.
5. Verify that the correct domain names are entered in the Exceptions field on the Proxy Settings dialog box.

Configuring Advanced Settings
To use Integrated Windows Authentication, follow these steps:

1. Select Tools ➔ Internet options ➔ Advanced.
2. Scroll to the Security section and verify that Enable Integrated Windows Authentication is selected.
3. Click OK and restart the browser to activate the changes.
Confirming the Changes

Once the steps in the previous sections are complete, you should be able to specify the URL for a SAS web application and use the application without a prompt for credentials.

Do not start and use a browser from the machine that is used for SAS Web Server. This does not work. You must use another computer to confirm that the steps were performed correctly.

Configuring the Mozilla Firefox to Use SPNEGO

To configure Mozilla Firefox to use SPNEGO, follow these steps:

1. From a browser window, navigate to about:config.
2. Click I’ll be careful, I promise! to accept the security warning.
3. In the Search field, enter network.negotiate.
4. Double-click the network.negotiate-auth.trusted-uris Preference Name, enter http://hostname.example.com, in the Enter string value field, and then click OK.

Note: The values in the Enter string value field are comma-separated.

To verify the configuration, see “Confirming the Changes” on page 237.

Configuring the Google Chrome to Use SPNEGO

Configuring Security Settings

To configure the security settings, follow these steps:

1. Click the Chrome menu key on the browser toolbar, and then select Settings.
2. Select Show advanced settings.
3. Scroll to the Network section and click Change proxy settings.
4. In the Internet Properties dialog box, select Security.
5. Select Local intranet and then click Sites.
6. Configure the intranet domain settings:
   a. Verify that the check boxes for the following items are selected:
      i. Include all local (Intranet) sites not listed in other zones
      ii. Include all sites that bypass the proxy server
   b. Click Advanced and add your domain name to the Websites list to ensure that Internet Explorer recognizes any site with your domain name as the intranet.
7. Configure intranet authentication:
   a. In the Security level for this zone area, click Custom level.
Scroll to the **User Authentication** section, select **Automatic Logon only in Intranet Zone** and click **OK**.

**Configuring Connection Settings**

If your site uses a proxy server, follow these steps:

1. In the Internet Properties dialog box, select **Connections**.
2. Click **LAN settings**.
3. Verify that the proxy server address and port number are correct.
4. Click **Advanced**.
5. Verify that the correct domain names are entered in the **Exceptions** field on the Proxy Settings dialog box.

**Configuring Advanced Settings**

To use Integrated Windows Authentication, follow these steps:

1. In the Internet Properties dialog box, select **Advanced**.
2. Scroll to the **Security** section and verify that **Enable Integrated Windows Authentication** is selected.
3. Click **OK** and restart the browser to activate the changes.

**Confirming the Changes**

To verify the configuration, see “Confirming the Changes” on page 237.

(Optional) **Configuring User Delegation**

**About User Delegation**

User delegation is a feature that allows a SAS application to reuse the end-user credentials to access other applications that are hosted on different servers. Delegation allows a user to be trusted for delegation of credentials to the SAS server tier. By default, user delegation is not enabled and must be configured. In order to configure user delegation, the SAS server tier must be configured for Kerberos authentication. This includes the workspace server and operating system. For more information about operating system specifications and other details, see “Integrated Windows Authentication” in *SAS Intelligence Platform: Security Administration Guide*.

**Confirming Integrated Windows Authentication Is Enabled for Microsoft Internet Explorer**

To confirm that IWA is enabled in the Microsoft Internet Explorer, see “Configuring Advanced Settings” on page 236.

In addition, define the SAS middle-tier machine as a trusted site by following these steps:

1. Select **Tools** ▶ **Internet options** ▶ **Security**.
2 Select Trusted Sites, and then click Sites.

3 Enter the middle tier host name in the **Add this website to the zone:** field and click Add.

4 Click Close, and then click OK.

**Note:** For Internet Explorer to pass a forwardable ticket to the SAS middle-tier machine, the service account in Active Directory holding the SPNs must be trusted for delegation.

### Configuring User Delegation for Mozilla Firefox

To configure user delegation for Mozilla Firefox, complete the following:

1 From a browser window, navigate to about:config.

2 Click **I'll be careful, I promise!** to accept the security warning.

3 In the **Search** field, enter network.negotiate.

4 Double-click the **network.negotiate-auth.delegation-uris** Preference Name, enter http://hostname.example.com, in the **Enter string value** field, and then click OK.

### Configuring User Delegation for Google Chrome

By default, Chrome disables the delegation of Kerberos credentials. The Windows registry must be updated. Microsoft recommends performing a system backup before editing the registry. Complete the following steps to enable Kerberos delegation:

1 Open the Windows registry editor.

2 Add the following REG_SZ key: Software\Policies\Google\Chrome \AuthNegotiateDelegateWhitelist.

3 Set the value to the SAS Web Server host name: hostname.example.com.

For more information, see http://dev.chromium.org/administrators/policy-list-3#AuthNegotiateDelegateWhitelist.

### Configuring the Middle Tier

To enable delegation of credentials to the server tier, configure the JAAS login module, which is used by Platform Foundation Services (PFS), to use Kerberos. Add the following entries to PFS in the jaas.config file for each SASServer _m_ instance:

```properties
PFS {
    com.sas.services.security.login.OMILoginModule  required
    "host"="hostname.example.com"
    "port"="port_number"
    "repository"="Foundation"
    "domain"="DefaultAuth"
    "trusteduser"="sastrust@saspw"
    "trustedpw"="{sas002}1D5793391C1104E20E3CF4CD2A793E2B"
    "aliasdomain"="DefaultAuth"
    "idpropagation"="sspi"
}
```
Updating the Metadata

For each logical SAS Workspace Server that you plan to configure for IWA authentication (for example, SASAPP - Logical Workspace Server), verify that the **Authentication service** fields are set by following these steps:

1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, navigate to **Environment Management ➤ Server Manager ➤ SASAPP**.
3. Right-click **SASAPP - Logical Workspace Server** and select **Properties**. The SASAPP - Logical Workspace Server Properties dialog box is displayed.
4. Click the **Options** tab.
5. Verify that the following values are specified for the **Authentication service** fields:
   - **Security package**: Negotiate
   - **Service principal name (SPN)**: Leave blank
   - **Security package list**: Kerberos
6. Click **OK** to close the SASAPP - Logical Workspace Server Properties dialog box.

Fallback to SAS Form-based Authentication

**Fallback Authentication**

Container security can be used to achieve single sign-on through IWA, client certificate authentication, and other authentication methods. You might need to support multiple authentication methods concurrently. SAS 9.4 supports a custom fallback authentication security module. On initial request, the security module attempts to authenticate using a primary authentication method, such as SPNEGO. If that authentication method fails, the security module will fallback to the default authentication, SAS form-based authentication.

**Note**: If your web browser does not support IWA, authentication will fallback to SAS form-based authentication, which is provided by SAS Logon Manager.

**Configuring Fallback Authentication**

To configure IWA fallback authentication to SAS form-based authentication, follow these steps:

1. Configure IWA for the middle tier. For more information, see “Support for Integrated Windows Authentication” on page 231.
2. Edit the `SASHome\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\META-INF\context.xml` file and locate the existing `Context definition`. Add the following valve definition:
<Valve
  className="com.sas.vfabriccsvr.authenticator.SasFallbackAuthenticatorValve"
  authMethod="SPNEGO" />

Note: As an alternative to updating the context.xml file, you can edit the deployed file, SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\conf\Catalina\localhost\SASLogon.xml. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

The security module supports the following parameters:

Table 18.6  Fallback Security Module Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uriPattern</td>
<td>/SASLogon/login.*</td>
<td>Specifies whether to process requests with a URI, including query string, matching this regular expression.</td>
</tr>
<tr>
<td>authMethod</td>
<td>None</td>
<td>Specifies the primary authentication method to use: BASIC, DIGEST, or SPNEGO.</td>
</tr>
</tbody>
</table>

3 Edit the SASHome\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig file and remove the comment that encloses the custom error page for code 401, which is located near the bottom of the file.

```xml
<error-page>
  <error-code>401</error-code>
  <location>/WEB-INF/view/jsp/default/ui/401Fallback.jsp</location>
</error-page>
```

Note: As an alternative to updating the web.xml.orig, you can edit the deployed file, SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\sas_webapps\sas.svcs.logon.war\WEB-INF\web.xml. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

4 In the same file, remove the <security-constraint>, <login-config>, and <security-role> code, if you added it for web authentication.
Support for TLS with Client Certificate Authentication

Overview of TLS with Client Certificate Authentication in the Middle Tier

Transport Layer Security (TLS) configuration allows clients to authenticate with the SAS middle tier using a client certificate that is installed in their web browser. When a client certificate is used for authentication and installed in a web browser, you are not required to provide a user name and password to log on.

Client Certificate

To use TLS with client certificate authentication, a client certificate must be loaded into your web browser. To use the client certificate, follow these steps:

1. Create a user in SAS Management Console, specifying the following on the Account tab:
   - In the User ID field, either enter the Subject distinguished name (DN) from the certificate, or the common name (CN) from within the DN.
   - For the Authentication Domain drop-down menu, select web. If the web option does not exist, create it.

   For information about creating a new user, see SAS Management Console: Guide to Users and Permissions.

2. From the CA that you obtained a client certificate, download the CA certificate.
   - Save the certificate to a file (for example, the root-certificate.pem file). In the next section, you must tell the server to trust certificates signed by this CA.

Configuring Middle-Tier Services for the Second Maintenance Release for SAS 9.4 and Previous Releases

For the second maintenance release for SAS 9.4 and previous releases, import the certificate of the CA that signed the client certificate into the SAS Private JRE cacerts truststore in the SASHOME\SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security directory:

keytool -importcert -file root-certificate.pem -keystore cacerts -storepass changeit -alias clientca
Configuring Middle-Tier Services for the Third Maintenance Release for SAS 9.4


Configuring TLS for SAS Web Server and SAS Web Application Server

About This Configuration

In this configuration, SAS Web Server is installed in front of SAS Web Application Server. The user agent (web browser) performs the TLS handshake and exchanges certificates with the web server. The client certificate can be passed through to SAS Web Application Server via HTTP headers. The following sections assume that the SAS middle tier was installed with TLS enabled for SAS Web Server. For more information, see "Configuring SAS Web Server Manually for HTTPS" on page 254.

Configuring SAS Web Server to Pass the Client Certificate to SAS Web Application Server

To configure SAS Web Application Server with SAS Web Server acting as a proxy, follow these steps:

1  Copy the PEM-encoded CA certificate for the CA that signed your client certificate, root-certificate.pem, to the SAS-configuration-directory\Lev\Web\WebServer\ssl directory.

2  Edit the SAS-configuration-directory\Lev\Web\WebServer\conf\extra\httpd-ssl.conf file as follows:
   a  Add the following statements above the VirtualHost directive:

   # initialize the special headers to a blank value to avoid http # header forgeries
   RequestHeader set SSL_CLIENT_CERT ""
   RequestHeader set SSL_CLIENT_VERIFY ""
   # increase sizes to accommodate larger headers
   LimitRequestFieldSize 16384
   SendBufferSize 16384

   b  Add the following statements inside the VirtualHost directive:

   <Location /SASLogon/login>
   SSLVerifyClient optional
   SSLVerifyDepth 10
   SSLCACertificateFile "ssl/root-certificate.pem"
   RequestHeader set SSL_CLIENT_CERT %{SSL_CLIENT_CERT}s
   RequestHeader set SSL_CLIENT_VERIFY %{SSL_CLIENT_VERIFY}s
   </Location>
The SSLVerifyClient optional parameter in the <Location> directive requests the client certificate from the browser on /SASLogon context only.

The following table provides details about the options that are set in the httpd-ssl.conf file:

<table>
<thead>
<tr>
<th>Option</th>
<th>Expected Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSLCertificateFile</td>
<td>&quot;/directory/pem-encoded-certificate.crt&quot;</td>
<td>Specifies the location of the PEM encoded certificate.</td>
</tr>
<tr>
<td>SSLCertificateKeyFile</td>
<td>&quot;/directory/key-file.key&quot;</td>
<td>Specifies the location of the key file, if it is not combined with the certificate.</td>
</tr>
<tr>
<td>SSLCertificateChainFile</td>
<td>&quot;/directory/certificate-chain.pem&quot;</td>
<td>Specifies the location of a file containing the concatenation of PEM encoded certificate authority (CA) certificates that form the certificate chain for the server certificate.</td>
</tr>
<tr>
<td>SSLCACertificateFile</td>
<td>&quot;/directory/CA-certificate.pem&quot;</td>
<td>Specifies the location of the CA certificates for client authentication.</td>
</tr>
<tr>
<td>SSLVerifyClient</td>
<td>verification-type</td>
<td>Specifies the client certificate verification type. Types are none, optional, require, and optional_no_ca.</td>
</tr>
</tbody>
</table>

3 Configure SAS Web Application Server to receive and process the client certificate by editing the SASHome\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\META-INF\context.xml file and adding the following configuration:

```
<Valve className="com.sas.vfabriiccsvr.authenticator.SSLAuthenticator" fallThrough="false" />
```

Note: As an alternative to updating the context.xml file, you can edit the deployed file, SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\conf\Catalina\localhost\SASLogon.xml. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

Note: This process is used when Client Certificate Authentication is fully configured.

The security module supports the following parameters:
### Table 18.7 SSLAuthenticator Security Module Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uriPattern</td>
<td>/SASLogon/login</td>
<td>Process requests with a URI, including query string, matching this regular expression.</td>
</tr>
<tr>
<td>fallThrough</td>
<td>false</td>
<td>Controls flow upon unsuccessful authentication. If true, control passes to the next security module in the pipeline; if false, a 401 error code is returned.</td>
</tr>
<tr>
<td>sslClientCert</td>
<td>SSL_CLIENT_CERT</td>
<td>HTTP header that contains the Base64-encoded x509 client certificate.</td>
</tr>
<tr>
<td>sslClientVerify</td>
<td>SSL_CLIENT_VERIFY</td>
<td>HTTP header that contains the status of the certificate as NONE, SUCCESS, GENEROUS, or FAILED:reason.</td>
</tr>
</tbody>
</table>

4 Edit the `SAS-configuration-directory\Lev\Web\WebAppServer \SASServer\m\conf\server.xml` file and locate the following block:

```
<Realm allRolesMode="authOnly" className="org.apache.catalina.realm.LockOutRealm">
    <Realm className="org.apache.catalina.realm.UserDatabaseRealm" 
        resourceName="UserDatabase" />
</Realm>
```

Replace the previous block with the following:

```
<Realm allRolesMode="authOnly" className="org.apache.catalina.realm.LockOutRealm">
    <Realm className="com.sas.vfabrictcsvr.realm.TrustedX509CertificateRealm" 
        x509UsernameRetrieverClassName="com.sas.vfabriccsrvr.
        realm.X509SubjectCnRetriever" />
</Realm>
```

**Note:** The `x509UsernameRetrieverClassName` configuration must be on one line. It is shown on more than one line in the preceding code sample for display purposes only.

**Note:** Alternatively, you can authenticate users against LDAP using the JNDIRealm. You can use the `x509UsernameRetrieverClassName` attribute on the JNDIRealm also, in order to use the Subject CN from the certificates as the user name. For more information, see “Configuring the Realm for SAS Web Application Server” on page 214.
Configuring TLS for Stand-alone SAS Web Application Server

About This Configuration

In this configuration, SAS Web Application Server is installed stand-alone without SAS Web Server in your configuration. TLS configuration for stand-alone SAS Web Application Server is a manual process. Unlike the configuration that includes SAS Web Server, the certificates for SAS Web Application Server are handled through Java keystores.

It is assumed that you have full control of handling certificates and Java keystores for SAS Web Application Server. Make sure that you have server identity certificate in the keystore.jks file for the incoming TLS request.

For the second maintenance release for SAS 9.4 and previous releases, a default trusted keystore that contains a list of well-known CA certificates is in the cacerts file that comes with the JRE and is used by SAS Web Application Server, which is SAS Private JRE. If any of your client or server certificates are signed by your own CA, insert that CA certificate into the cacerts file and also into the trusted CA area in the browser.

For the third maintenance release for SAS 9.4, use the SAS Deployment Manager to import your CA certificates into the trusted CA bundle. You need to specify the location of your self-signed or site-signed CA certificate to the SAS Deployment Manager, and it will update the SAS Private JRE for you. For more information, see "Add Pre-Existing Certificates to the Trusted CA Bundle" in SAS Intelligence Platform: Installation and Configuration Guide.

Configuring a Stand-alone SAS Web Application Server

To configure a stand-alone SAS Web Application Server, follow these steps:

1. Edit the SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\conf\server.xml file to enable one-way TLS. Update the <Connector> definition, based on the following sample:

   ```xml
   <Connector
       acceptCount="100" connectionTimeout="20000" executor="tomcatThreadPool"
       maxKeepAliveRequests="15"
       redirectPort="${bio.https.port}" useBodyEncodingForURI="true"
       scheme="https" secure="true" SSLEnabled="true"
       sslProtocol="TLS" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
       keystoreFile="/local/install/certs/serverids.jks"
       keystorePass="password" />
   ``

The following parameters that are needed to support one-way TLS:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Specifies the HTTPS port of SAS Web Application Server.</td>
</tr>
<tr>
<td>scheme</td>
<td>Specifies the communications protocol and should be set to https.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keystoreFile</td>
<td>Specifies the keystore file that contains the server identity certificate.</td>
</tr>
<tr>
<td>keystorePass</td>
<td>Specifies the password for the keystore access.</td>
</tr>
</tbody>
</table>

2. Verify that one-way TLS is working with the default authentication.

3. Follow the steps in the “Web Authentication” on page 210 task, but specify CLIENT-CERT as the auth-method in the web.xml file for SAS Logon Manager.

4. Edit the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\conf\server.xml` file and locate the following block:

   ```xml
   <Realm allRolesMode="authOnly" className="org.apache.catalina.realm.LockOutRealm">
     <Realm className="org.apache.catalina.realm.UserDatabaseRealm" 
        resourceName="UserDatabase" />
   </Realm>
   ```

   Replace the previous block with the following:

   ```xml
   <Realm allRolesMode="authOnly" className="org.apache.catalina.realm.LockOutRealm">
     <Realm className="com.sas.vfabrictcsvr.realm.TrustedX509CertificateRealm" 
       x509UsernameRetrieverClassName="com.sas.vfabrictcsvr.
        realm.X509SubjectCnRetriever" />
   </Realm>
   ```

   Note: The `x509UsernameRetrieverClassName` configuration must be on one line. It is shown on more than one line in the preceding code sample for display purposes only.

   Note: Alternatively, you can authenticate users against LDAP using the JNDIRealm. You can use the `x509UsernameRetrieverClassName` attribute on the JNDIRealm also, in order to use the Subject CN from the certificates as the user name. For more information, see “(Optional) Validating the Previous Steps” on page 214.

---

### SAS Web Server Authentication

#### About SAS Web Server Authentication

Starting with the first maintenance release for SAS 9.4, SAS Web Server authentication is supported. This section provides instructions for configuring the system so that SAS Web Server securely passes the authenticated user name to SAS Web Application Server.

SAS Web Server is based on the Apache HTTP Server version 2.2.x. You can use any of the Apache modules (for example, Shibboleth) for third-party
authentication. These commercial and open-source modules can be used with SAS Web Server to perform authentication and enable single sign-on services.

**Web Authentication**

Web authentication must be enabled before configuring authentication in SAS Web Server. By default, the first maintenance release for SAS 9.4 does not enable web authentication. To enable web authentication, follow the steps in the "Web Authentication" section, with the following exceptions:

- Do not edit either of the following SAS Logon Manager installation files:
  - `SASHome\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig`
  - `SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\sas_webapps\sas.svcs.logon.war\WEB-INF\web.xml`.

- Do not add any realms to the SAS Logon Manager installation files in the previous list item.

**Note:** Ensure that you follow the steps in the section that corresponds to the maintenance release that you have installed.

**Configuring Authentication in SAS Web Server**

To configure authentication in SAS Web Server, follow these steps:

1. Edit the `SAS-configuration-directory\Lev\Web\WebServer\conf\httpd.conf` file. Add a line to import the authentication module in the section where all the other modules are imported. In some cases, you will be including a module-specific configuration file instead, which does the import and other necessary configuration.

2. Configure SAS Web Server to authenticate requests to `/SASLogon/login`, based on one of the following:

   - For HTTP, specify the Location directive in the `httpd.conf` file. In the following example, replace `authentication_type` with the correct AuthType and any other configuration required by the authentication module that you are using:

     ```
     <Location /SASLogon/login>
     AuthType authentication_type
     require valid-user
     </Location>
     ```

   - For HTTPS, specify the Location directive in the `\extra\httpd-ssl.conf` file, inside the VirtualHost directive.

3. Depending on the type of authentication that you are using, perform whatever additional configuration that is necessary.

   **Note:** Because the additional configuration is dependent the type of authentication that is used in upon your environment, it cannot be fully described here.

4. Restart the web server.

5. Verify that authentication is working by opening a web browser and entering a URL that is similar to the following example:
You should see the user name in the SAS Web Server access log. The output should be similar to the following:

```
ip_address - user_name [20/Mar/2014:14:28:26 -0400] "GET /SASLogon/login" 302 -
```

Note: You can configure secure requests in the `extra/httpd-ssl.conf` file. After accessing the secure URL from a web browser, you should see the user name in the ssl_request log. By default, the user name is not displayed. To display it, add `%u` to the Log directive.

### Setting the REMOTE_USER Variable

The advantage of web server authentication is that you can use any web server authentication plug-in, as long as it sets the REMOTE_USER environment variable. The REMOTE_USER environment variable is passed to SAS Web Application Server by rewriting it into an HTTP header. Add the following highlighted text to the configuration that you created in Step 2 on page 248.

```xml
<Location /SASLogon/login>
  AuthType xxxxxxxx
  require valid-user
  RewriteEngine On
  RewriteCond %{LA-U:REMOTE_USER} (.+)
  RewriteRule . - [E=RU:%1,NS]
  RequestHeader set X-Remote-User "%{RU}e" env=RU
</Location>
```

### Configuring a Secret Password

SAS Web Server is configured to pass the authenticated subject to the application server via the HTTP headers. To prevent anyone from spoofing the web server, you can include a secret password in the headers and as an option, either use HTTPS back to the web application server, or set up two-way TLS.

To configure a secret password, follow these steps:

1. Create a Base64-encoded authorization string, using the format `username:password`. For example, `username:password` would be encoded as `dXNlcm5hbWU6cGFzc3dvcmQ=`.
   
   Note: You can leave username blank.

2. Add a standard Basic Authorization header with the secret password to the SAS Web Server configuration, substituting your encoded authorization string. Here is an example:

```xml
<Location /SASLogon/login>
  AuthType xxxxxxxx
  require valid-user
  RewriteEngine On
  RewriteCond %{LA-U:REMOTE_USER} (.+)
  RewriteRule . - [E=RU:%1]
  RequestHeader set X-Remote-User "$%{RU}e" env=RU
  RequestHeader set Authorization "Basic dXNlcm5hbWU6cGFzc3dvcmQ="
</Location>
```
3 The secret password needs to be specified in the application server configuration. It is suggested that you encode the password, instead of using plaintext. To get an encoded password string, from a command prompt, navigate to the SASHome\SASWebApplicationServer\9.4 directory and run the following command:

On Windows:

```
java -cp tomcat-6.0.35.B.RELEASE\lib\tcServer.jar;tomcat-6.0.35.B.RELEASE\bin\tomcat-juli.jar;tomcat-6.0.35.B.RELEASE\lib\tomcat-coyote.jar
com.springsource.tcserver.security.PropertyDecoder
-encode "tc-server-passphrase" password
```

On UNIX:

```
java -cp './tomcat-6.0.35.B.RELEASE/lib/tcServer.jar:./tomcat-6.0.35.B.RELEASE/bin/tomcat-juli.jar:./tomcat-6.0.35.B.RELEASE/lib/tomcat-coyote.jar'
com.springsource.tcserver.security.PropertyDecoder
-encode 'tc-server-passphrase' password
```

Note: The tc-server-passphrase passphrase should match the value of the com.springsource.tcserver.security.PropertyDecoder.passphrase property in the catalina.properties file.

4 Edit the SAS-configuration-directory\Levn\Web\WebAppServer\SASServern_m\conf\catalina.properties file. At the end of the file, add a new property containing the encoded secret password. Here is an example:

```
pw.sas.valve.PrincipalFromRequestHeadersValve=s2enc://encoded_password
```

Configuring the Security Module for SAS Web Application Server

SAS Web Application Server must be configured to accept the user name that is sent by SAS Web Server in the HTTP header. The web application server then uses the user name to create a principal and sets the user name in the request that is sent to the SAS Logon Manager.

Edit the SASHome\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\META-INF\context.xml file and add the following valve configuration.

```
<Valve className="com.sas.vfabriccsvr.authenticator.PrincipalFromRequestHeadersValve"
  secretPassword="${pw.sas.valve.PrincipalFromRequestHeadersValve}" />
```

Note: As an alternative to updating the context.xml file, you can edit the deployed file, SAS-configuration-directory\Levn\Web\WebAppServer\SASServern_m\conf\Catalina\localhost\SASLogon.xml. This avoids the need to rebuild and redeploy the application, but you must make sure your changes are not overwritten if the application is redeployed at a later date.

Note: The className configuration must be on one line. It is shown on more than one line in the preceding code sample for display purposes only.
The security module supports the following parameters:

Table 18.8 SAS Web Application Server Security Module Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uriPattern</td>
<td>/SASLogon/login.*$</td>
<td>Specifies the process requests with a URI, including query string, matching this regular expression.</td>
</tr>
<tr>
<td>fallThrough</td>
<td>false</td>
<td>Specifies the controls flow upon unsuccessful authentication. If true, control passes to the next security module in the pipeline. If false, a 401 error code is returned.</td>
</tr>
<tr>
<td>secretPassword</td>
<td>None</td>
<td>Specifies the secret password to expect in the Basic Authorization header (Optional).</td>
</tr>
<tr>
<td>userHeader</td>
<td>X-Remote-User</td>
<td>Specifies the HTTP header containing the authenticated subject name.</td>
</tr>
<tr>
<td>roleName</td>
<td>ROLE_USER</td>
<td>Specifies the role to associate with the principal. This value is usually not needed.</td>
</tr>
</tbody>
</table>
Middle-Tier Security

SAS Anonymous Web User

About the SAS Anonymous Web User .......................................................... 253
Using the SAS Anonymous Web User with SAS Authentication .......... 253

Configuring SAS Web Server Manually for HTTPS ......................... 254
Use of TLS with SAS Web Applications ............................................. 254
Reconfiguring to Use HTTPS ............................................................... 254

Configuring SAS Web Application Server to Use HTTPS .............. 259
About Configuring SAS Web Application Server to Use HTTPS ...... 259
Reconfiguring to Use HTTPS ............................................................... 259
Setting the Secure Attribute for Session Cookies ......................... 262

FIPS 140-2 Compliance ............................................................................. 262
About FIPS Compliance in the SAS Middle-Tier ................................. 262
Before You Begin ....................................................................................... 262
Configuring SAS Web Server ............................................................... 263
Configuring SAS Web Application Server ............................................ 263

Whitelist of Websites and Methods Allowed to Link to SAS Web Applications ................................................................. 266
About the Whitelist ................................................................................... 266
Modifying the Whitelist for URLs and HTTP Request Methods ....... 266

Configuring the Cross Domain Proxy Servlet through a Whitelist ..... 268
About the Whitelist ................................................................................... 268
Modifying the Whitelist ........................................................................... 268

Enabling Support for Forward Proxy Authentication .................... 269

SAS Anonymous Web User

About the SAS Anonymous Web User

The SAS Anonymous Web User (webanon) is an optional account that can be used to grant web clients anonymous access to certain SAS Web Infrastructure Platform applications (SAS BI Web Services and SAS Stored Process Web Application). This anonymous account is configured with the SAS Deployment Wizard and is applicable only when SAS authentication is being used. If web authentication is used, the web application server processes authentication requests, and this anonymous account has no effect.
Using the SAS Anonymous Web User with SAS Authentication

If the webanon account is configured, it is used when a web service is configured for SAS authentication, and credentials are not supplied. If the webanon account is not configured, there are no credentials for authentication, and the request fails.

Note: If the webanon account does not exist in your deployment, contact SAS Technical Support for assistance with creating the account.

In a default deployment, this anonymous account is configured as an internal user account. To determine whether to enable the webanon user account, administrators must decide whether they want to require clients to provide credentials for all requests. When clients provide credentials to an incoming request, these credentials are always used for authentication whether the account has been enabled or not.

The webanon user is defined in the following locations:

- in metadata. In default deployments, the SAS Anonymous Web Service User is an internal user account that is known only to SAS and that is authenticated internally in metadata. When internal authentication is used, it is not necessary for this user to have a local or network account.
- in the operating system of the metadata server machine, only if you selected the External authentication option for this user during a custom installation.

Configuring SAS Web Server Manually for HTTPS

Use of TLS with SAS Web Applications

Transport Layer Security (TLS) is a successor protocol to SSL. It is used to provide network security and privacy. In addition to providing encryption services, TLS uses trusted certificates to perform client and server authentication, and it uses message authentication codes to ensure data integrity.

This documentation assumes that you have a basic understanding of TLS and that you know how to obtain and use trusted certificates.

The best practice is to acquire CA-signed certificates before you install and configure SAS software. You can specify the location of the certificate to the SAS Deployment Wizard and it can configure SAS Web Server to use it. For more information, see “Using HTTPS” on page 30.

Reconfiguring to Use HTTPS

Note: The manual TLS configuration changes must be reverted to the original non-TLS values before applying any maintenance releases or upgrades to the system. Then the manual TLS configuration steps can be reapplied to the upgraded system.
If you did not choose to configure with secure sockets during the initial installation and configuration with the SAS Deployment Wizard, you can manually configure SAS Web Server to use HTTPS. Follow these steps:

1. Create a private key, generate a certificate signing request, and get a signed certificate. For more information, see *Encryption in SAS*.

2. Stop SAS Web Server and all SAS Web Application Server instances.

3. If the directory `SAS-configuration-directory\Lev\Web\WebServer\ssl` does not exist, then create it.

   Put the certificate file and key file in this directory. Be sure to change the default permissions of the .key file to be read-only for the user that SAS Web Server is configured to run as.

4. Edit the `SAS-configuration-directory\Lev\Web\WebServer\conf\httpd.conf` file and make the following changes:
   a. Remove the # from the following line:
      ```
      #Include C:/SAS/Config/Lev\Web\WebServer/conf/extra/httpd-ssl.conf
      ```
   b. To use SAS Environment Manager to monitor SAS Web Server, locate the following line:
      ```
      Listen 80
      ```
      Replace the previous line with the following line:
      ```
      Listen localhost:80
      ```

5. Edit the `SAS-configuration-directory\Lev\Web\WebServer\conf\extra\httpd-ssl.conf` file and make the following changes:
   a. Locate the following line and make sure it refers to the HTTPS port that you want the server to listen on:
      ```
      Listen 443 https
      ```
      **Note:** Be aware that on UNIX platforms, you must start SAS Web Server as root in order to listen on ports below 1024.
   b. Locate the following line and make sure it refers to the same HTTPS port:
      ```
      <VirtualHost _default_:443>
      ```
   c. Locate the following lines for the certificate file and key file and enter the correct filenames:
      ```
      SSLCertificateFile "ssl/myhost.crt"
      SSLCertificateKeyFile "ssl/myhost.key"
      SSLCertificateChainFile "ssl/myhost.crt"
      ```

6. (Optional) To verify that security has been configured correctly, start SAS Web Server. Then, access the secure SAS Web Server from your web browser.

7. For each instance of SAS Web Application Server, edit the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\conf\server.xml` file and make the following changes to the Connector element:
   - Change the `proxyPort` attribute to specify the HTTPS listen port.
Change the scheme to https.

8 Use SAS Management Console to update the protocol and port number for each web application. For more information, see “Specifying Connection Properties” on page 64.

**CAUTION!** Do not modify the connection properties for DP-SAS-Environment-Manager. However, the connection properties for Environment Mgr Mid-Tier should be changed.

Note the following items:

- The connection properties for Environment Mgr Mid-Tier should be changed.
- When you log on to SAS Management Console to update the connection properties (by navigating to the Plug-ins tab and selecting Application Management ➤ Configuration Manager), view the Properties of each web application that is listed, to determine whether there is connection information that needs to be updated.
- To change the connection properties for SAS Visual Analytics, expand the sub-trees and apply the change to each SAS Visual Analytics application and service. Also, for Visual Analytics, change the connection properties for Search Interface to SAS Content.
- Some SAS users prefer to update these values using a SAS DATA step. This approach is beyond the scope of this task. If you decide to modify these connections using a SAS script rather than SAS Management Console, the SAS_THEME table in SharedServices DB will not be modified. It is possible to manually update this database entry. However, the simplest solution is to use SAS Management Console to modify the Themes Connection, even if you use a script to configure the rest of these values.
- Change the port and protocol for SASTheme_default. From SAS Management Console, navigate to the Plug-ins tab and select Application Management ➤ Configuration Manager ➤ SASTheme_default. View the Properties to determine whether there is connection information that needs to be updated.

9 Use SAS Management Console to update the SAS Content Server connection information. For more information, see “Manual Configuration Tasks” on page 135.

10 For the second maintenance release of SAS 9.4 and previous releases, if the certificate that you use is not signed by a certificate authority (CA) that would be located in the JRE default truststore (for example, VeriSign), then add all the CA certificates in the chain to the SAS Private JRE truststore (the cacerts file). Do this for all middle tier machines before starting any servers.

You must also import the certificate chain for server-tier machines to support any Java clients such as PROC SOAP. Also do this for client tier products.

To add certificates into the SAS Private JRE truststore, complete the following steps:

a Know the location of your self-signed or site-signed certificate.

b Run the SAS Deployment Wizard in install mode only to install SAS Web Server.
c Import your self-signed or site-signed certificate into the SAS Private JRE default truststore (cacerts).

Importing a certificate into a Java keystore or truststore is accomplished with the Java `keytool - importcert` command. The location of cacerts in the SAS Private JRE is as follows:

```
SAS-installation-directory/SASHome/
SASPrivateJavaRuntimeEnvironment/9.4/jre/lib/security
```

For example, on Windows, the command that you run is similar to this:

```
cd C:\Program Files\SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security
../..\bin\keytool -importcert -keystore cacerts -file mycert.crt
```

**TIP** The default password for cacerts is `changeit`. It is publicly documented on the Oracle website.

For more information, see your Java documentation at http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html.

d Run the SAS Deployment Wizard in configure mode only to configure SAS Web Server.

For the third maintenance release of SAS 9.4, use the SAS Deployment Manager to import your CA certificates into the trusted CA bundle. You need to specify the location of your self-signed or site-signed CA certificate to the SAS Deployment Manager, and it will update the SAS Private JRE for you. For more information, see “Add Pre-Existing Certificates to the Trusted CA Bundle” in SAS Intelligence Platform: Installation and Configuration Guide.

11 Configure the server tier and client tier.

**Windows Specifics:** For server and client tiers machines, add any required CA certificates to the Windows truststore.

12 Start each SAS Web Application Server instance.

13 For SAS Visual Analytics deployments, perform the following steps with SAS Management Console to confirm that the SAS LASR Authorization Service URI is updated:

a Select Environment Management ➤ Server Manager.

b For each SAS LASR Analytic Server, select the server to display the connection information in the right panel. Right-click the connection and select Properties.

c Select the Options tab. Make sure the Use LASR authorization service check box is selected and that the URI includes the HTTPS port number. Click OK.

**Note:** You must perform these steps so that the HTTPS connection information is saved in metadata.

14 Depending on which products you have installed, you might have to update the `SAS-configuration-directory\Lev1\Web\WebServer\htdocs\sas\sas-environment.xml` file. For more information, see “Customizing the SAS Environment File” on page 303.
To access SAS Environment Manager console with SSL enabled on SAS Web Server, edit the following files and locate the instances of the URLs listed below that begin with http://server:port. Modify them to point to https://server:ssl-port. The default port for HTTPS on Windows systems is 443. On UNIX systems, the default port is 8343.

a Edit the SAS-configuration-directory\Lev\Web \SASEnvironmentManager\server-version-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\spring\security-web-context.xml file. Locate the following lines and enter the correct information for your environment in place of the highlighted text:

```xml
<constructor-arg value="https://server:port/SASLogon/
<property name="loginUrl" value="https://server:ssl-port/SASLogon"/>
<property name="casUrl" value="https://server:ssl-port/SASLogon" />
```

b Edit the SAS-configuration-directory\Lev\Web \SASEnvironmentManager\server-version-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\web.xml file. Locate the following lines and enter the correct information for your environment:

```xml
<param-value>https://server:ssl-port/SASWebDoc</param-value>
<param-value>https://server:ssl-port/SASEnvironmentMgrMidTier</param-value>
<param-value>https://server:ssl-port/SASLogon/TimedOut.do
```

c Restart SAS Environment Manager.

16 Update the SAS Content Server JVM options with the new HTTPS URI values that were specified in Step 15. Otherwise, users will not be able to access the SAS Content Server administration console.

The following JVM options must be updated:

- -Dsas.scs.cas.host
- -Dsas.scs.cas.port
- -Dsas.scs.cas.scheme
- -Dsas.scs.svc.host
- -Dsas.scs.svc.port
- -Dsas.scs.svc.scheme

For a description of each JVM option and more information, see Table 10.1 on page 118.

See Also

- Encryption in SAS
- SAS Intelligence Platform: Installation and Configuration Guide
Configuring SAS Web Application Server to Use HTTPS

About Configuring SAS Web Application Server to Use HTTPS

Transport Layer Security (TLS) is a successor protocol to SSL. This documentation assumes that you have a basic understanding of TLS and that you know how to obtain and use trusted certificates. Configuring secure communication between SAS Web Server and SAS Web Application Server is optional. You do not have to configure TLS between SAS Web Server and SAS Web Application Server.

Reconfiguring to Use HTTPS

In deployments that use SAS Web Server, the SAS Deployment Wizard does not include an option to configure SAS Web Application Server for HTTPS. The communication path between SAS Web Server and SAS Web Application Server uses HTTP.

Note: The following examples configure SAS Web Application Server with self-signed certificates. Since the certificates used by SAS Web Application Server are not exposed to web browsers directly, self-signed certificates are sufficient and allow longer expirations than signed certificates.

Note: The manual TLS configuration changes must be reverted to the original non-TLS values before applying any maintenance releases or upgrades to the system. Then the manual TLS configuration steps can be reapplied to the upgraded system.

In order to use HTTPS between SAS Web Server and SAS Web Application Server, follow these steps:

1. Create a JKS format keystore with the key and self-signed certificate, export the certificate, and convert it to a PEM encoded file. The following example creates a certificate that is valid for 10 years:

   keytool -genkeypair -keyalg RSA -alias myhost -keystore myhost.jks
   -storepass changeit -validity 3650

   Note: Enter the fully qualified machine name at the prompt asking for your first and last name.

   Note: Make sure the alias is unique for this certificate.

   keytool -exportcert -alias myhost -keystore myhost.jks
   -storepass changeit -file myhost.crt

   openssl x509 -in myhost.crt -inform DER -out myhost.pem -outform PEM

   Note: Each keytool command must be on one line. They are shown on more than one line in the preceding code sample for display purposes only.

   Note: Make sure the alias is the same value that is used in the first keytool command above.
2 For the second maintenance release of SAS 9.4 and previous releases, add the self-signed certificate to the JRE default truststore. If there are multiple machines, add the certificate to the JRE default truststore on each machine and choose a different alias for each certificate.

```
keytool -importcert -keystore "SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security\cacerts" -storepass changeit -alias myhost -file myhost.crt
```

**Note:** The `keytool` command must be on one line. It is shown on more than one line in the preceding code sample for display purposes only.

**Note:** Make sure the alias is the same value that is used in the first keytool command above.

**Note:** The cacerts file can be found in the SASHOME\SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security directory.

For information about the `openssl` and `keytool` commands, see the vendor documentation.


3 Edit the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServern_m\conf\server.xml` file. Duplicate the existing `Connector` element and complete the following:

**a** Add the following attributes:

- `secure="true"`
- `SSLEnabled="true"`
- `sslProtocols="TLSv1,TLSv1.1,TLSv1.2"`
- `keystoreFile="/path-to-myhost.jks"`
- `keystorePass="changeit"`

**b** Change `port="${bio.http.port}"` to `port="${bio.https.port}"`.

**Note:** Once you have completed your changes and confirmed that SAS Web Application Server is using HTTPS, edit the server.xml file again and remove the `Connector` element that was left using HTTP.

**Note:** This step must be repeated for each server that uses HTTPS.

**Note:** Make sure the `keystorePass` value is the same value used in Step 1 on page 259 when generating the JKS file, especially if you changed the default value (which is “changeit”).

4 For SAS Web Application Server, set the following JVM options, ensuring that you choose the correct HTTPS port that SAS Web Application Server is listening on. For SASServer1_1:

```
-Dsas.scs.port=8443
-Dsas.scs.scheme=https
-Dsas.auto.publish.port=8443
-Dsas.auto.publish.protocol=https
```

For SASServern_m set the following option:
-Dsas.auto.publish.port=https-port

**Note:** The HTTPS port used by SAS Web Application Server can be found in the catalina.properties file by looking for the *bio.https.port* property.

Also, for SASServer\_m, add the following option:

-Dsas.auto.publish.protocol=https

**Note:** For Windows deployments, edit the following files:

- `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\_m\bin\setenv.bat`
- `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer\_m\conf\wrapper.conf`

For UNIX deployments, edit the `SAS-configuration-directory/Lev\Web/WebAppServer/SASServer\_m/bin/setenv.sh` file.

5 For SAS Web Server, make the following changes:

a Edit `SAS-configuration-directory\Lev\Web\WebServer\conf\sas.conf` and change the `BalancerMember` directives to use `https` as the protocol and the HTTPS port that SAS Web Application Server is listening on. See the following example:

BalancerMember `https://myhost.example.com:8443`
  route=myhost.example.com_SASServer1_1

**Note:** There are `BalancerMember` directives for both a single server and a server cluster. Each cluster member must have a different port number. For example, the `BalancerMember` directive for the second server in a cluster can specify:

BalancerMember `https://myhost.example.com:8154`
  route=myhost.example.com_SASServer2_1

b Edit `SAS-configuration-directory\Lev\Web\WebServer\conf\sas.conf` and add the following directives to the top of the file:

```
SSLProxyEngine on
SSLProxyVerify require
SSLProxyVerifyDepth 10
SSLProxyCACertificateFile "/path-to/myhost.pem"
```

If there are multiple machines, choose one of the following options:

- If there are multiple middle tiers (for example, horizontal clustering or different web applications on different middle tiers using the same web server), create a `chain.pem` file with each of the self-signed certificates created for each middle-tier machine to use in the preceding `SSLProxyCACertificateFile` attribute.

- If there are multiple machines running SAS Web Application Server, use the `SSLProxyCACertificatePath` directive instead of the `SSLProxyCACertificateFile` directive that is shown above. Copy all the certificates to a directory that is accessible by SAS Web Server. The files in this directory have to be PEM-encoded and are accessed through hash filenames. Usually, you cannot place the certificate files there. You must create symbolic links named `hash-value.N`. You should always make sure this directory contains the appropriate symbolic links by completing the following:
1 Create a directory to place all the PEM files
2 Determine the hash value that TLS is expecting by running the following command:
   openssl x509 -noout -hash -in myhost.pem
3 Use the returned hash value to make a link back to the original .pem file:
   ln -s myhost.pem hashvalue.0
4 Repeat these steps for each PEM file.
6 Restart SAS Web Application Server and SAS Web Server.

Setting the Secure Attribute for Session Cookies

The secure attribute for cookies directs a web browser to only send the cookie through an encrypted HTTPS connection. To configure SAS Web Application Server to return the session ID with the secure attribute, complete the following:

1 Edit the SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\conf\server.xml file. Add secure="true" to the existing Connector element.
2 Edit the SAS-configuration-directory\Lev\Web\WebAppServer\SASServer_m\conf\web.xml file. Add the following session-config:
   <session-config>
     <session-timeout>30</session-timeout>
     <cookie-config>
       <secure>true</secure>
     </cookie-config>
   </session-config>

FIPS 140-2 Compliance

About FIPS Compliance in the SAS Middle-Tier

The following sections describe how to configure components in the middle tier to use cryptographic modules that are FIPS 140-2 compliant. Completing these procedures do not result in middle tier components that are FIPS 140-2 compliant, only that the components are using a FIPS 140-2 compliant cryptographic module.


Before You Begin

One of the tasks in this section is to configure SAS Web Application Server to use the following native libraries:
- APR library
- JNI wrappers for APR used by Tomcat (tc native)
- OpenSSL libraries

In the first maintenance release for SAS 9.4 and earlier, the binaries for the APR libraries that were shipped with SAS have a known issue that prevents them from being used. If your deployment is not current with the second maintenance release for SAS 9.4 or later, contact SAS Technical Support for assistance with getting the native libraries for your platform.

Note: Some subsequent deployment steps (for example, adding a SAS web application, performing an update in place, and reconfiguring a SAS web application) overwrites the web application server and web server configuration files. Therefore, the steps in this section must be reverted before doing any of these subsequent deployment steps. After the deployment steps are complete, the manual steps for configuring HTTPS and FIPS will need to be redone. Make a copy of the files in the steps below before modifying them to assist in reverting these steps, if necessary.

### Configuring SAS Web Server

SAS Web Server must be configured to use HTTPS. This is performed most easily during initial configuration with the SAS Deployment Wizard. Selecting the option to use HTTPS with SAS Web Server causes the server to use OpenSSL though the mod_ssl module for Apache HTTP Server. OpenSSL has a FIPS module that is certified as FIPS 140-2 compliant. As a result, the server can initialize the OpenSSL software in FIPS mode with a change to the server’s configuration file.

1. Edit the `SAS-configuration-directory\Levn\Web\WebServer\conf\extra\httpd-ssl.conf` file and add the following statement before the VirtualHost directive:

   ```
   SSLFIPS on
   ```

2. Restart the server and verify from any of the `SAS-configuration-directory\Levn\Web\WebServer\logs\error_*.log` files that the server successfully initialized in FIPS mode. Log entries similar to the following indicate successful configuration:

   ```
   [Thu Apr 09 16:43:01 2015] [notice] Operating in SSL FIPS mode
   [Thu Apr 09 16:43:11 2015] [notice] Digest: generating secret for digest ...
   [Thu Apr 09 16:43:11 2015] [notice] Digest: done
   [Thu Apr 09 16:43:12 2015] [notice] Operating in SSL FIPS mode
   ```

   Note: In this mode, the server establishes connections only with clients that use the TLSv1 protocol and strong encryption.

### Configuring SAS Web Application Server

The Apache Portable Runtime (APR) is a native web server library that can be used by SAS Web Application Server to leverage native library support for OpenSSL. Using a native library typically results in better performance than approaches that use Java. SAS Web Application Server can be started in FIPS mode by setting `FIPSMode="on"` on the APR listener. This option is new to
Tomcat 7 (that is part of SAS Web Application Server). Three native components are required:

To modify an existing SAS Web Application Server instance to use the APR, follow these steps:

1. Perform the steps in “Configuring SAS Web Application Server to Use HTTPS”. However, look at Connector settings in Step 3b. You can make the changes all at once.

   To export the private key from the myhost.jks file to use for the SSLCertificateKeyFile attribute in the Connector settings in Step 3b, run the following commands:

   ```
   keytool -v -importkeystore -srckeystore myhost.jks -srcalias myhost
              -destkeystore myp12file.p12 -deststoretype PKCS12
   openssl pkcs12 -in myp12file.p12 -nodes -nocerts -out myhostkey.pem
   ```

   **Note:** The `keytool` command must be on one line. It is shown on more than one line in the preceding code sample for display purposes only.

2. Edit the script files for SAS Web Application Server to use the APR libraries that are provided by SAS.

   For Windows deployments, edit the `SASServer1_1\conf\wrapper.conf` file to include lines similar to the following example:

   ```
   # Java Library Path
   wrapper.java.library.path.1=%CATALINA_BASE%\bin\winx86_64
   wrapper.java.library.path.2=c:\Program Files\SASHome\SASWebServer\9.4\httpd-2.2.26.0-64\bin
   ```

   **Note:** The `wrapper.java.library.path.2` command is shown on more than one line in the preceding code sample for display purposes only.

   Also, open the Windows Control Panel and select **System ➤ Advanced system settings ➤ Environment Variables**. Add the APR library path to the PATH environment variable.

   For the second maintenance release of SAS 9.4 and previous releases on UNIX, edit the `SASServer1_1/bin/tcruntime-ctl.sh` file to include lines similar to the following examples, based on your release of SAS 9.4:

   ```
   LD_LIBRARY_PATH="/install/cfgsas1/SASHome/SASWebServer/9.4/httpd-2.2.26.0-64/lib*
   export LD_LIBRARY_PATH
   ```

3. Edit `SASServer1_1\conf\server.xml` and make the following changes:
a Add the following listener to the Server element:

```xml
<Listener
    SSLEngine="on"
    className="org.apache.catalina.core.AprLifecycleListener"
    FIPSMode="on" />
```

b Change the `Connector` element to use `Http11AprProtocol` and specify the other TLS parameters. Here is an example:

```xml
<Connector acceptCount="100" connectionTimeout="20000"
            executor="tomcatThreadPool" maxKeepAliveRequests="15"
            port="8443" scheme="https" secure="true"
            proxyName="hostname.example.com" proxyPort="8343"
            redirectPort="8443" useBodyEncodingForURI="true"
            SSLCertificateFile="${catalina.base}/ssl/myhost.crt"
            SSLCertificateKeyFile="${catalina.base}/ssl/myhostkey.pem"
            SSLCertificateChainFile="${catalina.base}/ssl/chain.pem"
            SSLPassword="******"
            SSLEnabled="true" />
```

**Note:** Depending on the topology of your deployment and the types of certificates that are used in your environment, you might not have to specify the `SSLCertificateChainFile` option.

**Note:** This step must be repeated for each server that uses HTTPS.

**TIP** For information about the `Connector` element parameters, see [http://tomcat.apache.org/tomcat-7.0-doc/config/http.html#SSL_Support_-_APR/Native](http://tomcat.apache.org/tomcat-7.0-doc/config/http.html#SSL_Support_-_APR/Native).

4 Restart SAS Web Application Server and monitor the `logs\server.log` file. Log entries similar to the following indicate successful configuration:

```
[org.apache.catalina.core.AprLifecycleListener] APR capabilities: IPv6 [true], sendfile [true], accept filters [false], random [true].
[org.apache.catalina.core.AprLifecycleListener] Initializing FIPS mode...
[org.apache.catalina.core.AprLifecycleListener] Successfully entered FIPS mode
[org.apache.catalina.core.AprLifecycleListener] OpenSSL successfully initialized (OpenSSL 1.0.1c-fips 10 May 2012)
```

The previous steps are based on the procedure that is provided by VMware at [http://pubs.vmware.com/vfabric51/index.jsp?topic=com.vmware.vfabric.to-server.2.7/admin/manual-fips-140-mode.html](http://pubs.vmware.com/vfabric51/index.jsp?topic=com.vmware.vfabric.to-server.2.7/admin/manual-fips-140-mode.html). The steps are modified to include directory paths that are used in a SAS deployment and to configure SAS Web Application Server to use HTTPS.
Whitelist of Websites and Methods Allowed to Link to SAS Web Applications

About the Whitelist

Starting with the third maintenance release of SAS 9.4, websites that link directly to your SAS web applications, via URLs, must be added to a whitelist, or security filter, of allowed sites. The default configuration includes only SAS applications. This provides protection against browser-based vulnerabilities referred to as Cross Site Request Forgeries (CSRF). Any website that performs such activities as retrieving reports, using a single sign-on session, or linking to a SAS web application, needs to be explicitly added to the whitelist. Users that link to a SAS application from a company intranet or portal page that is not hosted in the SAS installation, you will encounter access denied messages.

The whitelist of the URLs that are allowed to link to SAS web applications can be specified during installation, using the SAS Deployment Wizard. Changes or updates to the whitelist can also be made using SAS Management Console. The list must be a comma-separated list of URLs, with each list entry including the protocol, host, and port number of the allowed URL. Wildcard characters can also be specified. The whitelist can also be manually edited. For more information, see “Modifying the Whitelist for URLs and HTTP Request Methods” on page 266.

Note: The port number must be specified if the whitelisted site uses port numbers other than the standard 80 for HTTP or 443 for HTTPS.

You can also whitelist certain HTTP request methods. This causes the SAS web applications to allow the specified types of requests, regardless of whether they originate from another website. If no HTTP request methods are whitelisted, then all types of HTTP requests are subject to CSRF protection. SAS web applications that use whitelisted request methods are susceptible to security attacks. For example, a user could be tricked into logging off or running a SAS stored process in the Stored Process Server if GET requests are skipped in security processing. The list of methods can also be manually edited. For more information, see “Modifying the Whitelist for URLs and HTTP Request Methods” on page 266.

Modifying the Whitelist for URLs and HTTP Request Methods

**CAUTION!** You can choose to disable whitelist checking. This could expose your environment to CSRF attacks. Disabling the security filter could expose your environment to attacks, which could compromise the security of your environment or your data.

To add, change, delete, or disable the whitelist for URLs and HTTP request methods, complete the following steps:

1. Log on to SAS Management Console.

2. On the **Plug-ins** tab, navigate to **Application Management** ➤ **Configuration Manager**.
3 Right-click **SAS Application Infrastructure** and select **Properties**.

4 Click the **Advanced** tab.

5 Complete one of the following:

- To add a URL or HTTP request method to the whitelist, if one has not been previously defined:
  
  1. Click **Add** and specify one of the following properties and required values:

     **Table 19.1 Whitelist Property Descriptions**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas.web.csrf.referers.allowNull</td>
<td>Specify whether to allow requests that are missing referer headers. When the</td>
</tr>
<tr>
<td></td>
<td>security filter is set to false, all incoming HTTP requests are required to</td>
</tr>
<tr>
<td></td>
<td>have a valid origin or referer header. You might set this property to false</td>
</tr>
<tr>
<td></td>
<td>for specific SAS web applications that do not have API endpoints and for which</td>
</tr>
<tr>
<td></td>
<td>you know that clients could always send the required header. For example, if a</td>
</tr>
<tr>
<td></td>
<td>site has an additional proxy server that always added these headers from</td>
</tr>
<tr>
<td></td>
<td>trusted clients. The default value is true.</td>
</tr>
<tr>
<td>sas.web.csrf.referers.blacklist</td>
<td>Specify a list of sites to block, that would otherwise be allowed by a wildcard</td>
</tr>
<tr>
<td></td>
<td>rule. For example, if you add *.example.com to the whitelist but wanted to</td>
</tr>
<tr>
<td></td>
<td>disallow test.example.com, you can add test.example.com to the blacklist.</td>
</tr>
<tr>
<td>sas.web.csrf.referers.knownHosts</td>
<td>Specify the allowed sites as a comma-separated list. This allows users to add</td>
</tr>
<tr>
<td></td>
<td>additional known hosts to the list of known hosts automatically calculated by</td>
</tr>
<tr>
<td></td>
<td>the system. To enable hostname1.example.com and hostname2.example.com, enter</td>
</tr>
<tr>
<td></td>
<td>the following: <a href="http://hostname1.example.com/">http://hostname1.example.com/</a>, <a href="http://hostname2.example.com/">http://hostname2.example.com/</a>.</td>
</tr>
<tr>
<td></td>
<td>Note: Omitting the trailing slash could allow sites to use a prefix attack to</td>
</tr>
<tr>
<td></td>
<td>bypass these protections.</td>
</tr>
<tr>
<td></td>
<td>Note: You can restrict a value to an application on the whitelisted site by</td>
</tr>
<tr>
<td></td>
<td>including the application's path in the value. Here is an example: http://</td>
</tr>
<tr>
<td></td>
<td>hostname.example.com/my-application/.</td>
</tr>
<tr>
<td>sas.web.csrf.referers.performCheck</td>
<td>Specify whether the security filter should be run. When set to false, no</td>
</tr>
<tr>
<td></td>
<td>checking will be performed regardless of the value of any other setting.</td>
</tr>
<tr>
<td></td>
<td>The default value is true.</td>
</tr>
</tbody>
</table>
### Configuring the Cross Domain Proxy Servlet through a Whitelist

#### About the Whitelist

Starting with the third maintenance release of SAS 9.4, in scenarios where applications are using the SAS mid-tier as a proxy for accessing external URLs, additional security has been added through a whitelist and logging. If an attempt is made to access a domain that is not on the whitelist, an error message will be generated. All external URLs that are accessed will be logged at warning level. These logs can be used for security audits.

The whitelist must be a comma-separated list of URLs, with each list entry including the protocol, host, and port number of the allowed URL.

**Note:** Omitting the trailing slash could allow sites to use a prefix attack to bypass these protections.

#### Modifying the Whitelist

To add, change, or delete the whitelist for URLs, complete the following steps:

1. Log on to SAS Management Console.
2. On the **Plug-ins** tab, navigate to **Application Management** ➔ **Configuration Manager**.
3. Right-click **SAS Application Infrastructure** and select **Properties**.

### Property Name and Value

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sas.web.csrf.referers.skipMethods</td>
<td>Specify the HTTP request method to exclude from security filtering as a comma-separated list. For example, to skip the GET, OPTIONS, and TRACE methods, enter the following: GET, OPTIONS, TRACE</td>
</tr>
</tbody>
</table>
4 Click the **Advanced** tab.

5 Complete one of the following:

- To add a URL to the whitelist, if one has not been previously defined:
  1. Click **Add**.
  2. In the **Property Name** field, enter `sas.web.cdps.knownHosts`. In the **Property Value** field, specify the allowed sites as a comma-separated list. To enable `hostname1.example.com` and `hostname2.example.com`, enter the following: `http://hostname1.example.com/, http://hostname2.example.com/`.

   **Note:** Omitting the trailing slash could allow sites to use a prefix attack to bypass these protections.
  3. Click **OK** to close the Define New Property dialog box.
  4. Click **OK** to close the SAS Application Infrastructure Properties dialog box.

- To change or delete a URL on the whitelist:
  1. Change or delete the value in the **Property Value** field that corresponds to the `sas.web.cdps.knownHosts` property.
  2. Click **OK** to close the SAS Application Infrastructure Properties dialog box.

6 To enable these properties to take effect, restart SAS Web Application Server.

---

**Enabling Support for Forward Proxy Authentication**

The SAS middle tier environment can be configured to forward external URL requests through a proxy. Starting with the third maintenance release of SAS 9.4 is support for the following proxy authentication protocols:

- no authentication
- basic authentication
- NT LAN Manager (NTLM)

The following table displays the JVM options that are required for each configuration:

<table>
<thead>
<tr>
<th>JVM Option</th>
<th>Proxy Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Dsas.flex.ntlmKerberosSupport</td>
<td>NTLM</td>
<td>When set to true, specifies whether Flex applications are supported.</td>
</tr>
<tr>
<td>JVM Option</td>
<td>Proxy Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-Dhttp.auth.ntlm.domain</td>
<td>NTLM</td>
<td>Specifies the domain name.</td>
</tr>
<tr>
<td>-Dhttp.auth.ntlm.workstation</td>
<td>NTLM</td>
<td>Specifies the workstation name. This option is optional.</td>
</tr>
<tr>
<td>-Dhttp.proxyUser</td>
<td>Basic authentication</td>
<td>Specifies the user name.</td>
</tr>
<tr>
<td>-Dhttp.proxyPassword</td>
<td>Basic authentication</td>
<td>Specifies the password.</td>
</tr>
<tr>
<td>-Dhttp.proxyHost</td>
<td>No authentication</td>
<td>Specifies the host name, or address, of the proxy server for HTTP connections.</td>
</tr>
<tr>
<td></td>
<td>Basic authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTLM</td>
<td></td>
</tr>
<tr>
<td>-Dhttps.proxyHost</td>
<td>No authentication</td>
<td>Specifies the host name, or address, of the proxy server for HTTPS connections.</td>
</tr>
<tr>
<td></td>
<td>Basic authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTLM</td>
<td></td>
</tr>
<tr>
<td>-Dhttp.nonProxyHosts</td>
<td>No authentication</td>
<td>Specifies the hosts that should be accessed without going through the proxy. Applies to both HTTP and HTTPS connections. The value of this property is a list of hosts, separated by the '</td>
</tr>
<tr>
<td></td>
<td>Basic authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTLM</td>
<td></td>
</tr>
<tr>
<td>-Dhttp.proxyPort</td>
<td>No authentication</td>
<td>Specifies the port number that the proxy server uses for HTTP connections.</td>
</tr>
<tr>
<td></td>
<td>Basic authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTLM</td>
<td></td>
</tr>
<tr>
<td>-Dhttps.proxyPort</td>
<td>No authentication</td>
<td>Specifies the port number that the proxy server uses for HTTPS connections.</td>
</tr>
<tr>
<td></td>
<td>Basic authentication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTLM</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

“Specifying JVM Options” on page 38
Tools and Utilities

Chapter 20

Using the SAS Web Infrastructure Platform Utilities ............................ 273

Chapter 21

SAS Configuration Scripting Tools ..................................................... 287
# Using the SAS Web Infrastructure

## Platform Utilities

### Using the DAVTree Utility to Manage WebDAV Content

273

- About the DAVTree Utility
- Start the Utility and Connect to a WebDAV Location
- Add Resources to WebDAV
- Edit a Text File in WebDAV
- Copy or Move a File in WebDAV
- Advanced Features

### Using the Package Cleanup Utility to Remove Packages

276

- Overview of the Package Cleanup Utility
- Deleting Packages
- List Packages
- Arguments
- Utility Logging and Debugging
- Examples

### Using JMX Tools to Manage SAS Resources

281

- About JMX and MBeans
- Accessing the SAS MBeans
- Understanding How to Use the SAS MBeans

---

## Using the DAVTree Utility to Manage WebDAV Content

### About the DAVTree Utility

The DAVTree utility is a stand-alone Java application that provides a tree view of WebDAV resources. The utility enables you to manipulate content by copying files to a WebDAV repository or by creating text files such as forms and templates.

The utility presents information in a tree view. When you select a resource item in the tree on the left side of the window, the WebDAV properties for the resource are displayed on the right side.
Here is an example DAVTree interface:

In the interface, you see only the content that you are authorized to see.

Start the Utility and Connect to a WebDAV Location

To use this utility, follow these steps:

1. Run the following command on Windows:
   
   ```
   SAS-configuration-directory\Lev\Web\Utilities\DAVTree.bat
   ```

   On UNIX:
   
   ```
   SAS-configuration-directory/Lev/Web/Utilities/DAVTree.sh
   ```

   The DAVTree utility appears.

2. Select File ▶️ Open.

   The DAV Location dialog box appears.

3. In the URL field, enter the URL for a WebDAV location. For example, enter the following URL and substitute the server name and port number of your WebDAV server (SAS Content Server):

   ```
   http://server:port/SASContentServer/repository/default/
   ```

4. If the WebDAV server was set up with a proxy, enter the proxy host and port.

5. Click OK. You are prompted for credentials.

6. Enter your administrator credentials in the logon dialog box.

   You can later connect to a different WebDAV location by repeating steps 2 through 6 and providing the URL for the new location.

Add Resources to WebDAV

Copy Files to DAVTree

You can copy both text files and binary files to the repository. To copy a file, click and drag the file from the file system to a folder in the DAVTree interface. This action can be performed on Windows systems and on UNIX systems that provide a graphical interface.
Note: To delete a resource, select the resource in the tree and then select Edit ▶ Delete. You are prompted to confirm the deletion.

Create a Text File

1 Position the cursor on the folder where you want to create the text file.

2 Select Edit ▶ Add.

You are prompted to confirm the action, and then an Add dialog box appears. Here is an example dialog box with data entered in the fields.

3 Select Resource.

4 In the field to the left of the Resource radio button, enter the name of the text file. If a file already exists with the name that you provide, the file is overwritten.

   The example shows a file with the name myFile.txt.

5 In the field below the Resource radio button, enter the text that you want the file to contain. Press Enter to start a new line.

   The example shows a file that contains the text string "Contents of myFile.txt."

6 If you want to define a custom WebDAV property, click New property. Two text fields appear in the gray properties panel. In the left field, add the property name. In the right field, enter the property value.

7 Click OK.

Create a Folder

1 Position the cursor on the folder where you want to create the new folder.

2 Select Edit ▶ Add.

   You are prompted to confirm the action, and then an Add dialog box appears.

3 Select Collection.

4 In the field to the left of the Collection radio button, enter the name that you want to give the folder.

5 Click OK.
Edit a Text File in WebDAV

To edit a text file, follow these steps:

1. Right-click the text file and select Edit. The Edit File dialog box appears and displays the contents of the file.
2. Make your changes to the text.
3. Click Save.

Copy or Move a File in WebDAV

To move a file from one location to another in WebDAV, in DAVTree click and drag the file to the desired location.

To copy rather than move a file, press and hold the Ctrl key while dragging.

Advanced Features

The DAVTree utility can be used as a diagnostic tool. The utility provides features such as locking files, versioning files, and modifying WebDAV properties.

**CAUTION! These are advanced WebDAV functions.** These functions are not described in this document. These functions should be performed only by someone who has WebDAV expertise.

Using the Package Cleanup Utility to Remove Packages

Overview of the Package Cleanup Utility

The Package Cleanup utility provides a simple, command-line interface for deleting or listing packages that have been published in a publication channel or in a WebDAV repository.

The SAS Publishing Framework supports channels that you define in the SAS Metadata Repository. Once channels have been defined, users can publish packages to the channels. For example, portal users can subscribe to available channels, view the persisted packages, and publish content (files, links, stored processes, and information maps).

Channels can be defined with archive or WebDAV persistent stores. When a package is published to a channel that is defined with a persistent store, the package is first persisted to that location and then it is published to all subscribers of that channel. All persisted packages have an expiration date. However, expired packages are not deleted automatically; you must explicitly delete them. You can use the Package Cleanup utility for this purpose.

Here is the path to the utility:
On Windows:
SAS-configuration-directory\Lev\Web\Utilities\PackageCleanup.bat

On UNIX:
SAS-configuration-directory/Lev/Web/Utilities/PackageCleanup.sh.

The Package Cleanup utility enables you to review basic information about a persisted package and delete both the metadata and the actual package. Deletions are based on the expiration date of the package. This utility supports the deletion of packages from either type of persistent store (archive or WebDAV). The utility also supports the deletion of packages that are not defined in any channel.

The Package Cleanup utility also supports a listing feature. The utility can be used to display information about packages that are published in a particular channel, packages that are not defined in any channel, and packages that exist on a WebDAV server.

Note: You must have the appropriate permissions on a channel in order to delete packages from the channel. See the “Authorization Model” chapter in the SAS Intelligence Platform: Security Administration Guide.

Deleting Packages

Delete Packages

To delete packages, follow these steps:

1  Run the command and specify the deletion date. You can also provide one of the following arguments:
   - a channel name in order to delete packages that are defined in a specific channel
   - a WebDAV URL in order to delete packages that are in the specified WebDAV location

   Note: If you do not provide the channel or WebDAV URL, then the utility deletes only orphaned packages that are not defined for any channel or WebDAV URL.

   After you run the command, the utility displays a list of packages that match your deletion criteria and prompts you to confirm deletion.

2  Respond to the prompt to confirm deletion of the packages or to exit without deleting any packages.

Minimal Syntax for Deleting Packages

Here is the minimal syntax for deleting packages that are defined in a channel:

```
PackageCleanup
   -d expiration-date
   -ch channel-name
   -metauser Metadata-Server-username
   -metapass Metadata-Server-password
   -domain authentication-domain
```
The utility deletes all packages in the specified channel that expire before the
date and time specified.

Here is the minimal syntax for deleting packages that are not defined in a
channel:

```
PackageCleanup
   -d expiration-date
   -metauser Metadata-Server-username
   -metapass Metadata-Server-password
   -domain authentication-domain
```

Here is the minimal syntax for deleting packages that are defined in a WebDAV
server:

```
PackageCleanup
   -url WebDAV-URL
   -username WebDAV-Server-username
   -password WebDAV-Server-password
   -d expiration-date
   -metauser Metadata-Server-username
   -metapass Metadata-Server-password
   -domain authentication-domain
```

Delete Specific Packages

To delete a specific package, specify `-package package-name` (or `-pkg
package-name`) along with the date. The `PACKAGE` option enables you to
specify the name of the package to delete.

Change Prompt Behavior

When you run the utility command, the utility displays a list of packages that
match your deletion criteria and prompts you to confirm deletion of all the
packages that are listed.

You can override this default behavior in order to be prompted for each package
individually.

To override the default, specify `-prompteach`. You are then prompted to delete
each package that meets the deletion criteria. After each package is processed,
the utility displays a final list of all packages that were selected. You can then
choose to delete all of those packages or exit without deleting any packages.

You can also turn off prompting altogether by specifying `-noprompt`. When you
run the utility in batch mode, you must use the `-noprompt` option (unless shell
programming is provided to respond to the prompts). It is best to run with
prompts when you are learning how to use the application. With prompts, you
can review proper date formatting and correct package deletion candidates with
the option to exit without deleting any packages.

List Packages

To obtain a list of packages, run the command and specify the `-list` option.
You can also provide one of the following arguments:

- a channel name in order to list packages that are defined in a specific
  channel
a WebDAV URL in order to list packages that are in the specified WebDAV location

Note: If you do not provide the channel or WebDAV URL, then the utility displays only orphaned packages that are not defined for any channel or WebDAV URL.

The LIST option lists the following information for each package:

- package name
- date and time that the package was created
- date and time that the package expires

Here is the minimal syntax for listing packages that are defined in a channel:

    PackageCleanup
    -list
    -ch channel-name
    -metauser Metadata-Server-username
    -metapass Metadata-Server-password
    -domain authentication-domain

Here is the minimal syntax for listing packages that are not defined in a channel:

    PackageCleanup
    -list
    -metauser Metadata-Server-username
    -metapass Metadata-Server-password
    -domain authentication-domain

Here is the minimal syntax for listing packages that are defined in a WebDAV server:

    PackageCleanup
    -list
    -url WebDAV-URL
    -username WebDAV-Server-username
    -password WebDAV-Server-password
    -metauser Metadata-Server-username
    -metapass Metadata-Server-password
    -domain authentication-domain

Arguments

The utility supports the following arguments:

- **channel** | **-ch channel-name**
  Specify the channel that contains the packages that you want to list or delete.

- **deletionDate** | **-d "expiration-date"**
  Specify the expiration date and time for the packages to be deleted. You can also use this argument when you list packages. The utility deletes or lists packages that have an expiration date before the date and time that you specify. The date and time should be enclosed in quotation marks. Format: "yyyy.MM.dd at hh:mm"

- **list**
  The utility displays a list of packages (no deletion occurs).

- **metauser Metadata-Server-username**
  Specify the user name to use when connecting to the SAS Metadata Server.
Utility Logging and Debugging

By default, application activity is sent to the Java standard out console. If you want to log to a file, use the LOGFILE option. For example, you might specify -logfile c:\mylog.file. If the log file already exists, then the log lines are appended to the current file.

Use the DEBUG option to enable debugging-level information. This option provides debugging information for all of the Foundation Services as well as the utility. This option should be used only when you experience problems with the utility and want to determine the cause.

Examples

This example deletes all packages published to the Sales channel that have an expiration date before October 7, 2009, at 12:59 p.m.

PackageCleanup -ch Sales -d "2009.10.07 at 12:59 PM" -metauser userX -metapass passX -domain DefaultAuth
This example uses the PROMPTEACH option, which enables you to confirm deletion of each package individually.

```
PackageCleanup -ch Sales -d "2009.10.07 at 12:59 PM" -metauser userX -metapass passX -domain DefaultAuth -prompteach
```

This example deletes a specific package that is defined in the Sales channel. The PKG option is specified to identify the exact package to delete. In this example, the package is named s109513698.spk and has an expiration date of October 7, 2009, at 12:59 p.m.

```
PackageCleanup -ch Sales -d "2009.10.07 at 12:59 PM" -pkg s109513698.spk -metauser userX -metapass passX -domain DefaultAuth
```

This example deletes all packages that are not defined in any channel. Only packages that are not defined in a channel and have an expiration date before October 7, 2009, at 10:00 a.m. are deleted.

```
PackageCleanup -d "2009.10.07 at 10:00 AM" -metauser userX -metapass passX -domain DefaultAuth
```

This example deletes packages that have been published to a WebDAV server. The utility connects to the server using the specified URL and deletes all packages published to that location that have an expiration before October 7, 2009, at 05:00 a.m.

```
PackageCleanup -d "2009.10.07 at 05:00 AM" -url http://myhost.com/Sales/Packages -username davUserX -password davPasswordX -metauser userX -metapass passX -domain DefaultAuth
```

This example deletes a specific package from a WebDAV server. The PKG option is used to provide the name of the package to delete. The utility connects to the server using the specified URL and deletes the package named s3964865240.

```
```

This example lists packages (does not delete) by using the LIST option. Note that the -d argument is not required when listing packages. This example lists all packages that are published in the Sales channel.

```
PackageCleanup -list -ch Sales -metauser userX -metapass passX -domain DefaultAuth
```

This example uses the LIST option to list all packages with an expiration date before October 7, 2009, at 12:00 p.m.

```
PackageCleanup -ch Sales -d "2009.10.07 at 12:00 PM" -metauser userX -metapass passX -domain DefaultAuth -prompteach -list
```

---

**Using JMX Tools to Manage SAS Resources**

**About JMX and MBeans**

SAS servers implement common administrative interfaces. These interfaces enable you to perform basic administrative functions such as stopping, pausing,
and resuming servers. You can also use the interfaces to monitor the health of
the servers via real-time and historical metrics. Java Management Extensions
(JMX) is a Java technology that supplies tools for managing and monitoring
applications, system objects, devices (such as printers), and service-oriented
networks. JMX managed beans, known as MBeans, have been implemented to
provide a standard way of managing SAS resources.

Accessing the SAS MBeans

About Accessing the SAS MBeans
You can use any of the standard JMX monitoring tools to access the MBeans
that manage SAS resources. To use these tools, you must do the following:

1. Enable access to the MBeans from the web application server. See
   "Configure the Web Application Server to Enable JMX Client Access" on
   page 282.

2. Use an application to connect and access the SAS MBeans. Follow the
   specific instructions for your JMX tool. For information about using the
   JConsole tool, see "Manage SAS Resources Using JConsole" on page 282.

Configure the Web Application Server to Enable JMX Client Access
You configure the web application server to enable access to the MBeans by
setting specific Java system options.

Specify the following Java Virtual Machine (JVM) argument to access the
MBeans locally:

com.sun.management.jmxremote

Specify the following JVM argument to access the MBeans from a remote
system. Replace portNum with the port number to use for JMX RMI connections:

com.sun.management.jmxremote.port=portNum

Remote monitoring and management requires security to ensure that
unauthorized persons cannot control or monitor your application. It is
recommended that you set the following JVM arguments when MBeans are
accessed remotely:

com.sun.management.jmxremote.authenticate=true | false
com.sun.management.jmxremote.ssl=true | false

For information about these arguments, see the Java documentation.

Manage SAS Resources Using JConsole
JConsole is a JMX tool that is included with the standard Java Development Kit
(JDK). The information provided through JMX technology enables JConsole to
provide information about application performance and functions. You can use
JConsole to interact with the JMX MBeans that are available to manage SAS
resources. The console's simple user interface displays all MBeans in a tree
navigator on the left side of the window. When you select a specific MBean, its
attributes, operations, notifications, and other information are displayed on the
right side of the window.
To access information about SAS resources using JConsole, follow these steps:

1. Start JConsole by running the following command:

   ```
   JDK-HOME\bin\jconsole
   ```

2. Connect to the MBean server as follows:
   - If you are accessing the MBeans locally, the **Local** tab should display every JVM that is running on the local system that was started with the same user ID as JConsole. Select the appropriate JVM and click **Connect**.
   - If you are accessing the MBeans remotely, follow these steps:
     1. Select the **Remote** tab.
     2. Enter the host on which the JVM is running, along with the port where the RMI connector was registered.
     3. You might need to specify credentials if authentication to the MBean server is required.
     4. Click **Connect** to connect to the MBean server.

3. Select the **MBeans** tab. This tab displays a tree view of all the registered MBeans.

4. Expand the `com.sas.services` domain to see all MBeans registered in this domain.

5. Select the **ServerFactory** MBean.

6. In the right pane, select the **Operations** tab. You can now see the operations (listing, stopping, pausing, and so on) so that you can list the defined SAS servers and manage your running SAS servers. When you invoke one of the manage-server operations, a new MBean is registered. The MBean is connected to the specified, running SAS server. The newly registered MBean can then be used to manage and monitor that particular SAS server.

**Understanding How to Use the SAS MBeans**

**About the SAS MBeans**

There are three primary MBeans provided by the SAS Web Infrastructure Platform for managing and monitoring SAS resources:

- **ServerFactory MBean**
- **Spawner MBean**
- **Server MBean**

The following sections describe these MBeans.

**ServerFactory MBean**

The **ServerFactory** MBean is the starting point for managing SAS servers. This MBean is registered during deployment of the SAS Web Infrastructure Platform and is named as follows:
com.sas.services:type=ServerFactory

During initialization, the ServerFactory MBean connects to the SAS Metadata Server. This enables the MBean to list all SAS servers defined in the metadata. The MBean can then be used to register additional MBeans that enable the running servers to be managed and monitored directly. The ServerFactory MBean does not have any attributes, but supports three operations:

listDefinedServers()
  provides a list of SAS IOM servers that are defined in the Metadata Server. Information that is returned for each defined server includes the server name, host, port, and server type. To begin actively managing a server, specify the name of the server on the manageServerByServerName operation.

manageServerByServerName(String ServerName, String Host)
  registers a Server MBean that enables you to actively manage the specified IOM server. The newly registered MBean connects to the running IOM server and can then be used to manage and monitor that server. The host name can be left blank if the IOM server is defined to run on only one host. If defined to run on multiple hosts, the proper host name should be provided.

  The manageServerByServerName() operation does not work on a server that is spawned by the SAS Object Spawner.

manageServer(String Host, Integer Port, String Username, String Password)
  registers a Server MBean that enables you to actively manage the specified IOM server. The IOM server that is managed is identified by the host and port provided on the manageServer operation. The newly registered MBean can be used to manage and monitor that specific IOM server. This operation is useful when the IOM server is not defined in the Metadata Server.

Spawner MBean

The Spawner MBean is created whenever an IOM Spawner is identified in one of the ServerFactory MBean's manageServer operations. The name of the registered MBean uses the form:

com.sas.services:type=Server,serverType=Spawner,
    name="Server Name",
    host=Host Name, port=Port

The Spawner MBean enables you to manage and monitor the running Object Spawner. You can perform SAS Spawner operations such as stop, pause, and resume.

Here are some commonly used Spawner MBean attributes:

- the number of times the counters have been reset
- the amount of time the server has been idle
- the number of currently connected clients
- the server start time
- the number of currently abandoned servers
- the number of currently launched servers
- the total number of servers that have been launched
- the number of currently failed servers
- the process identifier of the server process
- the amount of time spent in server method calls
- the number of method calls that the server has processed

**Server MBean**

The Server MBean is created whenever a SAS server is identified in one of the ServerFactory MBean's manageServer operations or when a server is managed via the Spawner MBean's manageLaunchedServer(s) operation.

A server MBean can represent a SAS Workspace Server, a SAS Stored Process Server, a SAS Metadata Server, or a SAS OLAP Server. The name of the registered SAS Server MBean uses one of these three forms:

- `com.sas.services:type=Server, serverType=Workspace, logicalServer="LogicalServerName", name="Server Name", instanceid="Unique instance ID"`
- `com.sas.services:type=Server, serverType=StoredProcess, logicalServer="LogicalServerName", name="Server Name", instanceid="Unique instance ID"`
- `com.sas.services:type=Server, serverType=Table, logicalServer="LogicalServerName", name="Server Name", host=Host Name, port=Port Number`

The Server MBean enables you to manage and monitor the running SAS server. You can perform server operations such as stop, pause, and resume.

Here are some commonly used Server MBean attributes:

- the number of times the counters have been reset
- the amount of time the server has been idle
- the number of currently connected clients
- the server start time
- the last time the counters were reset
- the execution state of the server
- the amount of time spent in server method calls
- the number of method calls that the server has processed
- the number of clients that the server has serviced
- the process identifier of the server process
- the identity under which the server process is executing
The configuration scripting tools enable administrators to perform the following tasks:

- **Create the configuration for SAS Web Application Server rather than following the manual instructions.** If the automatic configuration option was disabled in the SAS Deployment Wizard, then the SAS Deployment Wizard provides an Instructions.html file that describes the configuration steps to perform the web application server configuration. You can use the configuration scripting tools to perform these steps automatically instead of manually.

- **Rebuild the web application server configuration.** The results are identical to what is performed by the SAS Deployment Wizard and SAS Deployment Manager.

The SAS configuration scripting tools also enable an administrator to perform the following additional tasks:

- Use a command line to perform a configuration operation on a single resource. For example, creating a server instance can be performed with a single command.

- Edit property files that are associated with specific resources and then update the resources with the configuration scripting tools.

- Use existing property files as templates for creating additional resources. For example, an administrator can copy the definitions for SASServer1 to a new file and then use it as a template to create a new server instance.
Special Considerations

- If you are rebuilding or reconfiguring a web application server, then make sure that all the web application servers are stopped.

- If you encounter errors while configuring a web application server, review the properties that are being used by the tool and rerun the tool. The tool can be run many times without deleting the configuration between runs, so long as the server is not running. If the server starts in between runs, there can be locks on files that prevent subsequent runs from succeeding.

Scripting Tool for SAS Web Application Server

Command Syntax

Start, Stop, and Restart Syntax

The syntax for the start, stop, and restart operations is as follows:

```
appsrvconfig.cmd start
appsrvconfig.cmd stop
appsrvconfig.cmd restart
```

Note: For UNIX operating environments, the command is `appsrvconfig.sh`.

The requested operation is performed on all the instances of SAS Web Application Server that are on the same machine.

The script is located in the `SAS-configuration-directory\Lev\Web\Scripts\AppServer` directory.

Command Syntax

The positional command syntax is as follows:

```
<operation> <resourceType> <targetName> <scope ...>
```

The following example shows the commands for starting a server and deploying an application:

```
start server SASServer1 global global
deploy application SASWIPAdmin9.4 server SASServer1
```

TIP You can deploy all applications with `deploy application all server SASServer1.`
Resource Types

The following table provides a list of resource types and identifies the operations and scope that apply to the resource type.

Table 21.1  Resource Types, Operations, and Scopes

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Operations</th>
<th>Scopes</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>configure, unconfigure, start, stop, restart</td>
<td>global</td>
</tr>
<tr>
<td>mailsession</td>
<td>configure, unconfigure</td>
<td>server</td>
</tr>
<tr>
<td>datasource</td>
<td>configure, unconfigure</td>
<td>server</td>
</tr>
<tr>
<td>loginmodule</td>
<td>configure, unconfigure</td>
<td>server</td>
</tr>
<tr>
<td>application</td>
<td>deploy, undeploy</td>
<td>server</td>
</tr>
<tr>
<td>jmserver</td>
<td>configure, unconfigure, start, stop, restart</td>
<td>global</td>
</tr>
<tr>
<td>jms</td>
<td>configure, unconfigure</td>
<td>server</td>
</tr>
<tr>
<td>balancer</td>
<td>configure, unconfigure</td>
<td>global</td>
</tr>
<tr>
<td>member</td>
<td>configure, unconfigure</td>
<td>global</td>
</tr>
<tr>
<td>proxypass</td>
<td>configure, unconfigure</td>
<td>global</td>
</tr>
<tr>
<td>proxyserver</td>
<td>configure, unconfigure, start, stop, restart</td>
<td>global</td>
</tr>
<tr>
<td>cache_locator</td>
<td>configure, unconfigure, start, stop, restart</td>
<td>global</td>
</tr>
<tr>
<td>cache_server</td>
<td>configure, unconfigure, start, stop, restart</td>
<td>global</td>
</tr>
</tbody>
</table>

Managing Credentials

Credentials are required to configure resources such as data sources and login modules. You can store credentials in the SAS-configuration-directory \Lev1\Web\Scripts\AppServer\props\credentials.properties file.

By default, the SAS Deployment Wizard does not persist credentials in the specified file. When you run the configuration scripting tool, you are prompted for all credentials that are required to configure the resources—but are not specified in the credentials.properties file.

If the option to cache credentials was enabled when the SAS Deployment Wizard was run, then the credentials are stored in the credentials.properties file. In this case, the configuration scripting tool reads the credentials from the file rather than prompting for them. When the Update passwords feature of the SAS Deployment Manager is used, the passwords for the login modules and mail...
sessions are updated in the credentials file. Passwords for data source definitions are not updated.

**Log File**

Details for the command execution are stored in the `SAS-configuration-directory\Lev1\Web\Scripts\AppServer\logs\config.log` file. The SAS Deployment Wizard invokes the configuration scripting tool, so this file already contains messages for an installed system. This file can be useful for troubleshooting middle-tier configuration tasks performed with the SAS Deployment Wizard and the SAS Deployment Manager.

**Rebuilding the Configuration for SAS Web Application Server**

You can rebuild the server configuration by running the configuration scripting tool. The tool can re-create the entire configuration and restore it to the originally configured state. The tool configures the resources according to the settings in the `props\appserver.properties` file.

**Executing a Batch Script**

You can supply a file that contains a series of commands for the configuration scripting tool to execute. You can supply a file with different commands to configure different resources. The following example shows the syntax for using the configuration scripting tool with a commands file that is named `cmds.txt`:

```
appsrvconfig.cmd cmds.txt
```

The following example shows the commands for undeploying and redeploying the SAS Web Application Themes:

```
undeploy application SASThemes9.4 server SASServer1
deploy application SASThemes9.4 server SASServer1
```

If you are creating a resource that requires credentials, such as a data source, remember to create property keys in the `credentials.properties` file.

**Executing a Single Command**

You can execute a single command on a single resource from a command line. The following example shows how to undeploy SAS Web Application Themes:

```
appsrvconfig.cmd undeploy application SASThemes9.4 server SASServer1
```

**Properties Reference**

**Global Properties**

A properties file is used by the configuration scripting tool to configure SAS Web Application Server. This properties file is found in `SAS-configuration-directory\Lev1\Web\Scripts\AppServer\props\appserver.properties`. Each of the global properties are described in the following list:
global.1.activeMQInstallDir
identifies the path to the JMS Broker software.

global.1.autoConfigure
is a Boolean value. If set to false, then manual configuration is requested and the SAS Deployment Wizard creates a sample domain and configures servers in off-line mode only. All configuration steps that are run outside of SAS Deployment Wizard and SAS Deployment Manager are automated regardless of this setting.

global.1.autoDeploy
is a Boolean value. If set to false, then the SAS Deployment Wizard does not deploy the SAS web applications. This property is not used by the configuration scripting tool. This property is used by SAS Deployment Wizard to generate documentation.

global.1.configLevWebDir
identifies the path to SAS-configuration-directory\Lev\Web.

global.1.configLevWebStagingDir
identifies the path to SAS-configuration-directory\Lev\Web\Staging.

global.1.containerType
identifies SAS Web Application Server. The supported value is vfabrictcsvr.

global.1.deployAgentPickList
identifies the path to the picklist for the deployment agent client. The picklist specifies the versions of libraries to load.

global.1.gemFireInstallDir
identifies the path to the Cache Locator software.

global.1.isDeleted
is a Boolean value. If set to true, then this resource has been marked as deleted.

global.1.isScsPrimary
is a Boolean value. If set to true, then the SAS Content Server that is deployed on this machine is the primary instance.

global.1.jmsSecurity
is a Boolean value. This property is not used by the configuration scripting tool. This property is used by SAS Deployment Wizard to generate documentation.

global.1.jreHome
identifies the path to SAS-home\SASPrivateRuntimeEnvironment\9.4\jre.

global.1.osType
identifies the operating system for the SAS middle-tier machines. Valid values are win, unix, or zos.

global.1.runasService
identifies whether SAS Web Application Server is managed as a Windows service.

global.1.scriptingDir
identifies the path to SAS-configuration-directory\Lev\Web\Scripts.
global.1.scriptingServerDirName identifies the directory name that the configuration scripting tool uses. For SAS Web Application Server, this value is AppServer.

global.1.tcServerInstallDir identifies the path to SAS-home\SASWebApplicationServer\9.4.

global.1.tcServerInstanceDir identifies the path to SAS-configuration-directory\Lev\Web\WebAppServer.

global.1.tcServerName identifies the product name for the server. The default value is SAS Web Application Server.

global.1.tcServerVendor identifies the vendor that supplied the web application server software. The default value is SAS.

global.1.tcServerVersion identifies the version of SAS Web Application Server. The default value is 9.4.

global.1.vjrDirectory identifies the path to SAS-home\SASVersionedJarRepository\eclipse.

global.1.webServerCommonDir identifies the path to SAS-configuration-directory\Lev\Web\Common\WebServer.

global.1.webServerHost identifies the host name for the SAS Web Server.

global.1.webServerHttpPort identifies the network port number that the SAS Web Server uses for HTTP.

global.1.webServerHttpsPort identifies the network port number that the SAS Web Server uses for HTTPS.

global.1.webServerInstanceDir identifies the path to SAS-configuration-directory\Lev\Web\WebServer.

global.1.webServerIsConfigured is a Boolean value. Indicates whether the SAS Deployment Wizard was requested to configure the SAS Web Server.

global.1.webServerOsType identifies the operating system for the SAS middle-tier machines. Valid values are win or unx.

global.1.webServerProtocol identifies the protocol that is used by the SAS Web Server. Valid values are http or https.

global.1.webServerRemoteInstanceDir identifies the path to SAS-configuration-directory\Lev\Web\WebServer. This property is used when SAS Web Server is deployed on a different operating system than SAS Web Application Server.

global.1.windowsServiceNamePrefix identifies the service name prefix when SAS Web Application Server is managed as a Windows service. A sample value is SAS [Config-Lev1].
Credential Properties

All properties that are related to credentials are stored in the credentials.properties file. The tool prompts you for these properties. This properties file does not need to be edited directly. These values are cleared from the file after the tool completes if the global property webappsrvScriptingCacheCredentials is set to false. When stored, these values are stored in SAS base-64 encoding, not clear-text. If you chose to store passwords in this file, then they are updated when you use the Update passwords feature of the SAS Deployment Manager.

datasource.create_resource_passwd
  is the data source user password.

datasource.create_resource_userid
  is the data source user name.

domain.createloginmodule_SASTrust_passwd
  is the SAS Trusted User password.

domain.createloginmodule_SASTrust_userid
  is the SAS Trusted User. This identity is used to configure the JAAS login module.

mailsession.create_SASMailSession_passwd
  is the mail session user password.

mailsession.create_SASMailSession_userid
  is the mail session user ID. This credential is used only if the mail session property mailsrvRequiresAuthentication is set to true.

Resource Properties

Each property file governs the configuration of a specific resource. The next section lists and describes a group of properties that are common to many resources. The subsequent sections identify properties that are specific to each resource type.

Properties Common to Many Resources

The following properties are common to a number of resource types.

delmed
  is a Boolean value. If set to true, then this resource has been marked as deleted.

delmedTargets
  is a comma-separated list of target servers that contain this resource that are marked for deletion. A Delete operation removes these targets and removes the resource if no targets remain.

targets
  is a comma-separated list of servers that this resource instance is targeted to.

thisOperation
  is a field that is used internally by SAS Deployment Wizard and SAS Deployment Manager to manage resource files. It is not used by the configuration scripting tool.
thisTarget
  is a field that is used internally by SAS Deployment Wizard and SAS Deployment Manager to manage resource files. It is not used by the configuration scripting tool.

Application Properties

These resources represent applications deployed in SAS Web Server. Each application is associated with a balancer. The properties are named in the following pattern application.n.property.

archive
  identifies the path to the EAR or WAR file for the application.

balancerName
  identifies load balancer name that the application belongs to.

classLoaderMode
  is a Boolean value. This property is not used by SAS Web Application Server.

classLoaderPolicy
  is a Boolean value. This property is not used by SAS Web Application Server.

deployEJB
  is a Boolean value. This property is not used by SAS Web Application Server.

deployWS
  is a Boolean value. This property is not used by SAS Web Application Server.

explode
  is a Boolean value. When false, it indicates that the archive file for the application is copied and then deployed. When true, the application is extracted from the archive and the application is deployed from the files.

isClustered
  is a Boolean value. When false, the application is not deployed to additional cluster members when they are created. When true, the application deployed to each additional cluster member that has the same balancerName value when the cluster member is created.

isDeleted
  is a Boolean value. If set to true, then this resource has been marked as deleted.

loadOrder
  This property is not used by SAS Web Application Server.

name
  identifies the name of the application, as it is used by other SAS software applications (for example, SASWebReportStudio4.4).

serverName
  identifies the server that the application is deployed to.

webapps
  identifies the WAR file and context root mapping for each web application in the archive.
Balancer Properties
These resources represent load balancers that are deployed in SAS Web Server. The properties are named in the following pattern `balancer.n.property`.

- **isDeleted**
  - is a Boolean value. If set to `true`, then this resource has been marked as deleted.

- **name**
  - identifies the name of the balancer. This value is referenced in the application properties.

- **sessionid**
  - identifies the session identifier name. The name is used as a cookie or request parameter for sticky sessions to ensure that subsequent requests by a user are directed to a single instance of SAS Web Application Server.

Cache Locator Properties
These resources represent the Cache Locator locator processes. A locator process is used as an alternative to multicast messaging. The properties are named in the following pattern `cache_locator.n.property`.

- **force**
  - is a Boolean value. When set to `true`, the configuration scripting tools configure the locator.

- **host**
  - identifies the host name for the cache locator.

- **isDeleted**
  - is a Boolean value. If set to `true`, then this resource has been marked as deleted.

- **locators**
  - identifies the list of cache locators that this locator can communicate with.

- **name**
  - identifies the name for this cache locator.

- **port**
  - identifies the network port number that the cache locator uses for communication.

Cache Server Properties
These resources represent the Cache Locator processes. A locator process is used as an alternative to multicast messaging. The properties are named in the following pattern `cache_server.n.property`.

- **directory**
  - identifies the path to the Cache Locator software.

- **force**
  - is a Boolean value. When set to `true`, the configuration scripting tools configure the server.

- **isDeleted**
  - is a Boolean value. If set to `true`, then this resource has been marked as deleted.
Data Source Properties

Data source properties are used to configure JDBC data sources. The properties are named in the following pattern `datasource.n.property`.

- **classpath** identifies the JAR files required for the JDBC driver.
- **driver** identifies the fully qualified JDBC driver class name.
- **isDeleted** is a Boolean value. If set to `true`, then this resource has been marked as deleted.
- **jndiName** identifies the data source JNDI name. This name is configured in application configuration files and should not be changed without corresponding changes to the applications that use this data source.
- **name** identifies the data source name. This name must be unique.
- **password** identifies the password that is used to connect to the database server.
- **serverName** identifies which SAS Web Application Server the data source is associated with.
- **url** identifies the JDBC URL for communication with the database server.
- **username** identifies the user ID that is used to connect to the database server.
- **validationQuery** identifies the test query that the SAS Deployment Wizard uses to check that the data source is configured correctly.

JMS Resource Properties

JMS resource properties are used to configure JMS queues, topics, and connection factories. The properties are named in the following pattern `jms.n.property`.

- **agedTimeout** This property is not used with SAS Web Application Server.
- **autoCreate** is a Boolean value. The name of the JMS system module to target this resource to.
- **connectionFactoryType** identifies whether this JMS resource is a connection factory for topics or queues.
- **connectionTimeout** identifies the number of seconds before connections to the JMS resource are closed due to inactivity.
- **deliveryMode** This property is not used with SAS Web Application Server.
host
   identifies the host name.

isDeleted
   is a Boolean value. If set to true, then this resource has been marked as deleted.

jndiName
   is the global JNDI name used to look up the destination within the JNDI namespace. This name is configured in application configuration files and should not be changed without corresponding changes to the applications that use this JMS resource.

moduleName
   This property is not used with SAS Web Application Server.

name
   is the name of this JMS resource.

port
   identifies the network port number for connection factory JMS resources. For other JMS resources, the value is zero.

purgePolicy
   This property is not used with SAS Web Application Server.

readAhead
   This property is not used with SAS Web Application Server.

reapTime
   This property is not used with SAS Web Application Server.

schemaName
   This property is not used with SAS Web Application Server.

scope
   This property is not used with SAS Web Application Server.

serverName
   identifies which SAS Web Application Server name the JMS resource is associated with.

sIBusDestType
   This property is not used with SAS Web Application Server.

type
   is the type of JMS resource to be configured. Supported values are ConnectionFactory, Queue, and Topic.

unusedTimeout
   This property is not used with SAS Web Application Server.

xAEnabled
   This property is not used with SAS Web Application Server.

JMS Server Properties

JMS server resource properties are used to configure Java Message Services servers. The properties are named in the following pattern

jmsserver.n.property.

host
   identifies the host name.
isDeleted
   is a Boolean value. If set to true, then this resource has been marked as deleted.

name
   is the name of this JMS server.

port
   identifies the network port number for the server.

**Login Module Properties**

JAAS login module properties are used to configure login modules. The properties are named in the pattern `loginmodule.n.property`.

**className**
   identifies the Java class that is used for the login module.

**flag**
   identifies whether authentication must succeed with the module (required) or one of the following: requisite, sufficient, optional.

isDeleted
   is a Boolean value. If set to true, then this resource has been marked as deleted.

options
   identifies the name and value pair mappings for options to use with the login module.

**policyName**
   identifies the login policy for the login module.

**serverName**
   identifies which SAS Web Application Server name the login module is associated with.

**trustedUserPassword**
   identifies the password for the trusted user. The password is encoded and stored in the credentials.properties file, if caching credentials was enabled when the SAS Deployment Wizard was run.

**trustedUsername**
   identifies the user ID for the account that is used to communicate with the SAS Metadata Server.

**Mail Session Properties**

Mail session properties are used to configure mail sessions. The properties are named in the pattern `mailsession.n.property`.

**host**
   identifies the host name of the simple mail transfer protocol server.

isDeleted
   is a Boolean value. If set to true, then this resource has been marked as deleted.

**jndiName**
   is the global JNDI name used to look up the mail session within the JNDI namespace. This name is configured in application configuration files and
should not be changed without corresponding changes to the applications
that use this resource.

name
   identifies the name of the mail session resource.

password
   identifies the password for the user ID. This property is used when the mail
   server requires authentication.

port
   identifies the network port number for the mail server.

serverName
   identifies which SAS Web Application Server name the mail session is
   associated with.

username
   identifies the user ID for logging on to the mail server. This property is used
   when the mail server requires authentication.

Member Properties

Member properties are used to configure SAS Web Server. The member
properties are used together with balancer properties to identify the instances of
SAS Web Application Server and the applications. The properties are named in
the following pattern `member.n.property`.

host
   identifies the host name of the instance of SAS Web Application Server.

isDeleted
   is a Boolean value. If set to `true`, then this resource has been marked as
deleted.

name
   identifies the name of the instance of SAS Web Application Server.

port
   identifies the network port number for the instance of SAS Web Application
   Server.

protocol
   is one of `http` or `https`.

route
   is a Boolean value. If set to `true`, then a routing directive is added to the
   SAS Web Server configuration file for this member.

target
   identifies the balancer that this member is associated with.

Proxy Properties

The proxy properties are used to configure SAS Web Server as a reverse proxy
for the applications that are deployed to SAS Web Application Server instances.
The properties are named in the following pattern `proxypass.n.property`.

balancerName
   identifies the balancer that is associated with the application.
isDeleted
   is a Boolean value. If set to true, then this resource has been marked as deleted.

name
   identifies the application context root to proxy.

pass
   is a Boolean value. If set to true, then SAS Web Server is configured to proxy the application.

Server Properties
Server properties are used to configure SAS Web Application Server instances. The properties are named in the following pattern server.n.property.

cacheLocatorPort
   identifies the network port number for the Cache Locator.

cacheLocators
   identifies the instances of the Cache Locator.

host
   identifies the host name for SAS Web Application Server.

httpPort
   identifies the network port number that this server uses for HTTP connections.

httpsPort
   identifies the network port number that this server uses for HTTPS connections.

isDeleted
   is a Boolean value. If set to true, then this resource has been marked as deleted.

jmxPort
   identifies the network port number that the server uses for Java Management Extensions communication.

jvmOptions
   is a list of JVM options for this server.

multiplier
   identifies the number of vertical cluster members to configure identically to this server.

name
   identifies the name for this SAS Web Application Server.

serverId
   identifies that the resource type is a server.

name
   identifies the name of SAS Web Application Server.

sessionCookieName
   identifies the value for the cookie that is associated with connections to this server. Sticky sessions and cookies are used to ensure that all connections for a user are routed to the same server instance.

shutdownPort
   This property is not used with SAS Web Application Server.
Appendices

Appendix 1  
Configuring the SAS Environment File  ................................................. 303

Appendix 2  
Administering Multicast Options .......................................................... 307
About the SAS Environment File

A SAS environment file defines the available set of SAS environments for SAS client applications, and is generated during the configuration of the SAS Web Infrastructure Platform. The SAS Logon Manager includes a servlet that provides default information for the initial deployment. The sas-environment.xml file is automatically deployed on SAS Web Server at http://hostname.example.com/sas/sas-environment.xml.

Your site might have requirements that application clients interact with separate development, test, and production environments. Or, you might choose to have separate SAS deployments to support distinct business units. In either scenario, when multiple environments are required, you can customize and deploy the sas-environment.xml file as needed.

Make sure that the file is available to SAS desktop clients. In environments that protect URLs with third-party products like IBM Tivoli Access Manager WebSEAL or CA SiteMinder, do not protect the URL to the file. The SAS desktop clients that use the file are unable to respond to a prompt for credentials. In these environments, you can deploy the file from a different HTTP server. Update the SAS desktop clients with the new location if you change it.

Configuring the SAS Environment File

Customizing the SAS Environment File

The sas-environment.xml is located in the SAS-configuration-directory \Lev1\Web\WebServer\htdocs\sas directory.
Here is a sample sas-environment.xml file that is configured for two environments:

```xml
<?xml version="1.0" encoding="UTF-8">
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2.0">
  <environment name="Red" default="false">
    <desc>test server Red for SAS Financial Management Studio</desc>
    <service-registry>http://red.example.com:80/SASWIPClientAccess/remote/ServiceRegistry</service-registry>
    <service-registry interface-type="soap">http://red.example.com:80/SASWIPSoapServices/services/ServiceRegistry</service-registry>
    <service-registry interface-type="rest">http://red.example.com:80/SASWIPClientAccess/rest</service-registry>
  </environment>
  <environment name="Blue" default="true">
    <desc>test server Blue for SAS Financial Management Studio</desc>
    <service-registry>http://blue.example.com:80/SASWIPClientAccess/remote/ServiceRegistry</service-registry>
    <service-registry interface-type="soap">http://blue.example.com:80/SASWIPSoapServices/services/ServiceRegistry</service-registry>
    <service-registry interface-type="rest">http://blue.example.com:80/SASWIPClientAccess/rest</service-registry>
  </environment>
</environments>
```

The service registry that is specified in the file enables desktop client applications to determine the location of required services on the middle tier. It also enables the applications to obtain a list of services available in the environment. Note that this sas-environment.xml file is used by SAS Web Server, and the configuration in the file refers to the host name and port number of SAS Web Server.

If Transport Layer Security (TLS) is configured at your site, specify the https protocol and the TLS port number for the service registry.

If your site has multilingual users, you can configure the sas-environment.xml file to include localized descriptions. In the next example, the Blue environment is specified in German:

```xml
<environment name="Blue">
  <desc>test2 Blue</desc>
  <desc xml:lang="de">Blau</desc>
  <service-registry>http://blue.example.com:80/SASWIPClientAccess/remote/ServiceRegistry</service-registry>
  <service-registry interface-type="soap">http://blue.example.com:80/SASWIPSoapServices/services/ServiceRegistry</service-registry>
  <service-registry interface-type="rest">http://blue.example.com:80/SASWIPClientAccess/rest</service-registry>
</environment>
```

When the customized sas-environment.xml file is available for multiple environments, see to the documentation for your SAS application or solution for instructions about how to enable the availability of these environments for the users. If you change the location of the sas-environment.xml file, be aware that SAS desktop applications such as SAS Enterprise Miner need to be updated.
with the new location. The SAS desktop applications that integrate with the middle tier use the -Denv.definition.location JVM option in INI files to identify the location of the sas-environment.xml file. Refer the documentation for the SAS desktop applications that you use. The SASHome/sassw.config file is also used to identify the location of the sas-environments.xml file. Update the SASENVIRONMENTSURL= value in the sassw.config file.

**Element Description**

The following list identifies and describes the elements that can be used in the sas-environment.xml file:

- **environment**
  - has a name attribute that cannot contain space characters. This attribute is used internally by SAS software to identify each of the environments that are available in the deployment. This element has an attribute that is named default. This attribute is used to identify a default environment for client applications. If this attribute is set to true for more than one environment element, then the last environment in the file with the attribute set to true is set as the default environment. It is not necessary to set the attribute to false for all other environments.

- **desc**
  - used in the client applications to provide a menu of environment choices. As shown in the previous example, this field can provide a localized message when the xml:lang attribute is set.

- **service-registry**
  - contains the URL to the service registry for the environment. Use the protocol, host name, and port number of SAS Web Server. By default, SAS Web Server is configured to provide access to SAS Web Infrastructure Platform.
Overview of Multicasting

Note: By default, multicasting is not used in the typical SAS deployment, and SAS Remote Services is turned off. If you created a custom application that uses SAS Remote Services, you can use multicasting and enable SAS Remote Services.

Multicast communication is no longer used to communicate among SAS 9.4 middle-tier applications in a single SAS deployment (the set of applications connected to the same SAS Metadata Server). However, you can still take advantage of this communication if you have developed custom applications by starting Remote Services. When installation is performed with the SAS Deployment Wizard, the wizard generates a default multicast address that is based on IP address of the SAS Metadata Server. The combination of multicast address and multicast UDP port number must be different for each SAS deployment and also different from any other multicast applications at your site.

The multicast communication includes all the information that is needed to bootstrap a custom middle-tier application. Because this information includes the SAS environment credentials (such as the sasadm account name and its password), time to live (TTL) and encryption options are provided to secure the multicast communication.
Multicast options are specified as JVM options. Multicast options provide the ability to tune and change the behavior of the multicast communication that occurs within the SAS deployment. The multicast address and UDP port number must match the values in the start-up script for SAS Web Application Server and the environment.properties file located in the `SAS-configuration-directory\Lev1\Web\Applications\RemoteServices` directory.

Administering multicast options typically involves the following:

- setting options such as the multicast address
- configuring security with a multicast authentication token
- configuring the bind address that is used for multicast communication

---

**How Much Multicast Network Traffic is Generated?**

The amount of multicast network traffic that is generated by SAS applications is fairly small. The greatest amount of traffic is generated during application start-up. When SAS Remote Services starts, the largest packet that it generates is 124 bytes. Once start-up is complete, the typical rate is less than 64 Kb per hour.

When the web application server starts, the largest packet is 256 bytes. Once start-up is complete, the typical rate for an entire SAS Enterprise Business Intelligence Server deployment (including SAS Remote Services) is less than 128 Kb per hour.

Once the applications are generating multicast traffic, the amount of traffic is steady regardless of the load on the SAS Web applications.

---

**Multicast Security**

A multicast group communications protocol is used to communicate among middle-tier SAS applications in a single SAS deployment (the set of applications connected to the same SAS Metadata Server). During installation, the SAS Deployment Wizard supplies you with a default multicast address and port number that it generates based on the machine's (metadata server) IP address. The combination of multicast IP address and multicast UDP port should be different for each SAS deployment and also different from those used by other multicast applications at your site.

The IP address and multicast UDP port number for the multicast host must match the values in the start-up script for SAS Web Application Server and the environment.properties file.

The multicast group communication includes all information needed to bootstrap SAS middle-tier applications. Because this includes sending the SAS environment credentials (such as the sasadm account name and its password), scoping and encryption options are provided in the SAS Deployment Wizard. The defaults are most appropriate for deployments in the firewall, isolated data center environment. After installation, if you choose to modify the scoping or...
encryption options, you can do so by specifying the options for the 
-Dmulticast.security parameter for the web application server.

For more information, see Appendix 2, “Administering Multicast Options,” on page 307.

Configuring Multicast Options

Applications That Use Multicast Communication

Multicast options should be changed in a synchronous manner among the following applications:

- SAS Remote Services
- SAS Web Application Server

Multicast Options Configuration Files for SAS Remote Services

You can make changes to the multicast options for the JVM that is used by SAS Remote Services. Edit the appropriate files as needed.

On Windows, in directory SAS-configuration-directory\Lev1\Web\Applications\RemoteServices, change the following files:

- RemoteServices.bat
- wrapper.conf
- environment.properties

On UNIX, edit the RemoteServices.sh and environment.properties files.

Multicast Options Configuration Files for SAS Web Application Server

You can make changes to multicast options for SAS Web Application Server. The options are specified as JVM options. For more information, see “Specifying JVM Options” on page 38.

Key Multicast Properties

The following table shows some key multicast properties.
### Table A2.1 Multicast Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Default Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>multicast.address</td>
<td>239.X.Y.Z</td>
<td>Not applicable</td>
<td>This value is provided by the SAS Deployment Wizard prompting mechanism and defaults to 239.X.Y.Z. Values for X, Y, and Z are the last three octets of the metadata server's IP address. In an IPv6 environment, the value defaults to ff14::/16.</td>
</tr>
<tr>
<td>multicast.port</td>
<td>8561</td>
<td>Not applicable</td>
<td>This value is provided by the SAS Deployment Wizard prompting mechanism and represents the port on which UDP communication occurs.</td>
</tr>
<tr>
<td>multicast_udp_ip_ttl</td>
<td>1</td>
<td>Decimal</td>
<td>Specifies how far a multicast packet should be forwarded from a sending host. 0 is restricted to the same host. 1 is restricted to the same subnet. 32 is restricted to the same site. 64 is restricted to the same region. 128 is restricted to the same continent. 255 is unrestricted. The IP multicast routing protocol uses the Time to Live (TTL) field of IP datagrams to decide how far a multicast packet should be forwarded from a sending host. The default TTL for multicast datagrams is 1, which results in multicast packets going only to other hosts in the local network. If all SAS applications participating in the multicast (this includes Remote Services, any Java applications in the middle tier, and BI Report Services) are on the same machine, the value should be 0. If your site has a SAS middle-tier application that resides on a different subnet but uses the same metadata server within the same SAS deployment, increase the value for this property.</td>
</tr>
<tr>
<td>multicast.security</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>By default (with no value), both encryption and authentication are enabled. Valid values are:  ■ ENCRYPT: encrypt but do not require authentication  ■ NONE: do not encrypt and do not require authentication</td>
</tr>
<tr>
<td>multicast.config.file</td>
<td>Not applicable</td>
<td>URL string (file://, http://, and so on)</td>
<td>By default, a JGroups configuration is provided. However, you can provide your own configuration by specifying the URL path to that configuration. This option enables you to specify a port range or change from IP multicast to the gossip router capabilities of JGroups.</td>
</tr>
</tbody>
</table>
Configuring a Multicast Authentication Token

Understanding the Multicast Authentication Token

By default, the multicast communication is protected with encryption because it conveys credentials. This default setting for encryption uses a fixed encryption key that is built into the software and is common to all SAS middle-tier software. This strategy prevents access to the multicast communication from unauthorized listeners. This setting might be sufficient for deployments where multicast communication is isolated from the user community with a firewall, a TTL option, or the deployment is in an isolated data center.

If your middle tier meets any of the following criteria, then you might want to set a multicast authentication token value:

- you have custom applications
- the middle-tier environment is not well isolated from end-user access
- the security procedures at your site require protection among administrative and operational staff in various roles
- you want more protection against eavesdroppers and unauthorized participants

For these deployments, set a multicast authentication token value that is known only to the appropriate personnel. A multicast authentication token is a password-like string that is needed to connect to the multicast group and create a site-specific encryption key. In a multi-tier configuration, the SAS Deployment Wizard displays a prompt for a multicast authentication token on each tier that has an application participating in multicast communication. The same authentication token value must be specified for each tier in the same SAS deployment (each tier associated with the same metadata server).

The multicast authentication token has an interaction with the multicast.security property. By default, clients that want to join a multicast group to receive messages are required to provide an authentication token for the join request. (This is true whether a custom token value is used or if the default token value that is built into the software is used.) If you determine this process is causing an impact on performance, or that it is unnecessary, you can disable the use of authentication tokens. If you set the multicast.security property to NONE, encryption and authentication are disabled. If you set the property to ENCRYPT, then encryption is enabled with no authentication of the join request.
Reconfiguring to Use a Multicast Authentication Token

Generate a Token and Set the Token for SAS Remote Services

1. Use SAS and the PWENCODE procedure to generate an encoded password to use as the multicast authentication token. Here is an example:
   `{SAS002}DA9A0A5C20629B7F34D2C88A165E5530`.

2. Edit the `SAS-configuration-directory\Lev1\Web\Applications \RemoteServices\RemoteServices.bat` file to add a `-DMULTICAST_AUTHENTICATION_TOKEN` JVM option.

   For Windows, add the option in the `runasScripts` section:
   ```
   :runasScripts
   set MULTICAST_AUTHENTICATION_TOKEN=token
   ```

   For UNIX, add the option to the `RemoteServices.sh` file after the `SERVERUSER` variable:
   ```
   SERVERUSER=sas
   MULTICAST_AUTHENTICATION_TOKEN="token"
   export MULTICAST_AUTHENTICATION_TOKEN
   ```

3. For Windows, also add the JVM option to the `wrapper.conf` file. Add it to the end of the `wrapper.java.additional.11` entry:
   ```
   wrapper.java.additional.11=-XX:+UseTLAB -XX:+UseConcMarkSweepGC
   -XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
   -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -Xss256k
   -XX:NewSize=16m -XX:MaxNewSize=16m -XX:PermSize=64m -XX:MaxPermSize=64m
   -DMULTICAST_AUTHENTICATION_TOKEN=token
   ```

   Note: Do not use carriage returns or line feed characters when editing long lines.


Setting the Token for the Report Output Generation Tool

1. Edit the `SAS-installation-directory\SASBIReportServices \4.4\outputgen.ini` file.

2. Add a `JavaArgs_nn` entry that is similar to the following:
   ```
   JavaArgs_13=-Dias.app.launch.picklist=picklist;"help\primary.picklist"
   JavaArgs_14=-DMULTICAST_AUTHENTICATION_TOKEN=token
   Classpath=-cp "<VJRHOME>/eclipse/plugins/sas.launcher.jar"
   ```
Configuring the JGroups Bind Address

Understanding the JGroups Bind Address

Some SAS middle-tier applications use JGroups to perform multicast communication between applications and to perform caching of application properties. The JGroups software binds to the IP address of first non-loopback network interface that it can detect on the machine. Many machines have multiple network interfaces (multihomed), and each network interface has its own IP address. In some cases, the web application server selects the value of InetAddress.getLocalHost().getHostName() as the bind address to use for multicast communication and SAS Remote Services selects a different IP address to bind to.

Multicast communication does not function correctly if the IP address selected by JGroups for SAS Remote Services does not match the IP address selected by the web application server. One indication of a mismatch is an error message that appears in the web application server log file. See the following example:

```
org.springframework.beans.factory.BeanDefinitionStoreException: Invalid bean definition with name 'dashboardServices' defined in ServletContext resource ['/WEB-INF/spring-config/services-config.xml']: Could not resolve placeholder 'metadata.user'
```

```
ERROR [main] - ***************************************************************
ERROR [main] - Required entry, '/sas/properties/environment', not found in the cache.
ERROR [main] - Possible causes include: the RemoteServices VM is not started or
ERROR [main] - there is a multicast address/port mismatch; using
ERROR [main] - address=239.168.68.1 and port=8561.
ERROR [main] - ***************************************************************
```

Set the bind address for SAS Remote Services, the web application server, and the SAS BI Report Services Report Generation tool if the previous error message is seen.

Setting the Bind Address for SAS Remote Services

1. For deployments on Windows, edit the SAS-configuration-directory\Lev1\Web\Applications\RemoteServices\wrapper.conf file. Add a wrapper.java.additional.nn entry that is similar to the following:

   `wrapper.java.additional.12=-Dlog4j.configuration="..."
   wrapper.java.additional.13=-Djgroups.bind_addr=ip-address`

2. Edit the SAS-configuration-directory\Lev1\Web\Applications\RemoteServices\RemoteService.bat file. Add the JVM option in the start2 section:

   `:start2
   start "SAS Remote Services" "$JAVA_JRE_COMMAND%"`
Setting the Bind Address for SAS Web Application Server

Specify the following JVM option for the server:

-Djgroups.bind_addr=ip-address

The option is used when the server is restarted.

3 Restart SAS Remote Services.
Recommended Reading

- *SAS Intelligence Platform: Overview*
- *SAS Intelligence Platform: System Administration Guide*
- *SAS Intelligence Platform: Security Administration Guide*
- *SAS Management Console: Guide to Users and Permissions*
- *SAS Integration Technologies: Overview*

SAS offers instructor-led training and self-paced e-learning courses to help you administer the SAS Intelligence Platform. For more information about the courses available, see support.sas.com/admintraining.

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books  
SAS Campus Drive  
Cary, NC 27513-2414  
Phone: 1-800-727-0025  
Fax: 1-919-677-4444  
Email: sasbook@sas.com  
Web address: sas.com/store/books
Glossary

alert
an automatic notification of an electronic event that is of interest to the recipient.

authentication
See client authentication.

authentication domain
a SAS internal category that pairs logins with the servers for which they are valid. For example, an Oracle server and the SAS copies of Oracle credentials might all be classified as belonging to an OracleAuth authentication domain.

authentication provider
a software component that is used for identifying and authenticating users. For example, an LDAP server or the host operating system can provide authentication.

base path
the location, relative to a WebDAV server’s URL, in which packages are published and files are stored.

blacklist
a list or register of entities, such as email addresses or software applications, that are denied a particular privilege, service, mobility, access or recognition. See also whitelist.

client authentication (authentication)
the process of verifying the identity of a person or process for security purposes.

client-side pooling
a configuration in which the client application maintains a collection of reusable workspace server processes. See also puddle.

content mapping
the correspondence of the SAS metadata folder structure to a content repository system. SAS metadata folders are generally mapped to a WebDAV such as the SAS Content Server repository, or to a local file system.

credentials
evidence that is submitted to support a claim of identity (for example, a user ID and password) or privilege (for example, a passphrase or encryption key).
deploy
to install an instance of operational SAS software and related components. The deployment process often includes configuration and testing as well.

foundation repository
the metadata repository that is used to specify metadata for global resources that can be shared by other repositories. For example, a foundation repository is used to store metadata that defines users and groups on the metadata server.

foundation services
See SAS Foundation Services.

hot deployment
the process of upgrading an application or component in a client-server environment while the server is running. Hot-deployed components are made available immediately, and do not require the server to be restarted.

identity
See metadata identity.

Java Development Kit (JDK)
a software development environment that is available from Oracle Corporation. The JDK includes a Java Runtime Environment (JRE), a compiler, a debugger, and other tools for developing Java applets and applications.

Java RMI
See remote method invocation.

Java Virtual Machine (JVM)
a software application that can execute Java bytecode, on either a client or a server, enabling Java programs to be run on many different hardware and software platforms.

JDK
See Java Development Kit.

JVM
See Java Virtual Machine.

metadata identity (identity)
a metadata object that represents an individual user or a group of users in a SAS metadata environment. Each individual and group that accesses secured resources on a SAS Metadata Server should have a unique metadata identity within that server.

middle tier
in a SAS business intelligence system, the architectural layer in which web applications and related services execute. The middle tier receives user requests, applies business logic and business rules, interacts with processing servers and data servers, and returns information to users.
pool
a group of server connections that can be shared and reused by multiple
client applications. A client-side pool consists of one or more puddles. See also puddle.

portal
a web application that enables users to access websites, data, documents,
applications, and other digital content from a single, easily accessible user
interface. A portal's personalization features enable each user to configure
and organize the interface to meet individual or role-based needs. See also portlet.

portlet
a web component that is managed by a web application and that is
aggregated with other portlets to form a page within the application. Portlets
can process requests from the user and generate dynamic content.

puddle
a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers. See also client-side pooling.

remote method invocation (RMI, Java RMI)
a Java programming feature that provides for remote communication
between programs by enabling an object that is running in one Java Virtual
Machine (JVM) to invoke methods on an object that is running in another
JVM, possibly on a different host. See also Java Virtual Machine.

remote service deployment
a service deployment that supports shared access to a set of SAS
Foundation Services that are deployed within a single Java Virtual Machine
(JVM), but which are available to other JVM processes. Applications use the
remote service deployment to deploy and access remote foundation
services. See also service deployment.

repository
a storage location for data, metadata, or programs.

RMI
See remote method invocation.

SAS Application Server
a logical entity that represents the SAS server tier, which in turn comprises
servers that execute code for particular tasks and metadata objects.

SAS batch server
a SAS Application Server that is running in batch mode. In the SAS Open
Metadata Architecture, the metadata for a SAS batch server specifies the
network address of a SAS Workspace Server, as well as a SAS start
command that will run jobs in batch mode on the SAS Workspace Server.

SAS BI Web Service
a web service that adheres to the XML for Analysis (XMLA) specification for
executing SAS Stored Processes.
SAS Content Server
a server that stores digital content (such as documents, reports, and images) that is created and used by SAS client applications. To interact with the server, clients use WebDAV-based protocols for access, versioning, collaboration, security, and searching.

SAS Foundation Services (foundation services)
a set of core infrastructure services that programmers can use in developing distributed applications that are integrated with the SAS platform. These services provide basic underlying functions that are common to many applications. These functions include making client connections to SAS application servers, dynamic service discovery, user authentication, profile management, session context management, metadata and content repository access, information publishing, and stored process execution. See also service.

SAS Management Console
a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Repository
a container for metadata that is managed by the SAS Metadata Server.

SAS Web Infrastructure Platform
a collection of middle-tier services and applications that provide infrastructure and integration features that are shared by SAS web applications and other HTTP clients.

SAS Workspace Server
a SAS server that provides access to SAS Foundation features such as the SAS programming language and SAS libraries.

server-side pooling
a configuration in which a SAS object spawner maintains a collection of reusable workspace server processes that are available for clients. The usage of servers in this pool is governed by the authorization rules that are set on the servers in the SAS metadata.

service
one or more application components that an authorized user or application can call at any time to provide results that conform to a published specification. For example, network services transmit data or provide conversion of data in a network, database services provide for the storage and retrieval of data in a database, and web services interact with each other on the World Wide Web. See also SAS Foundation Services.

service configuration
a set of values that can be customized for a particular service in SAS Foundation Services. By editing a service configuration, you can override the default configuration for the foundation service. See also SAS Foundation Services.

service deployment
a collection of SAS Foundation Services that specifies the data that is necessary in order to instantiate the services, as well as dependencies upon other services. Applications query a metadata source (a SAS Metadata
Server or an XML file) to obtain the service deployment configuration in order to deploy and access foundation services. See also SAS Foundation Services.

**session context**
- a context that serves as a control structure for maintaining state within a bound session. 'State' includes information about the latest status, condition, or content of a process or transaction. Session Services and User Services use the session context to facilitate resource management and to pass information among services.

**single sign-on (SSO)**
- an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. For example, single sign-on can enable a user to access SAS servers that run on different platforms without interactively providing the user's ID and password for each platform. Single sign-on can also enable someone who is using one application to launch other applications based on the authentication that was performed when the user initially logged on.

**SSO**
- See single sign-on.

**theme**
- a collection of specifications (for example, colors, fonts, and font styles) and graphics that control the appearance of an application.

**trust**
- to accept the authentication or verification that has been performed by another software component. See also trust relationship, trusted user.

**trust relationship**
- a logical association through which one component of an application accepts verification that has already been performed by another component. See also trusted user.

**trusted user**
- a privileged service account that can act on behalf of other users on a connection to the metadata server.

**unrestricted identity**
- a user or group that has all capabilities and permissions in the metadata environment due to membership in the META: Unrestricted Users Role (or listing in the adminUsers.txt file with a preceding asterisk).

**user context**
- a set of information about the user who is associated with an active session. The user context contains information such as the user's identity and profile.

**Web Distributed Authoring and Versioning (WebDAV)**
- a set of extensions to the HTTP protocol that enables users to collaboratively edit and manage files on remote web servers.

**WebDAV**
- See Web Distributed Authoring and Versioning.
WebDAV repository
a collection of files that are stored on a web server so that authorized users can access them. See also Web Distributed Authoring and Versioning, SAS Content Server.

whitelist
a list or register of entities, such as email addresses or software applications, that are accepted for a particular privilege, service, mobility, access or recognition. See also blacklist.
Index

A

alert notification
  SMS 62
alerts
  default delivery type 60
anonymous access 253
anonymous web user 144
audit 76
audit profiles 70
auditing 67, 69, 70
  audit profiles 70
  for web applications 67
  internal accounts 70
  relational tables for 68
  authenticated users 75
authentication 144, 210
  See also Web authentication
  SAS Anonymous Web User 254
  SAS authentication for Java 145
token for multicast security 311
authentication requests 101
authorization
  for SAS Content Server 133

B

backups
  SAS Content Server 120
bind address 313
branding 149

C

cascading style sheets (CSS) 148
  migrating 160
channels
  deleting packages 277
clear text 144
client access
  enabling for JMX 282
clustering 191
  for web application servers 190
colors
changing in themes 155
comment management
  predefined role 55
concurrent sign-in sessions 115
correction
  auditing for web applications 67
  Chrome for SPNEGO 237
  cluster of web application servers 190
custom sign-out message 102
data sources for middle tier 21
Firefox for SPNEGO 237
HTTP sessions 195
Internet Explorer for SPNEGO 236
Job Execution Service 22
multicast options 309
properties for SAS Web Report
  Studio 57
reconfiguring Web application server 93
removing configuration content 85
sample middle-tier deployment
  scenarios 187
SAS environment file 303
scripting tools 287
shared between middle and server tiers 15
SharedServices DSN 21
SMTP mail server for middle tier 16
Web application server, to enable JMX client access 282
web services 140
Configuration Manager 56
deleting web services 140
example 57
properties for SAS Web Report
  Studio 57
summary of steps for 57
connecting to library server
  Android 179
  iPad 178
connection properties 65
  internal and external 64
content
  See also SAS Content Server
loading manually 123
moving and sharing 120
updating manually 124
creating a PostgreSQL database 20
custom sign-out message 102
custom themes
See themes
custom web applications
See Web applications

D
data sources 21
configuring for middle tier 21
configuring SharedServices DSN 21
data store 131
Garbage Collection utility 132
database persistence 120
DAVTree utility 273
adding resources to WebDAV 274
advanced features 276
connecting to a WebDAV location 274
copying or moving files in WebDAV 276
editing text files in WebDAV 276
starting 274
debugging
Package Clean-Up utility 280
Web application logging levels 94
Default theme 148
deleting a PostgreSQL database 20
demilitarized zone (DMZ) 192
deployment
manually deploying content to SAS Content Server 121
redeploying web applications 91
sample middle-tier scenarios 187
SAS Deployment Manager 85
themes 149, 150
themes, in test environment 158
Web applications 38
directives 84
adjusting URLs manually 124
DMZ (demilitarized zone) 192
documentation 13

E
e-mail
configuring SMTP server 16
EAR files
names 88
e-mail
sending to users 75
environment

F
files
adding to SAS Content Server 130
deleting 130
permissions for WebDAV files 127
firewalls 192
folders
creating 129
deleting 130
permissions for WebDAV folders 127
forcing users to log off 75
Forward Proxy Authentication
Flex Commons 269

G
Garbage Collection utility 132
generated web services 140, 144
global properties
setting for SAS applications 59
global single sign-on time-out interval 110
graphics
changing in themes 156
guest access 111

H
HTTP sessions
affinity 192
auditing 69
configuring 195
time-out interval 105
HTTP transport layer security 144

I
images 148
changing in themes 156
migrating 161
internal accounts
auditing 70
internal connection 64
IOM Spawners 284

Java
configuring web services for 140
SAS authentication for 145
web authentication for 145
Java Mail Session 16
Java Management Extensions
See JMX (Java Management Extensions)
Java Runtime Environment (JRE) 6
JConsole
managing SAS resources 282
JDBC 21
JGroups 313
JMX (Java Management Extensions) 281
enabling client access 282
JConsole 282
managing SAS resources 281
MBeans 281, 283
JSR 168 13
JVM options 300
default values 195
Forward Proxy Authentication 269
SAS Content Server 118
SAS Workflow 11

K
kiosk
See guest access

L
loading content manually 123
locked settings 59
log files
changing location of 95, 99
location 14
logging 94
changing logging levels 94
for SAS Web Infrastructure Platform
Data Server 19
for Web applications 93
Package Clean-Up utility 280
SAS Information Retrieval Studio 166
schema validation errors 180
service settings for Web applications 93
logging off
forcing users to log off 75

M
MBeans 281, 283
accessing 282
Server MBean 285
ServerFactory MBean 283
Spawner MBean 284
metadata
deleting themes from 160
middle tier
configuration shared with server tier 15
configuring data sources for 21
configuring SMTP mail server for 16
log files 14
sample deployment scenarios 187
SAS Web Infrastructure Platform
Data Server with 17
middle-tier environment 3
SAS Content Server 11
SAS Web Infrastructure Platform 6
SAS Workflow 11
starting web applications 13
Web applications 11
migrating themes 160
cascading style sheets (CSS) 160
images 161
theme descriptors 161
theme templates 161
mobile devices
See also SAS Mobile BI
connecting to library server 178, 179
monitoring users 75
moving content 120
multicast options 308
configuring 309
multicast properties 309
multicast security 307, 308
authentication token for 311

N
naming themes 157
online documentation
See documentation

Package Clean-Up utility 276
arguments 279
changing prompt behavior 278
deleting packages 277
deleting specific packages 278
examples 280
listing packages 278
logging and debugging 280
syntax for deleting packages 277
packages
deleting 277
deleting specific packages 278
listing 278
passwords 289, 293
performance
clustering web application servers 191
network topology 187
SAS Workflow 11
permissions
WebDAV folders and files 127
persistence, database 120
pgAdmin III 19
preferences 54
product-specific branding 149
production environment moving themes to 158
prompts
Package Clean-Up utility 278
properties
global properties for SAS applications 59
SAS Application Infrastructure 59
SAS Web Report Studio 57
proxy configurations
configuring HTTP sessions in environments with 195
rebuilding themes 158
rebuilding Web applications 86
exploded directories 87
rebuilding one or more 87
when to rebuild 86
redeploying web applications 91
SAS Web Application Server 91
relational tables for auditing 68
reports
See SAS Web Report Studio
resources
adding to WebDAV repository 274
managing SAS resources with JConsole 282
managing SAS resources with JMX tools 281
roles
Comments:Administrator 55
Job Execution Services 26

SAS Anonymous Web User 253
SAS authentication 254
SAS Application Infrastructure properties 59
SAS applications
global properties for 59
SAS authentication for Java 145
SAS Anonymous Web User 254
SAS BI Dashboard 12
SAS BI Portlets 13
SAS BI Web Services for Java 7
SAS Comment Manager
Comments:Administrator role 55
predefined role 55
SAS Content Server 7, 11, 118, 120
adding files to 130
Administration Console 125
authorization for 133
backing up 120
data store 131
database for storage 120
deploying content manually 121
loading content manually 123
moving and sharing content 120
preventing file types 120
updating content manually 124
SAS Content Server Administration Console 125
accessing 126
adding files to SAS Content Server 130
creating folders 129
deleting folders or files 130
interface 126
permissions for WebDAV folders and files 127
SAS Default theme 148
SAS Deployment Manager 85
accessing 86
auditing for web applications 67
custom sign-in, sign-out, and time-out messages 102
HTTP session time-out interval 105
rebuilding Web applications 86
removing configuration content 85
update passwords 289, 293
SAS environment file 303
configuring sas-environment.xml 303
SAS Foundation Services 9
SAS Information Delivery Portal 12
SAS Information Retrieval Studio 165
SAS Intelligence Platform 3
SAS Logon Manager 7
auditing 69
concurrent sign-in sessions 115
internal and external connections 65
SAS Mail Service 16
SAS Management Console
assigning default theme from Configuration Manager 159
SAS Mobile BI 173
advanced properties 180
blacklisting 174
customizing 181
enabling the whitelist 175
image resizing 182, 184
logging 180
security 173
troubleshooting 183
whitelisting 174
SAS Preferences Manager 7, 54
SAS Remote Services Application
multicast options 309
SAS resources
managing with JConsole 282
managing with JMX tools 281
SAS servers 283, 285
SAS Shared Web Assets 7
SAS Stored Process Web application 7
SAS Visual Analytics Administrator
Mobile Devices tab 175
SAS Visual Analytics Transport Service
image resizing 184
SAS Web Administration
viewing audit reports 76
SAS Web Administration Console 7, 73
accessing 74
forcing users to log off 75
monitoring users 75
sending email to users 75
update Job Execution Service 78
users appearing in 75
viewing information about web applications 83
SAS Web Application Server
about 35
automatic configuration 36
checking prerequisite servers 40
installing 36
manual configuration 36
monitoring 40
reconfiguring 288
redeploying web applications 91
SAS Web Application Themes
See themes
SAS web applications
sign out URL 108
SAS Web Infrastructure Platform 6, 54
Configuration Manager 56
default alert notification delivery type 60
global properties for SAS applications 59
SAS Comment Manager 55
SAS Preferences Manager 54
SAS Web Administration Console 73
SAS Web Infrastructure Platform Data Server 17
creating a database 20
deleting a database 20
logging for 19
password policy 19
pgAdmin III 19
SAS Web Infrastructure Platform Services 7, 10
SAS Web Report Studio 12
configuring properties 57
sas-environment.xml, configuring 303
search facility 165
Search Interface to SAS Content 165
security
HTTP transport layer 144
logon audit 69
mobile 173
multicast 307, 308
SAS Anonymous Web User 253
SAS Comment Manager 55
TLS 254
transport layer 146
web services 144
WS-Security message-level 144
Server MBean 285
server tier
  configuration shared with middle tier 15
ServerFactory MBean 283
servers
  See SAS servers
session affinity 192
session time-out interval 105
SharedServices database 21
SharedServices DSN 21
  configuring 21
sign-out message
  configuring custom message 102
sign-out URL 108
SMS alert notification 62
SMTP mail server
  configuring for middle tier 16
sources
  See data sources
Spawner MBean 284
static content
  caching 188
system users 75

T

test environment
  deploying themes in 158
testing themes 158
text files
  editing in WebDAV 276
theme descriptors 148
  migrating 161
theme templates 148
  changing 157
  migrating 161
themes 147
  assigning as default theme 159
cascading style sheets (CSS) 148
  changing colors 155
  changing graphics 156
  changing theme templates 157
components 148
creating and deploying 149
creating work area for 151
Default theme 148
defining and deploying 150
deleting from metadata 160
deploying in test environment 158
designing 150
images and 148
migrating 160
migrating cascading style sheets (CSS) 160
migrating images 161
migrating theme descriptors 161
migrating theme templates 161
moving to production environment 158
  naming 157
  rebuilding 158
testing 158
time-out interval 105, 110
TLS (Transport Layer Security)
  for web applications 254
transport layer security 146
  web services 144
transport services
  See SAS Mobile BI
tuning Web application servers 195

U

UpdateDefaultTheme.sas program 159
updating content manually 124
URLs
  adjusting directive URLs manually 124
users
  appearing in SAS Web
    Administration Console 75
  authenticated 75
  forcing users to log off 75
  monitoring with SAS Web
    Administration Console 75
  sending email to 75
  system users 75

W

warning message
  inactive user sessions 103
web application servers 5
  configuring a cluster of 190
  enabling JMX client access 282
tuning 195
Web applications deployed in single server 188
Web application servers
  bind address and JGroups 313
  multicast options 309
  reconfiguring 93, 287
web application themes
  See themes
web applications
  auditing for 67
  configuring custom sign-out messages 102
  custom sign-in, sign-out, and time-out messages 102
  deployed across web application server cluster 190
  deployed in single web application server 188
directives 84
disable concurrent sign-in sessions 115
HTTP session time-out interval 105
redeploying 91
SAS BI Dashboard 12
SAS Documentation for the Web 13
SAS Information Delivery Portal 12
SAS Web Report Studio 12
settings 84
themes 147
TLS 254
viewing information about 83
Web applications 11
  changing location of log files 95, 99
  changing logging levels 94
deploying 38
EAR file names 88
inactive user sessions 103
logging for 93
rebuilding 86
SAS Deployment Manager and 85
SAS Web Administration Console 73
starting 13
warning message 103
web authentication 144, 210
  for Java 145
  RESTful web services 145
  transport layer security 146
Web authentication
  See authentication
Web Service Maker 140, 144
web services
  CA SiteMinder 145
  configuring 140
deleting 140
generated 140, 144
security for 144
third-party authentication 145
XMLA 144
webanon account 253
WebDAV
  See also DAVTree utility
  adding resources to repository 274
  content management with DAVTree utility 273
copying or moving files 276
deleting packages 277
editing text files 276
permissions for folders and files 127
WebDAV/Dump utility 120
WebDAV/Restore utility 120
work area
  creating for themes 151
WS-Security message-level security 144

X

XMLA web services 144