SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide, Fourth Edition

SAS® Documentation
October 9, 2024

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide, Fourth Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide, Fourth Edition

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

October 2024

9.4-P6:bimtag

Contents

What's New in Middle-Tier Administration for the SAS 9.4 Intelligence Platform

Overview

SAS is currently shipping SAS 9.4M8.

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 9.4M8 includes the following changes and enhancements:

- The SAS Middle-Tier is configured with OpenSSL 3.0 and the Java TLS implementation that supports TLS 1.3 and FIPS.
- SAS Web Infrastructure Platform Data Server is based on PostgresSQL 14 and is configured with a FIPS 140 compliant cryptographic module. See "Configure SAS Web Infrastructure Platform Data Server in SAS 9.4M8" on page 362.
- VMware GemFire is replaced with OpenSource Geode. Because of this, the configuration directory references *geode* in its directory path and configuration files have changed their names to include *geode*.
- The SAS Web Server supports only TLS v1.3 and TLS v1.2.
- SAS Logon Manager uses Central Authentication Service (CAS) version 6.6 on page 118 to enable single sign-on. This newer version of CAS includes enhanced security that affects "Fallback Authentication" on page 292.

The February 15, 2022 release of SAS 9.4M7 includes the following changes and enhancements:

- SAS Web Application Server is based on Pivotal tc Server prior to the February 15, 2022 release, and is based on Apache Tomcat thereafter. See SAS 9.4 Support for Web Application Servers and HTTP Servers.

The May 2019 release of SAS 9.4M6 includes the following changes and enhancements:

- "Ability to Make New Fonts Available"
- "Using SAS Theme Designer"

SAS 9.4M6 includes the following changes and enhancements:

- "SAS Private JRE"
- "SAS Visual Analytics Apps"
- "SAS Web Infrastructure Platform Data Server"
- "Supported TLS Version for SAS Web Server "

SAS 9.4M5 includes the following changes and enhancements:

- "Enhancements to Configuring SAS Environment Manager for HTTPS"
- "Upgrade to the Apache Tomcat Version"

Starting with SAS 9.4M4, many security enhancements have been made. The following support has been added:

- "Support to Preserve Your TLS Customizations"
- "Documentation Enhancements"
- "Supported Version of OpenSSL"
- "Upgrades to Middle-Tier Components and Applications"

SAS 9.4M3 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"

Starting with the May 2015 release of SAS 9.4M2, guest access support is available through SAS Logon Manager. See "Support for Guest Access".

The October 2014 release of SAS 9.4M2 includes updates to SAS Information Retrieval Studio. See "Updates to SAS Information Retrieval Studio for TLS".

SAS 9.4M2 includes the following changes and enhancements:

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

SAS 9.4M1 includes 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 initial SAS 9.4 middle-tier software includes the following changes and enhancements:

- "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"

Ability to Make New Fonts Available

New font services enable you to manage the list of available fonts. As an administrator, you can add custom fonts to be used in reports and exploration. For more information, see "Make More Fonts Available" on page 184.

Using SAS Theme Designer

All SAS web applications are now displayed with HTML5. You can customize the appearance of the SAS web applications using SAS Theme Designer. For information about custom themes for SAS web applications, see *SAS Theme Designer: User's Guide*.

SAS Private JRE

The SAS Private JRE is now based on Java 11. SAS 9.4M5 and previous releases require a SAS Private JRE that is based on Java 7.

SAS Visual Analytics Apps

SAS Mobile BI was renamed SAS Visual Analytics Apps.

SAS Web Infrastructure Platform Data Server

SAS Web Infrastructure Platform Data Server is now based on PostgreSQL 14.

Supported TLS Version for SAS Web Server

SAS Web Server supports TLS v1.3 starting with SAS 9.4M8 and TLS v1.2. Both TLS v1.0 and TLS v1.1 are no longer supported by the SAS Web Server configuration.

Enhancements to Configuring SAS Environment Manager for HTTPS

SAS Environment Manager requires less manual configuration for HTTPS. For more information, see "Configure SAS Environment Manager for HTTPS Starting with SAS 9.4M5" on page 330.

Upgrade to the Apache Tomcat Version

The version of SAS Web Application Server used in the SAS 9.4M5 middle-tier environment has been upgraded. It is now based on Apache Tomcat version 8.5.13.

Starting with the February 15, 2022 release of SAS 9.4M7, the SAS Web Application Server is based on Apache Tomcat 9.0.55.

Support to Preserve Your TLS Customizations

Starting with SAS 9.4M4, you do not have to revert manual changes made to the following servers before upgrading or applying maintenance:

- SAS Web Server
- SAS Web Application Server
- SAS Environment Manager

For more information, see "Preserve TLS and Existing Customer Reverse Proxy Customizations" on page 342.

Support to Preserve Your Existing Reverse Proxy Customizations

Starting with SAS 9.4M4, manual updates made for the existing reverse proxy are preserved.

For more information, see "Preserve TLS and Existing Customer Reverse Proxy Customizations" on page 342.

Documentation Enhancements

In SAS 9.4M4, information has been moved from the *SAS 9.4 Intelligence Platform: Installation and Configuration Guide, Second Edition* to this document. The new sections are:

- "Configure SAS Environment Manager for HTTPS" on page 325
- Appendix 3, "Validate the Secured Middle-Tier Environment," on page 425

Supported Version of OpenSSL

In SAS 9.4M4, the OpenSSL libraries provided by SAS have been updated. For SAS 9.4 and all maintenance releases of SAS 9.4, updated versions of OpenSSL for UNIX and z/OS are provided and updated through hot fixes. See the SAS Security Bulletin on OpenSSL for the most current information about the versions of OpenSSL used in SAS products and about the advisories under consideration.

For a quick reference of the OpenSSL version supported for each version of SAS Foundation, see Mapping Between SAS Version and OpenSSL Version.

Note: On Windows deployments, SAS uses the code delivered with Windows for TLS. On UNIX and z/OS deployments, SAS redistributes a copy of OpenSSL.

Upgrades to Middle-Tier Components and Applications

The following upgrades have been made:

- Java upgrade on all platforms to enable the enforcement of using TLSv1.2 and TLSv1.3
- JMS Broker upgrade to version 5.12.2 on all platforms to fix Java deserialization and several other security issues
- SAS Web Server upgrade to version 5.5.2, which includes a newer version of OpenSSL and an updated mod_proxy_connect module that supports SAS IOM SSL communication

Support for Enabling Auditing of Internal Accounts

Beginning in SAS 9.4M3, 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 85.

Support for Forcing Users to Log Off

Starting with SAS 9.4M3, 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 91.

Support for Enabling Audit Profiles

Starting withSAS 9.4M3, you can enable Spring audit profiles by updating the **-Dspring.profiles.active** JVM option.

For more information, see "Enable Audit Profiles" on page 86.

Support for Management of the Trusted CA Bundle

Starting with SAS 9.4M3 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 "Configure Middle-Tier Services for SAS 9.4M3" on page 295.

Security Support for SAS Web Applications

Beginning in SAS 9.4M3 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 an allowlist, or security filter, of allowed sites. You can also allowlist certain HTTP request methods.

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

For additional information, see:

- "Allowlist of Websites and Methods Allowed to Link to SAS Web Applications" on page 364
- "Configure the Cross Domain Proxy Servlet Through an Allowlist" on page 369
- "Enable Support for Forward Proxy Authentication" on page 370

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

Starting with SAS 9.4M3, 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.

Note: The improvement in start-up time varies 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.

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.

For more information, see "Configure Guest Access" on page 129.

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*\Levn\Web\Applications\SearchInterfacetoSASContent\url_list.txt** file.

For more information, see "Configure for TLS" on page 188.

Enhancements for SAS Content Server

Beginning in SAS 9.4M2, SAS Content Server enables you to prevent certain file extensions and MIME types from being uploaded. This is done 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 with SAS 9.4M2, 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, "Administer the SAS Content Server," on page 139.

Enhancements for Managing Devices

Starting with SAS 9.4M2, 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 14, "Manage Devices," on page 193.

Support for Customizing Web Application Content

Beginning in SAS 9.4M1 is the ability to add custom content to a SAS web application.

Enhancements for Managing Devices

Starting with SAS 9.4M1 is added functionality for managing mobile devices that use SAS Mobile BI. Devices are managed either by inclusion or exclusion.

For more information, see Chapter 14, "Manage Devices," on page 193.

Support for TLS with Client Certificate Authentication

Beginning in SAS 9.4M1, 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 294.

Enhancements to SAS Logon Manager

In SAS 9.4M1, 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, "Administer SAS Logon Manager," on page 117.

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 Server, see Chapter 3, "Administer SAS Web Server," on page 35.

For more information about SAS Web Application Server, see Chapter 4, "Administer SAS Web Application Server," on page 41.

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 primary node in the cluster. If the primary 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 and Product Changes*.

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. 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 22.

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.

For more information, see Chapter 9, "Administer SAS Logon Manager," on page 117.

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.

For more information, see Chapter 10, "Administer the SAS Content Server," on page 139.

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.

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

Work in the Middle-Tier Environment

Understand the Middle-Tier Environment

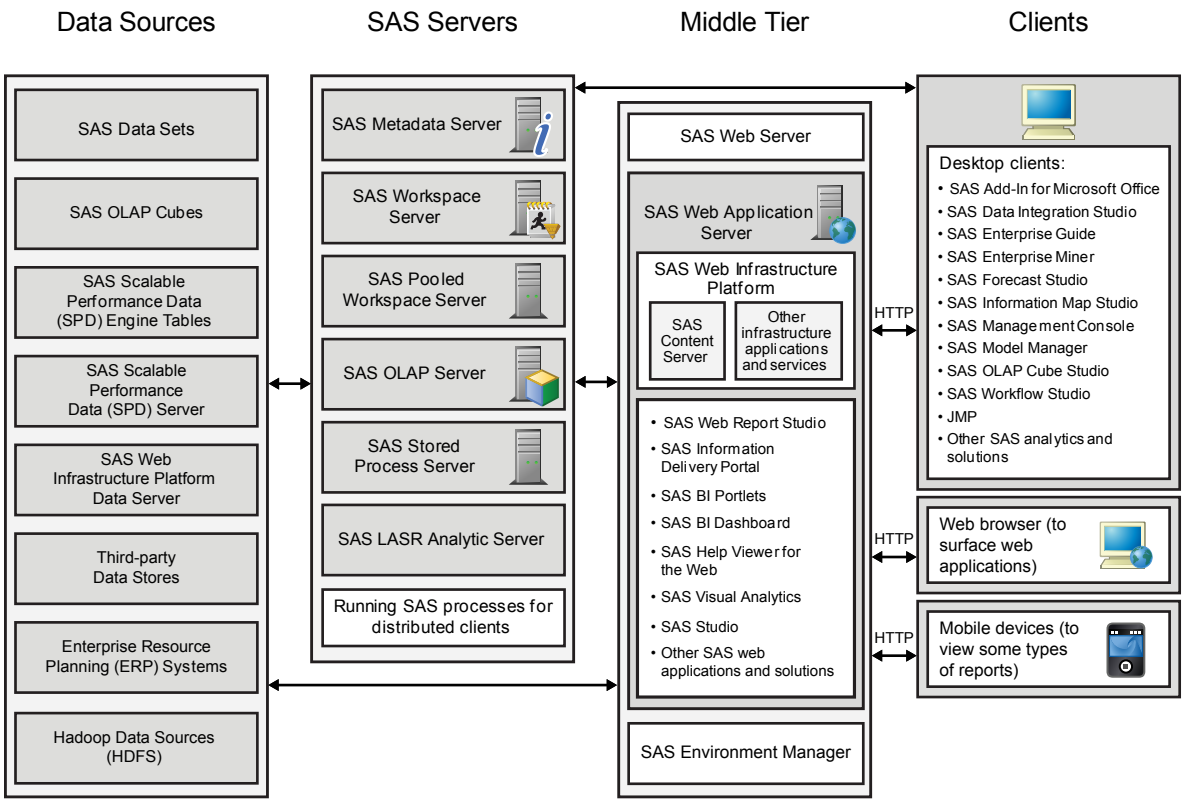
The middle tier of the SAS Intelligence Platform enables users to access intelligence data and functionality with a web browser. This tier provides web-based interfaces for report creation and information distribution, while passing analysis and processing requests to the SAS servers.

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.

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. SAS Web Application Server can also be manually configured to support HTTPS.

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.

SAS delivers regular updates for the Java environment. See http://support.sas.com/security/alerts.html.

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. 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 1.1 *Services and Applications in the SAS Web Infrastructure Platform*

Application or Service	Features
SAS Authorization Service	This service is used by some SAS web applications that manage authorization through web services.
SAS BI Web Services for Java	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.

Application or Service	Features
SAS Content Server	Stores digital content (such as documents, reports, and images) that can be created and used by the SAS web applications.
SAS Deployment Backup and Recovery Tool	Enables deployment-wide backup and recovery services. For more information, see *SAS Intelligence Platform: System Administration Guide*.
SAS Identity Services	Provides SAS web applications with access to user identity information.
SAS Logon Manager	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.
SAS Preferences Manager	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.
SAS Principal Services	Enables access to core platform web services for SAS applications.
SAS Shared Web Assets	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.

Application or Service	Features
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.
SAS Notification Template Editor	Enables administrators to create and edit messages that are sent as notifications to end users of SAS applications.
SAS Web Administration Console	Provides features for monitoring and administering middle-tier components. This browser-based interface enables administrators to perform the following tasks: • Monitor users who are logged on to SAS web applications, and send email to them. • 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. • Manage permissions for folders and documents that are managed by SAS Content Services. • Manage templates and letterheads that are used as part of messages that are sent as notifications to end users of SAS applications. • View configuration information for each middle-tier component.
SAS Web Infrastructure Platform Permission Manager	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.

Application or Service	Features
SAS Web Infrastructure Platform Services	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.
SAS Workflow	Provides the web services that implement workflow management. The SAS Workflow services are used by SAS applications and solutions for tightly integrated workflow management.

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 SAS. This suite of Java application programming interfaces provides core middleware infrastructure services. These services include the following:

- client connections to SAS 9.4 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 Studio

SAS Studio is a development application for SAS that you access through your web browser. With SAS Studio, you can access your data files, libraries, and existing programs, and you can write new programs. You can also use the predefined tasks in SAS Studio to generate SAS code.

For more information, see A Guide to the SAS Studio Documentation and Programming Documentation for SAS and SAS Viya.

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.

Start 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 Logs and Log Locations

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

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

Application or Server	Log Location
Cache Locator	***SAS-configuration-directory*\Levn\Web\gemfire\instances\ins_*port-number*\gemfire.log** file Starting in SAS 9.4M8 ***SAS-configuration-directory*\Levn\Web\geode\instances\ins_*port-number*\gemfire.log**
JMS Broker	***SAS-configuration-directory*\Levn\Web\activemq\data\activemq.log** file
SAS Environment Manager Agent	***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\agent-*version*-EE\log** directory ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\agent-*version*-EE\log\agent.log**

Application or Server	Log Location
SAS Environment Manager Server	***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\logs** directory ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\logs\server.log** ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\logs\bootstrap.log**
SAS Web Application Server	***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\logs** directory ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\logs\server.log** ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\logs\catalina.out** (for UNIX systems) ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\logs\gemfire.log** ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\logs\localhost_access_log.*date*.txt** ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\logs\wrapper.log** (for Windows systems)
SAS web applications	***SAS-configuration-directory*\Levn\Web\Logs\SASServer*n_m*** directory
SAS Web Infrastructure Platform Data Server	***SAS-configuration-directory*\Levn\WebInfrastructurePlatformDataServer\Logs** directory *Note:* In a multi-machine deployment, the default log location is on the server tier.
SAS Web Server	***SAS-configuration-directory*\Levn\Web\WebServer\logs** directory ***SAS-configuration-directory*\Levn\Web\WebServer\logs\access.log** ***SAS-configuration-directory*\Levn\Web\WebServer\logs\error.log** ***SAS-configuration-directory*\Levn\Web\WebServer\logs\ssl_request.log** (if using SSL)

For additional information about SAS Web Application Server logging, see "Administer Logging for SAS Web Application Server" on page 47.

For additional information about specific web application logs, see *SAS Intelligence Platform: Web Application Administration Guide*.

For information about SAS server logging, see "The SAS Logging Facility" in *SAS Logging: Configuration and Programming Reference*.

Chapter 2

Interact with the Server Tier

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

Overview

The SAS Web Infrastructure Platform includes a SAS Mail Service. The mail service 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.

Mail Properties

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.

Modify Java Mail Session Settings

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 ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer1_1\conf\server.xml** file. If you have more than one server instance, edit the server.xml file for each server. Change the following line:

```
<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:

Table 2.1 *SMTP Authentication Properties*

Property Name	Property Value
mail.smtp.auth	true
mail.smtp.user	*username*
password	*password* *Note:* The password can be encoded. For more information, see "Encode the SMTP Authentication Password" on page 19.

Encode the SMTP Authentication Password

You can encrypt the *password* that was specified above instead of using plaintext.

SAS 9.4M2 and Previous Releases

To get an encrypted 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.

SAS 9.4M3 and SAS 9.4M4

1. Set TCHOME=***SASHOME*\SASWebApplicationServer\9.4** on Windows.

   On UNIX:

   ```
   TCHOME=SASHOME/SASWebApplicationServer/9.4; export TCHOME
   ```

2. Run the following command on Windows:

   ```
   java -cp %TCHOME%\lib\com.springsource.org.bouncycastle.jce-1.46.0.jar;
        %TCHOME%\tomcat-7.0.55.A.RELEASE\lib\tcServer.jar;%TCHOME%\
        tomcat-7.0.55.A.RELEASE\bin\tomcat-juli.jar;%TCHOME%\
        tomcat-7.0.55.A.RELEASE\lib\tomcat-coyote.jar
   ```

```
-Dcom.springsource.tcserver.security.PropertyDecoder.
decoder_prefix=s2enc:// com.springsource.tcserver.security.PropertyDecoder
-encode "app-server-passphrase" password
```

On UNIX:

```
java -cp $TCHOME/lib/com.springsource.org.bouncycastle.jce-1.46.0.jar:
     $TCHOME/tomcat-7.0.55.A.RELEASE/lib/tcServer.jar:$TCHOME/
     tomcat-7.0.55.A.RELEASE/bin/tomcat-juli.jar:$TCHOME/
     tomcat-7.0.55.A.RELEASE/lib/tomcat-coyote.jar
     -Dcom.springsource.tcserver.security.PropertyDecoder.
     decoder_prefix=s2enc:// com.springsource.tcserver.security.PropertyDecoder
     -encode 'app-server-passphrase' password
```

Note: The previous commands must be on one line. They are shown on more than one line for display purposes only.

SAS 9.4M5 and SAS 9.4M6

1. Set TCHOME=***SASHOME*\SASWebApplicationServer\9.4** on Windows.

   On UNIX:

   ```
   TCHOME=SASHOME/SASWebApplicationServer/9.4; export TCHOME
   ```

2. Run the following command on Windows:

   ```
   java -cp %TCHOME%\lib\tcServer3.jar:
   %TCHOME%\bin\tomcat-juli.jar:%TCHOME%\lib\tomcat-util.jar
   -Dcom.springsource.tcserver.security.PropertyDecoder.decoder_prefix=s2enc://
   -Dcatalina.home=%TCHOME% com.springsource.tcserver.security.PropertyDecoder
   -encode "app-server-passphrase" password
   ```

   On UNIX:

   ```
   java -cp $TCHOME/lib/tcServer3.jar:
   $TCHOME/bin/tomcat-juli.jar:$TCHOME/lib/tomcat-util.jar
   -Dcom.springsource.tcserver.security.PropertyDecoder.decoder_prefix=s2enc://
   -Dcatalina.home=$TCHOME com.springsource.tcserver.security.PropertyDecoder
   -encode 'app-server-passphrase' password
   ```

   Note: The previous commands must be on one line. They are shown on more than one line for display purposes only.

3. Encode the *password*, instead of using plaintext, by running one of the following commands:

   On Windows:

   ```
   "SASHOME\SASWebApplicationServer\9.4\tcruntime-admin.bat" encode
   value-to-encrypt passphrase
   ```

   On UNIX:

   ```
   SASHOME/SASWebApplicationServer/9.4/tcruntime-admin.sh encode
   value-to-encrypt passphrase
   ```

4. Add the encoded output as a variable in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\catalina.properties** file.

Starting with the SAS 9.4M7 February 15, 2022 Release

Note: When you update-in-place to SAS 9.4M7 (after February 15, 2022) these secret passwords are not preserved and the following steps must be performed again.

With the move to Apache Tomcat in the February 15, 2022 release of SAS 9.4M7, encrypting and encoding passwords is handled differently. A shell script, *decoder.bat* on Windows, and *decoder.sh* on UNIX, is provided to encode and decode the password. To run the shell script, you must first set two environment variables: *JAVA_HOME*, and *CATALINA_HOME*:

```
export JAVA_HOME=SAS-home-directory/SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre

export CATALINA_HOME=SAS-home-directory/SASWebApplicationServer/9.4/apache-tomcat-9.0.65
```

Note: Any Java Runtime would work.

Once the above environment variables are set, navigate to the location of the *decoder* script:

```
cd SAS-config-directory/Levn/Web/WebAppServer/SASServern_m/bin
```

The decoder script encrypts a given value using a passphrase provided by the user. That passphrase can be provided in one of two ways. It is either provided directly on the command line by using the *-pw* option, for example: *-pw passphrase*. Or, it is provided indirectly by specifying a file from which to read the passphrase by using the *-pwf* option, for example: *-pwf passphrase file*. The passphrase can be found in the ***SAS-config-directory*\Levn\Web\ WebAppServer\SASServern_m\conf \secure.file** file.

You can use the *--help* option to see valid arguments for the script.

Below is an example of the command that uses a file from which to read the passphrase:

- On Windows:

  ```
  .\decoder.bat -encode -config sasmtr01 -pwf SAS-config-directory\Levn\Web\
  WebAppServer\SASServern_m\conf\secure.file -v value-to-encrypt
  ```

- On UNIX:

  ```
  ./decoder.sh -encode -config sasmtr01 -pwf SAS-config-directory/Levn/Web/
  WebAppServer/SASServern_m/conf/secure.file -v value-to-encrypt
  ```

Note: The command is displayed over multiple lines, but it should be entered on a single line.

The scheme is enabled by two values in the catalina.properties file. The first is unchanged from prior versions, but the second is different and needs to be updated after an update to SAS 9.4M7.

The command is displayed over multiple lines, but should be entered on a single line.

```
com.sas.vfabrictcsvr.decoder.PropertyDecoder.passphrase.file=${catalina.base}
/conf/secure.file
org.apache.tomcat.util.digester.
PROPERTY_SOURCE=com.sas.vfabrictcsvr.decoder.PropertyDecoder,
org.apache.tomcat.util.digester.EnvironmentPropertySource
```

Configure Security

You can configure Transport Layer Security (TLS) 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

Overview

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 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.

data
: 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.

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

  The _webinfdsvrc_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.

Shared Services
: 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 denylist and allowlist 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 Shared Services database and the Administration database. SAS Environment Manager is configured for access to the EVManager 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 characters, mixed case characters, and most special characters.
3. The password cannot contain the following:
   - single quotation mark (')
   - double quotation mark (")
   - dollar sign ($)

- exclamation point (!)

4. The password cannot include any trailing spaces. Leading spaces are allowed and blank spaces among the characters are allowed.
5. The **updatePasswords** task for the SAS Web Infrastructure Platform Data Server does not support SAS003, SAS004, or SAS005 passwords. If an encrypted password is to be used for this task, only SAS001 and SAS002 encrypted passwords can be used.

Administer 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/.

   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.
3. Restart SAS Web Infrastructure Platform Data Server.

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 Tool

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

You can use the version of the pgAdmin tool that is recommended to work with the version of PostgreSQL that is deployed with your SAS 9.4 software. Starting with SAS 9.4M8, SAS Web Infrastructure Platform Data Server is based on PostgreSQL 14. (Previous SAS 9.4 maintenance releases included PostgreSQL versions 9.1, 9.4, 9.5, and 12.) You can download the PgAdmin tool from https://www.pgadmin.org/download/.

Create 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 has all of the privileges that are granted to the database.

To create a database, run the following command:

```
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). This role 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:

```
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**.

Delete 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:

```
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**.

Back Up or Restore 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 information, see "Using the Deployment Backup and Recovery Tool" in *SAS Intelligence Platform: System Administration Guide*.

JDBC Data Sources

Overview

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

If you have not already done so, make sure that you review "Configuring an Alternate Database for SAS Web Infrastructure Platform Data Server " 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 2.2 *JDBC Connection Parameters for SAS Web Infrastructure Platform Data Server*

Connection Parameter	Setting
JNDI name:	sas/jdbc/SharedServices
JDBC URL:	**`jdbc:postgresql://`***serverName:port*/**`SharedServices`** 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.
JDBC driver class:	org.postgresql.Driver

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*\Levn\Web\WebAppServer\SASServer1_1\conf\server.xml** file:

```
<Resource auth="Container" driverClassName="org.postgresql.Driver"
    factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" initialSize="10"
    jdbcInterceptors="org.apache.tomcat.jdbc.pool.interceptor.ConnectionState;
    org.apache.tomcat.jdbc.pool.interceptor.StatementFinalizer"
    jmxEnabled="true" maxActive="100" name="sas/jdbc/SharedServices"
    password="${pw.sas.jdbc.SharedServices}" testOnBorrow="true"
    type="javax.sql.DataSource"
    url="jdbc:postgresql://hostname.example.com:9432/SharedServices"
    username="SharedServices" validationInterval="30000"
    validationQuery="select 1"/>
```

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

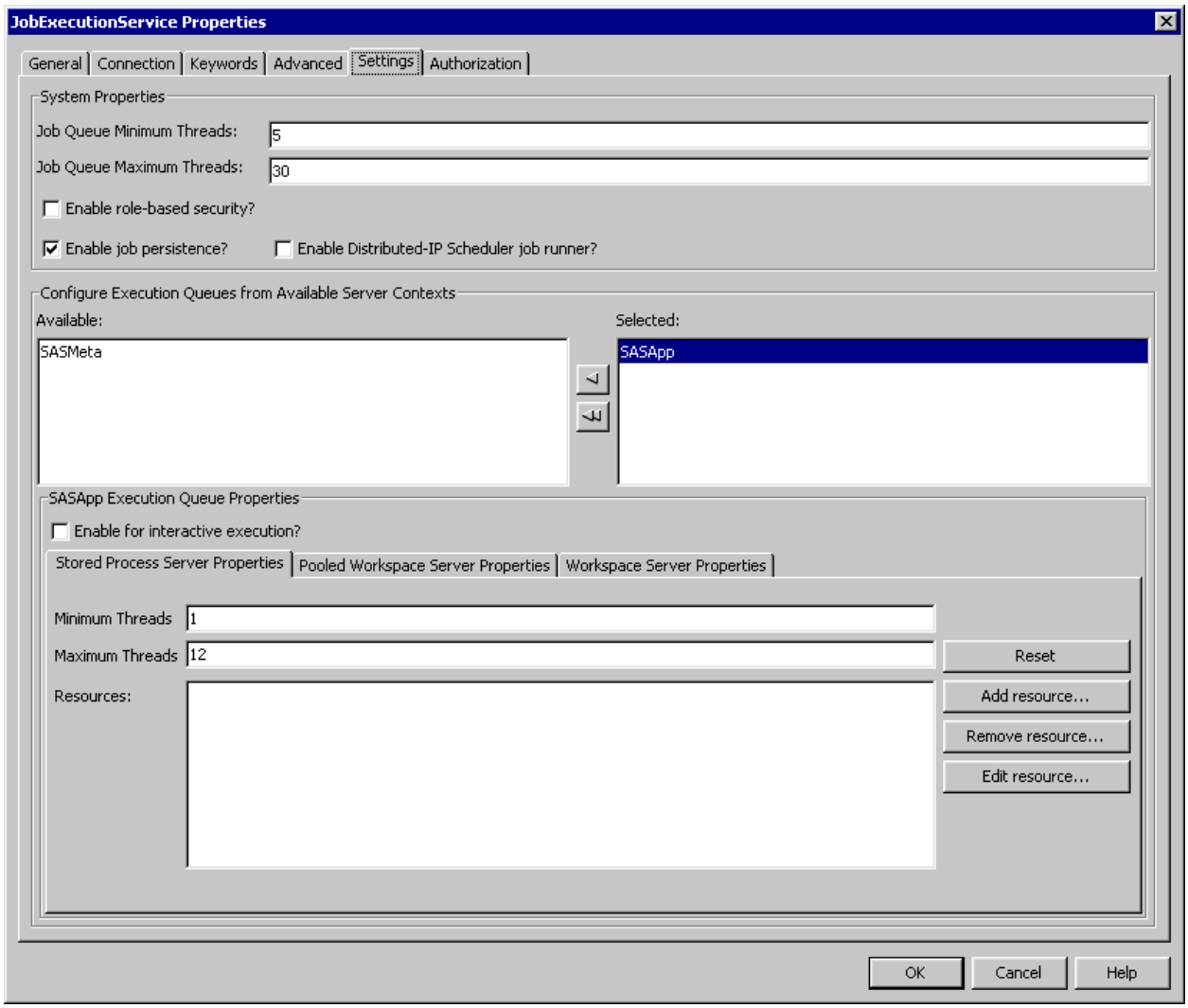


Table 2.3 Job Execution Service Settings Descriptions

Setting	Default Value	Description
Job Queue Minimum Threads	5	Minimum number of job queue threads to create for incoming job requests.
Job Queue Maximum Threads	30	Maximum number of job queue threads to create if the demand requires additional resources.
Enable role-based security	Disabled	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 30.

Setting	Default Value	Description
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 "Configure Auditing for SAS Web Applications" on page 83.

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 "Update the Job Execution Service Configuration" on page 93.

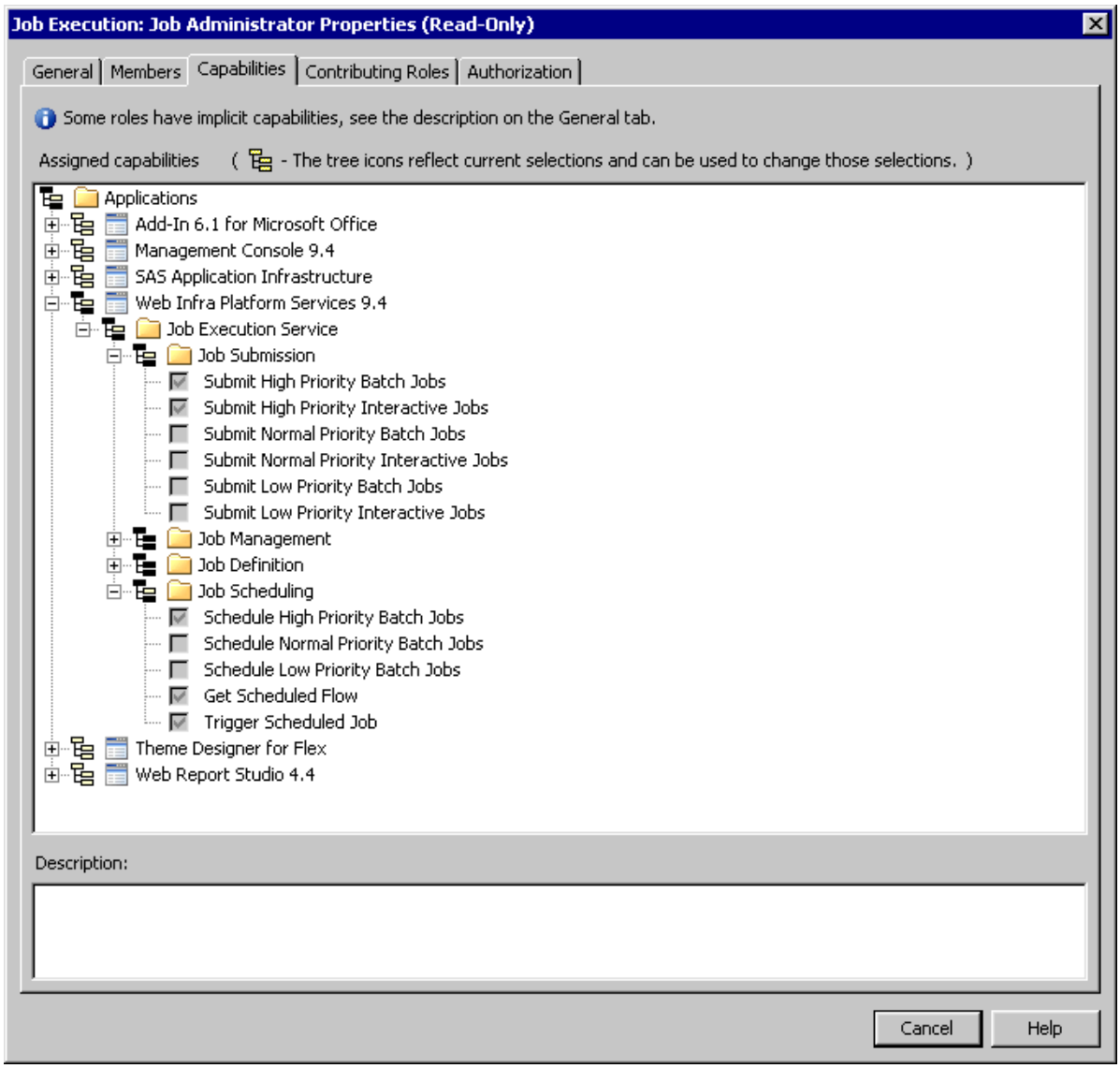
The default configuration for the job execution services does not check role-based permissions. If role-based security is enabled, then the job execution service checks that the identity submitting the request has sufficient permission.

Table 2.4 *Job Execution Service Roles*

Role	Capabilities
Job Execution: Job Administrator	Can submit jobs of high, normal, and low priority and perform all job-related operations.
Job Execution: Job Designer	Can add, update, or remove jobs and tasks from metadata.
Job Execution: Job Scheduler	Can schedule jobs.
Job Execution: Job Submitter	Can submit normal priority jobs for execution.

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

Administer SAS Web Server

Overview

SAS Web Server is an HTTP 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

Install 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.

Use 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 1 *RSA Private Key without a Pass phrase*

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQC4vPQMyiVKvjIERVNfa34iVxeauzcUa8zc2xBHRlJ43uAvvWuL
63yeGl8QQoT55yqhAWhs62i24lE34t2ituhCm0QYbU1KiyB9PNyfOk3/2E7Y7o1T
```

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

Example Code 2 *Encrypted RSA Private Key*

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,FB353F5E4F1719EB

LigQnszN4joO24QonLHCEl7d4LlLa6uMEqdxhl1PX8O4o+pbY5cEQJBbCiRlEmfg
Io5V/YZUa+uGG82ULsAUy3zWTHP+OjxpTV/3gjLwbmD3+JM5Dd0jFLGenfPF5hld
```

SAS Deployment Wizard also prompts for the certificate. A certificate file from a certificate authority typically begins as follows:

Example Code 3 *Certificate Authority-Signed Certificate*

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

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

Understanding the SAS Web Server Configuration

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

bin
This directory includes a command for starting and stopping the server. For more information, see "Use 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

Use 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**.

```
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`**.

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

Use the appsrvconfig Command

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

```
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`*`\Levn\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 397.

Use 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
```

Use 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.

Monitor SAS Web Server

View 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 Apache httpd Server *version*.

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

See Also

SAS Environment Manager: User's Guide

View 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

Chapter 4

Administer SAS Web Application Server

Overview

SAS Web Application Server is a lightweight server that provides enterprise-class features for running SAS web applications. 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. 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.

Install 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 "Add a Horizontal Cluster Member" on page 220.

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.

For a list of the default SAS Web Application Server assignments, see "SAS Web Application Server Assignments" in *SAS Intelligence Platform: Web Application Administration Guide*.

See Also

"Add a Vertical Cluster Member" on page 218

Server Directories

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

***SAS-configuration-directory*\Levn\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.

sas_webapps
This directory is used for the SAS web applications. SAS software manages the addition and removal of web applications from the directory.

Specify 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*\Levn\Web\WebAppServer\SASServern_m\conf\wrapper.conf** file and the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\bin\setenv.bat** file. If you have multiple instances of SAS Web Application Server, make the same changes in each of the files.

Note: After you modify the wrapper.conf file for SAS 9.4M7 February 16, 2022 and later, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

For UNIX deployments, JVM options are specified in the ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServern_m/bin/setenv.sh** file. If you have multiple server instances, make the changes in each setenv.sh file. If you add a new JVM option:

1. Name the option something other than JVM_OPTS (for example, TS_JVM_OPTS).
2. Add the new option to the following line, making sure that the new option comes after $JVM_OPTS:

```
JAVA_OPTS="$JVM_OPTS $TS_JVM_OPTS $AGENT_PATHS $JAVA_AGENTS $JAVA_LIBRARY_PATH"
```

Note: After specifying any needed JVM options, you must restart the SAS Web Application Server for changes to take effect.

Deploy 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 107.

Understanding SAS Web Application Server Management

Use the tcruntime-ctl Command

Each server instance provides a **tcruntime-ctl** command in the **`bin`** directory.

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:

- For UNIX environments:

  ```
  tcruntime-ctl.sh start|stop|restart|status
  ```

- For Windows environments:

  ```
  tcruntime-ctl.bat start|stop|restart|status
  ```

Starting with SAS 9.4M7 February 15, 2022 release, the tcruntime-ctl.bat and catalina.bat shell scripts cannot be used to start or stop a Windows service. You manage the web application server instances through the Windows Services Manager or the SAS provided **SASServiceCtl.ps1** Windows PowerShell script. The SASServiceCtl.ps1 script for each SAS Web Application Server instance is located here: *\SAS-configuration-directory\Levn\Web\WebAppServer\SASServerN_M\bin.* To execute the SAS Service Control script and operate on Windows services, follow the powershell.exe command with the *-file* switch and the name of the script, the *-action* switch followed by start or stop and the *-name* switch followed by the full name of the service to be operated on. For example:

```
> powershell -file SASServiceCtl.ps1 -action start
-name SAS-SASConfig-Lev1-SASServer1_1-WebAppServer
```

Note: You do not need to use the script to start or stop the web application server instance as you can achieve the same result using the Windows Services Manager. The script primarily exists to support uninstalling and installing the Service; start and stop are just ancillary capabilities of the script.

Note: Starting with SAS 9.4M7 February 15, 2002 release, the *restart* option is no longer available from the UNIX script. Use the *stop* option and subsequent *start* option instead.

Note: On Windows, the **status** option does not indicate whether the server is running or stopped.

Note: Prior to February 15, 2022 release of SAS 9.4M7, SAS Web Application Server instances are based on Pivotal tc Server. Thereafter, SAS Web Application Server instances are based on Apache Tomcat. In releases of SAS 9.4M7 where SAS Web Application Server instances are managed by Apache Tomcat, the following error might occur when stopping an instance:

```
ERROR [Catalina] No shutdown port configured. Shut down server through OS signal. Server
The stop command failed. Attempting to signal the process to stop through OS signal. Tom
PID file was not removed. To aid diagnostics a thread dump has been written to standard
```

Contrary to what this error message states, the SAS Web Application Server instance has been stopped. Also, no shutdown port is configured by design, so this is not a problem. To eliminate or reduce the chance of receiving this error message, extend

the time-out value for each instance of the SAS Web Application Server. To extend the time-out value, use the catalina.sh script rather than the appsrvconfig.sh script to stop each instance. See "Troubleshooting the sas.servers Script" in *SAS Intelligence Platform: System Administration Guide* for information about eliminating or reducing the chance of receiving this error message.

Use 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*\Levn\Web\Scripts\AppServer** directory. The **appsrvconfig.sh** command in UNIX environments and **appsrvconfig.cmd** command in Windows environments can be used for starting, stopping, and restarting all SAS Web Application Server instances on the machine. The command will also start, stop, and restart 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*\Levn\Web\Scripts\AppServer\props**. The file is generated and then executed. The file does not exist until the **appsrvconfig** command is used.

Information about using the **appsrvconfig** command for configuration tasks is provided in SAS Configuration Scripting Tools on page 397.

Use 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.

Note: Starting with SAS 9.4M7 February 15, 2022 release, the tcruntime-ctl.bat and catalina.bat shell scripts cannot be used to stop or start a Windows service.

Use 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

Administer Logging for SAS Web Application Server

Logging Configuration File

The SAS Web Application Server uses log4j to perform logging. The log4j configuration file defines the level of logging and it is read at server start-up.

- Starting with SAS 9.4 M7 February 15, 2022 release and SAS 9.4 M8, SAS web applications use Log4j v2 for its logging framework. Because of this, the log4j configuration filename is *log4j2.xml* and it is found here:

  `SAS-config-directory/Levn/Web/WebAppServer/SASServern_m/Log4j2/conf`

- Prior to the SAS 9.4 M7 February 15, 2022 release, SAS web applications use Log4j v1 for its logging framework. The *log4j.xml* file is found here:

  `SAS-config-directory/Levn/Web/WebAppServer/SASServern_m/lib`

Customizations can be performed by editing the *log4j2.xml* configuration file, such as:

- Change the logging levels
- Add a logging category
- Change the layout pattern for the log message

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.

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 than 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.

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

1. Navigate to the SAS Web Application Server's log4j file:

   ```
   cd SAS-configuration-directory/Levn/Web/WebAppServer/SASServern_m/Log4j2/conf
   ```

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

   ```
   <Logger name="com.atomikos" level="error" additivity="false">
            <AppenderRef ref="FILE"/>
        <AppenderRef ref="CONSOLE"/>
      </Logger>
          <Logger name="org.apache.catalina" level="info" additivity="false">
            <AppenderRef ref="FILE"/>
        <AppenderRef ref="CONSOLE"/>
      </Logger>
   ```

3. Restart the SAS Web Application Server so that it uses the new configuration.

Note:

- Prior to the SAS 9.4M7 February 15, 2022 release, SAS Web Application Server uses Log4j v1 for its logging framework. For information about the log4j v1 configuration file, see http://logging.apache.org/log4j/1.2/index.html and http://logging.apache.org/log4j/1.2/manual.html.
- Starting with SAS 9.4M7 February 15, 2022 release and SAS 9.4 M8, SAS Web Application Server uses Log4j v2 for its logging framework. For information about the log4j v2 configuration file, see http://logging.apache.org/log4j/2.x/manual/index.html.

SAS server logging is also based off of log4j, and syntax is similar. For more information, see “Administering Logging for SAS Servers” in *SAS Intelligence Platform: System Administration Guide*.

Monitor 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.

In SAS Environment Manager, SAS Web Application Server is represented as SASWebApplicationServerTomcat 9.47.

See Also

SAS Environment Manager: User's Guide

Check Prerequisite Servers

Overview

Beginning with SAS 9.4M2, a LifeCycle Listener is provided with SAS Web Application Server. The LifeCycle Listener 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.

Enable the Prerequisite Checker

To enable the LifeCycle Listener feature, edit the ***SAS-configuration-directory*\Levn\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:

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

For information about configuring the LifeCycle Listener for clustered servers, see "Configure the Prerequisite Checker for Clustered Servers " on page 228.

Prerequisite for Update in Place

Before performing an update in place on a server where updates were previously performed, additional steps must be completed. For example, if you are updating your system from SAS 9.4M3 to SAS 9.4M4, and you previously performed an update in place from SAS 9.4M2 to SAS 9.4M3, additional steps must be performed prior to the update in place.

Complete the following steps:

1. Check if the ***SAS-configuration-directory*/Levn/Web/WebAppServer/Backup** directory exists.
2. If the directory exists, create a backup of it by renaming it to something else, such as Backup.old.
3. Start the update in place process.

Chapter 5

Administer Cache Locator

Overview

The Cache Locator 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 are 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 standalone client applications like the SAS Web Infrastructure Platform Scheduling Services (wipschedbatch.bat).

Note: In SAS 9.4M8, VMware GemFire is replaced with OpenSource Geode. Because of this change, the configuration directory references *geode* in its directory path, and configuration files have changed their names to include *geode*. See "Configure JVM Options for the Cache Locator" on page 52 for directory and file names.

Install 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.

Configure JVM Options for the Cache Locator

Add or modify JVM options in the appropriate file, which is based on your SAS 9.4 version of the software and operating system. This information is in the tables below. For the specific JVM options that are required for a task, see the following:

- "Member Timeout JVM Option" on page 54
- "Set the Bind Address" on page 56
- "Modify the Configuration to Accommodate a Firewall" on page 56

Note: On a multi-machine deployment, two Cache Locators are installed. A Cache Locator is installed on the first middle-tier machine and on the server-tier machine that includes SAS Web Infrastructure Platform Scheduling Services. You must add or modify JVM options for the Cache Locator in both places.

Table 5.1 *SAS 9.4M8 Uses OpenSource Geode for Cache Locator*

Directory: ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_*port-number***	
UNIX deployment scripts:	
geode-locator.sh	Use this script with one of the following arguments: **start**, **stop**, or **status**.
geode-start-locator-sas.sh	Use this file to specify JVM options.
Windows deployment scripts:	
SAS [Config-Lev1] Cache Locator on port 41415 Windows Service	Use Windows service to operate the Cache Locator on Windows deployments.

Directory: ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_*port-number***	
geode-service.bat	Use this file to specify JVM options. *Note:* You must re-install the Geode service after you make changes to this file. See the note below this table for steps.
z/OS deployment scripts:	
geode-start-locator-sas-zos.sh	Use this script with one of the following arguments: **start**, **stop**, or **status**.
geode-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.

Note: For Windows environments in SAS 9.4 M8, you must perform additional steps.

1. Re-install the Geode service by calling the **geode-service.bat**. Open a Windows command prompt and "Run as Administrator".
2. Run the following commands after you edit the values for SASCONF, SASLEV, and SASPORT that are correct for your environment:

```
REM Edit these to fit your situation
set SASCONF=D:\SAS\Config
set SASLEV=Lev1
set SASPORT=41415
REM No editing needed below this line
cd /d "%SASCONF%%SASLEV%\Web\geode\instances\ins_%SASPORT%\"
net stop "SAS [Config-%SASLEV%] Cache Locator on port %SASPORT%"
cmd /d /c .\geode-service.bat uninstall "SAS [Config-%SASLEV%] Cache Locator on port
cmd /d /c .\geode-service.bat install "SAS [Config-%SASLEV%] Cache Locator on port %S
net start "SAS [Config-%SASLEV%] Cache Locator on port %SASPORT%"
```

Note: The log file for the Cache Locator is **gemfire.log** and it is located here: ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_*port-number***. Be aware that this log file is different from the log file with the same name that is written to ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServer1_1/logs**.

Table 5.2 *SAS 9.4M7 and Prior Releases Use VMware GemFire for Cache Locator*

Directory: ***SAS-configuration-directory*/Levn/Web/gemfire/instances/ins_*port-number***	
UNIX deployment scripts:	
gemfire-locator.sh	Use this script with one of the following arguments:**start**, **stop**, or **status**.
gemfire-start-locator-sas.sh	Use this file to specify JVM options.
Windows deployment scripts:	
wrapper.conf	This file is used when you operate the **SAS [Config-Lev1] Cache Locator** service. Use this file to specify JVM options.

Directory: ***SAS-configuration-directory*/Levn/Web/gemfire/instances/ins_*port-number***	
z/OS deployment scripts:	
gemfire-start-locator-sas-zos.sh	Use this script with one of the following arguments: **start**, **stop**, or **status**.
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.

Note: The log file for the Cache Locator is **gemfire.log** and it is located here: ***SAS-configuration-directory*/Levn/Web/gemfire/instances/ins_*port-number***. Be aware that this log file is different from the log file with the same name that is written to ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServer1_1/logs**.

Member Timeout JVM Option

The Cache Locator uses the member-timeout server configuration, specified in milliseconds, to detect unresponsive members. The default value of the member timeout JVM option is 5000 milliseconds (5 seconds).

Note: In SAS 9.4M7, the member timeout JVM option is set to 10 minutes: **-Dgemfire.member-timeout=600000**. This change is a result of errors that occur regarding lost connections to GemFire or JMS Broker during upgrades to SAS 9.4M7.

You might need to modify the member timeout if you encounter issues during upgrades. Set the **-Dgemfire.member-timeout=integer** property in each installed instance of the member components. Use the following tables to locate the file to which you add or modify the JVM option. Replace the highlighted values with appropriate values for your deployment.

Table 5.3 *Windows Updates for Adjusting the Member Timeout Value*

Component	File Location and Property Examples

Cache Locator	In SAS 9.4M8:
	In the ***SAS-configuration-directory*\Levn\Web\geode\instances\ins_*port*\geode-service.bat** file, add the following JVM option:
	`SET "javaArgs=...;-Dgemfire.member-timeout=integer`
	Note: In SAS 9.4M8 you must re-install the Geode service after you make changes to the **geode-service.bat** file. See the **Note** in "Configure JVM Options for the Cache Locator" on page 52 for steps to do this.
	In SAS 9.4M7 and earlier releases:
	In the ***SAS-configuration-directory*\Levn\Web\gemfire\instances\ins_*port*\wrapper.conf** file, add the following JVM option:
	`wrapper.java.additional.N=-Dgemfire.member-timeout=integer`
	where *N* is the next sequential number from the previous parameter in the file.

Component	File Location and Property Examples
SAS Web Application Server	In the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\wrapper.conf** file, add the following JVM option: `wrapper.java.additional.N=-Dgemfire.member-timeout=integer` where *N* is the next sequential number from the previous parameter in the file. *Note:* After you modify the wrapper.conf file for SAS 9.4M7 February 16, 2022 and later release, you must rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.
	In the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\bin\setenv.bat** file, add the following JVM option: `set JVM_OPTS="... -Dgemfire.member-timeout=integer"`
SAS Distributed In-Process Scheduler Job Runner	In the ***SAS-configuration-directory*\Levn\Web\Applications\SASWIPSchedulingServices9.4\dip\wrapper.conf** file, add the following JVM option: `wrapper.java.additional.N=...Dgemfire.member-timeout=integer` where *N* is the next sequential number from the previous parameter in the file. *Note:* After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you must rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.
SAS Web Infrastructure Platform Scheduling Services	In the ***SAS-configuration-directory*\Levn\Web\Applications\SASWIPSchedulingServices9.4\servicetrigger.ini** file, add the following JVM option: `JavaArgs_N=-Dgemfire.member-timeout=integer` where *N* is the next sequential number from the previous parameter in the file.
Report Output Generation Tool (for SAS Web Report Studio)	In the ***SAS-installation-directory*\SASBIReportServices\4.4\outputgen.ini** file, add the following JVM option: `JavaArgs_N=-Dgemfire.member-timeout=integer` where *N* is the next sequential number from the previous parameter in the file.

Table 5.4 *UNIX Updates for Adjusting the Member Timeout Value*

Component	File Location and Property Examples
Cache Locator	In SAS 9.4M8: In the ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_port/geode-start-locator-sas.sh** file, add the following JVM option: `JAVA_ARGS="... -Dgemfire.member-timeout=integer"` In SAS 9.4M7 and earlier releases: In the ***SAS-configuration-directory*/Levn/Web/gemfire/instances/ins_port/gemfire-start-locator-sas.sh** file, add the following JVM option: `JAVA_ARGS="... -Dgemfire.member-timeout=integer"`

Component	File Location and Property Examples
SAS Web Application Server	In the ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServern_m/bin/setenv.sh** file, update the JVM options: `JVM_OPTS="... -Dgemfire.member-timeout=integer"`
SAS Distributed In-Process Scheduler Job Runner	In the ***SAS-configuration-directory*/Levn/Web/Applications/SASWIPSchedulingServices9.4/dip/DIPJobRunner.sh** file, add the following line in the **JVM DIP Jobrunner specific VM properties** section: `PROPS="$PROPS -Dgemfire.member-timeout=integer"`
SAS Web Infrastructure Platform Scheduling Services	In the ***SAS-configuration-directory*/LevnWeb/Applications/SASWIPSchedulingServices9.4/servicetrigger.ini** file, add the following line: `JavaArgs_N=-Dgemfire.member-timeout=integer` where *N* is the next sequential number from the previous parameter in the file.
Report Output Generation Tool (for SAS Web Report Studio)	In the ***SAS-installation-directory*/SASBIReportServices/4.4/outputgen.ini** file, add the following line: `JavaArgs_N=-Dgemfire.member-timeout=integer` where *N* is the next sequential number from the previous parameter in the file.

Set 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.

When you specify the network bind address to use for network traffic, add the following JVM option:

`-Dgemfire.bind-address=preferred-ip-address`

For information about how to specify the options, see "Configure JVM Options for the Cache Locator" on page 52.

If SAS Web Application Server is deployed on the same machine, specify the same JVM option for the server: **-Dgemfire.bind-address=preferred-ip-address**. For more information, see Specifying SAS Web Application Server JVM Options on page 44.

Modify the Configuration to Accommodate a Firewall

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.

Update 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 are 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 "Determine the Number of Required Ports" on page 59. 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 0, 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*\Lev*n*.

Table 5.5 *Windows Updates to Accommodate a Firewall*

Component	File Location and Property Examples

Cache Locator	In ***SAS-configuration-directory*\Levn\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`
	In SAS 9.4M8, the file is ***SAS-configuration-directory*\Levn\Web\geode\instances\ins_*port*\geode-service.bat**. Add the Java options as follows:
	`SET "javaArgs=...;-Dgemfire.membership-port-range=40000-50000;-Dgemfire.tcp-port=45500`
	Note: For Windows environments in SAS 9.4M8, you must re-install the Geode service after you make changes to the **`geode-service.bat`** file. See the **Note** in "Configure JVM Options for the Cache Locator" on page 52 for steps to do this.

Component	File Location and Property Examples

SAS Web Application Server	In ***SAS-configuration-directory*\Levn\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`
	Note: After you modify the wrapper.conf file for SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

	In ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\bin\setenv.bat**, add JVM options as follows: `set JVM_OPTS="... -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45600"` *Note:* You must choose a unique port for the **Dgemfire.tcp-port** JVM option for each SAS Web Application Server that runs on the same host.
SAS Distributed In-Process Scheduler Job Runner	In ***SAS-configuration-directory*\Levn\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` *Note:* After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

SAS Web Infrastructure Platform Scheduling Services	In ***SAS-configuration-directory*\Levn\Web\Applications\SASWIPSchedulingServices9.4\servicetrigger.ini**, add the following lines:
	`JavaArgs_16=-Dgemfire.membership-port-range=40000-50000`
	`JavaArgs_17=-Dgemfire.tcp-port=45800`

Report Output Generation Tool (for SAS Web Report Studio)	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`

Table 5.6 *UNIX Updates to Accommodate a Firewall*

Component	File Location and Property Examples
Cache Locator	In ***SAS-configuration-directory*/Levn/Web/gemfire/instances/ins_port/gemfire-start-locator-sas.sh**, update the Java arguments as follows: `JAVA_ARGS="... -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45500"` *Note:* In SAS 9.4M8, the directory and file is ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_port/geode-start-locator-sas.sh**.

Component	File Location and Property Examples
SAS Web Application Server	In ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServer*n_m*/bin/setenv.sh**, update the Java arguments as follows: `JVM_OPTS="... -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45600"` *Note:* You must choose a unique port for the **Dgemfire.tcp-port** JVM option for each SAS Web Application Server that runs on the same host.
SAS Distributed In-Process Scheduler Job Runner	In ***SAS-configuration-directory*/Levn/Web/Applications/SASWIPSchedulingServices9.4/dip/DIPJobRunner.sh**, add the following line in the section **JVM DIP Jobrunner specific VM properties**: `PROPS="$PROPS -Dgemfire.membership-port-range=40000-50000 -Dgemfire.tcp-port=45700"`

SAS Web Infrastructure Platform Scheduling Services	In ***SAS-configuration-directory*/LevnWeb/Applications/SASWIPSchedulingServices9.4/servicetrigger.ini**, add the following lines:
	`JavaArgs_16=-Dgemfire.membership-port-range=40000-50000`
	`JavaArgs_17=-Dgemfire.tcp-port=45800`

Report Output Generation Tool (for SAS Web Report Studio)	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`

Determine 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:

	A	B	C	D	E
Scenario	**Total Members in the Deployment**	**Machine***	**Ports per Member (Column A * 4)**	**Members per Machine**	**Total Ports (Column C * Column D)**
1 Web application server instance	5	Server tier	20	3	60
		Middle tier	20	2	40
3 Web application server instances	7	Server tier	28	3	84
		Middle tier	28	4	112

Scenario	A	B	C	D	E
	Total Members in the Deployment	Machine*	Ports per Member (Column A * 4)	Members per Machine	Total Ports (Column C * Column D)
5 Web application server instances	9	Server tier	36	3	108
		Middle tier	36	6	216

* 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.

Perform an Update in Place

After performing an update in place, you might have to update the files in Table 5.5 on page 57 and Table 5.6 on page 58 with any changes that you made.

Chapter 6

Administer JMS Broker

Overview

JMS 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.

You can secure JMS Broker resources automatically with the SAS Deployment Wizard.

Install 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.

Understand 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.

Monitor 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 **ActiveMQ** *version*. Statistics for the broker itself as well as the queues and topics are also gathered.

Toggle Secure JMS Broker Resources

If you previously configured secure JMS Broker, you need to only change the password for jmsuser.password in the ***SAS-configuration-directory*\Levn\Web\activemq\conf\credentials-enc.properties** file.

Important: In the following task, you must use only alphanumeric characters for the password. Special characters, such as open parenthesis "(", close parenthesis ")", dollar sign "$", forward slash "/", caret "^", and ampersand "&", are not supported and will cause a configuration failure.

To activate secure JMS Broker resources, follow these steps:

1. Add your JMS Broker user name and password to a credentials file.

   a. Encrypt your password by running the following command:

   ```
   SAS-configuration-directory\Levn\Web\activemq\bin\activemq encrypt
   --password activemq --input mypassword
   ```

   Note: The previous command must be entered on one line. It is shown on more than one line for display purposes only.

   The hashed password is displayed.

   For more information about the ActiveMQ encryption tool, see http://activemq.apache.org/encrypted-passwords.html.

b. Edit the ***SAS-configuration-directory*\Levn\Web\activemq\conf\credentials-enc.properties** file.

If the file does not exist, complete the following steps:

i. Open the ***SAS-configuration-directory*\Levn\Web\activemq\conf\credentials.properties** file and copy the passwords from the file.

ii. Encrypt the passwords.

iii. Create the ***SAS-configuration-directory*\Levn\Web\activemq\conf\credentials-enc.properties** file and copy the encrypted passwords to the file.

c. Specify the encrypted password as follows:

```
jmsuser.username=username
jmsuser.password=ENC(encrypted_password)
```

2. Edit the activemq.xml file so that secure JMS Broker resources are activated.

a. Edit the ***SAS-configuration-directory*\Levn\Web\activemq\conf\activemq.xml** file.

b. Add the following definition to the file:

```
<bean class="org.jasypt.encryption.pbe.StandardPBEStringEncryptor"
id="configurationEncryptor">
    <property name="algorithm" value="PBEWithMD5AndDES"/>
    <property name="password" value="activemq"/>
</bean>
<bean class="org.jasypt.spring31.properties.
EncryptablePropertyPlaceholderConfigurer" id="propertyConfigurer">
    <constructor-arg ref="configurationEncryptor"/>
    <property name="placeholderPrefix" value="@{"/>
    <property name="placeholderSuffix" value="}"/>
    <property name="location"
  value="file:${activemq.base}/conf/credentials-enc.properties"/>
</bean>
```

Note: The previous bean definitions must be entered on one line. They are shown on more than one line for display purposes only.

c. Add the following definition as a child to the <broker xmlns="http://activemq.apache.org/schema/core" brokerName="localhost" dataDirectory="${activemq.data}"> definition:

```
<plugins>
    <simpleAuthenticationPlugin>
        <users>
          <authenticationUser groups="JMS_USERS" password="@{jmsuser.password}"
                              username="@{jmsuser.username}"/>
         </users>
    </simpleAuthenticationPlugin>
    <authorizationPlugin>
        <map>
            <authorizationMap>
                <authorizationEntries>
                    <authorizationEntry admin="JMS_USERS" queue="&gt;"
                                        read="JMS_USERS" write="JMS_USERS"/>
                    <authorizationEntry admin="JMS_USERS" read="JMS_USERS"
```

```
                                                    topic="&gt;" write="JMS_USERS"/>
                        </authorizationEntries>
                    </authorizationMap>
                </map>
            </authorizationPlugin>
        </plugins>
```

Note: The previous authenticationUser and authorizationEntry definitions must be entered on one line. They are shown on more than one line for display purposes only.

3. Change the ActiveMQ client password:

   a. The authorization string for *password* needs to be specified in the catalina.properties file. The passphrase must match what is found in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\secure.file** file. It is suggested that you encode the *password*, instead of using plaintext, by running one of the following commands:

      On Windows:

      ```
      "SASHOME\SASWebApplicationServer\9.4\tcruntime-admin.bat" encode
      value-to-encrypt passphrase
      ```

      On UNIX:

      ```
      SASHOME/SASWebApplicationServer/9.4/tcruntime-admin.sh encode
      value-to-encrypt passphrase
      ```

   b. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\catalina.properties** file.

   c. For each SAS server that you want to change the password for, update the following properties:

      - pw.sas.jms.TopicConnectionFactory=pbkdf2://*encoded_password*
      - pw.sas.jms.QueueConnectionFactory=pbkdf2://*encoded_password*

4. Restart the SAS middle tier by running the following command:

   ```
   SAS-configuration-directory\Levn\sas.servers restart
   ```

Part 3

Middle-Tier Applications

Chapter 7

Administer the SAS Web Infrastructure Platform

Overview

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.

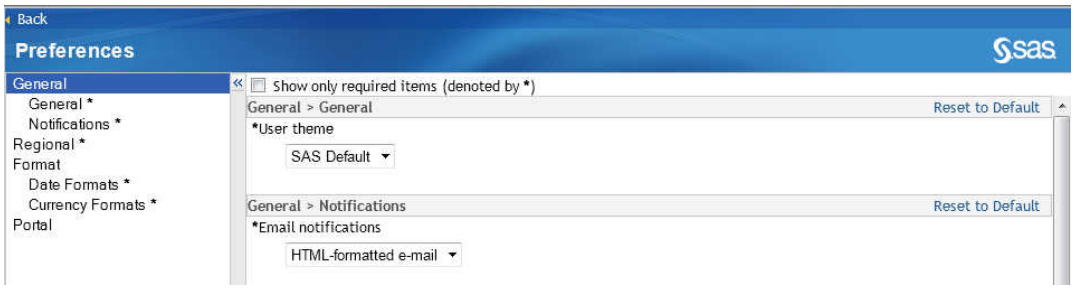
You can invoke the application by using the following URL:

`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.

Note: The **My alerts portlet** alert notification option does not only refer to the My alerts portlet that is available with SAS Information Delivery Portal. You do not have to have SAS Information Delivery Portal to choose this option. Selecting this option indicates that you prefer to store alert notifications in a database file that you can retrieve and display using mechanisms provided by various SAS 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.

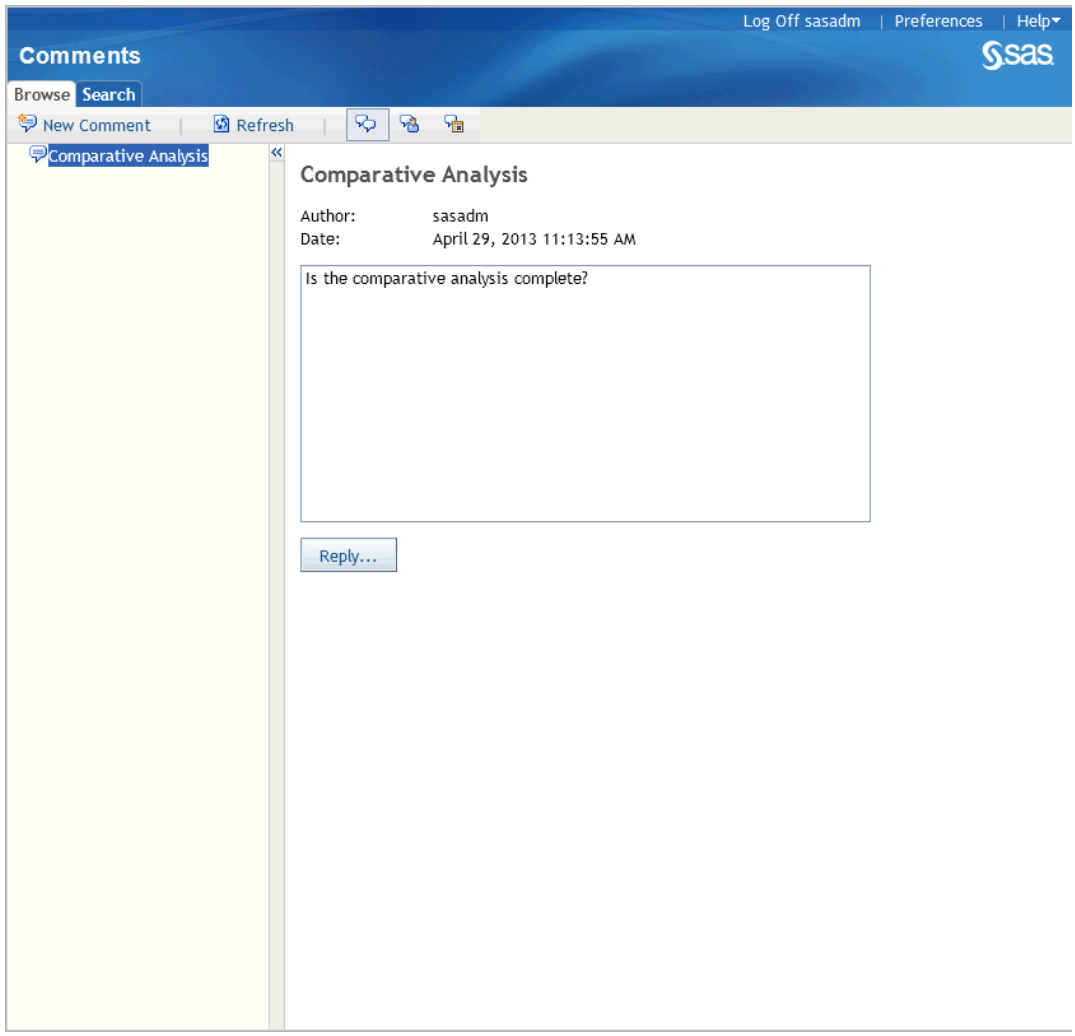
Note: Due to possible conflicts that can occur when multiple users delete comments in the same comment thread, the best practice is to limit the number of users to just a few.

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



Note: Attachments that are uploaded through SAS Comment Manager can be compromised and as a result can cause a stored XSS attack when those attachments are viewed within the browser. Many file content types are allowed to be uploaded and viewed in a non-secure sandbox iframe by default. If you need to add a file type to the list, use the Configuration Manager in SAS Management Console. See "Add a Property to Configure File Types to Upload in SAS Comment Manager That Cannot Be Viewed Securely" on page 77 for instructions.

Use Configuration Manager

Overview

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 "Specify Connection Properties" on page 79.
- 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 "Set Global Properties for SAS Applications" on page 73.
- 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 "Set Global Properties for SAS Applications" on page 73.
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 start 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 "Use JMX Tools to Manage SAS Resources " on page 391.
- Alternatively, stop and then start 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:

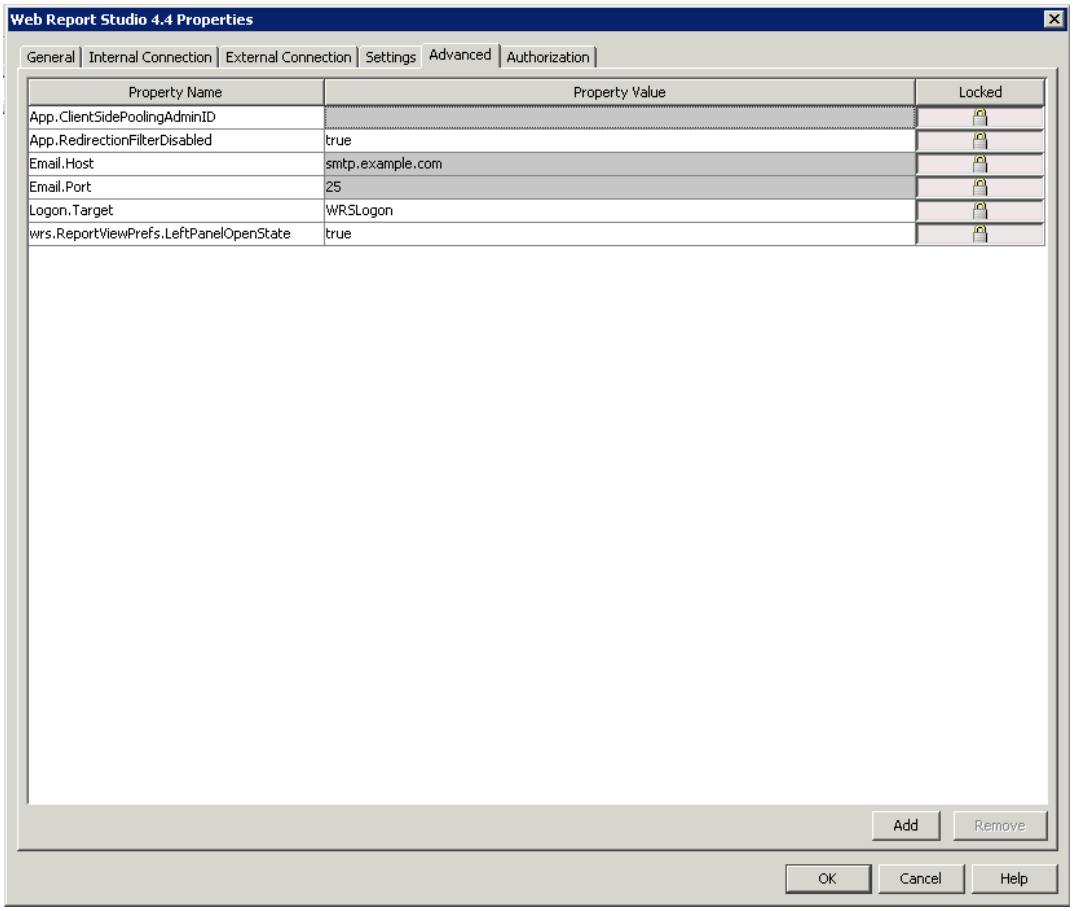
   Property Name: **`wrs.ReportViewPrefs.LeftPanelOpenState`**

   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 71.

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.

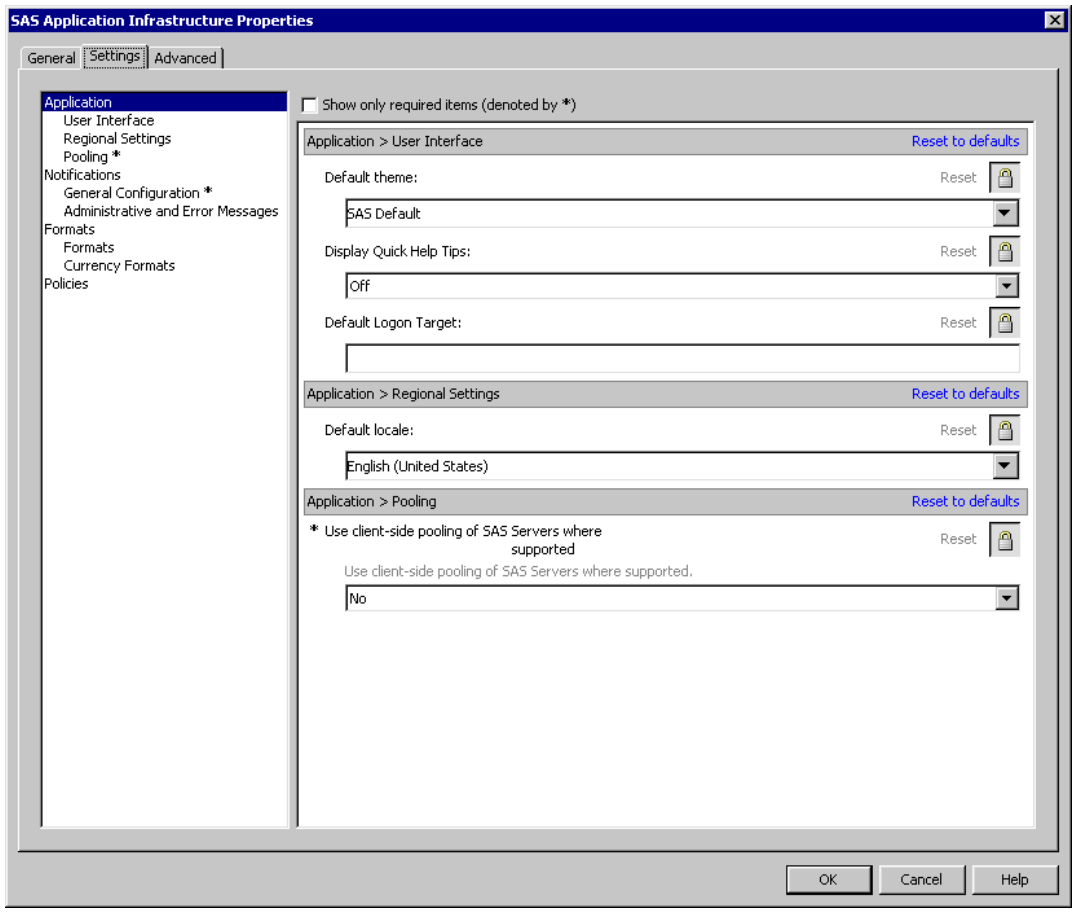
Set Global Properties for SAS Applications

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.

Change 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 7.1 *SAS Application Infrastructure Settings*

Setting	Default Value	Description
Application > User Interface		
Default theme	SAS Default	This setting controls the default theme that is used by the SAS web applications. For information about creating an alternative theme, see Chapter 12, "Administer SAS Web Application Themes," on page 169.
Display Quick Help Tips	Off	
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		

Setting	Default Value	Description
Use client-side pooling of SAS Servers where supported	No	For information about the advantages and disadvantages, see "Benefits and Risks of Server-Side 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 "Use the SMS Text Message Alert Notification Type" on page 77.
Character set for e-mail messages	UTF-8	
Allow multi-part e-mail messages	Yes	
Alert prefix type	Default	
Alert prefix	SAS Alert:	
E-mail digest frequency	4	
Notifications > Administrative and Error Messages		
Sender of messages	noreply@*smtpserver*	Used as the sender email address for administrative messages.
Recipient of administrative messages	varies	Administrative and error messages are sent to all email addresses in the list.
Formats > Formats		
Short date format	varies	Use the menu to set the default format for date, time, and datetime values.
Time format		
Long date format		
Time/Date format		
Formats > Currency Formats		
Currency display format	varies	Use the menu to set the default format for currency values.
Currency number format		

Setting	Default Value	Description
Policies		For information about policies, see "Configure Middle-Tier Security Policies" on page 133.

Add a Property to Configure File Types to Upload in SAS Comment Manager That Cannot Be Viewed Securely

Attachments that are uploaded through SAS Comment Manager can be compromised and as a result can cause a stored XSS attack when those attachments are viewed within the browser. Many file content types are allowed to be uploaded and viewed in a non-secure sandbox iframe by default. If you need to add a file type to the list, use the Configuration Manager in SAS Management Console.

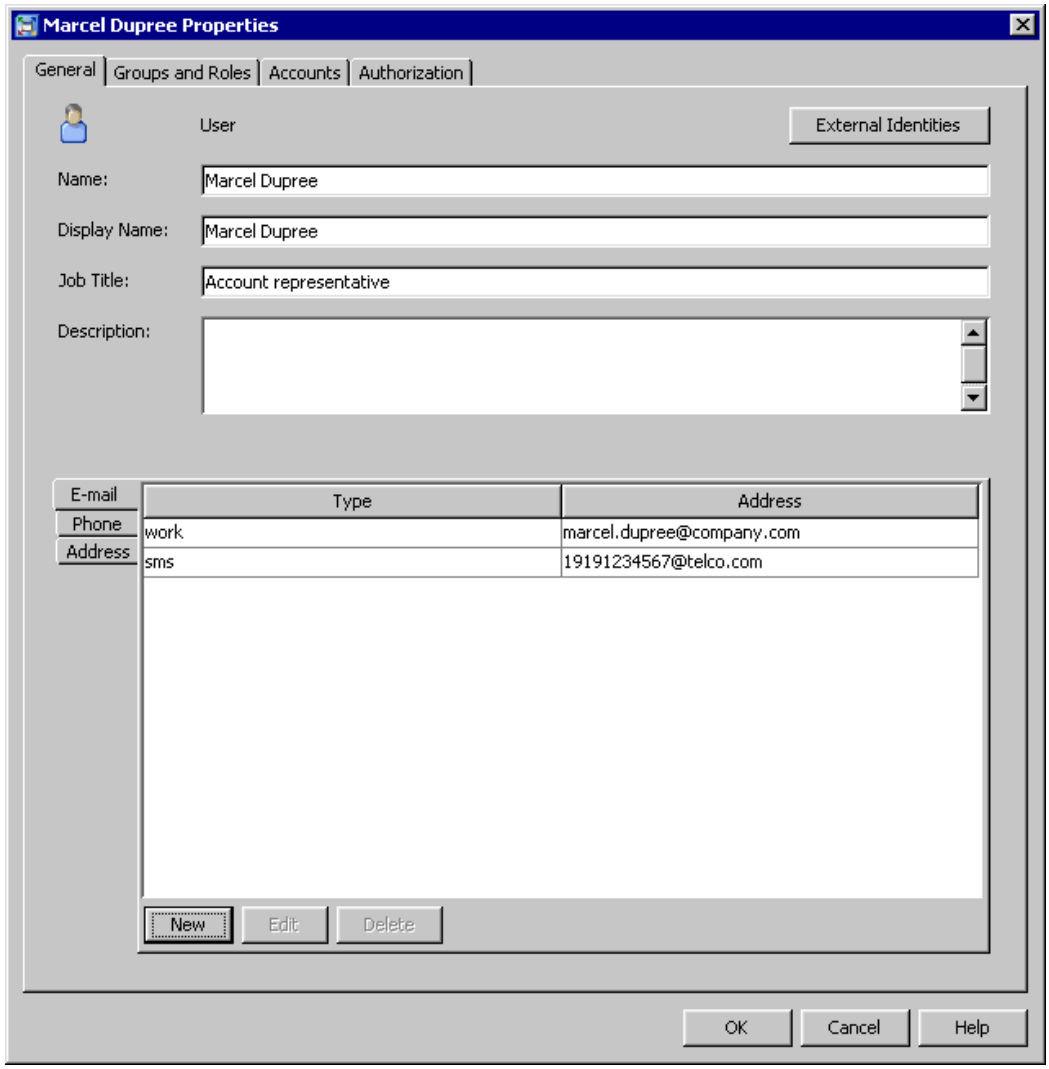
1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**.
3. Right-click **SAS Application Infrastructure** and select **Properties**.
4. Click the **Advanced** tab.
5. Click **Add**.
6.
   - For **Property Name** type `sas.web.app.insecureViewingCommentAttachmentTypes`.
   - For **Property Value** type the file content type that you want to add to the default list. Use the OR character (|) to separate multiple values. Here are examples of values that you add: `text/HTML | text/javascript | image/bmp | application/json | text/css | text/csv | image/gif | text/XML | image/png`
7. Click **OK** in the **Define Property Window**.
8. Click **OK**.
9. Restart SAS Web Application Server.

Use the SMS Text Message Alert Notification Type

The alert notification service can send alerts though 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 text 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 Email Address



Make sure that you know the SMS text message email gateway for the provider. Some SMS email 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 text message text messages have a SMS-style email address, you might need to set two properties related to SMS text messages.

Table 7.2 *Advanced Properties for SMS Text Messages*

Property Name	Default Value	Description
Notifications.SMSMessageLength	120 characters	Modify this value as needed to increase or decrease the size of SMS text messages that SAS software sends to the mail server.
Policy.EnforceSMSMessageLength	false	If set to true, then messages are truncated to the length of the previous property.

Specify Connection Properties

Overview of Internal and External Connections

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

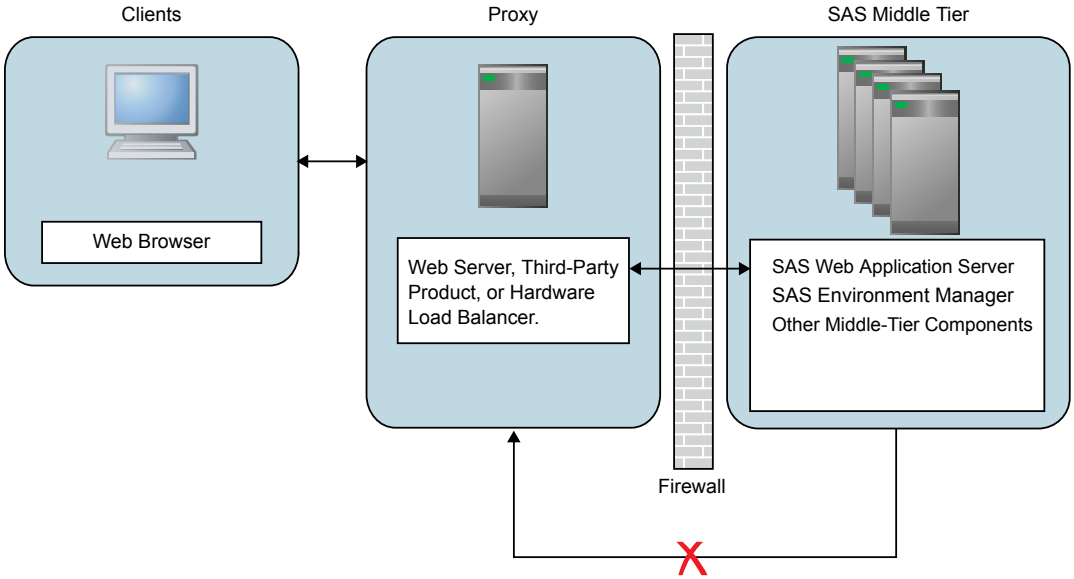
- Communication Protocol
- Host Name
- Port Number
- Service

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 **Use internal connection information** check box is selected on the **External Connection** tab, 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.

Some SAS users prefer to update connection properties using a SAS DATA step. This approach is beyond the scope of this section. If you choose to modify connections using a SAS script rather than SAS Management Console, then 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.

You will also need to change the port and protocol for SASTheme_default. From SAS Management Console, navigate to the **Plug-ins** tab and select **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Themes** ⇨ **SASTheme_default**. View the **Properties** to determine whether there is connection information that needs to be updated.

Change Internal Connection Properties

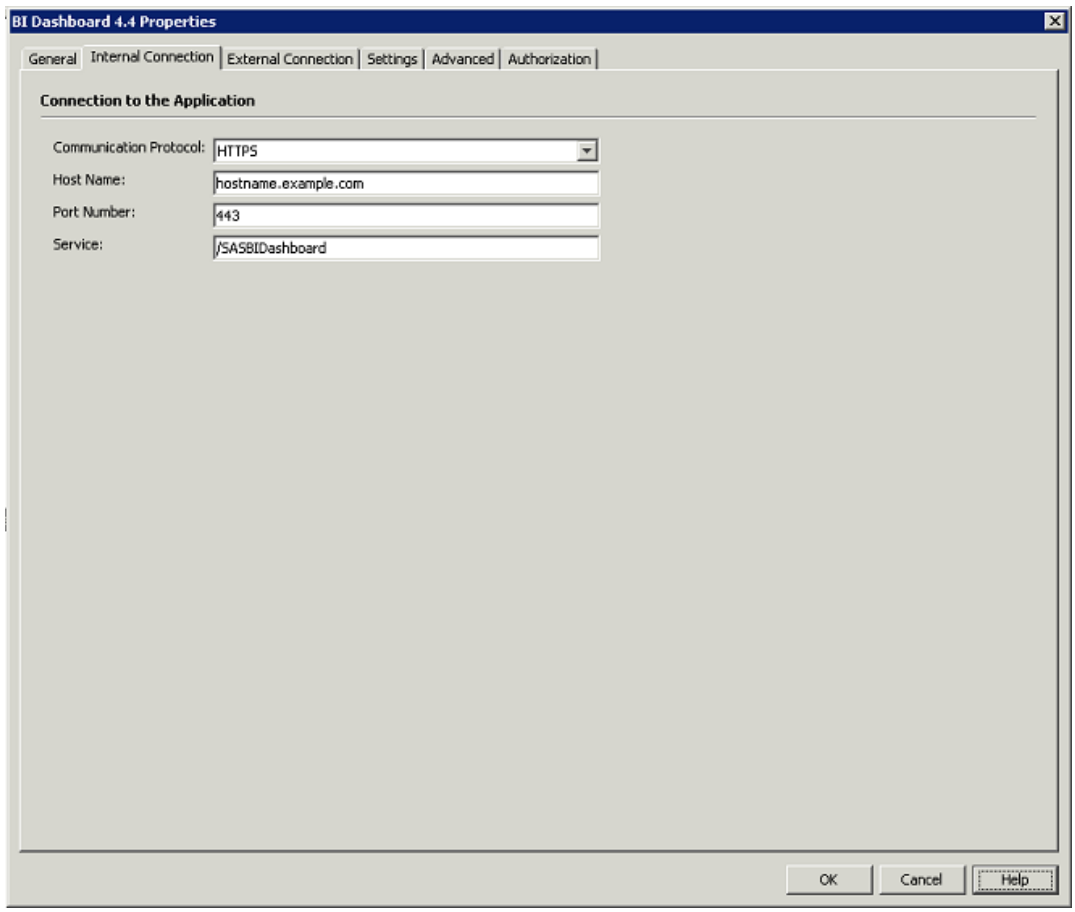
The **Internal Connection** tab in the properties dialog box for SAS applications enables you to modify the parameters for connecting to a SAS web application. The selections that are displayed on the tab determine the URL that is used to access the application's resources or services.

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.

Figure 7.7 *Internal Connection Tab for 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. You can use the **`SAS-configuration-directory/Levn/Web/WebServer/conf/sas.conf`** file to

identify the specific web applications to proxy. Each web application is identified in a pair of ProxyPass and ProxyPassReverse directives.

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].

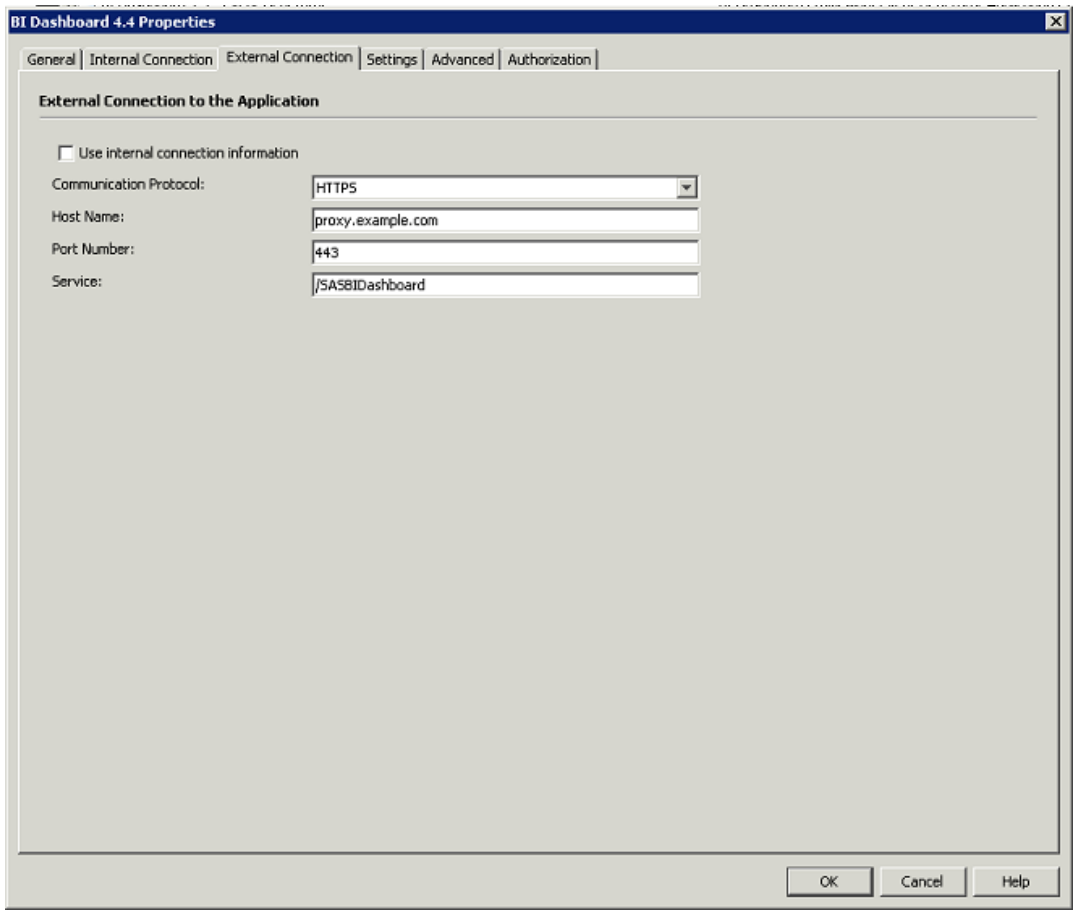
In addition, for SAS 9.4M3 and SAS 9.4M4 only, you must edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\classes\identity-service.properties** file. Locate the following line and enter the correct information for your environment:

```
url.base=https\://server/
```

Change 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.

Figure 7.8 External Connection Tab for BI Dashboard Properties



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
```

Some web applications might require you to configure additional options when setting up an external reverse proxy. For more information, see "Optional Configuration for the Cross Domain Proxy Servlet" on page 370.

Note: After specifying this JVM option, you must restart the SAS Web Application Server in order for the change to take effect.

In addition, for SAS 9.4M3 and SAS 9.4M4 only, you must edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\classes\identity-service.properties** file. Locate the following line and enter the correct information for your environment:

```
url.base=https\://server/
```

See Also

"Specify JVM Options" on page 44

Configure Auditing for SAS Web Applications

Overview

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.

Logon, logoff, and unsuccessful logon 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 logon, logoff, unsuccessful logon 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 "Configure Auditing for SAS Web Applications" on page 83.
- Purge the audit archive tables as needed.

Enable Auditing for Additional Services

Prior to SAS 9.4M3, all SAS products that include the SAS Web Infrastructure Platform provide audit records for logon and logoff activity, and unsuccessful logon 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:

```
<!-- Job Execution Service auditing
<bean class="com.sas.svcs.aop.auditing.jes.SuccessfulSubmitJobAuditAdvice">
     <property name="auditRecorder" ref="auditService" />
</bean>
```

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.

   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.
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 SAS 9.4M3, 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

"Specify JVM Options" on page 44

Enable Audit Profiles

Starting with SAS 9.4M3, audit advice beans can be enabled. Audit advice beans are disabled by default. 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:

Spring Profile Name	Description
sas.audit	Enables all audit profiles.
sas.audit.svcs.content	Enables all Content Server audit profiles.
sas.audit.svcs.content.service	Enables Content Server audits that deal with services.
sas.audit.svcs.jes	Enables all Job Execution Service audits (definition, execution, and retrieval).
sas.audit.svcs.jes.definition	Enables Job Execution Service audits that deal with job definition.
sas.audit.svcs.jes.execution	Enables Job Execution Service audits that deal with job execution.
sas.audit.svcs.jes.retrieval	Enables Job Execution Service audits that deal with job retrieval.
sas.audit.svcs.mail	Enables all Mail Service audits.
sas.audit.svcs.scheduling	Enables all Scheduling audits.
sas.audit.svcs.impersonation	Enables all Impersonation audits.
sas.audit.svcs.wss	Enables all Workspace Data audits (data, data set, file, fileref, format, language, libref, utilities, and workspace).
sas.audit.svcs.wss.data	Enables all Workspace Data audits that deal with data.
sas.audit.svcs.wss.dataset	Enables all Workspace Data audits that deal with data set.
sas.audit.svcs.wss.file	Enables all Workspace Data audits that deal with file.
sas.audit.svcs.wss.fileref	Enables all Workspace Data audits that deal with fileref.

Spring Profile Name	Description
sas.audit.svcs.wss.format	Enables all Workspace Data audits that deal with format.
sas.audit.svcs.wss.language	Enables all Workspace Data audits that deal with language.
sas.audit.svcs.wss.libref	Enables all Workspace Data audits that deal with libref.
sas.audit.svcs.wss.utilities	Enables all Workspace Data audits that deal with utilities.
sas.audit.svcs.wss.workspace	Enables all Workspace Data audits that deal with workspace.

See Also

"Specify JVM Options" on page 44

Archive Process for Audit Records

Overview

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.

By default, audit archiving is enabled. To disable audit archiving, set the sas.svcs.audit.archive.disabled property to **`true`**.

SAS_AUDIT_ARCHIVE_RULE Table

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*

Column Name	Description
OBJECT_TYPE_ID	Object type. Each object type is assigned an ID in table SAS_TYPE_OBJECT.
ACTION_TYPE_ID	Type of change. Each action type is assigned an ID in table SAS_TYPE_ACTION.

Column Name	Description
FREQUENCY_NO	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.

Archive Jobs

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"
```

Note: In a clustered environment, set the CRON JVM option on the primary node.

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 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.

Audit Object Types and Actions

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*

Audit Action	Object Type ID Value	Action Type ID Value
User log on	-1	8
User log off	-1	9
Sent E-mail	-1	44
Add job	11	0
Submit job	10	3
Retrieve job	11	45
Cancel job	10	47
Release job	10	48
Update job	11	1

Audit Action	Object Type ID Value	Action Type ID Value
Remove job	11	37
Start scheduled job	86	3
Remove scheduled job	86	37

Purge 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:

```
DELETE FROM sas_audit_entry_archive;

DELETE FROM sas_audit_archive;
```

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

```
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.

Use the SAS Web Administration Console

Overview

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

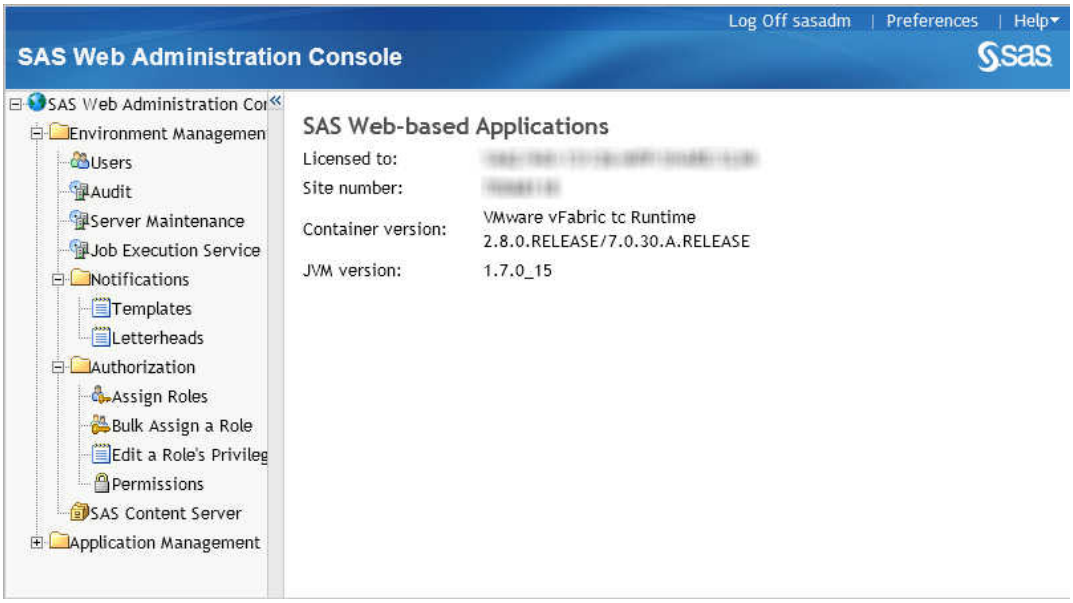
- monitoring information about users who are currently logged on to SAS web applications
- viewing audit reports that show user logon and logoff activity and failed logon attempts
- performing server maintenance, as a part of system maintenance
- updating the Job Execution Service configuration on page 93
- 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 "Use the SAS Content Server Administration Console " on page 147.

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 "Use the SAS Content Server Administration Console " on page 147.

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 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. In SAS 9.4M6 and earlier, 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.

   Starting with SAS 9.4M7, a single message is sent to all recipients (both authenticated users and system users). To send separate emails, you must select one recipient at a time.
7. Click **Send**.

Force Users to Log Off

Starting with SAS 9.4M3, 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.

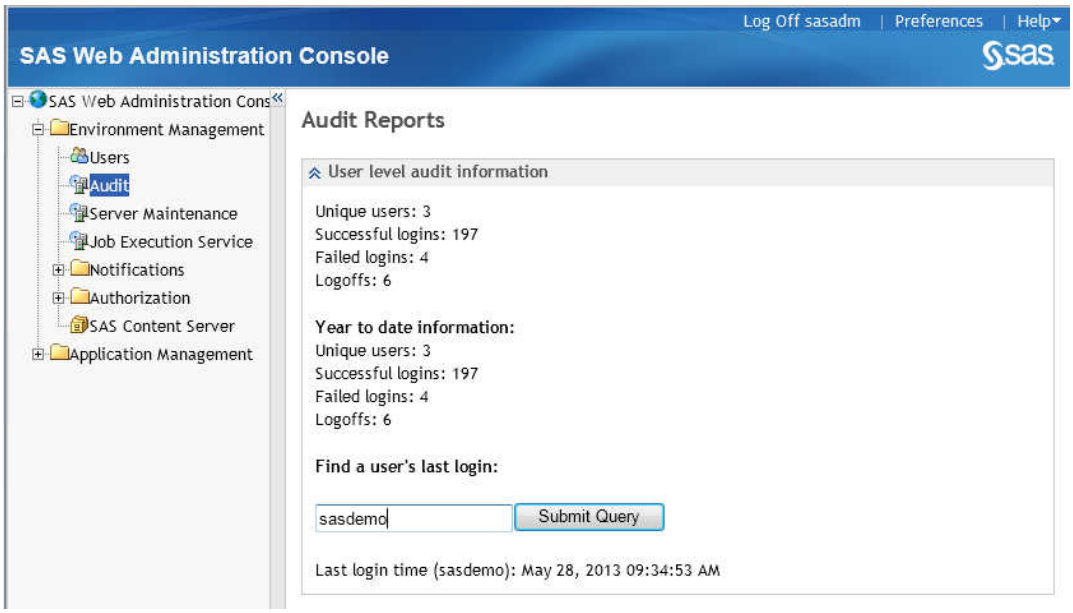
   You can select multiple check boxes in order to force off several users. To select all of the check boxes, select the check box in the heading of the last column.
3. Click in the heading of the last column and select **Force Log Off**.

   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.

View 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.

Figure 7.10 *Audit Reports Page*



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**.

Perform 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:

1. Connect to the load balancer manager at http://saswebserver.example.com/balancer-manager.

   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 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.

Update 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.

Manage Notification Templates and Letterheads

Overview

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 95.
   - Edit the subject line (for HTML and text formats only).
   - Edit the template body in the HTML, text, and SMS text message 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 95.
   - 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 94.

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.

Manage Web Infrastructure Platform Privileges and Roles

Overview

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:

1. Select **Environment Management** ⇨ **Authorization** ⇨ **Assign Roles** in the navigation pane.
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.

Manage Web-layer Permissions

Overview

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, "Administer the SAS Content Server," on page 139.

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.
   a. If the requesting user has a direct grant or denial, that determines the outcome.
   b. If a group to which the requesting user belongs has a direct denial, the outcome is a denial.
   c. 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.

Review and Set 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**.

View 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.

Chapter 8

Administer SAS Web Applications

Overview of 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 "Rebuild the SAS Web Applications" on page 102.
- **Redeploy web applications.** You can redeploy web applications that have been rebuilt. See "Redeploy the SAS Web Applications" on page 107.
- **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 re-install 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**.

Rebuild 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 106.
- 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 103.
- 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.
- When you rebuild SAS Web Applications, the sas.conf file in the ***SAS-configuration-directory*/Lev1/Web/WebServer/conf/** directory is overwritten. If you customized this file (for example, when configuring HTTPS for the SAS Web Application Servers), back up the sas.conf file before rebuilding. Then restore the sas.conf backup file after rebuilding the SAS Web Applications with the following steps.

  Note: If you are unsure whether the sas.conf file has been modified, back up the sas.conf file before rebuilding the SAS Web Applications. After rebuilding the SAS Web Applications, compare the sas.conf backup file and the current sas.conf file. If there are changes, restore the sas.conf backup file after rebuilding the SAS Web Applications with the following steps.

  1. Stop the SAS Web Server.
  2. Replace the sas.conf file in the ***SAS-configuration-directory*/Lev1/Web/WebServer/conf/** directory with the backup file.
  3. Restart the SAS Web Server.

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 SAS Metadata Server, SAS Deployment Agent, and SAS Cache Locator are 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 104.
9. 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 "Redeploy the SAS Web Applications" on page 107.

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 8.1 *Product Configuration, Web Application, and File Names*

Product Configuration	Application	File Name
BI Dashboard *version*	SAS BI Dashboard	**sas.bidashboard*version*.ear**
BI Portlets *version*	SAS BI Portlets	**sas.biportlets*version*.ear**
Environment Manager Middle Tier *version*	SAS Environment Manager	**sas.environmentmanager*version*.ear**
Flex Application Themes *version*	SAS Flex Application Themes	**sas.flexthemes*version*.ear**
	SAS Theme Designer for Flex	**sas.themedesigner*version*.ear**

Product Configuration	Application	File Name
Help Viewer for Midtier App *version*	SAS Help Viewer for Midtier Applications	**sas.webdocmd*version*.ear**
Information Delivery Portal *version*	SAS Information Delivery Portal	**sas.portal*version*.ear**
	SAS Package Viewer	**sas.packageviewer*version*.ear**
SAS Themes	SAS Web Application Themes	**sas.themes.ear**
Visual Analytics *version*	SAS Visual Analytics Administrator	**sas.visualanalyticsadministrator*version*.ear**
	SAS Visual Analytics Designer	**sas.visualanalyticsdesigner*version*.ear**
	SAS Visual Analytics Explorer	**sas.visualanalyticsexplorer*version*.ear**
	SAS Visual Analytics Graph Builder	**sas.visualanalyticsgraphbuilder*version*.ear**
	SAS Visual Analytics Hub	**sas.visualanalyticshub*version*.ear**
	SAS Visual Analytics Services	**sas.visualanalyticsservices*version*.ear**
	SAS Visual Analytics Viewer	**sas.visualanalyticsviewer*version*.ear**
Web Infrastructure Platform *version*	SAS Content Server	**sas.wip.scs*version*.ear**
	SAS Stored Process	**sas.storedprocess*version*.ear**
	SAS Web Administration Console	**sas.wip.admin*version*.ear**
	SAS Web Infrastructure Platform Applications	**sas.wip.apps*version*.ear**
	SAS Web Infrastructure Platform Resources	**sas.wip.resources*version*.ear**
	SAS Web Infrastructure Platform Services	**sas.wip.services*version*.ear**
	SAS Workflow	**sas.workflow*version*.ear**
	SAS Authorization Service	**sas.authorization.services.war**
	SAS Identity Services	**sas.identity.services.war**
	SAS Principal Services	**sas.principal.services.war**
Web Report Studio *version*	SAS Web Report Studio	**sas.webreportstudio*version*.ear**

Web Application Custom Content

Overview

You can add custom content to a SAS web application by doing any of the following:

- creating and saving your custom content in the appropriate custom content directory structure
- running the SAS Deployment Manager
- 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 103.

3. Redeploy the web applications. For more information, see "Redeploy Web Applications" on page 107.

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

Redeploy the SAS Web Applications

Overview

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 redeploy the SAS web applications, then manual TLS changes are reverted. For information about manually configuring TLS for the SAS Web Application Server, see "Configure SAS Web Application Server for HTTPS" on page 318.

When you redeploy the SAS Web Applications, the **sas.conf** file in the ***SAS-configuration-directory*/Lev1/Web/WebServer/conf/** directory is overwritten. If you customized this file (for example, when configuring HTTPS for the SAS Web Application Servers) back up the **sas.conf** file before redeploying. Then restore the **sas.conf** backup file after redeploying the SAS Web Applications with the following steps.

Note: If you are unsure whether the **sas.conf** file has been modified, back up the **sas.conf** file before redeploying the SAS Web Applications. After redeploying the SAS Web Applications, compare the **sas.conf** backup file and the current **sas.conf** file. If there are changes, restore the **sas.conf** backup file after redeploying the SAS Web Applications with the following steps.

1. Stop the SAS Web Server.
2. Replace the **sas.conf** file in the ***SAS-configuration-directory*/Lev1/Web/WebServer/conf/** directory with the backup file.
3. Restart the SAS Web 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. Stop the web application server, if it is running.
2. Start the SAS Deployment Manager.
3. Select **Deploy Web Applications** and click **Next**.
4. Specify the configuration directory and the level (for example, Lev1) on the Select Configuration Directory/Level page. Click **Next**.
5. Enter the user ID and password for an unrestricted administrative user (for example, sasadm@saspw) on the Specify Connection Information page. Click **Next**.

6. 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**.
7. 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 104.
8. 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*\Levn\Web\WebAppServer\SASServer1_1\sas_webapps\Backup** directory.
- Context file backups are in the ***SAS-configuration-directory*\Levn\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. This backup folder can be quite large because it contains the entire contents of the SASServer[n]_1 directory. You should consult this folder if you need to reapply any customizations after updating your software. After the customizations have been applied and validated, you can delete these backup directories.

Additional Steps for Horizontal Clusters

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

1. Ensure that all SAS servers and spawners from tiers other than the horizontal cluster nodes are running.
2. For each middle-tier node (horizontal cluster node), stop all SAS sessions, daemons, spawners, servers, and agents.
3. If not already installed, apply any hot fixes and re-apply the latest Security Update to the horizontal cluster node.
4. Start the SAS Deployment Agent on the middle-tier node.
5. Use the **Update Existing Configuration** task in SAS Deployment Manager. This task copies the rebuilt web applications from the primary middle tier node and deploys them to the horizontal cluster node. Perform the following steps in SAS Deployment Manager:

   a. On the Select SAS Deployment Manager Task page, under **Administration Tasks**, click **Update Existing Configuration**, and then click **Next**.

   b. On the Select Configuration Directory/Level page, specify the configuration directory and the level (for example, **Lev1**), and then click **Next**.

   c. On the Specify Connection Information page, enter the user ID and password for an unrestricted administrative user, and then click **Next**.

   d. On the Summary page, click **Start**.

6. If you applied any SAS Web Server or SAS Environment Manager hot fixes to the middle-tire node, perform any additional post-update steps. See the SAS Support site for more details.
7. If the SAS Environment Manager Agent is not started, start it now.

Reconfigure 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*.

Administer 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*\Lev1\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:

Task	Log4j Configuration File	Dynamic Logging Changes
Change the logging levels.	✓	✓

Task	Log4j Configuration File	Dynamic Logging Changes
Add a logging category.	✓	✓
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.	✓	

Note:

- Prior to the SAS 9.4M7 February 15, 2022 release, SAS web applications use Log4j v1 for its logging framework. For information about the log4j v1 configuration file, see http://logging.apache.org/log4j/1.2/index.html and http://logging.apache.org/log4j/1.2/manual.html.
- Starting with SAS 9.4M7 February 15, 2022 release and SAS 9.4 M8, SAS web applications use Log4j v2 for its logging framework. For information about the log4j v2 configuration file, see http://logging.apache.org/log4j/2.x/manual/index.html.

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 than 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.

Use 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:

```
<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*

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

Name in Web Administration Console	Context Root for Logging Control Console
Visual Analytics Viewer *version*	SASVisualAnalyticsViewer/admin/Logging
Visual Data Builder *version*	SASVisualDataBuilder/admin/Logging
Theme Designer for Flex	SASThemeDesignerForFlex/admin/Logging
Web Infra Platform Permission Manager *version*	SASPermissionManager/admin/Logging
Web Infra Platform Services *version*	SASWIPServices/admin/Logging
Web Infra Platfrm ClntAccss *version*	SASWIPClientAccess/admin/Logging
Web Infra Platfrm Soap Svcs *version*	SASWIPSoapServices/admin/Logging
Web Report Studio *version*	SASWebReportStudio/admin/Logging
Workflow Services *version*	SASWorkflowServices/admin/Logging
Workflow Web Services *version*	SASWorkflowWebServices/admin/Logging

* This application is not listed in the Web Administration Console, but you can make changes to the dynamic logging settings by accessing the context root and using the Logging Control Console. For more information, see "Access the Logging Control Console" on page 114.

Use SAS Web Administration Console

You can use SAS Web Administration Console to change logging levels at run time. 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 96.

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.

Access 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.

To access the console, enter the following URL in your web browser and substitute the fully qualified host name and port number of your server:

http://*hostname*:*port*/*context_root*

For example: http://*hostname.example.com*:*port*/SASEnvironmentMgrMidTier/admin/Logging.

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

Change 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>false</param-value>
    </init-param>
</servlet>
```

3. Restart SAS Web Application Server.

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**

Change 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:

```
<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.

- Prior to the SAS 9.4 M7 February 15, 2022 release, SAS web applications use Log4j v1 for its logging framework. The log4j conversion characters are described at https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html. Two conversion characters 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.
- Starting with SAS 9.4 M7 February 15, 2022 release and SAS 9.4 M8, SAS web applications use Log4j v2 for its logging framework. The log4j conversion characters are described at https://logging.apache.org/log4j/2.x/manual/

layouts.html#PatternLayout. The `%u` conversion character outputs the UUID and the `%s` conversion character is not valid.

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

Chapter 9

Administer SAS Logon Manager

Overview

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 time-out** policy in Configuring Middle Tier Security Policies.

Note: The maximum length for user names and passwords is 500 characters.

Central Authentication Service

For SAS 9.4, SAS Logon Manager uses the Central Authentication Service (CAS) to enable single sign-on. This allows users to access multiple SAS web applications without delay or interruption.

Clients authenticate to CAS using credentials. The authentication mechanism is SAS Metadata Server via the SAS Metadata Repository JAAS logon module. When a client authenticates to CAS, the credentials (user name and password) are exchanged for a ticket-granting ticket (TGT). This TGT can then be used to obtain service tickets for specific services that the client wants to invoke.

CAS is configured to support Single Logout (SLO). This means that CAS invalidates all client application sessions in addition to its own SSO session. When you log out of a SAS web application, a notification is sent to invalidate all other SAS HTTP sessions. For more information, see Apereo CAS - Identity & Single Sign-On . Note that this link routes you to a newer version of CAS.

Starting with SAS 9.4 M8, SAS uses CAS version 6.6 that calls Apache Commons Validator to validate SAS Web URLs. Only SAS web applications with valid URLs and valid domain names receive a SLO message.

Important: Starting with SAS 9.4 M8, **`local`** is not a valid internal domain name. Examples of valid internal domain names are: **`localdomain`** and **`localhost`**.

Trouble Accessing SAS Logon Manager?

After the middle tier is deployed, SAS Logon Manager builds a list of approved services, based on those that are registered in SAS Metadata Server. For security reasons, only those registered applications are authorized to use SAS Logon Manager. If the middle-tier servers are started before all of the applications in the environment are configured, the list of authorized services might not be complete.

If you attempt to log on to a web application that is not on the cached list, the following message is displayed: `The application is not authorized to use SAS Logon Manager.`

If this message is displayed, you need to refresh the cached list by restarting the SAS middle-tier servers, starting with the machine containing the SASServer1 web application. Run the following command:

```
SAS-configuration-directory\Levn\sas.servers restart
```

Customize 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. Starting with SAS 9.4M8, the location of the files to update is ***SAS-home-directory*\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\classes\templates\fragments**. Edit the files that you want to change:

   - logon_custom.html
   - logoff_custom.html
   - timeout_custom.html

   Note: When you migrate from a previous version of SAS 9.4, you need to copy the custom messages from the former **.jsp** files into the **.html** files.

   In SAS 9.4M7 and earlier releases, the location of the files to update is ***SAS-home-directory*\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\view\jsp\default\ui**. 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 file 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 time-out 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 redeployment.)
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 located here:

- In SAS 9.4M8, ***`SAS-home-directory`***`\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\classes\templates\fragments`
- In SAS 9.4M7 and earlier releases, ***`SAS-home-directory`***`\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\view\jsp\default\ui`

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.

Display 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 that is 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 time-out** 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
- SAS Preferences
- SAS Web Administration Console
- SAS Stored Process

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, 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.

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. The value that you specify must be smaller than the value or values that are specified for the session time-out intervals in the web.xml files.

To set the interval for the inactive session warning, follow these steps:

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, follow these steps:

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.

Configure 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. (See notes for **SAS Logon Manager** in the following table for a variation to this default value.) 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 time-out** 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 or add 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 9.1 *Files to Modify for the Time-Out Interval*

Web Application	File Location
SAS Logon Manager *	***SAS-installation-directory*\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig** *Note:* In SAS 9.4M6 and SAS 9.4M7 the default session-timeout is set to 5 minutes for this application in the **web.xml.orig** file. This overwrites the default session-timeout of 30 minutes, which is defined in the ***SAS-configuration-directory*/Levn/Web/WebappServer/SASServer1_1/conf/web.xml** file. Remove the session-timeout element in **web.xml.orig** to default to 30 minutes. Or, change the session-timeout value in **web.xml.orig** to something other than 30 minutes. *Note:* In SAS 9.4M8 the session time-out element is not included in the **web.xml.orig** file. Therefore, the session-timeout defaults to 30 minutes, which is defined in the ***SAS-configuration-directory*/Levn/Web/WebappServer/SASServer1_1/conf/web.xml** file. Add the session-timeout element to **web.xml.orig** to overwrite the default value of 30 minutes.

Web Application	File Location
SAS Deployment Backup and Recovery Tool	***SAS-installation-directory***\SASDeploymentBackupandRecoveryTool\9.4\configurable\wars\sas.svcs.admin.backup\WEB-INF\web.xml.orig
SAS Environment Manager Middle-Tier Configuration *	***SAS-installation-directory***\SASEnvironmentManagerMidTier\9.4\Configurable\wars\sas.admapp.fldmod\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASEnvironmentManagerMidTier\9.4\Configurable\wars\sas.admapp\WEB-INF\web.xml.orig
SAS Help Viewer for Midtier Applications	***SAS-installation-directory***\Documentation\9.4\Static\wars\sas.webdoc\WEB-INF\web.xml ***SAS-installation-directory***\Documentation\9.4\Static\wars\sas.webdoc\WEB-INF\web.spring-enabled.xml
SAS BI Dashboard	***SAS-installation-directory***\SASBIDashboard***version***\Configurable\wars\sas.bidashboard\WEB-INF\web.xml.orig
Event generation framework in SAS BI Dashboard	***SAS-installation-directory***\SASBIDashboard***version***\Configurable\wars\sas.eventsgenerationframework\WEB-INF\web.xml.orig
SAS BI Portlets	***SAS-installation-directory***\SASBIPortlets***version***\Configurable\wars\sas.biportlets\WEB-INF\web.xml-thirdparty.orig ***SAS-installation-directory***\SASBIPortlets***version***\Configurable\wars\sas.biportlets\WEB-INF\web.xml-idp.orig
JSR 168 for SAS BI Portlets	***SAS-installation-directory***\SASBIPortlets***version***\Configurable\wars\sas.jsr168remoteportlet\WEB-INF\web.xml.orig
Flex Themes for SAS*	***SAS-installation-directory***\SASFlexApplicationThemes***version***\Configurable\FlexThemes\wars\sas.flexthemes\WEB-INF\web.xml.orig
SAS Theme Designer for Flex	***SAS-installation-directory***\SASFlexApplicationThemes***version***\Configurable\ThemeDesigner\wars\sas.themedesigner\WEB-INF\web.xml.orig
SAS Package Viewer	***SAS-installation-directory***\SASInformationDeliveryPortal***version***\Configurable\wars\sas.packageviewer\WEB-INF\web.xml.orig
SAS Information Delivery Portal	***SAS-installation-directory***\SASInformationDeliveryPortal***version***\Configurable\wars\sas.portal\WEB-INF\web.xml.orig
SAS BI Web Services *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.biws\WEB-INF\web.xml.orig
SAS Preferences *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.preferences\WEB-INF\web.xml.orig
SAS Shared Applications *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.shared.apps\WEB-INF\web.xml.orig

Web Application	File Location
SAS Stored Process *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.storedprocess\WEB-INF\web.xml.orig
SAS Web Infrastructure Platform Permission Manager *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.permissions\WEB-INF\web.xml.orig
SAS Content Server*	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.scs\WEB-INF\web.xml.orig
SAS Authorization Services	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.authorization.services\WEB-INF\web.xml.orig
SAS Web Infrastructure Platform Client Access *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.access\WEB-INF\web.xml.orig
SAS Identity Services *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.identity.services\WEB-INF\web.xml.orig
SAS Web Administration Console *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.admin\WEB-INF\web.xml.orig
SAS Notification Template Editor *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.templateeditor\WEB-INF\web.xml.orig
SAS Principal Services *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.principal.services\WEB-INF\web.xml.orig
SAS Web Infrastructure Platform Services *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.services\WEB-INF\web.xml.orig
SAS SOAP Services *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.wip.soapservices\WEB-INF\web.xml.orig
SAS Workflow Web Service *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.workflow.webservice\WEB-INF\web.xml.orig
SAS Workflow *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.workflow\WEB-INF\web.xml.orig
SAS Shared Web Assets *	***SAS-installation-directory***\SASWebInfrastructurePlatform\9.4\Static\wars\sasweb\WEB-INF\web.xml
SAS Web Report Studio	***SAS-installation-directory***\SASWebReportStudio*version*\Configurable\wars\sas.webreportstudio\WEB-INF\web.vfabrictcsvr.xml.orig

Web Application	File Location
SAS Visual Analytics	***SAS-installation-directory***\SASVisualAnalyticsServices***version***\Configurable\wars\sas.lasr.authorization\WEB-IN F\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsServices***version***\Configurable\wars\sas.va.linkservices\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsServices***version***\Configurable\wars\sas.bitransportservices\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualDataBuilder***version***\Configurable\wars\sas.visualdatabuilder\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsHub***version***\Configurable\wars\sas.visualanalyticshub\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsAdministrator***version***\Configurable\wars\sas.adminclient\WEB-INF\web.xml.orig ***SAS-installation-directory***\SearchInterfacetoSASContent***version***\configurable\wars\sas.searchsas\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsExplorer***version***\Configurable\wars\sas.visualanalyticsexplorer\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsExplorer***version***\Configurable\wars\VAEForecastService\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsReportViewer***version***\Configurable\WEB-INF\web.xml.orig ***SAS-installation-directory***\SASVisualAnalyticsDesigner***version***\Configurable\WEB-INF\web.xml.orig

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

Note: SAS Studio uses the user preference time-out interval instead of a file. For more information, see "Setting the Start Up Preferences" in *SAS Studio: User's Guide*.

TIP It is recommended that you keep the SAS Visual Analytics time-out intervals in a synchronous manner.

Customize 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.
6. Enter the sign-out URL to which users should be redirected in the **Property Value** field.
7. Click **OK** to close the Define New Property window.
8. Click **OK** to close the SAS Application Infrastructure Properties window.
9. To enable these properties to take effect, restart SAS Web Application Server.

Starting with SAS 9.4M8, additional steps are required. 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.
6. Enter the sign-out URL to which users should be redirected in the **Property Value** field.
7. Click **OK** to close the Define New Property window.
8. Click **Add** to enter another property.
9. Enter **`ServiceUrl.Allowed`** in the **Property Name** field.
10. Enter the sign-out URL to which users should be redirected in the **Property Value** field. This value is identical to the **`Logoff.Url`** property value and is necessary to register the **`Logoff.Url`**.
11. Click **OK** to close the Define New Property window.
12. Click **OK** to close the SAS Application Infrastructure Properties window.
13. To enable the redirection after the logoff, add the following property to the ***`SAS-configuration-dir`*`/Levn/Web/WebAppServer1_1/sas_webapps/sas.svcs.logon.war/WEB-INF/classes/application.properties`** file:

    ```
    cas.logout.follow-service-redirects=true
    ```

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

Change 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 120.

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

Change 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:

   ***SAS-installation-directory*/SASWebInfrastructurePlatform/9.4/Static/wars/sas.svcs.logon/themes/default/css**

2. 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 "Rebuild the SAS Web Applications" on page 102.

4. Use the SAS Deployment Manager to redeploy the SAS Web Infrastructure Platform web application. See "Redeploy the SAS Web Applications" on page 107.

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 a hot fix or maintenance release.)

Configure 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 time-out** 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 longer than 10 minutes to ensure that the single sign-on session is extended if the user is actively using an application. By design, the request filter (TgtKeepAliveFilter) extends the single sign-on session only once every 10 minutes to cut down on the number of requests to SASLogon.

Specify 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=`***inverval-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. After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

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

Configure Guest Access

Overview

Starting with 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 71.
- 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. To invoke the SAS Visual Analytics viewer using guest access, specify the following URL: http://*hostname*/SASVisualAnalyticsHub/guest.jsp. To log on and view a single report using guest access, use a URL similar to the following (which has parameters to specify the name and location of the report): http://*hostname*/SASVisualAnalyticsViewer/VisualAnalyticsViewer_guest.jsp?reportName=*report-name*&reportPath=*report-path*&appSwitcherDisabled=true. For more information, see *SAS Visual Analytics: User's Guide*.
- 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 15, "Best Practices for Configuring Your Middle Tier," on page 213.
- 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 292.

Limit Content

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 use either of the following approaches to exclude content from guest access:
  - Where access is granted to the SASUSERS or PUBLIC group, add denials for the SAS Anonymous Web User.
  - Replace grants to the SASUSERS or PUBLIC group with grants to the Visual Analytics Users group and the SAS Services group.

  For information about permissions, see "Authorization Overview" 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 is 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 logon functionality).

Limit Functionality

Guest access functionality corresponds to the capabilities of the SAS Anonymous Web User.

- For the home page, the **`sas.home.allow.anonymous.user.personalization`** property ensures that guest access does not include individualized capabilities. For more information about the property, see "Configuration Properties for the Home Page" in *SAS Intelligence Platform: Web Application Administration Guide*.
- For SAS Visual Analytics, the Visual Analytics: Basic role provides an appropriate set of guest access capabilities. Do not permanently give the Personalization capability to the Visual Analytics: Basic role. Failure to conform to this guideline causes each user's experience to reflect the activities of the previous user.

CAUTION:

Any capabilities that the SASUSERS or PUBLIC group has can expand guest access functionality. This expansion of functionality can cause unintended results. If your deployment supports guest access, it is important to review the capabilities of the SASUSERS and PUBLIC groups. The SAS Anonymous Web User is an implicit member of those groups.

Enable or Disable Guest Access

Enable Guest Access during Installation

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

- Create a SAS Anonymous Web User (webanon). For more information about the webanon account and creating the account after the initial installation has finished, see "SAS Anonymous Web User" on page 372.
- Enable guest access for the SAS web application.

In addition, once the installation completes, in the SAS Management Console Configuration Manager, set the **`App.AllowGuest`** property to **`true`** on the **Search Interface to SAS Content** node. If the property does not exist, add it. For more information about the property, see "Configuration Properties for the Home Page" in *SAS Intelligence Platform: Web Application Administration Guide*.

If you enable guest access during installation, the home page, the web viewer, and transport service (SAS Visual Analytics App) allow users to connect as the guest user. Users can choose to sign in to those applications as a guest.

Note: For the home page and the web viewer, users can instead explicitly specify a guest access URL. Here is an example:

```
http://host/SASVisualAnalyticsHub/guest.jsp
```

For the exact URL, see the ***`SAS-configuration-directory`***`\Documents\Instructions.html` file on the middle-tier machine.

Enable Guest Access Post-Installation

If you need to configure guest access as a post-installation task, follow these steps:

1. If the webanon account does not already exist in your deployment, create that service identity. For more information, see "SAS Anonymous Web User" on page 372.
2. In the SAS Management Console Configuration Manager, set the **`App.AllowGuest`** property to **`true`** on the **Visual Analytics Hub**, **Visual Analytics Viewer**, **Visual Analytics Transport Service**, and **Search Interface to SAS Content** nodes. If the property does not exist, add it. For more information about the property, see "Configuration Properties for the Home Page" in *SAS Intelligence Platform: Web Application Administration Guide*.
3. Restart the SAS Web Application Server.

Disable Guest Access

To disable guest access for an individual application, set the application's **`App.AllowGuest`** property to **`false`**, and restart the SAS Web Application Server.

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

To disable guest access system-wide, set the **`App.DisallowGuestAccess`** property to **`true`** for SAS Logon Manager.

Customize the Home Page for Guests

TIP If guest access is enabled, it is a good practice to periodically access the home page as a guest in order to verify that only the intended resources and functionality are available to guests.

To customize the home page for guests, follow these steps:

1. Identify the changes that you want to make.

   TIP This step helps you minimize the period of time in which another user might sign in as the guest user and inadvertently affect the guest access configuration.

   a. Access the home page as yourself, and familiarize yourself with the available customizations. You can make changes such as the following:

      - Add, remove, or reorganize collections, shortcuts, and links.
      - Change application settings by clicking your name (in the upper right) and then selecting **Settings**.

      Note: These instructions are for the modern mode. For details or classic mode instructions, see the online Help.

   b. Access the home page as a guest, and examine the current configuration. Notice that if you click **SAS Anonymous Web User** in the banner, the **Settings** menu is not available.

      TIP This planning step helps you minimize the interval in which another user might sign in as guest and inadvertently affect the configuration.

2. Temporarily enable the SAS Anonymous Web User to modify the home page.

   a. Set the property sas.home.allow.anonymous.user.personalization to `true`.

   b. Restart the SAS Web Application Server.

3. Access the home page as a guest. Notice that if you click **SAS Anonymous Web User** in the banner, the **Settings** menu item is available. This is because anonymous user personalization is now enabled.

4. Make the changes that you identified in step 1.

5. Set sas.home.allow.anonymous.user.personalization to `false`. Restart the SAS Web Application Server.

6. Access the home page as guest.

   a. Verify that the results are as expected.

   b. Verify that you (as the webanon account) cannot make any further customizations.

Customize the Web Viewer for Guests

To customize the web viewer for guests, follow these steps:

1. Temporarily add the Personalization capability to the Visual Analytics: Basic role.

   a. Log on to SAS Management Console as someone who has user administration capabilities (for example, sasadm@saspw).

   b. On the **Plug-ins** tab, select **User Manager**.

c. In the right pane, right-click the **Visual Analytics: Basic** role, and select **Properties**.

d. On the **Capabilities** tab, expand the **Visual Analytics** node, and select the check box for the **Personalization** capability. Click **OK**.

2. Access the web viewer as a guest.

3. Immediately after your session is established, remove the Personalization capability from the **Visual Analytics: Basic** role.

   TIP Minimizing the period of time in which the Personalization capability is granted to the **Visual Analytics: Basic** role reduces the risk of another user inadvertently affecting the guest access configuration.

4. Change web viewer settings for the SAS Anonymous Web User as needed, and then sign out.

5. Access the web viewer as a guest.

   a. Verify that the results are as expected.

   b. Verify that you (as the webanon account) cannot make any further customizations.

Configure Middle-Tier Security Policies

The policies identified in the following table are configured with SAS Management Console. For more information, see "Set Global Properties for SAS Applications" on page 73.

Table 9.2 *Middle-Tier Security Policies*

Policy Name	Default Value	Description
Check for metadata updates	Check on navigation	This is a deprecated property. Do not change the value unless you are directed to by SAS Technical Support.
Profile refresh interval	600000	This is a deprecated property. Do not change the value unless you are directed to by SAS Technical Support.
Allow client password storage	Yes	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.
Log user off on time-out	No	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 **`No`**, 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 **`Yes`** and 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 **`Yes`** reproduces the behavior provided in SAS 9.3 and earlier releases.

Policy Name	Default Value	Description
Allow user sign-in from web sign-out page	Yes	Determines whether to display a **Sign In** 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.
Allow user sign-in from web time-out page	Yes	Determines whether to display a **Sign In** 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.
Display custom sign-in message	No	Determines whether to display a custom message or custom page on the standard sign-in page.
Display custom sign-out message	No	Determines whether to display a custom message or custom page on the standard sign-out successful page.
Display custom time-out message	No	Determines whether to display a custom message or custom page on the standard session timed out page.
Display sign-out security message	Yes	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.
Display time-out security message	Yes	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 "Configure the HTTP Session Time-Out Interval" on page 122.
Display failed sign-in hints	No	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 **No**, the system-generated exceptions and errors are still displayed. For example, if the system is quiesced or if the SAS Metadata Server is paused. If the value is **No**, the only message that is displayed for any user input failure is the invalid credentials message.
Enable autocomplete feature on sign-in page	No	Determines whether to use the autocomplete feature that is provided by the web browser on the sign-in page.
Allow clients to keep service sessions alive	Yes	Determines whether desktop client applications keep resources alive. If set to **No**, then resources time out in a similar manner to web applications. If set to **Yes**, then desktop client applications ping the server to keep the resources available.

Disable 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 have only one active session at a time. That is, a user can access multiple SAS web applications through a single sign-on session, but cannot have more than one single sign-on session. 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.

Note: The *sasadm* and *sastrust* accounts cannot be restricted.

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.

Note: Prior to SAS 9.4M4, you cannot disable concurrent sign-in sessions if you are running SAS Visual Analytics. For more details, contact SAS Technical Support.

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 window, 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.

Disable 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 is locked 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.

See Also

"Specify JVM Options" on page 44

Disable Cross-Frame Scripting

To disable the embedding of SAS Logon Manager in an iframe, follow these steps:

1. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\sas_webapps\sas.svcs.logon.war\WEB-INF\spring-configuration\filters.xml** file and locate the following block:

```
<property name="headersToSet">
    <props>
        <prop key="X-UA-Compatible"><![CDATA[IE=EmulateIE8]]></prop>
    </props>
</property>
```

   Replace the previous block with the following:

```
<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.

Enable the X-Frame-Options Header

To set the X-Frame-Options header value, follow these steps:

1. Log on to SAS Management Console.
2. On the **Plug-ins** tab, select **Application Management** ⇨ **Configuration Manager**.

3. Right-click **SAS Application Infrastructure**, and select **Properties**.
4. In the SAS Application Infrastructure Properties window, click the **Advanced** tab.
5. Click **Add** to define a new property.
6. Enter **`WebApp.XFrameOptions`** in the **Property Name** field and the desired header value in the **Property Value** field. Then, click **OK**.

To enable your changes to take effect, restart SAS Web Application Server.

Chapter 10

Administer the SAS Content Server

Overview

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 10.1 *SAS Content Server JVM Options*

JVM Option	Description	Applies To
`-Dsas.scs.svc.internal.url`	Specifies the internal URI of the SAS Content Server (for example, http://*host*:*port_number*). *Note:* Specify only the information for the primary SAS Web Server.	Central Authentication Service
`-Dsas.retry.internal.url`	Set to **`True`** when you have separate internal or external URLs.	Web Application Server
`-Dsas.web.html.cdps.use.internal.urls`	Set to **`True`** when you have separate internal or external URLs.	Web Application Server
`-Dsas.scs.cas.scheme`	Specifies **`http`** or **`https`**.	Central Authentication Service
`-Dsas.scs.cas.host`	Specifies the host name of the Central Authentication Service (CAS) server.	Central Authentication Service

JVM Option	Description	Applies To
`-Dsas.scs.cas.port`	Specifies the port number of the CAS server. *Note:* This value should be specified as a string. For SAS 9.4M1, if a value other than the default port is being used, the value must be preceded by a colon. For example, the default port for HTTP is 80 and you specify it as: -Dsas.scs.cas.port="". Specify a non-default port as the following: -Dsas.scs.cas.port=":8125". Beginning with SAS 9.4M2, both the default and non-default port should be specified as follows: Dsas.scs.cas.port=*port_number*.	Central Authentication Service
`-Dsas.scs.svc.host`	Specifies the host name of the reverse proxy or load balancer.	Web Application Server
`-Dsas.scs.svc.port`	Specifies the port number of the reverse proxy or load balancer. *Note:* This value should be specified as a string. For SAS 9.4M1, if a value other than the default port is being used, the value must be proceeded by a colon. For example, the default port for HTTPS is 443 and you specify it as: -Dsas.scs.cas.port="". Specify a non-default port as the following: -Dsas.scs.cas.port=":9230". Starting with SAS 9.4M2, both the default and non-default port should be specified as follows: Dsas.scs.cas.port=*port_number*.	Web Application Server
`-Dsas.scs.svc.scheme`	Specifies the scheme of the reverse proxy or load balancer.	Web Application Server

For information about how to update the Central Authentication Service (CAS) options, see "Configure the SAS Content Server to Use an Existing Customer Reverse Proxy " on page 159.

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, 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.

Move Content or Back 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 "Back Up the SAS Content Server" in *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 "WebDAVDump and WebDAVRestore Utilities" in *SAS Intelligence Platform: System Administration Guide*.

Filter Files by Extension and MIME Type

By default, any file type can be uploaded to SAS Content Server. Starting with SAS 9.4M2, you can now prevent certain file extensions and MIME types from being uploaded to the SAS Content Server via WebDAV. This offers an additional layer of security by preventing potentially malicious files from being made available to a wider, unsuspecting audience. Files such as .exe or .bat, which can be used to execute malicious code on a target system, are of primary concern; there might be others that you want to restrict. Similarly, MIME types like application/octet-stream or application/exe could indicate that the files being uploaded are suspect and should be filtered. There are a number of potential candidate extensions and MIME types. Decide what types of content you want to store.

To prevent certain file extension from being uploaded, follow these steps:

1. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer 1_1\sas_webapps\sas.svcs.scs.war\WEB-INF\config.xml** file.
2. Locate the following block:

```
<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.

Deploy 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 10.2 *Selecting Automatic Options or Manual Performance of Tasks*

Options Selected in SAS Deployment Wizard	Results and Instructions for Manual Tasks
SAS Web Server: Automated or Manual Configuration Option **Web Applications: Automatic Deployment** **Deploy web applications automatically** is selected	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.

Options Selected in SAS Deployment Wizard	Results and Instructions for Manual Tasks
`SAS Web Server: Automated or Manual Configuration Option` **`Web Applications: Automatic Deployment`** **Deploy web applications automatically** is not selected	SAS Web Server and SAS Web Application Servers are configured automatically. Instructions are provided on how to manually deploy SAS web applications, load content to the SAS Content Server, and adjust any web applications that store SBIP URLs.
Manually configure SAS Web Server, SAS Web Application Server, deploy the web applications, load the content to the SAS Content Server, and adjust any web applications that store SBIP URLs.	Instructions are provided on how to perform all tasks manually.

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*

Configuration of Web Application Server	Deployment of Web Applications	Load Content	Update Content	Adjust URLs
Automatic	Automatic	Automatic	Automatic	Automatic
Automatic	Manual	Manual	Manual	Manual
Manual	Manual	Manual	Manual	Manual

The following table shows the files associated with loading content to the SAS Content Server or updating content. The file name 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*

```
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] Error connecting to the metadata server: Access denied.
[GetObjectProperties]    Host: hostname.example.com
[GetObjectProperties]    Port: 8561
[GetObjectProperties]    User: sasadm@saspw
[GetObjectProperties]    m_mdFactory: com.sas.metadata.remote.MdFactoryImpl@74db2c
[GetObjectProperties] Error finding foundation repository: Encountered metadata exception.

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 1 *PWENCODE Procedure Example*

```
proc pwencode in='changeit' method=sas005;
run;
```

The SAS log shows the value to copy and paste into the script:

```
{SAS005}ADD8AB7108595A7D1A69190D78CDFE6145C1EB849CC7A43D
```

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:

  ***SAS-configuration-directory*\Lev1\Web\Utilities\manualLoadContent-*OrderNumber*.bat**

- On UNIX:

  ***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 the information about 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 file name always includes the date and timestamp. For Windows machines, the log file name 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**

Use the SAS Content Server Administration Console

Overview

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 "Use the SAS Web Administration Console " on page 89.

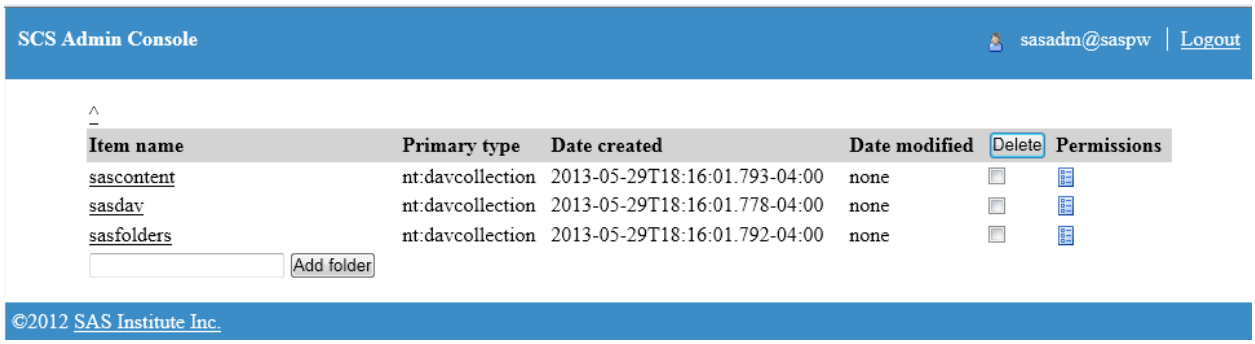
Log on to the console with an unrestricted user ID (for example, sasadm@saspw). In order to use the console, you must be logged on as an unrestricted user. This provides full administrator rights to use the console.

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 ^ / sasdav / Users breadcrumb trail, click **sasdav** to return to the sasdav 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**.

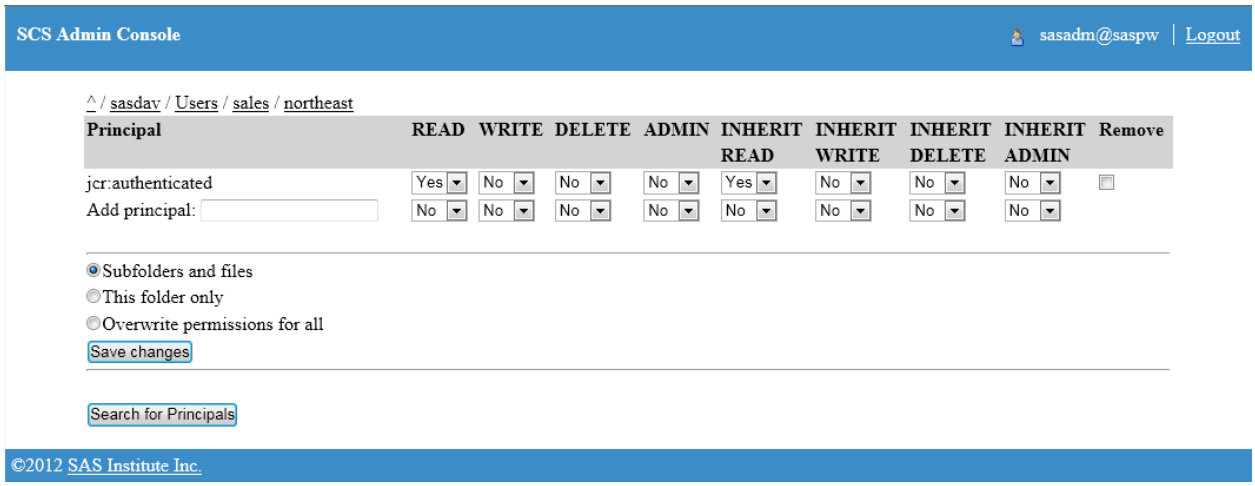
Note: You might see a principal named jcr:authenticated. This principal refers to any user who can log on to a SAS web application. By default, authenticated users have Read and Inherit Read permissions only.

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*



The following permissions are available for you to apply to objects:

Table 10.4 *Permissions for Objects*

Permissions	Purpose
Read	Allows the principal to read the object. For folders, this permission allows the principal to see the members of the folder.
Write	Allows the principal to write an object. For folders, this permission allows the principal to create new objects in a folder.
Delete	Allows the principal to delete the object.
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).

Permissions	Purpose
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*

Permissions for Folders	Results
Subfolders and files	Changed permissions are applied to subfolders and files that exist below the current folder.
This folder only	Changed permissions are applied to subfolders and files that exist in the current folder.
Overwrite permissions for all	Changed permissions are applied to all folders and files.

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*\Levn\Web\Utilities\DAVTree.bat**

  On UNIX, the utility command is **DAVTree.sh**.

  For more information about using DAVTree, see "Use the DAVTree Utility to Manage WebDAV Content " on page 383.

- 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.

Enable the Data Store

Overview

Beginning in SAS 9.4M2, 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.

An added benefit of the file data store is that there are no file size restrictions. Databases have limits on the maximum file size that is allowed. See the table below for those file sizes:

Table 10.6 *Maximum File Size*

Database	Limit
PostgreSQL	512MB
Oracle	4Gb*DB_BLOCK_SIZE (8Tb to 128Tb)
MySQL	1Gb
MSSQL	2Gb
DB2	1Gb

Configure 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*\Levn\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*\Levn\Web\WebAppServer\SASServer1_1\conf\wrapper.conf** file.

- On UNIX: Open the ***SAS-configuration-directory*/Levn/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.

Note: After you modify the wrapper.conf file for 9.4M7 Feb 16th 2022 and later, you need to rebuild the Windows service for each SAS Web Application server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

4. Restart all web application servers. The SAS Content Server now stores all files larger than the specified size (default is 1024 bytes) in the data store.

Use the Garbage Collection Utility

Once the data store is configured, you must 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 contains the status code, either 200(OK) or 500(Internal Server Error). In addition, the body of the response contains 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/. 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_number*/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_number/SASLogon/v1/tickets -D-
   ```

   Upon successful completion, the following headers are returned:

   ```
   201 Created
   Location: https://host:port_number/SASLogon/v1/tickets/
   TGT-18-umUeNL4yUkWHES2VdtKki5mFzatga43kNNCe3niguLWaUxl1aK-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'
http://host:port_number/SASLogon/v1/tickets/
TGT-3-1VhZYBorOfUdC0wYTCBU9n3yteYaO5wcuBy9Nvof9mkeDQ6PIg
-casST-2-zSX47dBu731WsTYExBZ5-casd72933
```

3. 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=*ticket* parameter:

```
curl -X POST http(s)://webserver-host:webserver-port/SASContentServer/admin/
collectDataStoreGarbage?ticket=ticket
```

Implement 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.

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 10.7 *Summary of WebDAV Folders on the SAS Content Server*

Folder	Americas Sales	Sales Executives
/sasdav/Catalog Sales	Read, Write	Read, Write
/sasdav/Field Sales	Read	Read, Write
/sasdav/Sales Execs	(none)	Read, Write

To create this sample configuration, follow these steps:

1. In SAS Management Console, define the users, groups, and log on credentials that need to access the SAS Content Server. When you define log on 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 10.8 *Example Users, Groups, and Logins*

Group Metadata Identities	User Metadata Identities	User ID	Authentication Domain
America Sales	salesusr1	salesusr1	DefaultAuth
Sales Executives	execusr1	execusr1	DefaultAuth
SAS Trusted User	sastrust	sastrust	DefaultAuth

   For example, the America Sales group contains a user named salesusr1 as a member, and salesusr1 has an associated logon 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 10.9 *WebDAV Permissions for /sasdav/Catalog Sales*

Group	Read	Write	Delete	Inherit Read	Inherit Write	Inherit Delete
Americas Sales	Yes	Yes	No	Yes	Yes	No
Sales Executives	Yes	Yes	No	Yes	Yes	No

Table 10.10 *WebDAV Permissions for /sasdav/Field Sales*

Group	Read	Write	Delete	Inherit Read	Inherit Write	Inherit Delete
Americas Sales	Yes	No	No	Yes	No	No
Sales Executives	Yes	Yes	No	Yes	Yes	No

Table 10.11 *WebDAV Permissions for /sasdav/Sales Execs*

Group	Read	Write	Delete	Inherit Read	Inherit Write	Inherit Delete
Americas Sales	No	No	No	No	No	No
Sales Executives	Yes	Yes	No	Yes	Yes	No

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. A common change that requires that you update the information is when changing from HTTP to HTTPS manually. (When HTTPS is configured with the SAS Deployment Wizard, the wizard sets all the connections automatically.)

Reconfigure 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
- SASBIPortlets*version* Local Services
- SASJSR168RemotePortlet*version* Local Services
- SASLASRAuthorization*version* Local Services
- SASPackageViewer*version* Local Services
- SASPortal*version* Local Services
- SASStoredProcess*version* Local Services
- SASStudioMidTier*version* Local Services
- SASVisualAnalyticsTransport*version* 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 manually reconfigured SAS Web Server from HTTP to HTTPS.
- 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.

Reconfigure 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.

Configure 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.

1. Change the following JVM options to point to the existing customer reverse proxy:

   ```
   -Dsas.scs.cas.scheme=protocol
   -Dsas.scs.cas.host=proxy.example.com
   -Dsas.scs.cas.port=port_number
   -Dsas.scs.svc.scheme=protocol
   -Dsas.scs.svc.host=proxy.example.com
   -Dsas.scs.svc.port=port_number
   ```

2. Change the **`-Dsas.scs.svc.internal.url=http://host:port_number`** JVM option to point to the internal URI of the SAS Web Server. If the option does not exist, add it.

Note: For installations that are not current with SAS 9.4M2 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 about how to set the options, see "Specify JVM Options" on page 44 . For a description of the options, see "Overview" on page 140.

Chapter 11

Administer the SAS BI Web Services

Overview

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.

See Also

- *SAS BI Web Services: Developer's Guide*
- *SAS Stored Processes: Developer's Guide*

Manage 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.

Configure 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 163. This property overrides its analogous common configuration property.

DynamicPromptsSupport
See "Enable dynamic prompts validation" on page 163. This property overrides its analogous common configuration property.

MaxSTPExecTime
See "SAS Stored Process timeout" on page 163. 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.

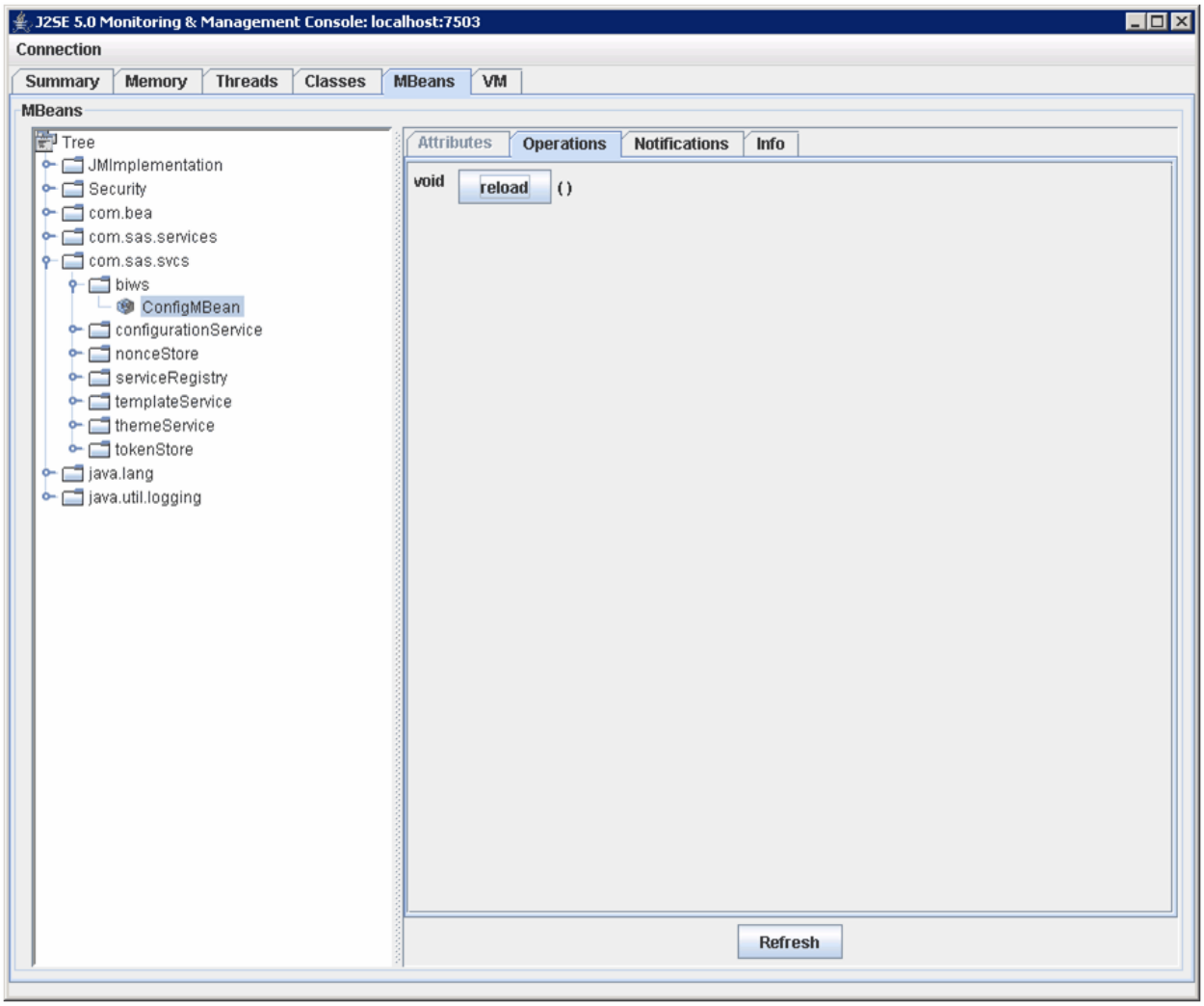
ValidateResponseWithSchema
Setting this property to true causes the resulting output that is created by the service execution to be validated against the service's schema. The default is false because this operation can be CPU intensive.

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 the WebServiceMaker web service when SAS clients are used because these clients authenticate by using one-time passwords.

Secure 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.

The following security constraints should be applied to the web.xml.orig deployment descriptor. This file is located in the *SASHOME* **\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.biws\WEB-INF** directory. Change the file by adding the security constraints as follows:

```
<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>
```

Rebuild and redeploy the SAS Web Infrastructure Platform web application with the SAS Deployment Manager.

For guidance, see "Web Authentication" on page 245.

Edit 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:

```
<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <!-- comment out or remove this line
      <url-pattern>/*</url-pattern>
    -->
    <!-- add the following two lines -->
    <url-pattern>/j_spring_cas_security_proxyreceptor</url-pattern>
    <url-pattern>/j_sprint_cas_security_check</url-pattern>
</filter-mapping>
```

Rebuild and redeploy the SAS Web Infrastructure Platform web application with the SAS Deployment Manager.

Use of TLS with BI WebServices

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.

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. See "Overview" on page 161.

If you are using web authentication and HTTPS, see information from "Support for TLS with Client Certificate Authentication" on page 294 for guidance.

Chapter 12

Administer SAS Web Application Themes

Overview

Introduction to SAS Web Application Themes

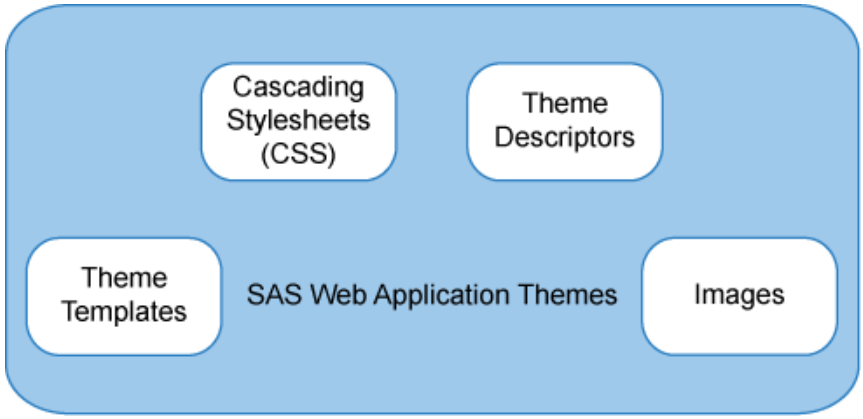
SAS Web Application Themes provide a way to define a consistent look and feel across SAS web applications. You can use themes to apply uniform visual customizations and company branding to all SAS web applications that support the theme infrastructure. A typical custom theme might include a banner with 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 "Change the Appearance of the Sign In Page" on page 127.

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.

Note: Previously, SAS Web Report Studio 3.1 used product-specific branding. Product-specific branding is not available for SAS Web Report Studio 4.4. Use themes to create branding in SAS Web Report Studio 4.4. A few properties for branding that existed in SAS WebReport Studio 3.1 are supported in SAS Web Report Studio 4.4. For information about these properties and usage, see "Customizing Report Styles for SAS Web Report Studio" in *SAS Intelligence Platform: Web Application Administration Guide*.

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 172.)
2. "Step 2: Create a Work Area for the Theme" (See page 173.)
3. "Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates" (See page 177.)
4. "Step 4: Rebuild SAS Web Application Themes" (See page 179.)
5. "Step 5: Deploy SAS Web Application Themes in Your Test Environment " (See page 180.)
6. "Step 6: Test the New Theme" (See page 180.)
7. "Step 7: Move the New Theme from Test to Production Environment" (See page 180.)
8. "Step 8: Assign the Default Theme" (See page 180.)

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:

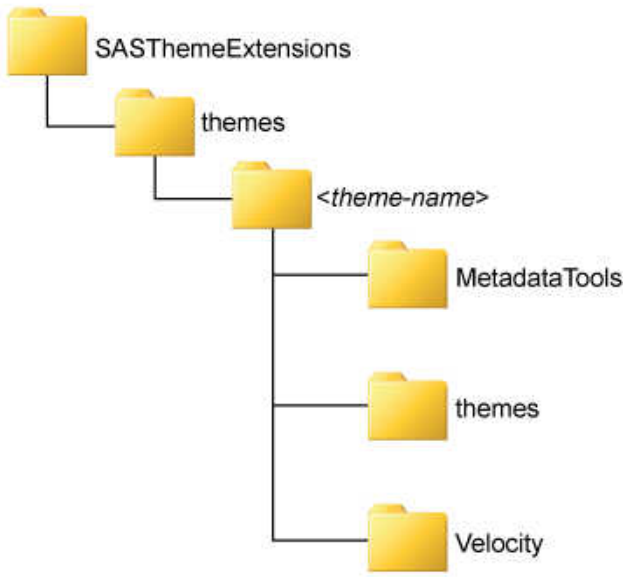
- for Windows: **NewTheme.bat *theme-name* true**
- for UNIX: **NewTheme.sh *theme-name* true**

To use the Color Palette option, the **true** parameter is required in the command.

Note: The theme name must not contain spaces.

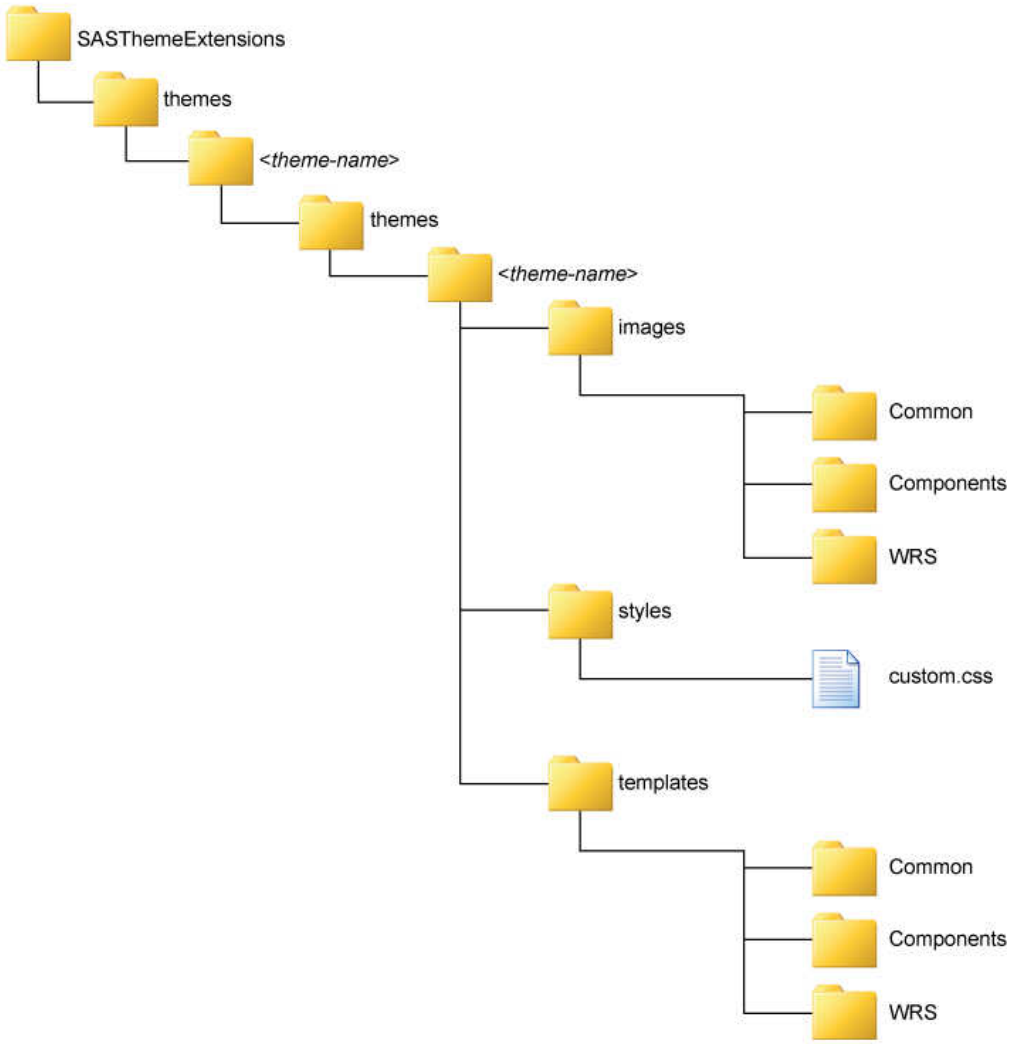
The following figure shows the *theme-name* directory, which is the root directory for theme resources. The ***theme-name*\MetadataTools** directory contains SAS programs for managing the theme. The **Velocity** directory contains several subdirectories with files.

Figure 12.2 Subdirectories within SASThemeExtensions Directory



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:

`\`*`theme-name`*`\themes\`*`theme-name`*`\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.

***theme-name*\themes*theme-name*\styles**
contains a cascading style sheet file named **custom.css** that can be used to define additional style elements for the theme. This file is empty when the work area is created.

***theme-name*\themes*theme-name*\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:

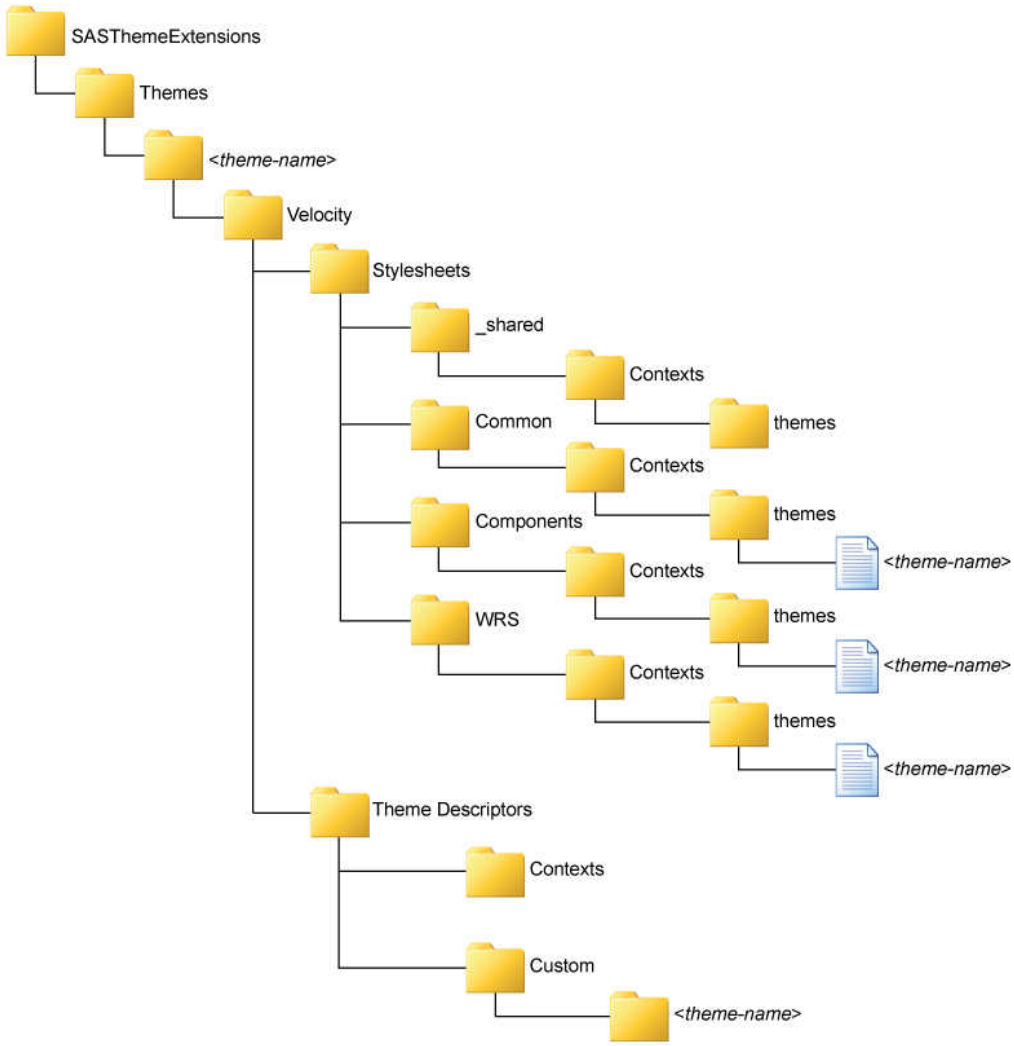
Common
contains theme templates for page elements that are commonly used in SAS web applications.

Components
contains theme templates for the collection of components that are shared by SAS web applications.

WRS
contains theme templates for elements in SAS Web Report Studio pages.

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**.

theme-name*\Velocity\Stylesheets\Components\contexts\themes*theme-name
contains a CSS template file named **components.***theme-name***.vtl** that is used to build style sheets for the collection of components that are shared by SAS web applications.

theme-name*\Velocity\Stylesheets\WRS\contexts\themes*theme-name
contains a CSS template file named **wrs.***theme-name***.vtl** that is used to build style sheets for SAS Web Report Studio.

***theme-name*\Velocity\ThemeDescriptors\contexts**
contains a context file named *theme-name***.themeDescriptor.vctxt** that defines context values that can be used in theme descriptor templates.

theme-name*\Velocity\ThemeDescriptors\contexts\custom*theme-name
contains theme descriptor template files for building the XML files that define the available collections of style sheets, theme templates, and images, including **ComponentsThemes.vtl**, **CustomThemes.vtl**, **SASThemes.vtl**, **SolutionsThemes.vtl**, and **WRSThemes.vtl**. The **SemanticThemes.vtl** file is added in SAS 9.3M2.

If you were to build the new theme at this point, it would be a fully functional duplicate of the Default theme.

Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates

Change 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 ***SAS-configuration-directory*\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\Components\html\Banner.html** show that the context key for the title text is **Banner_Title_Text_Color** and it displays its context value.

Callout #	Context Key	Context Value	Hex Color	Color Chip
1	Banner_Background_Color	${Color44}	#3b8ccb	
2	Banner_Title_Text_Color	${Color24}	#f5f7f9	
3	Banner_Title_Separator_Color	${Color35}	#a2beda	
4	Banner_SecondaryTitle_Text_Color	${Color24}	#f5f7f9	
5	Banner_UtilityBar_Background_Color	${Color49}	#003399	
6	Banner_UtilityBar_UserRole_Text_Color	${Color14}	#cccccc	
7	Banner_UtilityBar_Navigation_Links_Text_Color	${Color24}	#f5f7f9	
8	Banner_UtilityBar_Menu_Text_Color	${Color35}	#a2beda	
9	Banner_Divider_Background_Color	${Color9}	#e9e9e7	
10	Banner_Divider_Border_Bottom_Color	${Color15}	#c8c8c8	

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.

Change 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.

Change 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:

```
<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 103.

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 "Redeploy the SAS Web Applications" on page 107.

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 177. 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 171 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 179.
- Deploy SAS Web Application Themes to your web application server. See "Step 5: Deploy SAS Web Application Themes in Your Test Environment " on page 180.

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 181.
- Use the **`UpdateDefaultTheme.sas`** program. See "Assign the Default Theme with the UpdateDefaultTheme.sas Program" on page 181.

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.

Deploy New Theme in a High-Availability Middle-Tier Environment

Complete the following steps to deploy custom themes on multiple middle-tier nodes:

1. On the primary middle-tier node, define and deploy the custom theme. For more information, see "Steps for Defining and Deploying a New Theme " on page 171.
2. On each additional node, complete the following steps:
   a. Ensure that all of the required components on the primary middle-tier machine are running: SAS Web Infrastructure Platform Data Server, SAS Metadata Server, SAS Web Server, SAS Cache Locator, SAS JMS Broker, SAS Web Application Server instances, and SAS Deployment Agent.
   b. For each middle-tier node (horizontal cluster node), stop all SAS sessions, daemons, spawners, servers, and agents.
   c. Start the SAS Deployment Agent on the middle-tier node.
   d. From the SAS Deployment Manager, follow these steps:

i. On the Select SAS Deployment Manager Task page, under **Administration Tasks**, click **Update Existing Configuration**, and then click **Next**.

ii. On the Select Configuration Directory/Level page, specify the configuration directory and the level (for example, **Lev1**), and then click **Next**.

iii. On the Specify Connection Information page, enter the user ID and password for an unrestricted administrative user, and then click **Next**.

iv. On the Summary page, click **Start**.

3. If you made any manual customizations prior to deploying the custom themes, such as configuring HTTPS or setting up an external reverse proxy, check to ensure that the changes were preserved. If any of your manual changes were reverted, you will have to manually add them back.

4. If you have multiple SAS Web Servers in your high-availability environment, check the **SAS-configuration-directory\Levn\Web\WebServer\conf\sas.conf** files on each server. Verify that the ProxyPass statements are defined for the custom theme.

   For example, every sas.conf file should have entries similar to the following:

```
ProxyPass /SASTheme_Custom_Theme balancer://hostname_Cluster1/
     SASTheme_Custom_Theme
ProxyPassReverse /SASTheme_Custom_Theme balancer://hostname_Cluster1/
     SASTheme_Custom_Theme
```

Delete 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.

Migrate 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 173.

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.

Migrate 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.

Migrate 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.

Migrate 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.

Migrate 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.

Make More Fonts Available

When a report or exploration is printed to a PDF file, font substitution occurs for fonts that are unavailable to the transport service that generates the PDF. Starting in May 2019, you can make more fonts available to a service by adding your own custom fonts. The **Manage Fonts** capability can be assigned to entities who are responsible for adding, deleting, and updating fonts. By default, the *Fonts Administrator* role is assigned this capability. The SAS Administrator account, by default, is assigned this role.

For instructions for working with roles and capabilities, see the *SAS Management Console: Guide to Users and Permissions*.

To make more fonts available, complete the following steps:

1. Copy the fonts to the default directory for custom fonts, **`SAS_SYSTEM_FONTS_LOCATION/custom`**.

   Note: The font file must be installed on the middle-tier machine that hosts the transport service. If you have a clustered middle tier, the font file must be installed on every middle-tier machine.

   Note: As an administrator, you must make sure that you have the license to distribute the fonts to your users.

2. In SAS Management Console, override the default font location by defining the **`Fonts.Custom.Location.Override`** property.

   The following task defines the **`Fonts.Custom.Location.Override`** property so that all applications can access it.

   a. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager**.

   b. Right-click **SAS Application Infrastructure** and select **Properties**.

   c. Click the **Advanced** tab.

   d. Click **Add**.

   e. In the **Property Name** field, enter **`Fonts.Custom.Location.Override`**. In the **Property Value** field, enter the custom folder location. b

   f. Click **OK** to close the Define New Property dialog box.

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

   h. To enable the new property to take effect, restart SAS Web Application Server.

3. As a member of the *Fonts Administrator* role, run the Refresh Rest API: **SERVER:HOST/SASEnvironmentMgrMidTier/appResources/fonts/refresh**.

Note: If you change or delete a font, run the Refresh REST API to update the available fonts.

Note: In order to see the new fonts, refresh the application where the font is being used or viewed. You can refresh the browser where the application is open or close the application and reopen it.

Note: SAS Output Delivery System (ODS) destinations, such as stored processes, register fonts using Fontreg.

Chapter 13

Administer the Search Facility

Overview

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.

Overview of Search Index Providers

SAS Information Retrieval Studio

Overview

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.

Configure for TLS

Beginning with the October 2014 release for SAS 9.4 and until SAS 9.4M2, 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*\Levn\Web\Applications\SearchInterfacetoSASContent\url_list.txt** file.
2. Locate the HTTP URL. The file contents 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=
   {SAS005}ADD8AB7108595A7D1A69190D78CDFE6145C1EB849CC7A43D
   ```

   Note: The previous line must be entered on one line. It is shown on more than one line for display purposes only.

   Note: The url_list.txt file should specify the host name of the SAS Web Server. If the SAS Web Server is on the same machine as the SAS middle-tier software, then url_list.txt should specify an internal host. Otherwise, if SAS Web Server is on a separate machine, then url_list.txt should specify an external host.
3. Change **http** to **https**.
4. Change the port from **:80** to **:443**.
5. 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.

Specify 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:

   App.AllowGuest
   : specifies whether guest access is enabled. The default is **`false`**. To allow searching from applications that are configured for guest access, add this property and specify **`true`**.

   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 14

Manage Devices

Overview

SAS Visual Analytics App (previously called SAS Mobile BI) is a free app that enables mobile device users to view and interact with SAS Visual Analytics reports. The app supports all charts, graphs, gauges, tables, and other report objects from SAS Visual Analytics.

The app also enables mobile device users to view SAS Web Report Studio reports. See "View SAS Web Report Studio Reports on Mobile Devices" on page 210.

SAS also provides a free software development kit (SDK) that enables organizations to create mobile apps that include SAS Visual Analytics content. See "Mobile Software Development Kits" on page 211.

Supported Devices

Supported mobile devices include the following:

- Apple iPads and iPhones
- Android tablets and smartphones
- Windows 10 tablets and computers

How Mobile Content Is Protected

As an administrator, you can control how a mobile device running SAS Visual Analytics App can access reports and data located on a SAS Visual Analytics server. You can use features, capabilities, and properties (alone or in combination) to control access to the server data and reports from the app.

Protections for mobile content include the following:

- Users must authenticate in order to establish a connection to the server.

  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 245.

- The user ID and password used for authentication are stored on the mobile device by SAS Visual Analytics App. The app uses application programming interfaces (APIs) provided by the mobile device's operating system to store and retrieve this information:
  - Android uses the AccountManager API.
  - iOS uses the Apple Keychain APIs.
  - Windows uses the Microsoft Credentials Locker.
- SAS metadata security is enforced on all reports.
- Device access to the server can be managed by exclusion or inclusion. See "Enable or Prevent Access by Using the Allowlist and Denylist" on page 197.
- Require SAS Visual Analytics App users to lock the app with a passcode. See "Lock SAS Visual Analytics App with a Passcode " on page 201.
- To minimize access by users with revoked credentials, require server credentials after a user has not logged on for a specified period of time. See "Use the Time-out Setting to Prevent Access" on page 202.
- To minimize persistence of mobile data, specify that a user must maintain a network connection to the server while viewing a report in SAS Visual Analytics App. See "Prevent Report Data from Being Cached on the Device" on page 203.
- Functionality in SAS Visual Analytics App can be limited per user. Functionality includes whether a user can subscribe to and view reports, share links to reports by using email, add or view comments, and see and use the Favorites or Recent views. See "Capabilities for End Users" on page 204.
- 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 "Configure SAS Web Server Manually for HTTPS" on page 312.

Access and Use SAS Visual Analytics App

About SAS Visual Analytics App

SAS Visual Analytics App enables you to view and interact with SAS Visual Analytics reports on your mobile device. You can also share observations with others while on the go. The SAS Visual Analytics App (previously called SAS Mobile BI) supports all charts and graphs that are available in SAS Visual Analytics.

This free mobile app is supported on Apple iPhones and iPads; Android smartphones and tablets; and PCs and tablets running Microsoft Windows.

Download SAS Visual Analytics App

You can download SAS Visual Analytics App (previously called SAS Mobile BI) for free from the following locations:

- Apple App store
- Google Play
- Microsoft Windows store

View Sample Reports

When you download the app for the first time, the Welcome screen gives you the option to view sample reports without setting up the app. This providers an effortless way to try the app and experience how SAS Visual Analytics reports work on your device. You can also download additional sample reports.

About Searching for Reports

In the current release of SAS Visual Analytics App, the search feature within SAS Visual Analytics Apps does not discover any reports unless Search SAS is installed on the server. However, if SAS Visual Analytics Suite is installed, Search SAS is included in that installation and is available for use.

Access the Help

All of SAS Visual Analytics Apps provide online Help. To access the Help and other resources, see the SAS Visual Analytics App product documentation page on the SAS support site.

Be sure to view the Help for the platform and version of the app that you are using.

Prerequisites for Managing Mobile Devices

Assign the *Visual Analytics: Administration* role to administrators. This role grants the privileges that are needed to manage mobile devices.

Enable or Prevent Access by Using the Allowlist and Denylist

Overview

Note: Allowlist and *Denylist* are the terms used throughout this documentation. However, the SAS Visual Analytics App still displays *Whitelist* and *Blacklist* in its interface.

The *allowlist* manages the devices that can access servers by using SAS Visual Analytics App. A device must be on the allowlist in order to use SAS Visual Analytics App on your network. The allowlist affects devices, not users. If a device is lost, a SAS administrator can remove the device from the allowlist and prevent access to the reports and data.

The *denylist* manages the devices that cannot access servers by using SAS Visual Analytics App. All devices can use SAS Visual Analytics App on your network except those that are on the denylist. The denylist affects devices, not users. If a device is lost, a SAS administrator can add the device to the denylist and prevent access to the reports and data.

The following tasks should be performed using SAS Visual Analytics Administrator 7.5. For information about administering SAS Visual Analytics 7.5, see "Introduction" in *Using SAS Environment Manager Administration*.

Considerations

Here are the key points for using the denylist and allowlist:

- You can manage devices either by exclusion or by inclusion.
  - If you manage by exclusion, all devices can access servers through SAS Visual Analytics App, except those that are on the denylist. A denylist is a list of mobile devices that are not authorized to use SAS Visual Analytics App.
  - If you manage by inclusion, only devices that are on the allowlist can access servers through SAS Visual Analytics App. An allowlist is a list of mobile devices that are authorized to use SAS Visual Analytics App.

  Note: The denylist and allowlist that manage mobile devices are different from the security filters that are documented in "Allowlist of Websites and Methods Allowed to Link to SAS Web Applications" on page 364.
- A deployment enforces only one list (either the denylist or the allowlist) at a time.
- In a new deployment, the denylist is enforced and contains no items. Therefore, all devices can access servers through SAS Visual Analytics App.
- You can modify both lists. Making changes to a list that is not currently enforced can help accommodate a future change.
- The denylist and allowlist affect devices, not users. To manage what a particular user can see or do in SAS Visual Analytics App, use permissions and capabilities.

About the Mobile Devices Tab

Here are some details about the **Mobile Devices** tab in SAS Visual Analytics Administrator 7.4:

- 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 denylist, adding a device to the allowlist has no immediate effect.
- 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.
- 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 Visual Analytics App 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.
- When you right-click on a device in the denylist or allowlist, you can choose either a move action or a remove action. In terms of immediate effect, there is no difference between these two actions.

Add Devices by User ID

The easiest way to add a device to the allowlist or denylist is to add a device that has already connected (or attempted to connect) to the server. When the attempt is made, the **Logon History** tab logs the device owner's user ID, device ID, device type, and other information. You can sort the **User ID** column to locate the user ID of the person whom you want to add.

Restricting and enabling devices by user ID is a best practice because users can have more than one device. By identifying the user ID, you can be sure to add all devices used by that person.

TIP The only way to add a device running Windows 10 is by user ID.

About Device IDs

A device might appear multiple times in the denylist or allowlist if a different user ID attempts to log on with a device that has already been captured. 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 comes from an existing source (existing device ID and new user ID)
- a connection attempt that is accompanied by a device change (such as a new operating system version or application version)

Navigate to the Mobile Devices Page

Note: This page is available only if you are a member of the SAS Administrators group.

To manage devices that use SAS Visual Analytics App, select **Tools** ⇨ **Manage Devices** from the main menu in SAS Visual Analytics Administrator 7.4 . The administrator interface is available from http://*hostname.example.com*/SASVisualAnalyticsAdministrator.

Add a Device to a List from Logon History

TIP This option is disabled if the ID already exists on the respective list.

You can add a device that has already connected (or attempted to connect) to the denylist or allowlist by completing the following steps:

1. On the **Mobile Devices** tab, click the **Logon History** tab.
2. Right-click the device that you want to add and select the list to which you want to add the device.
3. In the Add Device window, click **Yes**.

Add One or More Devices to the Denylist or Allowlist

1. On the **Mobile Devices** tab, click the **Blacklist** or **Whitelist** tab, depending on which list you want to add devices.
2. To add one or more devices, click **+**.
3. In the Add to Blacklist or Add to Whitelist window, enter the device ID in the field. 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**.

Move One or More Devices between Lists

You can move devices from one list to the other (for example, from the denylist to the allowlist).

1. On the **Mobile Devices** tab, click the **Blacklist** or **Whitelist** tab, depending on which list you want to add devices.
2. Select one or more devices that you want to move, and click .
3. In the Move devices window, click **Yes**.

Remove One or More Devices from a List

1. On the **Mobile Devices** tab, click the tab that corresponds to the list from which you want to remove a device.

2. Select one or more devices that you want to remove, and click ✕.
3. In the Confirm Remove window, click **Yes**.

View Logon Event Information

1. On the **Mobile Devices** tab, click the **Logon History** tab.
2. View the device logon event information, including status. The **Status** column does not indicate the current status of a device connection.

TIP Use the **Filter by** drop-down list to filter the information about the tab.

View Previous Logon Events

1. On the **Mobile Devices** tab, click the **Logon History** tab.
2. To view records that were captured from devices on a prior application version or operating system version, select the **Include device history** check box.

Determine Which List Is Enforced

There are several ways to determine whether the denylist or allowlist is being enforced.

- In the toolbar at the top of the **Mobile Devices** tab, the **Enforced** drop-down list indicates which list is enforced.
- The list that is not being enforced displays the following message above the **Device ID** table: "⚠ This list is not currently being enforced.".
- Verify the current configuration in SAS Management Console. The denylist is enforced unless the viewerservices.enable.whitelist.support property is set to `true`. For more information, see "Configuration Properties: Transport Services" on page 206.

Determine When a Device Was Added to a List

1. On the **Blacklist** or **Whitelist** tab, right-click on the device, and select **Copy Device ID**.
2. On the **Management History** tab, select **Device ID** from the **Filter** drop-down list.
3. Click in the text field, and enter Ctrl-V from the keyboard. (You cannot perform the paste action from the pop-up menu.)
4. Click **Apply**.

TIP You can also determine the logon history for a device by pasting a device ID into the **Device ID** filter on the **Logon History** tab.

Change How Devices Are Managed

CAUTION:

These are deployment-level instructions that affect all access to SAS Visual Analytics App.

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 allowlist, the allowlist should contain all eligible devices. The denylist is ignored.
   - If you enforce the denylist, the denylist should contain all excluded devices. The allowlist 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**.

Lock SAS Visual Analytics App with a Passcode

Overview

The passcode feature locks SAS Visual Analytics App. This feature is separate from and in addition to the passcode feature that is provided by mobile devices. There are two types of app passcodes: required and optional.

A *required passcode* is a passcode that is required by the server. When the app first connects to the affected server, the server forces the app to require that the app user create a passcode. Then, whenever the app user opens the app or views a report that is associated with that server, the user must enter the passcode.

Note: By using an additional capability, the SAS administrator can exempt app users from using a passcode. By using a combination of two capabilities, all mobile devices that access the server must use a passcode except for those separately exempted.

An *optional passcode* is a passcode that the app user can choose to use to lock the app. The passcode is not required to access the server. The app user can disable the passcode at any time.

Considerations

Here are some key points to remember when working with passcodes:

- The passcode should be known only to the app user. If the app user loses the mobile device, no one else should be able to guess the passcode and use it to open the app.
- The passcode has a time-out feature. The SAS administrator can customize the passcodeTimeoutMinutes setting to configure this feature. This setting specifies, in minutes, how long a user must wait before reentering his or her passcode in SAS Visual Analytics App. The default is 15.

  If the app user (or another person) provides an incorrect passcode a specific number of times (passcodeAttempts), the app locks itself for a length of time (passcodeTimeoutMinutes). The app user can enter the passcode again after the time-out expires.

- The passcode has a lock-out feature. The SAS administrator can customize the passcodeAttempts setting to configure this feature. The setting limits the number of sequential, failed attempts to enter a passcode for SAS Visual Analytics App. The default is 5.

  If a user reaches the specified limit (passcodeAttempts), the user is timed out of the app for 15 minutes (or the value set for passcodeTimeoutMinutes). After the time-out interval, the user can make one more attempt to enter his or her passcode. If the password fails again, all custom content (data, reports, settings, and connection information) is removed from the mobile device. The app is reset to its default settings.
- If the app user forgets the passcode, the app user must delete and re-install the app on the device. Doing so deletes the reports and data.

For information about how app users set a passcode, see the SAS Visual Analytics App Help. See "Access the Help" on page 196.

Enable the Required Passcode

To set a required passcode, the SAS Visual Analytics administrator assigns users or groups to a role that has the **Require Passcode on Mobile Devices** capability.

Customize the Passcode Constraints

To customize the passcode behavior, use the following Transport Service properties:

- viewerservices.passcode.attempts
- viewerservices.passcode.timeout

See "How to Set Configuration Properties" on page 206.

Use the Time-out Setting to Prevent Access

Overview

If a user has been offline for a specified number of days, he or she must sign in to the server used by SAS Visual Analytics App. For example, if the user attempts to browse reports on the server or open a report in the report viewer, the app requires the user to enter the password for the requested server connection. If the user fails to sign in, then the app no longer downloads reports, updates subscribed reports, or opens reports for viewing.

This feature is useful when the device is missing, and it also provides security when the employee leaves the organization but keeps the device. The denylist and allowlist features require that the device must access the server before the list can look up the device to deny or permit access. The offline access time-out feature denies access by checking the employee's credentials, which the IT organization revokes when the employee leaves the organization.

Enable the Offline Access Time-Out Feature

To enforce a time limit for offline access, the SAS Visual Analytics administrator assigns users or groups to a role that has the **Limit Duration of Offline Access** capability.

Adjust the Time-Out Interval

To adjust the time limit, set the Transport Service viewerservices.offline.limit.days property. The default is 15 days. See "How to Set Configuration Properties" on page 206.

Prevent Report Data from Being Cached on the Device

Overview

When a user subscribes to a report, it appears in the **Subscriptions** view of SAS Visual Analytics App. However, depending on the security assigned to the user ID, the report data might not exist on the mobile device. Report data can be local or remote:

- *Local data* is stored on the mobile device.
- *Remote data* exists on the mobile device only while the report is open and the device is connected to a Wi-Fi or cellular network. If a report uses remote data, the report tile in the Subscriptions view displays a cloud icon.

How Cache Report Data Works

Each time a user opens a cached report, the app connects to the server. The Prepare Data notification is displayed while the data is downloaded. The report opens when the data is available on the mobile device. The data is available only while the user views the report.

After the user closes the report, the data is removed from the device. The thumbnail image on the report tile no longer appears. If the user is not connected to a network and tries to open the report, it does not open.

This feature affects the user ID that is used to access the server. When the user accesses the server via SAS Visual Analytics App using that user ID, all reports on that server use the caching report data feature.

Prevent Mobile Devices from Storing Report Data

By default, all authenticated users' mobile devices can cache report data. To prevent offline access to mobile data on a server, the SAS Visual Analytics administrator assigns users or groups to a role that has the **Purge Mobile Report Data** capability.

Limit Functionality in the App

You can limit some functionality that is available in the app by changing capability assignments for a role. Then, a user or group of users are given membership to that role. For a list of the mobile-specific capabilities, see "Capabilities for End Users" on page 204.

Note: Capabilities can be set on a per user per server basis. This means, for example, that a user ID might be enabled to add comments on one server, but not on another.

Capabilities for SAS Visual Analytics App

For instructions for working with roles and capabilities, see the *SAS Management Console: Guide to Users and Permissions*.

Predefined Roles

The following predefined roles are relevant for administering mobile devices:

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: Prior to SAS Visual Analytics 7.2, SAS Visual Analytics App 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 for mobile device users:

Visual Analytics *version*

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.

Export Data
enables users to export data for an object.

Note: The Export Data feature is supported only on Windows 10 for SAS Visual Analytics App.

Export or Print as PDF
enables users to export the report as a PDF and to print the PDF.

Email
enables users to send a link to a report via email.

Personalization
enables users to see the Favorites view, add and remove reports from their Favorites view, see recently viewed reports in a historical view, and see alerts.

Visual Analytics Transport Service *version*

Limit Duration of Offline Access

forces a user to sign in to the server used by SAS Visual Analytics App after a user has been offline for a specified number of days. See "Use the Time-out Setting to Prevent Access" on page 202.

Note: Because unrestricted users always have all capabilities, their offline access is always limited. In general, you should not use an unrestricted identity (for example, sasadm@saspw) to view reports.

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. See "Prevent Report Data from Being Cached on the Device" on page 203.

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 Visual Analytics App. For users who do not have this capability, an application passcode is not required. See "Lock SAS Visual Analytics App with a Passcode " on page 201.

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 206 and viewerservices.passcode.timeout on page 207.

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**

for SAS Visual Analytics 7.4 and earlier versions, 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**

for SAS Visual Analytics 7.4 and earlier versions, 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.

See Also

For capabilities information for SAS Visual Analytics 7.5, see "Capabilities for SAS Visual Analytics App" on page 204.

Configuration Properties: Transport Services

How to Set Configuration Properties

Note: Some transport service properties affect multiple components, such as SAS Visual Analytics App, HTML5, and SAS Office Analytics (SAS Enterprise Guide, SAS Add-In for Microsoft Office, and SAS Web Parts for Microsoft SharePoint).

To set configuration, follow these steps:

1. On the **Plug-ins** tab in SAS Management Console, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Visual Analytics** ***version*** ⇨ **Visual Analytics Services** ***version***.
2. Right-click **Visual Analytics Transport Service** ***version***, and select **Properties**.
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.
4. Restart SAS Web Application Server.

Reference for Selected Properties for SAS Visual Analytics App

viewerservices.enable.whitelist.support
: controls which approach is used to manage mobile devices. Valid values are:

  false — causes the denylist to be enforced and the allowlist to be ignored. With this setting, all mobile devices can use SAS Visual Analytics App except for those devices that are on the denylist. This is the default.

  true — causes the allowlist to be enforced and the denylist to be ignored. With this setting, only mobile devices that are on the allowlist can use SAS Visual Analytics App.

  For more information, see "Enable or Prevent Access by Using the Allowlist and Denylist" on page 197.

  CAUTION:
  Enabling the allowlist can disrupt existing users. Make sure that all valid devices are on the allowlist 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 "Add One or More Devices to the Denylist or Allowlist" on page 199.

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 Require Passcode on Mobile Devices capability. See "Lock SAS Visual Analytics App with a Passcode " on page 201.

viewerservices.passcode.timeout
specifies, in minutes, how frequently a user must re-enter his or her passcode. The default is `15`. See "Lock SAS Visual Analytics App with a Passcode " on page 201.

viewerservices.offline.limit.days
specifies, in days, how long a user can be offline before he or she must re-enter server credentials. The default is `15`. See "Use the Time-out Setting to Prevent Access" on page 202.

Reference for Selected Properties for All Transport Services Clients

Transport Services clients include SAS Visual Analytics App, HTML5, SAS Office Analytics (SAS Enterprise Guide, SAS Add-In for Microsoft Office, and SAS Web Parts for Microsoft SharePoint), and supports printing of reports.

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.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 to include in the results 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.

CAUTION:
Modifying the limits on the number of cells can cause the device to become unstable

viewerservices.image.default.max.bytes
sets the maximum size of images (PNG, BMP, JPEG, or GIF) that can be rendered in a report. 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`.

viewerservices.lasr.socketTimeout.milliseconds.interactions
sets the maximum wait time for attempts to contact SAS LASR Analytic Server. This property is applicable to live requests for tasks such as filtering, brushing, and drilling. The default is `300,000 milliseconds` (5 minutes), which is sufficient for most environments. If connections to the 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 report after contacting SAS LASR Analytic Server. The default is **300,000 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.validate.schema.write
: enables XML schema validation when reports are rendered. 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. 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. Also, this property checks for schema validation errors when reports are created. 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.

Printing.Footer.Content.Formatted
: enables standard footer text to be added to the bottom of all printed reports. The footer definition can include basic formatting options such as font selection and size. It can also include attributes such as bold, italic, or underline. Footers cannot contain reports. For more information, see "Supporting Footers in Printed Reports" in *SAS Visual Analytics: Administration Guide*.

Modify the Value Used for Resizing Images in the Middle Tier

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 rendered in a report. 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. The default value is sufficient for most environments. If this property is set to **0**, images are not resized before they are delivered.

You should modify this value only when you want to increase or decrease the number of image bytes that can be delivered. Modify the value for this property cautiously because it impacts the download time and memory.

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.

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 Visual Analytics App

For more troubleshooting information about SAS Visual Analytics App, see the SAS Visual Analytics App Help. See "Access the Help" on page 196.

Issue: A user cannot open reports in an offline device.

Explanation:

- The user ID might be required to use remote report data.
- The user ID might be affected by the offline-access time-out.

Resolution:

- If the user ID is subject to the remote report data authorization capability, make sure the user understands that he or she must be connected to a network while viewing the report. For more information, see "Purge Mobile Report Data" on page 205.
- If the user ID is subject to the offline-access time-out authorization capability, make sure the user can log on to the server connection in SAS Visual Analytics App. See "Use the Time-out Setting to Prevent Access" on page 202.

Issue: A user is prompted for an application passcode.

Explanation: The user is required to secure SAS Visual Analytics App with a passcode.

Resolution:

- To learn how to create a required passcode in SAS Visual Analytics App, see the SAS Visual Analytics App Help. See "Access the Help" on page 196.
- If you do not want to force the user to use a passcode, 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 205.

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 denylist or the allowlist) is in use.

View SAS Web Report Studio Reports on Mobile Devices

Getting Started

Important: Starting in May 2019, SAS Web Report Studio reports can no longer be viewed on mobile devices.

Mobile reporting for SAS Web Report Studio enables users to view certain types of relational reports in SAS Visual Analytics App. Supported reports are displayed in the native format of the device.

Note: When displaying SAS Web Report Studio reports on mobile devices, certain formats are not supported. Therefore, these reports are not displayed correctly.

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 "Access and Use SAS Visual Analytics App" on page 196.
- In the initial configuration, only an unrestricted user (such as sasadm@saspw) can denylist devices. For more information, see "Enable or Prevent Access by Using the Allowlist and Denylist" on page 197.

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

Mobile Software Development Kits

The Software Development Kits (SDKs), SAS SDK for Android or SAS SDK for iOS, enable your mobile apps to include SAS Visual Analytics content. You can preconfigure, customize, and manage the app experience by doing the following:

- Preconfigure your apps to include the server connections and report subscriptions so that your users do not have to perform these tasks.
- Substituting your organization's name and branding in SAS Visual Analytics App.

- Displaying SAS Visual Analytics reports in a custom-designed app.
- Integrating the mobile app with your mobile device management (MDM) service.

Your customized apps can connect to SAS Visual Analytics servers and can be managed by your organization's SAS administrators. The SAS SDK is free and available for iOS and Android operating systems. It can be downloaded from https://developer.sas.com.

Chapter 15

Best Practices for Configuring Your Middle Tier

Sample Middle-Tier Deployment Scenarios

Overview

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 SAS Workspace Servers, SAS Pooled Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers.

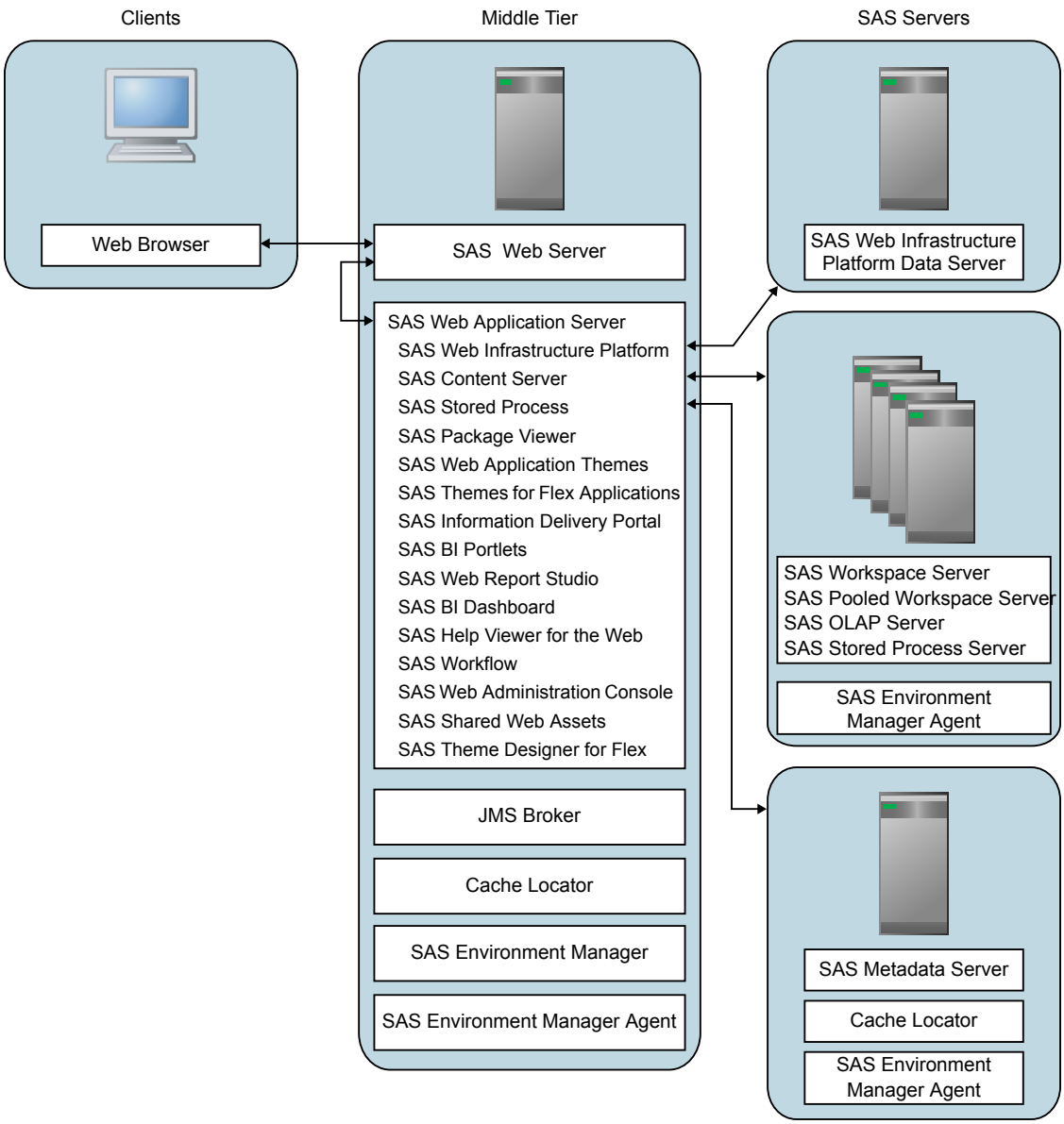
Scenario 1: Web Applications Deployed in a Single Web Application Server

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 15.1 *Scenario 1: Middle-Tier on a Single System*



Here are the advantages and disadvantages of this topology:

Table 15.1 *Scenario 1 Advantages and Disadvantages*

Topic	Advantages	Disadvantages
Security	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.	Adding firewalls to the network is a good next step.
Performance	SAS Web Server is automatically configured to cache static content.	This topology does not support hundreds of concurrent users.
Scalability	There are no advantages in this scenario, but the topology provides an upward path to clustering web application servers.	This topology does not support hundreds of concurrent users.
Availability	None	This topology has no provision for planned or unplanned down time.
Maintainability	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.	None

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.

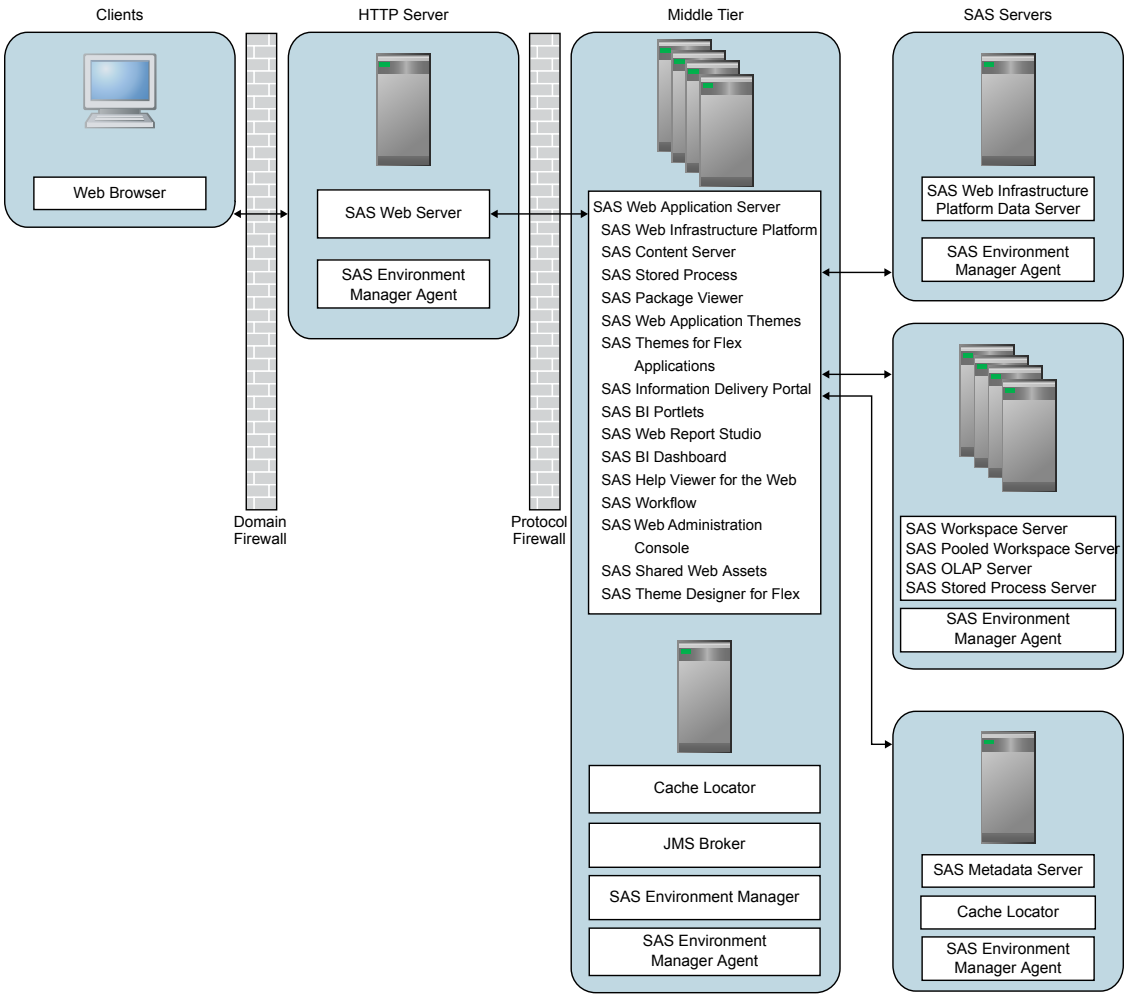
Scenario 2: Web Applications Deployed across a Web Application Server Cluster

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 15.2 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 15.2 *Scenario 2 Advantages and Disadvantages*

Topic	Advantages	Disadvantages
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.	Requires special configuration to open appropriate ports.
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.

Note: In a cluster environment, whether horizontal or vertical, the machines in the middle-tier environment can share the same **SASHome** directory.

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.

Add 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.

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 depends 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.

11. For SAS 9.4M4 and SAS 9.4M5, add the -Dgemfire-conserve-sockets=false JVM option to the Cache Locator start-up script:

    - For Windows deployments, add the option to the section that is labeled # *Java Additional Parameter* in the ***SAS-configuration-directory/Levn/Web/WebAppServer/SASServern_m/conf/wrapper.conf*** file.

      Note: After you modify the wrapper.conf file for 9.4M7 Feb 16th 2022 and later, you need to rebuild the Windows service for each SAS Web Application server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

    - For UNIX deployments, add the option to the section that is labeled # *Java Additional Parameter* in the ***SAS-configuration-directory/Levn/Web/***

gemfire/instances/ins_41415/gemfire-start-locator-sas.sh file.

Post-Configuration Steps

Complete the following steps if SAS Enterprise Miner and SAS Forecast Server are installed in your environment:

12. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\bin\setenv.bat** file (for Windows deployments) or the ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServern_m/bin/setenv.sh** file (for UNIX deployments) of the newly added cluster member.
13. Create an alias for the host name of the machine on which the cluster member resides.
14. Modify the *-Dsas.auto.publish.host* JVM option and set it to the alias that you created:

    ```
    -Dsas.auto.publish.host='alias'
    ```

    Modify all JVMs in the vertical cluster, so that a unique host name is specified for each cluster member in the *-Dsas.auto.publish.host* option.
15. Restart the SAS Web Application Server for changes to take effect.

See Also

"Specify JVM Options" on page 44

Add 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

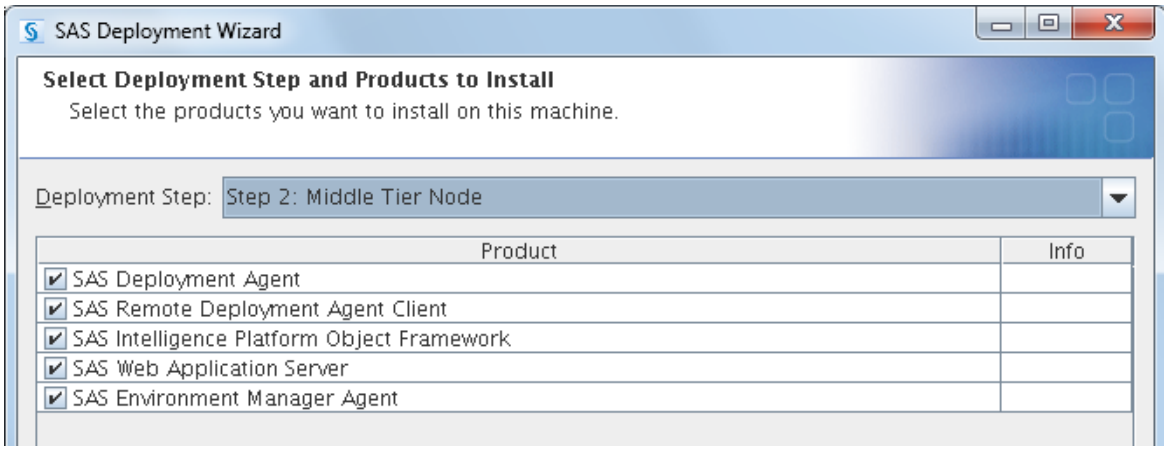
To add a horizontal cluster member:

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`**.

Figure 15.3 *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. For SAS 9.4M4 and SAS 9.4M5, add the -Dgemfire-conserve-sockets=false JVM option to the Cache Locator start-up script:

   - For Windows deployments, add the option to the section that is labeled # *Java Additional Parameter* in the ***`SAS-configuration-directory`***`/Levn/Web/WebApp Server/SASServern_m/conf/wrapper.conf` file.

     Note: After you modify the wrapper.conf file for 9.4M7 Feb 16th 2022 and later, you need to rebuild the Windows service for each SAS Web Application server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

   - For UNIX deployments, add the option to the section that is labeled # *Java Additional Parameter* in the ***`SAS-configuration-directory`***`/Levn/Web/gemfire/instances/ins_41415/gemfire-start-locator-sas.sh` file.

5. If you disabled clustering for SAS Content Server during the configuration, the **-Dcom.sas.server.isclustered** JVM option is 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, first restart the SAS Web Server, and then restart the SAS Web Application Server instances.

TIP Log on to SAS Environment Manager and add the new machine and servers to your inventory.

Maintain a Horizontal Cluster Member

Overview

This section provides an outline of the steps that are needed to maintain a middle-tier cluster node when applying maintenance and any other changes to the SAS middle tier web applications. Maintenance activities include installing hot fixes, adding new products, and performing an update in place. Where applicable, links are included to other sections of this document and other books for additional information.

Generally, to maintain or update the middle-tier horizontal cluster node, you need to apply updates to the primary middle-tier machine and then update all middle-tier horizontal cluster machines.

Install Hot Fixes

Steps to Perform on the Primary Middle-Tier Machine

1. Install all hot fixes on the primary machine using the SAS Deployment Manager. For more information about the installation process, see "Applying Hot Fixes" in *SAS Deployment Wizard and SAS Deployment Manager: User's Guide*.
2. Perform any post-installation tasks that are listed in the documentation that accompanies each hot fix that you applied.
3. Rebuild the SAS web applications. For more information, see "Rebuild the SAS Web Applications" on page 102.
4. Redeploy the SAS web applications. For more information, see "Redeploy the SAS Web Applications" on page 107.

Once you have completed the steps on the primary middle-tier machine, proceed to "Steps to Perform on Each Horizontal Cluster Node" on page 222.

Steps to Perform on Each Horizontal Cluster Node

CAUTION:

Do not proceed with this section until you perform the steps on the primary middle-tier machine. The steps in the above procedure must be performed on the primary middle-tier machine before starting the steps below. Also, do **not** perform the steps below on the primary middle-tier machine.

1. Ensure that all SAS servers and spawners from tiers other than the horizontal cluster nodes are running.
2. For each middle-tier node (horizontal cluster node), stop all SAS sessions, daemons, spawners, servers, and agents.
3. If not already installed, apply any hot fixes and re-apply the latest Security Update to the horizontal cluster node.
4. Start the SAS Deployment Agent on the middle-tier node.
5. Copy the rebuilt web applications from the primary middle tier and deploy them to the horizontal cluster node. Do this by updating the existing configuration from the SAS Deployment Manager with the following steps:

a. On the Select SAS Deployment Manager Task page, under **Administration Tasks**, click **Update Existing Configuration**, and then click **Next**.

b. On the Select Configuration Directory/Level page, specify the configuration directory and the level (for example, **Lev1**), and then click **Next**.

c. On the Specify Connection Information page, enter the user ID and password for an unrestricted administrative user, and then click **Next**.

d. On the Summary page, click **Start**.

6. If you applied any SAS Web Server or SAS Environment Manager hot fixes to the middle-tire node, perform any additional post-update steps. See the SAS Support site for more details.

7. If the SAS Environment Manager Agent is not started, start it now.

Perform an Update in Place

1. For each middle-tier node (horizontal cluster node), stop all SAS sessions, daemons, spawners, servers, and agents.

2. On the primary machine, stop all SAS sessions, daemons, spawners, servers, and agents.

3. On the primary machine, start the SAS Deployment Wizard using the SAS Software Depot that contains the newer versions of the software that you want to upgrade.

4. On the Select Deployment Task page, click **Install SAS software**, and then click **Next**.

5. On the Specify SAS Home page, click **Select a previously created SAS Home** and select the SASHome location that you want to upgrade. Click **Next**.

6. If an upgrade is detected (you selected a SASHome location that contains an older version of a product than what is in the SAS Software Depot) a screen displays what versions will be upgraded. Click **Next** to continue with the upgrade.

   Note: SASHome is upgraded and then the SAS Deployment Manager is launched to update the configuration using **Update Existing Configuration** from the SAS Deployment Manager target.

7. Once the update completes on the primary machine, repeat steps 3-6 on each middle-tier node.

   Note: On the cluster node machine, delete the files in the `Staging` directory before updating.

For more information about update in place, see *SAS Guide to Software Updates and Product Changes*.

Add SAS Products

For information about how to add SAS products on the primary machine, see "Adding SAS Products" in *SAS Intelligence Platform: Installation and Configuration Guide*.

For information about how to add SAS products on each middle-tier node (horizontal cluster node), see "Add SAS Products to a SAS Middle-Tier Horizontal Cluster" in *SAS Intelligence Platform: Installation and Configuration Guide*.

Make Other Changes to Your Middle-Tier Environment

To ensure that any SAS web application changes are updated on the middle-tier node (horizontal cluster node), complete the following:

1. On the primary machine, make the necessary changes to the middle-tier SAS web applications.
2. On the primary machine, run the SAS Deployment Manager and select **Rebuild Web Applications** on the target.
3. On the primary machine, run the SAS Deployment Manager and select **Deploy Web Applications** on the target.
4. For each middle-tier node, follow these steps:
   a. Delete the files in the Staging directory.
   b. Run the SAS Deployment Manager and select **Update Existing Configuration**.

Tune 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*.

Configure 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.

Chapter 16

High-Availability Features in the Middle Tier

Overview

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 217
- "Add a Horizontal Cluster Member" on page 220

Configure the Prerequisite Checker for Clustered Servers

If you enabled the LifeCycle Listener in "Enable the Prerequisite Checker" on page 50, 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*\Levn\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. It can also 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*\Levn\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.vfabrictcsvr.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)?randomize=false"
/>
```

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

Overview

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 difficultly, 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.

Install 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*\Levn\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 "Specify Connection Properties" on page 79.
2. Based on the network topology or protocol change, perform the tasks that apply from "Manual Configuration Tasks" on page 157.
3. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file for all SAS Web Application Servers. In the /Service/Connector element, specify the new connection information for **proxyName**.
   - For SAS 9.4M4 and previous versions:

     ```
     <Connector acceptCount="100" bindOnInit="false" connectionTimeout="20000"
         executor="tomcatThreadPool" maxHttpHeaderSize="16384"
         maxKeepAliveRequests="15" port="${bio.http.port}"
         protocol="org.apache.coyote.http11.Http11Protocol"
         proxyName="proxy.example.com" proxyPort="443"
         redirectPort="${bio.https.port}" scheme="https"
         useBodyEncodingForURI="true"
     />
     ```

   - Starting at SAS 9.4M5:

     ```
     <Connector acceptCount="100" bindOnInit="false"
     ciphers="TLS_ECDHE_ECDSA_WITH_AES_128
     _GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256
     _GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128
     _GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256
     _GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128
     _CBC_SHA" connectionTimeout="20000" executor="tomcatThreadPool"
     maxHttpHeaderSize="16384" maxKeepAliveRequests="15"
     maxSwallowSize="-1" port="${nio.http.port}"
     protocol="org.apache.coyote.http11.Http11NioProtocol"
     proxyName="proxy.example.com" proxyPort="443"
     redirectPort="${nio.https.port}" scheme="https"
     useBodyEncodingForURI="true"/>
     ```

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*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\web.xml**
   - ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\spring\security-web-context.xml**
5. Restart the newly configured SAS Web Server instance and SAS Web Application Servers.

Enable the Prerequisite Checker for Clusters

If you enabled the LifeCycle Listener in "Enable the Prerequisite Checker" on page 50, you should update its configuration to include the additional cluster members. For

information about configuring the LifeCycle Listener for clustered servers, see "Configure the Prerequisite Checker for Clustered Servers " on page 228.

JMS Broker

Overview

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.

Add 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*\Levn\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:

   ```
   <kahaDB directory="/shared/directory/kahadb"/>
   ```

3. Archive ***SAS-configuration-directory*\Levn\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\Levn\Web\WebAppServer\SASServern_m/conf/server.xml** file. Change all the **brokerURL** attributes for the resources to resemble the following example:

   ```
   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.

See Also

"Update the Connection to JMS Broker" on page 229

Enable the Prerequisite Checker for Clusters

If you enabled the LifeCycle Listener in "Enable the Prerequisite Checker" on page 50, you should update its configuration to include the additional cluster members. For information about configuring the LifeCycle Listener for clustered servers, see "Configure the Prerequisite Checker for Clustered Servers " on page 228.

JMS Broker Restart

If you bring JMS Broker back up after a shutdown, make sure there is no lock file present in JMS Broker configuration directory. If the ***SAS-configuration-directory*\Levn\Web\activemq\data\kahadb\lock** file exists on any of the machines on which JMS Broker is installed, delete the file before attempting to start the JMS Broker.

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.

Configure Additional Locator for UNIX for SAS 9.4 M8

In SAS 9.4M8, the cache locator is OpenSource Geode, which replaced VMware GemFire. Because of this change, the configuration directory references geode in its directory path, and configuration files have changed their names to include geode. See "Configure JVM Options for the Cache Locator" on page 52for directory and file names.

To configure an additional locator for UNIX deployments, follow these steps:

1. On the primary middle-tier machine, the locator software is archived here: ***SAS-configuration-directory*/Levn/Web/Scripts/AppServer/src/Config/vfabrictcsvr/apache-geode.zip**

   Extract the archive to the identical directory on the additional machine. After extracting the contents of the zip file, change the name of the high-level directory from **apache-geode** to **geode**.

2. Copy the following files from the ***SAS-configuration-directory*/Levn/Web/Scripts/AppServer/src/Config/vfabrictcsvr** directory on the primary middle-tier machine to the directories on the additional machine as follows:

   - Copy the **geode-sas** file to the ***SAS-configuration-directory*/Levn/Web/geode/bin** directory.
   - Copy the **sas-gemfire-startup-cleaner.jar** file to the ***SAS-configuration-directory*/Levn/Web/geode/lib** directory.

3. Update the following files to be executable:

```
dos2unix SAS-configuration-directory/Levn/Web/geode/bin/gfsh
chmod 755 SAS-configuration-directory/Levn/Web/geode/bin/gfsh
dos2unix SAS-configuration-directory/Levn/Web/geode/bin/geode-sas
chmod 755 SAS-configuration-directory/Levn/Web/geode/bin/geode-sas
```

4. Create an instance directory that is identical to the primary machine, for example: ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_41415**

5. Copy the files from the instance directory on the primary machine to the additional machine.

   Note: You do not need to copy the **.locator**, **.pid**, or **.dat** files, nor the **ConfigDiskDir*** directory.

6. Edit the ***SAS-configuration-directory*/Levn/Web/geode/instances/ins_41415/geode-locator.sh** file on the additional machine. Find the **LOCATORS** property and specify a comma-separated list of all the locators that includes the newly added locator:

```
LOCATORS=primary.example.com[41415],secondary.example.com[41415]
```

7. Update the following file to be executable:

```
chmod 755 SAS-configuration-directory/Levn/Web/geode/instances/ins_41415/
geode-start-locator-sas.sh
```

8. Start the locator:

```
SAS-configuration-directory/Levn/Web/geode/instances/ins_41415/
geode-locator.sh start
```

9. Update the following items with the same list of locators:

```
LOCATORS=primary.example.com[41415],secondary.example.com[41415]
```

   - **geode-locator.sh** file for all the previously installed locators
   - **-Dsas.cache.locators** JVM option for all SAS Web Application Server instances
   - **-Dsas.cache.locators** JVM option for all instances of the SAS Web Infrastructure Platform Scheduling Services. The change is made in the ***SAS-configuration-directory*/Levn/Web/Applications/SASWIPSchedulingServices9.4/servicetrigger.ini** file.

Configure Additional Locator for UNIX for SAS 9.4 M7 and Previous Releases

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*/Levn/Web/Scripts/AppServer/src/Config/vfabrictcsvr/gemfiren.zip** where *n* is the version number, if one exists. Extract the archive to the identical directroy on the additional machine.

2. Copy the following files from the ***SAS-configuration-directory*/Levn/Web/Scripts/AppServer/src/Config/vfabrictcsvr** directory on

the primary middle-tier machine to the directories on the additional machine as follows:

- The **gemfire-sas** file should be copied to the ***SAS-configuration-directory*/Levn/Web/gemfire/bin** directory.

  The **sas-gemfire-startup-cleaner.jar** file should be copied to the ***SAS-configuration-directory*/Levn/Web/gemfire/lib** directory.

3. Update the following files to be executable:

```
dos2unix SAS-configuration-directory/Levn/Web/gemfire/bin/gemfire
chmod 755 SAS-configuration-directory/Levn/Web/gemfire/bin/gemfire
dos2unix SAS-configuration-directory/Levn/Web/gemfire/bin/agent
chmod 755 SAS-configuration-directory/Levn/Web/gemfire/bin/agent
dos2unix SAS-configuration-directory/Levn/Web/gemfire/bin/cacheserver
chmod 755 SAS-configuration-directory/Levn/Web/gemfire/bin/cacheserver
dos2unix SAS-configuration-directory/Levn/Web/gemfire/bin/gfsh
chmod 755 SAS-configuration-directory/Levn/Web/gemfire/bin/gfsh
dos2unix SAS-configuration-directory/Levn/Web/gemfire/bin/gemfire-sas
chmod 755 SAS-configuration-directory/Levn/Web/gemfire/bin/gemfire-sas
chmod 755 SAS-configuration-directory/Levn/Web/gemfire/lib/*.so
```

4. Create an instance directory that is identical to the primary machine, for example: ***SAS-configuration-directory*/Levn/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. Edit the ***SAS-configuration-directory*/Levn/Web/gemfire/instances/ins_41415/gemfire-locator.sh** file on the additional machine. Find the **LOCATORS** property and specify a comma-separated list of all the locators that includes the newly added locator:

```
GF_JAVA=/SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/java
export GF_JAVA
LOCATOR_HOME=/SAS-configuration-directory/Levn/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
LOCATOR_HOST=secondary.example.com
```

7. Update the **gemfire-start-locator-sas.sh** file to be executable:

```
chmod 755 /SAS-configuration-directory/Levn/Web/gemfire/instances/ins_41415/
gemfire-start-locator-sas.sh
```

8. Start the locator:

```
/SAS-configuration-directory/Levn/Web/gemfire/instances/ins_41415/
gemfire-locator.sh start
```

   - **SAS 9.4M7** and earlier: **/SAS-configuration-directory/Levn/Web/gemfire/instances/ins_41415/ gemfire-start-locator-sas.sh start**

9. Update the following items with the same list of locators:

```
LOCATORS=primary.example.com[41415],secondary.example.com[41415]
```

- **gemfire-locator.sh** file for all the previously installed locators
- **-Dsas.cache.locators** JVM option for all SAS Web Application Server instances
- **-Dsas.cache.locators** JVM option for all instances of the SAS Web Infrastructure Platform Scheduling Services. The change is made in the ***SAS-configuration-directory*/Levn/Web/Applications/SASWIPSchedulingServices9.4/servicetrigger.ini** file.

Configure Windows for SAS 9.4 M7 and Previous Releases

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*\Levn\Web\Scripts\AppServer\src\Config\vfabrictcsvr\gemfiren.zip**, where *n* is the version number, if one exists.
2. Extract the archive to the identical ***SAS-configuration-directory*\Levn\Web** directory on the additional machine.
3. Create an instance directory that is identical to the primary machine (for example, ***SAS-configuration-directory*\Levn\Web\gemfire\instances\ins_41415**).
4. Copy the files from the instance directory on the primary machine to the additional machine.
5. Copy the entire SASPrivateJavaRuntimeEnvironment directory and subdirectories on the primary machine to the additional machine.

   Note: After you perform an update in place, the SAS Private JRE might need to be copied again, as it might be updated.
6. In the instance directory, update the following lines in the wrapper.conf file:

   ```
   set.GEMFIRE_HOME=../..
   set.INSTANCE_NAME=ins_41415
   set.INSTANCE_PORT=41415
   set.JAVA_HOME=SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre
   set.GEMFIRE_SERVICE_NAME=SAS [Config-Levn] 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.

   Note: After you modify the wrapper.conf file for 9.4M7 Feb 16th 2022 and later, you need to rebuild the Windows service for each SAS Web Application server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.
7. Install Windows service for SAS Cache Locator:

   ```
   cd gemfire\bin\winx86_64
   installservice.bat ins_41415
   ```
8. Start the locator with the Windows service name **SAS [Config-Levn] SAS Cache Locator 41415**.

9. 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.

     Note: After you modify the wrapper.conf file for 9.4M7 Feb 16th 2022 and later, you need to rebuild the Windows service for each SAS Web Application server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.
   - 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-directory*\Levn\Web\Applications\SASWIPSchedulingServices9.4\servicetrigger.ini** file.

Enable the Prerequisite Checker for Clusters

If you enabled the LifeCycle Listener in "Enable the Prerequisite Checker" on page 50, you should update its configuration to include the additional cluster members. For information about configuring the LifeCycle Listener for clustered servers, see "Configure the Prerequisite Checker for Clustered Servers " on page 228.

Perform an Update in Place

If you perform an update in place, the high availability Cache Locator servers will not automatically update. During the update in place, backup files are created that you can use to restore the high availability settings. The backup files are stored in the ***SAS-configuration-directory*/Levn/Web/Backup/gemfire.*backup-date*/** directory. To restore the high availability configuration, compare the changes that were made for high availability to the files in the backup directory to the files in the updated deployment.

Chapter 17

Enterprise Integration

Configure the Middle Tier to Use an Existing Customer Reverse Proxy

Note: If you have configured the middle tier to use an existing reverse proxy and you are upgrading to SAS 9.4M4, your manual configuration settings are preserved.

Note: There are some solutions that do not preserve manual reverse proxy settings for the WebDAV repository and connection properties. After updating to SAS 9.4M4, you must determine which solutions, if any, will need to have their manual reverse proxy settings for WebDAV and connection properties reset. To do this review, complete Step 3 on page 242 and Step 4 on page 243.

Note: You can disable the following filters to avoid configuration issues that are related to existing reverse proxy connection settings while upgrading:

- ServiceUrl.PerformCheck
- sas.web.cdps.performCheck

- sas.web.csrf.referers.performCheck

  Disable the filters by setting the properties to false. If you disable the filters, remember to enable them after the upgrade completes. Setting these to false bypasses an update content error. For more information, see "Modify the Allowlist for URLs and HTTP Request Methods" on page 365.

Note: If you configure the middle-tier environment to use an existing reverse proxy, then the reverse proxy must be configured to allow for certain request methods. For a list of the supported HTTP request methods, see "HTTP Request Methods Used by SAS 9.4 Software" on page 245.

Note: Starting with SAS 9.4M5, it is required that you configure SAS Environment Manager for HTTPS. For more information, see "Configure SAS Environment Manager for HTTPS Starting with SAS 9.4M5" on page 330.

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.

The changes made to this section are not reflected in the ***SAS-configuration-directory*\Levn\Web\Scripts\AppServer\props\appserver.properties** file that is used by the scripting tool for SAS Web Application Server. For more information, see "Scripting Tool for SAS Web Application Server" on page 398. If you use these scripts, the changes that you make must also be added to any related properties listed in that file. It is recommended that you create a backup of the file before editing it.

The existing web server proxy might or might not be enabled for Transport Layer Security (TLS). Complete the following steps for an existing web server proxy that is TLS-enabled:

1. Edit the ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServern_m/conf/server.xml** file and follow these steps:

   a. In the /Service/Connector element, change the value for **proxyName** and check the values for **proxyPort** and **scheme**.

      - For SAS 9.4M4 and earlier versions:

```
<Connector acceptCount="100" bindOnInit="false" connectionTimeout="20000"
    executor="tomcatThreadPool" maxHttpHeaderSize="16384"
    maxKeepAliveRequests="15" port="${bio.http.port}"
    protocol="org.apache.coyote.http11.Http11Protocol"
    proxyName="proxy.example.com" proxyPort="443"
    redirectPort="${bio.https.port}" scheme="https"
    useBodyEncodingForURI="true"
/>
```

      - Starting with SAS 9.4M5:

```
<Connector acceptCount="100" bindOnInit="false"
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128
_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256
_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128
_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256
_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128
_CBC_SHA" connectionTimeout="20000" executor="tomcatThreadPool"
maxHttpHeaderSize="16384" maxKeepAliveRequests="15"
maxSwallowSize="-1" port="${nio.http.port}"
```

```
protocol="org.apache.coyote.http11.Http11NioProtocol"
proxyName="proxy.example.com" proxyPort="443"
redirectPort="${nio.https.port}" scheme="https"
useBodyEncodingForURI="true"/>
```

b. In the /Engine/Host/Valve element, specify the IP address or regular expression for the reverse proxy by adding a trustedProxies attribute, and verify that httpServerPort, httpsServerPort, and protocolHeader specify the values for the reverse proxy. If they are not specified, add them as follows:

```
<Valve
className="org.apache.catalina.valves.RemoteIpValve"
httpServerPort="80" httpsServerPort="443"
internalProxies="(fe80|fd[0-9a-fA-F]{2})(:{1,2}[0-9a-fA-F]{0,4})
{0,7}(%[0-9a-zA-Z]+)?$|::1|0:0:0:0:0:0:0:1|10\.\d{1,3}\.\d{1,3}\.\d{1,3}|192\.168\.\d
{1,3}\.\d{1,3}|169\.254\.\d{1,3}\.\d{1,3}|127\.\d{1,3}\.\d{1,3}\.\d{1,3}|
172\.1[6-9]{1}\.\d{1,3}\.\d{1,3}|172\.2[0-9]{1}\.\d{1,3}\.\d{1,3}|172\.3[0-
1]{1}\.\d{1,3}\.\d{1,3}" trustedProxies="###\.###\.###\.###"
protocolHeader="X-Forwarded-Proto"/>
```

Note: For **trustrustedProxies=**, replace the **#** signs with the actual digits of the IP address of the proxy. In regular expressions, the **** sign is used to escape a character that otherwise has special meaning. In the preceding example, it replaces periods. If you want to specify more than one IP address, separate the addresses with a | sign. Here is an example: **trustedProxies="12\.345\.678\.90|23\.456\.789\.01"**

c. If you have more than one SAS Web Application Server instance, make the change for each one.

2. For customers of SAS Enterprise Guide and SAS Add-In for Microsoft Office, when configuring a reverse proxy for a SAS 9.4 environment, you need to set the WebServiceURIPolicy to External in your systems settings file: **%APPDATA%/SAS/SharedSettings/7.1/Engine/SystemSettings.xml**.

3. Use SAS Management Console to specify an external connection for each SAS web application.

   If your reverse proxy is dedicated to SAS, it can proxy all requests. Otherwise, examine the ***SAS-configuration-directory*\Levn\Web\WebServer\conf\sas.conf** file to identify the specific web applications to proxy. Each web application that is identified in a pair of ProxyPass and ProxyPassReverse directives must be proxied.

   Note: Do not edit the **sas.conf** file.

   Some applications, such as SAS Visual Analytics, are nested, and you must expand them to access all the underlying applications. For more information, see "Specify Connection Properties" on page 79.

   Note the following items:

   - To use SAS Management Console to modify the Themes Connection information, ensure that the SAS middle tier is running.
   - Some SAS users prefer to update these values using a SAS DATA step. This approach is beyond the scope of this task. If you choose to modify these connections using a SAS script rather than SAS Management Console, then the SAS_THEME table in the Shared Services 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** ⇨ **SAS Themes** ⇨ **SASTheme_default**. View the **Properties** to determine whether connection information needs to be updated.
- For more information about the properties that are needed when internal and external connections are different, see "Allowlist of Websites and Methods Allowed to Link to SAS Web Applications" on page 364.

4. Use SAS Management Console to update the SAS Content Server connection information. For more information, see "Manual Configuration Tasks" on page 157.
5. Use SAS Management Console to update the metadata property searchsas.auth.provider.url to point to the reverse proxy.

   From SAS Management Console, navigate to the Plug-ins tab and select **Application Management** ⇨ **Configuration Manager**. Right-click **Search Interface to SAS Content 3.#** and select **Properties**. On the **Advanced** tab, edit the searchsas.auth.provider.url property to point to the reverse proxy.
6. Restart SAS Web Application Server.
7. Update SAS Environment Manager by editing the following files and locating 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*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\spring\security-web-context.xml** file and complete the following:

      - Locate the bean definition with id="logoutFilter". It has two constructor arguments, each containing a URL. The first argument specifies the SAS Logon Manager logoff URL. Appended to that URL is the default entry point for SAS Environment Manager. Update the SAS Logon Manager logoff URL to go through the existing customer reverse proxy.
      - Locate the bean definition with id="casAuthenticationEntryPoint". It has a property name="loginUrl" with a value that specifies the SAS Logon Manager logon URL. This value is called by the web browser. Update the SAS Logon Manager logon URL to go through the existing customer reverse proxy.

        Note: Do not change the value of the URLs specified for id="casTicketValidator" and id="usernamePasswordCasAuthenticationProvider". These values use internal URLs that should not use the external reverse proxy.

   b. Edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\web.xml** file and update the SAS Logon URL in the following lines with the existing customer reverse proxy information for your environment. For the line with two constructor arguments, the SAS Logon URL is the first URL.

      - For SAS 9.4M4 and earlier releases, locate the following lines and enter the correct information for your environment:

```
<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>
```

- Starting with SAS 9.4M5, locate the following lines and enter the correct information for your environment:

```
<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.

8. If SAS Grid Manager for Platform is deployed in your environment, update the following variables to point to the reverse proxy configuration:

   - For Windows deployments, edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer14_1\conf\wrapper.conf** file and update these variables:

     ```
     set.CAS_SERVER_LOGIN_URL=http://proxy.example.com:port/SASLogon/login
     set.PWS_SERVER_URL=http://proxy.example.com:port
     ```

     Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later releases, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

   - For UNIX deployments, edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer14_1\bin\setenv.sh** file and update these variables:

     ```
     CAS_SERVER_LOGIN_URL=http://proxy.example.com:port/SASLogon/login
     PWS_SERVER_URL=http://proxy.example.com:port
     ```

   In addition, update the SAS Grid Manager for Platform resource in SAS Environment Manager. Set the **pws.ticket.url** value to http://*proxy.example.com:port*/PlatformWebServices, so the correct URL is used during authentication.

9. When SAS web applications return HTTP 502 proxy errors, you might have to change the time-out interval for your reverse proxy server. For example, complete the following steps for the Apache HTTP Server version 2.2 server:

   a. Edit the **/etc/httpd/conf/httpd.conf** file.

   b. Add the following parameter to the file:

      ```
      ProxyTimeout time-in-seconds
      ```

   c. Restart the Apache HTTP Server.

Note: For instructions about how to configure solutions, such as SAS Customer Intelligence Studio, that include SAS Visual Analytics when a reverse proxy server is used, see SAS Note 59077.

If you are using the scripting tool for SAS Web Application Servers, changes made above are not reflected in the appserver.properties file. The changes made must also be added to any related properties listed in that file.

Note: Before editing the appserver.properties file, back up the file.

For more information, see "Scripting Tool for SAS Web Application Server" on page 398.

HTTP Request Methods Used by SAS 9.4 Software

SAS 9.4 software uses the following HTTP request methods:

Table 17.1 *Supported HTTP Request Methods*

ACL	OPTIONS
CHECKIN	POST
CHECKOUT	PROPFIND
COPY	PROPPATCH
DELETE	PUT
GET	REPORT
HEAD	SEARCH
LABEL	UNCHECKOUT
LOCK	UNLOCK
MKCOL	VERSIONCONTROL
MOVE	

Note: If you configure the middle-tier environment to use an existing reverse proxy, the reverse proxy must be configured to allow for these request methods. For more information, see "Configure the Middle Tier to Use an Existing Customer Reverse Proxy" on page 240.

Web Authentication

Overview

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.

For more information, see "Introduction to Authentication Mechanisms" in *SAS Intelligence Platform: Security Administration Guide*.

Note: If you have server instances on multiple machines, then perform the following 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.

Note: Before you configure web authentication, 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.

Configure Web Authentication

Modify SAS Logon Manager Installation Files

Steps for SAS 9.4M1

1. Open the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\cas-servlet.xml** file. 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 preceding 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 sas.svcs file, you can edit the deployed file, ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_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. Locate the following block in the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\login-webflow.xml** file:

```
<action-state id="generateLoginTicket">
     <evaluate
        expression="generateLoginTicketAction.generate(flowRequestContext)" />
     <transition on="success" to="viewLoginForm" />
</action-state>
```

   Replace the preceding 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*\Levn\Web\WebAppServer\SASServer*n_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. Open the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\deployerConfigContext.xml.orig** file.

   - Add the following bean definition within **\beans\bean[id="authenticationManager"]\property[name="credentialsToPrincipalResolvers"]\list**:

     ```
     <bean class="org.jasig.cas.adaptors.trusted.authentication.principal.
         PrincipalBearingCredentialsToPrincipalResolver" />
     ```

   - Add the following bean definition within the **\beans\bean[id="authenticationManager"]\property[name="authenticationHandlers"]\list**:

     ```
     <bean class="org.jasig.cas.adaptors.trusted.authentication.handler.support.
         PrincipalBearingCredentialsAuthenticationHandler" />
     ```

   Note: The preceding bean definitions 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*\Levn\Web\WebAppServer\SASServer*n_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.

4. Add the following code above the closing **</web-app>** tag in the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig** file:

   ```
   <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*\Levn\Web\WebAppServer\SASServer*n_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.

Steps for SAS 9.4M2 and Later Releases

1. Open 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 SAS 9.4M3, a similar section is also provided for form-based log on.

   Note: If you plan to configure fallback authentication, do not make changes to the web.xml.orig file. For more information about configuring fallback authentication, see "Fallback to SAS Form-based Authentication" on page 292.

```
<!-- Enable SPNEGO authentication -->
<!--
<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>SPNEGO</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*\Levn\Web\WebAppServer\SASServer*n_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.

   Note: BASIC is case-sensitive.

Modify SAS Visual Analytics Transport Service Installation Files

The following steps apply to deployments that distribute reports for SAS Visual Analytics App users. If you choose to configure Transport Service for web authentication, make sure that you use BASIC for the **auth-method**. SAS Visual Analytics App supports BASIC authentication only. Guest access is not compatible with web authentication.

1. Open the ***SASHOME*\SASVisualAnalyticsServices*version*\Configurable\wars\sas.bitransportservices\WEB-INF\web.xml.orig** file. Remove the comment that encloses the <security-constraint> and <login-config> tags. 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>
               <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*\Levn\Web\WebAppServer\SASServer*n_m*\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.

2. (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.

Rebuild and Redeploy Web Applications

Use the SAS Deployment Manager to do the following:

1. Rebuild the SAS Web Infrastructure Platform. If you modified SAS Transport Service, also rebuild Visual Analytics Services.
2. Stop SAS Web Application Server. Redeploy the SAS Web Infrastructure Platform. If you modified SAS Transport Service, also redeploy Visual Analytics Services.

See "Rebuild the SAS Web Applications" and "Redeploy the SAS Web Applications" in Chapter 8, “Administer SAS Web Applications,” on page 101 for more details.

Create Authentication Domain and Confirm User Accounts in SAS Metadata

1. Start SAS Management Console and access the User Manager plug-in.

2. Create the web Authentication Domain. Right-click the User Manager plug-in and select **Authentication Domains**. Click **New** and specify **`web`** as the name. Click **OK**.

3. Open the properties of each user that requires access to web applications. Click **New** on the **Accounts** tab of the user. Add the user ID and change the Authentication Domain to **`web`**. Click **OK**.

   Note: If users in SAS metadata do not have a user ID on the **Accounts** tab that is associated with the web Authentication Domain, then a SAS identity will not be found after authentication to the web application server container succeeds and authorization takes place.

(Optional) Validate the Previous Steps

1. 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`*\`Levn`\`Web`\`WebAppServer`\`SASServern_m`\`conf`\`server.xml`** file.

   Edit the ***`SAS-configuration-directory`*\`Levn`\`Web`\`WebAppServer`\`SASServern_m`\`conf`\`tomcat-users.xml`** file to be similar to the following example:

   ```
   <?xml version="1.0"?>
   <tomcat-users>
       <role rolename="ROLE_USER" />
       <user username="sasdemo" password="Password1" roles="ROLE_USER" />
   </tomcat-users>
   ```

   Note: In SAS 9.4M2 and later, you must have specified BASIC authentication in the web.orig.xml file. This is Modify SAS Logon Manager Installation Step 2 on page 248.

   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`**. The specified user must have an account on the **Accounts** tab in metadata.

2. Verify that the LockOutRealm and the UserDatabaseRealm look similar to the following in the ***`SAS-configuration-directory`*\`Levn`\`Web`\`WebAppServer`\`SASServern_m`\`conf`\`server.xml`** file. (In SAS 9.4M1, you need to modify the file.)

   ```
   <Realm className="org.apache.catalina.realm.LockOutRealm"
   allRolesMode="authOnly">

   <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
   resourceName="UserDatabase" allRolesMode="authOnly"/>
   ```

3. 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

4. Remember to remove the user and role information when you complete this procedure.

Configure the Realm for SAS Web Application Server

1. Locate the existing **/Server/Service/Engine/Realm** definition in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file.

   Note: If you have more than one web application server instance, you must make the following changes to each one.

2. Modify the realm information so that it accesses the system that you want to use for identity management. Here is an example for accessing an LDAP server:

```
<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-8.0-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.

3. Start SAS Web Application Server.

4. 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.

Auto-Provision User Accounts

Starting with SAS 9.4M3, new users can be auto-provisioned into the SAS Metadata Server after their credentials have been verified via web authentication. Once configured, the auto-provisioning filter intercepts logins, checks to see whether the user exists in the SAS Metadata Server, and creates the user, if necessary.

Although SAS provides utilities for importing users into the SAS Metadata Server, these scripts must be run periodically to import new users and might create accounts for users that never use the software.

Note: The auto-provisioning filter works only with users that authenticate using web authentication.

1. Edit the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig** file as follows:

   a. Add the following code to the <filter> section of the file:

```
<filter>
    <filter-name>autoProvisioningFilter</filter-name>
    <filter-class>org.springframework.web.filter.
    DelegatingFilterProxy</filter-class>
</filter>
```

Note: The preceding filter-class definition must be entered on one line. It is shown on more than one line for display purposes only.

b. Add the following code to the <filter-mapping> section of the file:

```
<filter-mapping>
    <filter-name>autoProvisioningFilter</filter-name>
    <url-pattern>/login</url-pattern>
</filter-mapping>
```

Note: As an alternative to updating the web.xml.orig file, you can edit the deployed file, ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_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. Edit the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\spring-configuration\filters.xml** file, and add the following code before directAuthenticationFilter:

```
<bean id="autoProvisioningFilter" class="com.sas.svcs.security.authentication.
filters.UserAutoProvisioningFilter"
      p:urlGenerator-ref="svcs.urlGenerator"
      p:adminUser="@{server.admin.userid}"
      p:adminPassword="@{server.admin.password}"
      p:groupName="group_name" />
```

Note: The first line of the preceding bean definition must be entered on one line. It is shown on more than one line for display purposes only.

Note: The *p:groupName* field specifies the group to which new users are added in metadata.

Note: As an alternative to updating the filters.xml file, you can edit the deployed file, ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\sas_webapps\sas.svcs.logon.war\WEB-INF\spring-configuration\filters.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. Restart SAS Web Application Server.

Increase the Session Time-Out

When using web authentication, especially with form-based authentication, you might need to increase the session time-out in SAS Logon Manager.

- For SAS 9.4M8 and later releases, you do not need to perform these steps because the session time-out is set to 30 minutes by default.
- For SAS 9.4M6 and SAS 9.4M7 the default session-timeout is set to 5 minutes for this application in the **web.xml.orig** file. It should be set to 30 minutes or more. To increase the session time-out, follow these steps:

  1. Edit the ***SASHOME*/SASWebInfrastructurePlatform/9.4/Configurable/wars/sas.svcs.logon/WEB-INF/web.xml.orig** file.

     Note: As an alternative to updating the **web.xml.orig** file, you can edit the deployed file, ***SAS-configuration-directory*/Levn/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.

2. Locate the section below:

```
 <session-config>
<!-- Default to 5 minute session timeouts -->
<session-timeout>5</session-timeout>
</session-config>
```

3. Change the **<session-timeout>** value to 30 or more.

   Alternatively, remove the session-timeout element in **web.xml.orig**. This reverts the session-timeout to 30 minutes, which is defined in the ***SAS-configuration-directory*/Levn/Web/WebappServer/SASServer1_1/conf/web.xml** file.

4. Use the SAS Deployment Manager to rebuild SAS Logon Manager. See "Rebuild the SAS Web Applications" on page 102.

5. Use the SAS Deployment Manager to redeploy SAS Logon Manager. See "Redeploy the SAS Web Applications" on page 107.

Configure IBM WebSEAL Using Standard Junction

Configure 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*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file.

Change the Proxy Information for SAS 9.4M1 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 1c on page 242.

Update 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 "Specify Connection Properties" on page 79. Make sure that you also specify the **-Dsas.retry.internal.url=true** JVM option that is identified on that page.

Configure the WebSEAL Junction

For SAS 9.4, transparent junctions with no special consideration are supported. Beginning with SAS 9.4M2, 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 Server is listening, with a command that is similar to the following:

```
pdadmin> server task default-webseald-host_name create -t tcp -I -c iv-user,
iv-groups -b ignore -h saswebserver.example.com -p 80 /junction_name
```

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/*
/junction-name */SASFlexThemes/*
/junction-name */SASIdentityServices/*
/junction-name */SASJSR168RemotePortlet/*
/junction-name */SASLogon/*
/junction-name */SASPackageViewer/*
/junction-name */SASPermissionManager/*
/junction-name */SASPortal/*
/junction-name */SASPreferences/*
/junction-name */SASPrincipalServices/*
/junction-name */SASSharedApps/*
/junction-name */SASStoredProcess/*
/junction-name */SASTemplateEditor/*
/junction-name */SASTheme_default/*
/junction-name */SASThemeDesignerForFlex/*
/junction-name */sasweb/*
/junction-name */SASWebDoc/*
/junction-name */SASWebReportStudio/*
/junction-name */SASWIPServices/*
/junction-name */SASWorkflowServices/*
/junction-name */SASWorkflowWebServices/*
```

TIP These entries represent most of a SAS Enterprise Business Intelligence deployment. Look at the ***SAS-configuration-directory*\Levn\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 is 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:

```
pdadmin> server task default-webseald-host_name jmt load
```

Modify User Permissions for REST API Calls

In several SAS solutions (for example, SAS Visual Analytics 7.2 and higher), REST API calls with PUT or DELETE requests are used in some functionality. By default, PUT or DELETE requests are blocked by WebSEAL. The user permissions must be modified to allow PUT or DELETE requests to go through WebSEAL. User permissions should include Modify and Delete permissions. Without adding Modify and Delete permissions for the SAS user, REST API PUT and DELETE calls can fail with a 403 (not authorized) error.

To update user permissions from the WebSEAL resource administration tool (pdamin), follow these steps:

1. Invoke pdamin from the *PolicyDirector*/bin directory and log on using the sec_master account:

   ```
   ./pdadmin
       login  -a sec_master  -p password
   ```

2. Check out the object space defined for the **WebSEAL configuration object** list. The object space for the default configuration is /WebSEAL. You might have your own.

   ```
   object show /WebSEAL
   ```

3. Check out the attached ACL / effective ACL. The default configuration contains "default-webseal". Here is an example ACL:

   ```
   acl show default-webseal
   ```

4. The typical user has Trx permission (read and execute). Add Modify and Delete permissions to all SAS WebSEAL users. The following command is for the sasdemo user:

   ```
   acl modify default-webseal set user sasdemo Tdmrx
   ```

See Also

IBM Tivoli Access Manager for e-business WebSEAL Administration Guide

Configure ISAM WebSEAL Using Standard Junction

Create a Reverse Proxy

From IBM Security Access Manager (ISAM), follow these steps:

1. Log on to ISAM as administrator.
2. Click **Secure Web Settings** ⇨ **Reverse Proxy**.
3. In the Reverse Proxy window, click **New**.
4. In the New Reverse Proxy Instance window, follow these steps:
   a. On the **Instance** tab, enter an instance name and select an IP address from the drop-down menu. Then, click **Next**.
   b. On the **IBM Security Access Manager** tab, enter the administrator password and click **Next**.
   c. On the **Transport** tab, select the option to **Enable HTTP**. You can use the default HTTP Port that is pre-populated or enter a different port, and then click **Finish**.

      The newly created instance is displayed in the **Instance Name** list.
5. Set the authentication and realm name for the newly created instance by completing the following steps:
   a. Select the instance and click **Next**.
   b. On the **Authentication** tab, under **Basic Authentication**, select the transport protocol and enter the realm name. Then, click **Save**.

      Note: The value of realm name should be the same as the instance name that you previously specified.
6. Proceed to "Create a Junction" on page 256.

Create a Junction

1. Click **Secure Web Settings** ⇨ **Reverse Proxy** ⇨ ***instance-name***.
2. Click **Manage** and select **Junction Management** from the drop-down menu.
3. In the Junction Management window, click **New** ⇨ **Standard Junction**.
4. On the **Junction** tab, enter a junction name.

   Note: The name must be preceded by a forward slash.
5. On the **Servers** tab, click **New**. Enter the host name and TCP or TLS port of the machine that SAS Web Server is running on. Then, click **Save**.
6. On the **Identity** tab, follow these steps:
   a. For the **HTTP Basic Authentication Header** drop-down menu, select `Ignore`.

b. In the **HTTP Header Identity Information** section, choose the **`IV-USER`** and **`IV-GROUPS`** options.

c. For the **HTTP Header Encoding** drop-down menu, select **`UTF-8 URI Encoded`**.

d. Enable the **Include original junction path in cookies** option.

e. Click **Save**.

7. Follow the ISAM system messages to change the deployment and restart the instance.

Create a Junction Mapping Table

1. Open the ***`SAS-configuration-directory`*`\Levn\Web\WebServer\conf\sas.conf`** file and copy the Service Names for which ProxyPass URLs are defined to a word processing program or spreadsheet. Here is an example:

```
ProxyPass /FolderModule
```

2. Build the Junction Mapping Table (JMT) in the following format:

```
/junction-name */service-name/*
```

Note: Do not include /SASWIPClientAccess in the JMT. 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 JMT to handle this redirection. The additional entry is 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*
```

3. Create the JMT in ISAM by completing the following:

a. Copy the JMT that you created in Step 2.

b. Log on to ISAM as administrator.

c. Click **Secure Web Settings** ⇨ **Junction Mapping**.

d. In the Junction Mapping window, click **New**.

e. In the JMT Configuration File Create window, paste the JMT from Step 3a in the **Content** section.

f. In the **JMT Configuration File Name** field, enter a file name and click **Save**.

g. Follow the ISAM system messages to change the deployment and restart the instance.

4. Proceed to "Associate a Junction Mapping Table with a Reverse Proxy Instance" on page 258.

Associate a Junction Mapping Table with a Reverse Proxy Instance

1. Click **Secure Web Settings** ⇨ **Reverse Proxy** ⇨ ***instance-name***.
2. Click **Manage** and select **Configuration** from the drop-down menu.
3. In the Configuration window, select **Edit Configuration File** from the drop-down menu.
4. In the Advanced Configuration File Editor window, follow these steps:
   a. Press Ctrl-F to open the **Find** search box.
   b. Search for **`jmt-map`**.
   c. Modify the jmt-map property to point to the JMT that you created in Step 3f on page 257. Then, click **Save**.
   d. Follow the ISAM system messages to change the deployment and restart the instance.
5. Proceed to "Modify the Configuration File for the Instance" on page 258.

Modify the Configuration File for the Instance

1. Click **Secure Web Settings** ⇨ **Reverse Proxy** ⇨ ***instance-name***.
2. Click **Manage** and select **Configuration** from the drop-down menu.
3. In the Configuration window, select **Edit Configuration File** from the drop-down menu.
4. In the Advanced Configuration File Editor window press Ctrl-F to open the **Find** search box.
5. Search for **`Method disablement`**.
6. Modify the entries in the *Method disablement* section of the file, so that they are blank. Here is an example:

   ```
   http-method-disabled-local =
   http-method-disabled-remote =
   ```

7. Press Ctrl-F to open the **Find** search box.
8. Search for **`process-root-requests`**.
9. Modify the *Process root junction requests* section of the file as follows:

   ```
   process-root-requests = never
   ```

10. Search the file for **`script-filter`** and modify as follows:

    ```
    script-filter = no
    ```

11. To enable FIPS mode processing in SAS 9.4M8, change the value of **`jct-nist-compliance`** to **`yes`**. Here is an example:

    ```
    jct-nist-compliance = yes
    ```

12. Click **Save**.
13. Follow the ISAM system messages to change the deployment and restart the instance.

14. Proceed to "Add Users" on page 259.

Add Users

1. Click **Secure Web Settings** ⇨ **Policy Administration**. When prompted, enter the secure domain, user ID, and password. Click **Sign On**.
2. In the **Task List**, click **Expand User** ⇨ **Create User**.
3. Enter the following information:
   - User ID
   - Common Name
   - Surname
   - Password

     Note: The Password does not have to be the same password that is associated with the operating system user ID. It is recommended that you use a different password in ISAM.
   - Registry UID
4. Select the four options that are available, and then click **Create**.
5. Add the newly created User ID to the group, if the group is set up with permissions the user needs, by completing the following:
   a. Click **Search Users**. In the User Properties window, enter the new User ID, and then click **Search**.
   b. Click the User ID link to display its properties.
   c. Select the Groups tab and click **Add**.
   d. Click **Search** to find the groups. Select the appropriate group (sasusers).
   e. Click **Add**.

   For more information, see IBM Security Access Manager.
6. Proceed to "Add Web Authentication Domain to the User" on page 259.

Add Web Authentication Domain to the User

1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, click **User Manager**.
3. Add the user that was added to ISAM in the steps above. Right-click on the user and select **Properties**.
4. In the New User Properties window, select the Account, and click **New**.
5. In the New Login Properties window, enter a user ID, leave the **Password** fields blank, and then select **`web`** from the **Authentication Domain** drop-down list.
6. If **`web`** is not an available Authentication Domain option, follow these steps:
   a. On the **Plug-ins** tab, right-click **User Manager**, and select **Authentication Domains**.

b. In the Authentication Domains widow, click **New**.

c. In the New Authentication Domain window, enter **web** in the **Name** field, and then click **OK**.

d. Add the web authentication domain to the user by starting with Step 2 on page 259.

Note: If the user is going to use SAS Studio or do any work that uses SAS Workspace Server, the user's password for the DefaultAuth authentication domain must be entered into the user's **Accounts** tab in SAS Management Console. When this password is added to the **Accounts** tab, SAS Workspace Server can access to the operating system password, which is required for the server to complete tasks. This is also necessary since the ISAM password for a user does not have to be (and probably should not be) the same as the operating system password.

Modify Files in the Deployment

1. Configure BASIC web authentication by following the steps in the "Web Authentication" on page 245 procedure, with the following modification. Do not add users to tomcat-users.xml or configure a **Realm** in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file.

2. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file and locate the existing /Engine definition. Add the following valve definitions before </Engine>:

   - For all SASServer*n_m*, except SASServer2_1:

     ```
     <Valve
         className="com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve"
         userHeader="iv-user"/>
     <Valve
         className="com.sas.vfabrictcsvr.valves.WebSEALRequestWrappingValve"
         host="hostname"
         port="ISAM-reverse-proxy-port"/>
     ```

   - For SASServer2_1:

     ```
     <Valve
         className="com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve"
         userHeader="iv-user"
         asyncSupported="true"/>
     <Valve
         className="com.sas.vfabrictcsvr.valves.WebSEALRequestWrappingValve"
         host="hostname"
         port="ISAM-reverse-proxy-port"
         asyncSupported="true"/>
     ```

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

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

   Starting with SAS 9.4M8, modify or add the following JVM options:

   - For SASServer*n_m* where SAS Logon application is deployed, add the following option:

```
-Dsas.webseal.configured=true
```

- For SASServer1_1, change the following JVM options to point to the reverse proxy:

  ```
  -Dsas.scs.cas.scheme=protocol
  -Dsas.scs.cas.host=proxy.example.com
  -Dsas.scs.cas.port=port_number
  -Dsas.scs.svc.scheme=protocol
  -Dsas.scs.svc.host=proxy.example.com
  -Dsas.scs.svc.port=port_number
  ```

  Add the following option to specify the internal URL of the SAS Web Server:

  ```
  -Dsas.scs.svc.internal.url=http://host:port_number
  ```

- For all SASServer*n_m*, except SASServer1_1, change the following JVM options to point to the reverse proxy:

  ```
  -Dsas.scs.cas.scheme=protocol
  -Dsas.scs.cas.host=proxy.example.com
  -Dsas.scs.cas.port=port_number
  ```

For Windows deployments, add the JVM options to the \SASServer*n_m*\conf \wrapper.conf file. For UNIX deployments, add the JVM options to the file under / SASServer*n_m*/bin/setenv.sh file. See "Modify External Connections " on page 261.

Note: After you modify the wrapper.conf file for SAS 9.4M7 Feb 16, 2022 and later, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

4. Edit the ***SAS-configuration-directory*\Levn\Web\WebServer\conf \sas.conf** file and add the following under the RewriteEngine section:

   ```
   RewriteEngine On

   # one for SASStoredProcess

   RewriteCond %{QUERY_STRING} ^service=http%3A%2F%2F
   hostname%2Fjunction-name%2FSASStoredProcess%2Fj_spring_cas_security_check$

   RewriteRule ^/SASLogon/login$ /SASLogon/login?service=
   http://hostname/SASStoredProcess/j_spring_cas_security_check
   ```

Modify External Connections

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.

To change connection properties, follow these steps:

1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, expand **Application Management** ⇨ **Configuration Manager**.

3. Right-click the SAS web application name, and select **Properties**.
4. If the properties for an object have an **External Connection** tab, modify the External Connection to point to the following items:
   - the server that is running the ISAM instance
   - the port that was used when the ISAM reverse proxy instance was created
   - the service name used for the Internal Connection

   The External Connection should not be modified to point to the ISAM instance, but should instead remain pointing to the internal connection information (the default) for the following services:
   - BI Web Services for Java 9.4
   - DP-SAS-Environment-Manager
   - LASR Authentication Services (available in **Visual Analytics** ⇨ **Visual Analytics Services**)
   - Visual Analytics Hyperlink Service (available in **Visual Analytics** ⇨ **Visual Analytics Services**)
   - Web Infra Platform ClntAccess
   - Web Infra Platform Soap Svcs
   - Starting in SAS 9.4M8, if you are using SAS Workflow Studio: Workflow Web Services 9.4

     See Problem Note 69731: https://support.sas.com/kb/69/731.html.

ISAM WebSEAL Virtual Host Junction Creation Process

Overview

Within the WebSEAL instance, two types of junction can be created - the standard junction and the virtual host junction. SAS has supported the standard junction since SAS 9.1.3. Starting with SAS 9.4M7, support has been added for the virtual host junction.

The standard junction uses a junction name in the URL to identify the target host to send the request to. It also requires use of junction mapping table (JMT) to accommodate the lack of junction name in the server URL. When SAS 9.4 configuration includes a load balancer and ISAM WebSEAL, the standard junction with JMT might not work correctly.

Conversely, the virtual host junction uses the host name in the HTTP header to identify the junction rather than the junction name supplied with URL. The junction name in the URL is eliminated. Additional steps are required to create and configure the virtual host junction. The virtual host name, which conforms to your organization's convention, should be created, and it should be defined as a DNS alias to the WebSEAL instance's primary interface from DNS mapping tool.

The virtual host junction works better when your organization's infrastructure and SAS 9.4 configuration have a load balancer that is used with ISAM WebSEAL.

Configure the Virtual Host Junction

1. Log on to your ISAM V9 as an administrator.

   Note: This configuration process assumes that a WebSEAL instance is already defined and operational.

2. From Secure Web Setting, select the **Reverse Proxy from Manage** tab.
3. Select your WebSEAL instance and select **Junction Management** from the **Manage** drop-down menu.
4. In the Junction Management window, click **New** and select **Virtual Junction**.

   Enter the following parameters from the Junction Creation window.

   Note: The parameter values are only examples.

   a. On the **Junction** tab, set the fields that are specified in the following table:

   Table 17.2 *Junction Tab Fields and Values*

Field	Value
Junction Label	vhj1-label
Stateful Junction	Select this option to ensure session affinity.
Virtual Host Name	*vhost1*.unx.sas.com
Virtual Host Port	This is the WebSEAL instance listening port, 887.
Junction Type	If you configured HTTPS, TLS.

   b. On the **Server** tab, add the new Target Back-End Server where SAS 9.4 is deployed by setting the fields that are specified in the following table:

   Table 17.3 *Server Tab Fields and Values*

Field	Value
Hostname	*target.host.mycompany*.com
TCP or TLS Port	7980
UUID of the server	This value is filled in automatically.

   c. On the **Identity** tab, in the **HTTP Header Identity Information** field, select **IV-USER**.

   d. Click **Save**.

5. Starting with SAS 9.4M8, you must do the following:

   a. Click **Secure Web Settings** ⇨ **Reverse Proxy** ⇨ ***instance-name***.

b. Click **Manage** and select **Configuration** from the drop-down menu.

c. In the Configuration window, select **Edit Configuration File** from the drop-down menu.

d. In the Advanced Configuration File Editor window press Ctrl-F to open the **Find** search box.

e. Search for **`Method disablement`**.

f. Modify the entries in the *Method disablement* section of the file, so that they are blank. Here is an example:

```
http-method-disabled-local =
http-method-disabled-remote =
```

g. Click **Save**.

6. Complete the following items:

   a. In the ISAM console, from your WebSEAL instance, note the Primary Interface Address, which is the entry point of your WebSEAL instance.

   b. From your DNS mapping tool, add your virtual host (for example, *vhost1*.unx.sas.com) as an alias to your WebSEAL Primary Interface Address.

   When the WebSEAL instance receives the request with the virtual host name, it checks the virtual host junction that matches the virtual host name, finds the target server, and passes the request to the target server.

7. The same virtual host name should be used in the SAS 9.4 server definitions file (***`SAS-configuration-directory`***`\Levn\Web\WebAppServer\SASServern_m\conf\server.xml`) in place of WebSEAL host name.

   You can access the SAS 9.4 web application using virtual host name (for example, https:// *vhost1*.unx.sas.com:887/SASStoredProcess).

8. Proceed to "Configure SAS Web Applications and SAS Environment Manager to Use ISAM WebSEAL Virtual Host Junction" on page 264 to complete the configuration in the SAS deployment.

Configure SAS Web Applications and SAS Environment Manager to Use ISAM WebSEAL Virtual Host Junction

Add the Web Authentication Domain to the User

For a list of the steps that must be completed, see "Add Web Authentication Domain to the User" on page 259.

Modify the Files in the Deployment for SAS Web Applications

1. Configure BASIC web authentication. Follow the instructions in "Web Authentication" on page 245, with the following change. Do not add users to the ***`SAS-configuration-directory`***`\Levn\Web\WebAppServer\SASServern_m\conf\tomcat-users.xml` file or configure a realm in the

***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file.

2. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file and locate the existing /Engine definition. Add the following valve definitions before </Engine>:

   - For all SASServer*n_m*, except SASServer2_1:

     ```
     <Valve
         className="com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve"
         userHeader="iv-user"/>
     <Valve
         className="com.sas.vfabrictcsvr.valves.WebSEALRequestWrappingValve"
         host="vhost1"
         port="/ISAM-reverse-proxy-port"/>
     ```

     Note: In the previous definition, *vhost1* is the virtual host junction that is defined for the SAS web applications in ISAM.

   - For SASServer2_1:

     ```
     <Valve
         className="com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve"
         userHeader="iv-user"
         asyncSupported="true"/>
     <Valve
         className="com.sas.vfabrictcsvr.valves.WebSEALRequestWrappingValve"
         host="vhost1"
         port="/ISAM-reverse-proxy-port"
         asyncSupported="true"/>
     ```

     Note: In the previous definition, *vhost1* is the virtual host junction that is defined for the SAS web applications in ISAM.

Modify the JVM Options for SAS Web Applications

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

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

- Some web applications might require you to use the following additional JVM option when internal and external connection information must differ due to different access rules. This JVM option enables the Cross Domain Proxy Servlet to use internal URIs when external URIs are requested by applications:

  ```
  -Dsas.web.html.cdps.use.internal.urls=true
  ```

- If there is a basic authentication challenge with SASLogon during redirect, then add the following JVM option:

  ```
  -Dsas.scs.svc.internal.url=http(s)://hostname:port
  ```

  Where *hostname* is the SAS deployment web server machine, and *http(s)* and *port* are dependent on whether http or https is configured for the deployment, and whether the web server machine is on Windows (80 or 443) or UNIX (443 or 8343).

Starting with SAS 9.4M8, modify or add the following JVM options:

- For SASServer*n_m* where SAS Logon application is deployed, add the following option:

  ```
  -Dsas.webseal.configured=true
  ```

- For SASServer1_1, change the following JVM options to point to the reverse proxy:

  ```
  -Dsas.scs.cas.scheme=protocol
  -Dsas.scs.cas.host=proxy.example.com
  -Dsas.scs.cas.port=port_number
  -Dsas.scs.svc.scheme=protocol
  -Dsas.scs.svc.host=proxy.example.com
  -Dsas.scs.svc.port=port_number
  ```

- For all SASServer*n_m*, except SASServer1_1, change the following JVM options to point to the reverse proxy:

  ```
  -Dsas.scs.cas.scheme=protocol
  -Dsas.scs.cas.host=proxy.example.com
  -Dsas.scs.cas.port=port_number
  ```

For Windows deployments, add the JVM options to the **\SASServer*n_m*\conf\wrapper.conf** file. For UNIX deployments, add the JVM options to the file under the **/SASServer*n_m*/bin/setenv.sh** file.

Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

Modify the Files in the Deployment for SAS Environment Manager

To determine which values to change in the following two files for SAS Environment Manager, use the following table:

Table 17.4 *File Variables and Descriptions*

Variable	Description
http(s)	Depends on the protocol of the virtual host junctions as defined in ISAM.
vhost1	Specifies the virtual host junction that is defined for the SAS web applications in ISAM.
vhost2	Specifies the virtual host junction that is defined for SAS Environment Manager in ISAM.
port	Specifies the ISAM reverse proxy port number.

1. Edit the ***SAS-configuration-directory*/Lev1/Web/SASEnvironmentManager/server-5.8.0-EE/hq-engine/hq-server/webapps/ROOT/WEB-INF/spring/security-web-context.xml** file and complete the following:

a. From the **`<sec:filter-chain pattern="/**" filters="securityContextPersistenceFilter,`** section, remove **`basicAuthenticationFilter,`**.

b. Modify the following three locations in the file:

Important: Modify only these three instances, not every instance in the file.

-

```
 <!-- For logout -->
<bean id="logoutFilter" class="org.springframework.security.web.authentication.
logout.LogoutFilter">
      <constructor-arg value="http(s)://vhost1:port/SASLogon/logout
      ?sas_svcs_logon_LogonUrl=http%3A%2F%2Fvhost2%3Aport%2F" />
```

-

```
 <!-- For form-based authentication -->
<bean id="casAuthenticationEntryPoint" class="org.springframework.security.
cas.web.CasAuthenticationEntryPoint">
      <property name="loginUrl" value="http(s)://vhost1:port/SASLogon"/>
```

-

```
<sec:http entry-point-ref="casAuthenticationEntryPoint">
<sec:logout invalidate-session="true" logout-url="/logout"
logout-success-url="/cas/logout"/>
</sec:http>
        <bean id="casService"
      class="org.springframework.security.cas.ServiceProperties">
      <property name="service"
        value="http(s)://vhost2:port/j_spring_cas_security_check"/>
```

2. Edit the ***`SAS-configuration-directory`*`/Lev1/Web/SASEnvironmentManager/server-5.8.0-EE/hq-engine/hq-server/webapps/ROOT/WEB-INF/web.xml`** file and modify all instances of the following code in the file:

```
<context-param>
    <param-name>ModuleFrameworkURL</param-name>
    <param-value>http(s)://vhost1:port/SASEnvironmentMgrMidTier</param-value>

<param-name>timeout-url</param-name>
        <param-value>
http(s)://vhost1:port/SASLogon/TimedOut.do?sas_svcs_logon_LogonUrl=
http%3A%2F%2Fvhost2%3Aport%2F</param-value>
```

Modify the External Connections for Both SAS Web Applications and SAS Environment Manager

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.

To change connection properties, follow these steps:

1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, expand **Application Management** ⇨ **Configuration Manager**.
3. Right-click the SAS web application name, and select **Properties**.
4. If the properties for an object have an **External Connection** tab, modify the External Connection to point to the following items:

   For web applications to be authenticated through ISAM virtual host junction defined for SAS web applications (which is most of the SAS web applications listed):

   - the ISAM virtual host junction defined for the SAS web applications
   - the port that was used when the ISAM reverse proxy instance was created
   - the service name used for the Internal Connection

   For DP-SAS-Environment-Manager to be authenticated through ISAM virtual host junction defined for SAS Environment Manager:

   - the ISAM virtual host junction defined for SAS Environment Manager
   - the port that was used when the ISAM reverse proxy instance was created
   - the service name used for the Internal Connection

   The External Connection should *not* be modified to point to the ISAM instance, but should instead remain pointing to the internal connection information (the default) for the following services:

   - BI Web Services for Java 9.4
   - LASR Authentication Services (available in **Visual Analytics** ⇨ **Visual Analytics Services**)
   - Visual Analytics Hyperlink Service (available in **Visual Analytics** ⇨ **Visual Analytics Services**)
   - Web Infra Platform ClntAccess
   - Web Infra Platform Soap Svcs

Designate Virtual Host Junctions

Designate virtual host junctions as Allowed Sites for SAS deployment and authorize SAS Environment Manager to use SAS Logon Manager by completing the following steps:

1. Log on to SAS Management Console as an administrator.
2. On the **Plug-ins** tab, expand **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**.
3. Right-click and select **Properties** ⇨ **Advanced**.
4. Make the following modifications:

   a. Add the two virtual host junctions to sas.web.csrf.referers.knownHosts property:

   ```
   http(s)://vhost1:port,http(s)://vhost2:port
   ```

Table 17.5 *File Variables and Descriptions*

Variable	Description
http(s)	Depends on the protocol of the virtual host junctions as defined in ISAM.
vhost1	Specifies the virtual host junction that is defined for the SAS web applications in ISAM.
vhost2	Specifies the virtual host junction that is defined for SAS Environment Manager in ISAM.
port	Specifies the ISAM reverse proxy port number.

b. Add the following to the Advanced Properties:

```
Property Name: ServiceUrl.Allowed
Property Value: http(s)://vhost2:port/**
```

For more information, see this SAS Note: Problem Note 56451: The error "The Application is not authorized to use SAS Logon Manager" occurs when you try to log on to SAS® Environment Manager.

Restart the Deployment

For a list of the steps that must be completed, see "Restart the SAS Middle Tier" on page 324.

Note: SAS Environment Manager uses ISAM WebSEAL for authentication, but the user must log in twice because there are two virtual host junctions.

Support for Symantec SiteMinder (Formerly Known as CA Single Sign-On)

Overview

SAS 9.4 support for Symantec SiteMinder (formerly known as CA Single Sign-On) 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 Symantec SiteMinder environments.

Note: CA Single Sign-on is referred to as Symantec SiteMinder for the rest of this chapter.

Retrieve Required Symantec SiteMinder Applications

SAS 9.4 integration with Symantec SiteMinder depends on two software applications from CA:

- Symantec SiteMinder Web Agent (any version)
- Symantec 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.)

Also Included in the download package from Symantec SiteMinder are API JAR files. 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 17.6 *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.
smagentapi.jar	cryptoj.jar	This JAR file requires the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files.

You will create two host configurations:

- a web agent to use with SAS Web Server
- a separate host configuration for the agent to use with SAS Web Application Server

For information about configuring the agents and the policy servers, see the documentation for Symantec SiteMinder.

Configure the Java Cryptography Extension

Symantec SiteMinder Pure Java API requires that the Java environment that is used by SAS Web Application Server be updated with the JCE. Retrieve the API JAR files and copy them to the SAS Private JRE: ***SAS-home-directory/*** **SASPrivateJavaRuntimeEnvironment/9.4/jre/lib/security**.

For details, see "Configure the Java Cryptography Extension" on page 309.

Configure 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 276.

The custom security module for Symantec 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: Symantec SiteMinder provides a configuration utility. However, on Windows, it does not recognize SAS Web Server, so manual configuration is necessary.

Register the SAS Web Server Host

To register the machine with the Symantec SiteMinder policy server, perform the steps below. Use this table for sample values .

Table 17.7 *Sample Values for Agent Configurations*

Property Name	Value
Policy server	**`policyserver.example.com`**
Admin user name	**`siteminder`**
Admin password	**`Pass`**
Host configuration and host name	***`hostname`*****`_apache`** for the SAS Web Server ***`hostname`*****`_tc`** for SAS Web Application Server
Agent name	**`sasagent`**
Agent configuration	**`sasagentconf`**

- For Windows deployments:
  1. On Windows 64-bit platforms, copy the ICE_JNIRegistry.dll from **`C:\Windows\System32`** to **`C:\Windows\SysWOW64`**.
  2. Run the **smreghost.exe** command in the bin directory under the Web Agent installation to register the host with the policy servers:

```
CA-webagent-install-directory\webagent\win64\bin\smreghost.exe
-i policyserver.example.com
-u siteminder -p adminPassword -hc hostname_apache
-hn hostname_apache -o
-f CA-webagent-install-dir\webagent\win64\config\SmHost.conf
```

Note: You need at least **hostname_ws** and **hostname_tc** trusted host objects, as they cannot be shared.

- For UNIX deployments:
  - Source the environment file: **ca_wa_env.sh**:

    ```
    CA-webagent-install-directory/ca_wa_env.sh
    ```

The command generates the **SmHost.conf** file, if successful.

Configure SAS Web Server for the Web Agent in SAS 9.4M7 and SAS 9.4M8

TIP You can try to use the Symantec 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:

- For Windows deployments:
  1. Create a **WebAgent.conf** file in the ***SAS-configuration-directory*\Levn\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="\CA-webagent-install-dir\webagent\win64\config\SmHost.conf"
     AgentConfigObject="webserver_agent_conf"
     EnableWebAgent="YES"
     ServerPath="SAS-config-directory/Lev1/Web/WebServer/conf"
     LoadPlugin="CA-webagent-install-dir\webagent\win64\bin\HttpPlugin.dll"
     AgentIdFile="SAS-config-directory\Lev1\Web\WebServer\conf\AgentId.dat"
     ```

     Important: You must use forward slashes (/) for the ServerPath.

  2. Edit the ***SAS-configuration-directory*\Levn\Web\WebServer\conf\httpd.conf** file. Add lines that are similar to the following at the beginning of the LoadModule directive:

     ```
     LoadModule sm_module "CA-webagent-install-dir/webagent/win64/bin/mod_sm24.dll"
     SmInitFile "SAS-config-directory/Lev1/Web/WebServer/conf/WebAgent.conf"
     ```

     Note: In earlier releases, the name of the library for the LoadModule directives is **mod_sm22.dll**.

  3. In the same *httpd.conf* file, add the following lines within the **<IfModule alias_module>** section. Change the paths to match the location of the Web Agent software on your machine.

     ```
     Alias /siteminderagent/nocert/[0-9]+/(.*) "CA-webagent-install-dir/webagent/$1"
     <Directory "CA-webagent-install-directory/webagent/win64/$1">
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
     </Directory>

     Alias /siteminderagent/pwcgi/ "CA-webagent-install-directory/webagent/pw/"
     <Directory "CA-webagent-install-directory/webagent/win64/pw/">
        Options Indexes MultiViews ExecCGI
        AllowOverride None
        Order allow,deny
        Allow from all
     ```

```
</Directory>

Alias /siteminderagent/pw/ "CA-webagent-install-directory/webagent/pw/"
<Directory "CA-webagent-install-directory/webagent/win64/pw/">
   Options Indexes MultiViews ExecCGI
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

Alias /siteminderagent/ "CA-webagent-install-directory/webagent/win64/samples/"
<Directory "CA-webagent-install-directory/webagent/win64/samples/">
   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
```

4. Stop SAS Web Server.
5. Start SAS Web Server.

   Note: If the SAS Web Server fails to start, run the following executable from an administrator command prompt:

   ```
   editbin /STACK:524288 "SAS-Home-dir\SASWebServer\9.4\httpd-2.4.54\bin\httpd.exe"
   ```

   If the **editbin** executable is not installed, you can download and install Visual Studio Express 2022, which includes the **editbin** binary. You can find it at: **C:\Program Files\Microsoft Visual Studio \2022\Enterprise\VC\Tools\MSVC\14.34.31933\bin \Hostx64\x86\editbin**. Then restart the SAS Web Server.

6. Access *http://hostname:7980/SASLogon* to confirm a logon prompt is given, and log in with your credentials.

- For UNIX deployments:

  1. Run the following script with the *-i console* flag for console mode:

  ```
  /CA-webagent-install-dir/ca-wa-config.sh -i console
  ```

  Below are example values for the prompts:

  - Choose *1* to do host registration.
  - Admin user: *siteminder*
  - Shared Secret Rollover: *default of no*
  - Password: *adminPassword*
  - Trusted host name:*typically use hostname, or some unique string*
  - Host Configuration Object:*host_config*
  - Policy Server: *policyserver.example.com*
  - FIPS: *compatibility mode*
  - File Name: *Default of SmHost.conf*
  - Location: *Default value*
  - Web Server: *Apache*
  - Path: *SAS_config_directory/Lev1/Web/WebServer*

- Version: *2.4* (check your httpd's version)
- Type: *ASF/RedHat Apache* and select *1* for the web server that it finds
- Agent Object: *webserver_agent_config*
- SSL: *No advanced authentication*
- Enable: *Yes*
- *Continue*

2. Restart the SAS Web Server.
3. Access *http://hostname:7980/SASLogon* to confirm a logon prompt is given, and log in with your credentials.

Configure SAS Web Server for the Web Agent in SAS 9.4 M6 and Earlier Releases

TIP You can try to use the Symantec 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*\Levn\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="\CA-webagent-install-directory\webagent\config\SmHost.conf"
AgentConfigObject="webserver_sasagent_conf"
EnableWebAgent="YES"
ServerPath="SAS-config-directory/Lev1/Web/WebServer/conf"
LoadPlugin="CA-webagent-install-directory\webagent\bin\HttpPlugin.dll"
AgentIdFile="SAS-config-directory\Lev1\Web\WebServer\conf\AgentId.dat"
```

   Note: You must use forward slashes (/) for the ServerPath.

   For UNIX deployments, the library for the LoadPlugin property is named libHttpPlugin.so instead of HttpPlugin.dll.

2. Edit the ***SAS-configuration-directory*\Levn\Web\WebServer\conf\httpd.conf** file. Add lines that are similar to the following at the beginning of the LoadModule directives:

```
LoadModule sm_module "CA-webagent-install-dir/webagent/bin/mod_sm22.dll"
SmInitFile "SAS-config-directory/Lev1/Web/WebServer/conf/WebAgent.conf"
```

   For UNIX deployments, the name of the library is libmod_sm22.so instead of mod_sm22.dll.

   Note: If you installed the SAS Web Server hot fix that updated the Apache HTTP Server from version 2.2.*x* to version 2.4.*x*, the name of the library for the LoadModule directives is one of the following:

   - For Windows deployments: mod_sm24.dll
   - For UNIX deployments: libmod_sm24.so

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.

```
<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/pwcgi/ "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.

Troubleshoot 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*\Levn\Web\WebServer\conf**. Include the following lines:

```
components:  AgentFramework, HTTPAgent, WebAgent
data: Date, Time, Pid, Tid, TransactionID, Function, Message
```

2. Use the Symantec SiteMinder Administrative UI to set the trace properties for the agent configuration. The following table provides sample values:

Table 17.8 *Sample Values for Symantec SiteMinder Web Agent Troubleshooting*

Property Name	Value
TraceAppend	**Yes**
TaceConfigFile	**C:\SAS\Config\Levn\Web\WebServer\conf\WebAgentTrace.conf**

Property Name	Value
TraceFile	Yes
TraceFileName	**C:\SAS\Config\Levn\Web \WebServer\logs\webagent.trace**
TraceFileSize	**100**

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*\Levn \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.

Here 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_SASServern_m
</Proxy>

--------------------------------
  This modified sas.conf file must be added
  to the httpd.conf file o reverse proxy server.
  Here is an example:
--------------------------------
<IfModule mod_proxy.c>
  Include conf/sas.conf
</IfModule>
```

Configure 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.

Configure Web Authentication

Follow the steps in the "Web Authentication" procedure, but specify **SiteMinderAuthenticated** for the role-name element in the web.xml.orig file.

Register 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 SAS Web Server runs on the same host as SAS Web Application Server.

To register the machine with the Symantec SiteMinder policy server, follow these steps:

1. Verify the values for the **JAVA_HOME** and **SM_REGHOST_CLASSPATH** environment variables. You can use the *echo* command, for example:

   - On Windows:

     ```
     echo %JAVA_HOME%
     ```

   - On UNIX:

     ```
     echo $JAVA_HOME
     ```

   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 Symantec SiteMinder SDK **java** or **java64** directories.

2. Navigate to the bin directory where the Symantec SiteMinder SDK is installed. Run the **smreghost.exe** script to register the host with the policy servers:

   ```
   smreghost.exe -i policyserver.example.com -u siteminder -p Pass
        -hc hostname_tc -hn hostname_tc
        -o -f "SAS-config-directory\Lev1\Web\WebAppServer\SASServer1_1\conf\SmHost.conf"
   ```

   Note: On UNIX, the script is **smreghost.sh**. Make sure you have sourced the **ca_wa_env.sh** script.

   If successful, the command generates the **SmHost.conf** file.

Configure SAS Web Application Server for the Web Agent in SAS 9.4M8

SAS Web Application Server must be configured to accept the user name that is sent by SAS Web Server in the HTTP header. The SAS 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.

To configure the server, follow these steps:

1. Edit the ***SAS-home-directory*/SASWebInfrastructurePlatform/9.4/Static/wars/sas.svcs.logon/META-INF/context.xml** file. Add the following **Valve** configuration:

   ```
   <Valve className=
   "com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve"
   secretPassword="${pw.sas.valve.PrincipalFromRequestHeadersValve}"
   userHeader="sm_user" />
   ```

   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.

   Note: To set up a secret password, see "Configure a Secret Password" on page 303.

The security module supports the following parameters:

Table 17.9 *SAS Web Application Server Security Module Attributes*

Attribute	Default Value	Description
uriPattern	/SASLogon/login.*$	Specifies the process requests with a URI, including query string, matching this regular expression.
fallThrough	false	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.
secretPassword	None	Specifies the secret password to expect in the Basic Authorization header (Optional).
userHeader	X-Remote-User	Specifies the HTTP header containing the authenticated subject name.
roleName	ROLE_USER	Specifies the role to associate with the principal. This value is usually not needed.

Note: For the SAS Web Application Server where SAS Studio is deployed (as SASServer2_1 or SASServer1_1 if the deployment is a single managed server), add the following option in the valve definition:

```
asyncSupported="true"
```

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.

2. OPTIONAL: You can enable debug logs for authenticator app by adding the following tag to the ***`SAS-config-directory`***`/Lev1/Web/WebAppServer/SASServer1_1/log4j2/conf/log4j2.xml` file:

```
<Logger name="com.sas.vfabrictcsvr.authenticator" level="debug" additivity="false">
      <AppenderRef ref="FILE"/>
      <AppenderRef ref="CONSOLE"/>
    </Logger>
```

3. Restart SAS Web Application Server.

Configure SAS Web Application Server for the Web Agent in SAS 9.4 M7 and Earlier Releases

To configure the server, follow these steps:

1. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file and locate the existing **/Engine** definition. Add the following valve definition:

```
<Valve
    className="com.sas.svcs.security.vfabrictcsvr.siteminder.SiteMinderValve"
    role="SiteMinderAuthenticated"
    agentName="sasagent"
    validateClientIp="true"
    cookieName="SMSESSION"
    webagentConf="${catalina.base}/conf/WebAgent.conf" />
```

Table 17.10 *Symantec SiteMinder Security Module Attributes*

Attribute	Default Value	Description	Required
role	SiteMinderAuthenticated	Specifies the name of the role to add to authenticated principals.	No
agentName	None	Specifies the name of the agent that was specified in the Administrator UI. *Note:* If you are configuring Symantec SiteMinder for SAS Visual Analytics App, then the value that you specify for the agentName attribute must correspond to the value that you specify for the agent associated with the realms filtering the SASVisualAnalyticsTransport resources. For a list of resources, see Table 17.11 on page 282.	Yes
validateClientIp	true	Specifies whether to enable client IP checking when set to true. It might be necessary to disable client IP checking (by setting this value to false) in some external proxy configurations.	No
cookieName	SMSESSION	Specifies the name of the session cookie that is used by Symantec SiteMinder.	No
webagentConf	None	Specifies the path to the WebAgent.conf file.	Yes

Note: For the SAS Web Application Server where SAS Studio is deployed (as SASServer2_1 or SASServer1_1 if the deployment is a single managed server), add the following option in the valve definition:

```
asyncSupported="true"
```

2. Copy the sas.svcs.security.vfabrictcsvr.siteminder.jar file from the ***SASHOME*\SASWebApplicationServer\9.4\templates\sas\lib** directory to ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\lib**.

3. Create a WebAgent.conf file in the ***SAS-configuration-directory*\Levn\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 Symantec SiteMinder for SAS Visual Analytics App, then the value that you specify for the AgentConfigObject attribute must correspond to the agentName attribute that you specified in Step 1 on page 279.

4. In the same server.xml file, complete the following steps:

   a. Check the values for proxyName and proxyPort in the existing **/Connector** definition.

   b. Check the values for httpServerPort, httpsServerPort, protocolHeader, and internalProxies in the existing **RemoteIpValve** definition.

   If you are using an external proxy, change the values so that they match the proxy instead of SAS Web Server, and specify the IP address of the external proxy by adding a trustedProxies attribute. For more information, see "Configure the Middle Tier to Use an Existing Customer Reverse Proxy" on page 240.

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
```

Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

For UNIX deployments, edit the **SASServern_m\bin\setenv.sh** file and make changes that are similar to the following example:

```
CLASSPATH="/opt/CA/sdk/java/smagentapi.jar:/opt/CA/sdk/java/cryptoj.jar"
```

6. Restart SAS Web Application Server.

Troubleshoot 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*n*_*m*\lib\log4j.xml** file and add the following lines:

```
<category name="com.sas.svcs.security.vfabrictcsvr">
    <priority value="DEBUG"/>
</category>
```

2. Restart SAS Web Application Server and monitor the **SASServer*n*_*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
successfull
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 shortened to SiteMinderValue for readability in the example.

4. After a successful logon attempt, the log contains messages like the following example:

```
yyyy-mm-dd 14:02:31,404 DEBUG (tomcat-http--42) [SiteMinderValve] GET /SASLogon/
login from 192.168.99.37
yyyy-mm-dd 14:02:31,406 DEBUG (tomcat-http--42) [SiteMinderValve] Request has a
SMSESSION token
yyyy-mm-dd 14:02:31,412 DEBUG (tomcat-http--42) [SiteMinderValve] Resource
'/SASLogon/login' is protected by SiteMinder realm hostname.example.com_tcServer
Realm
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve]
200=hostname.example.com_tcserver
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve]
208=192.168.14.248
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve] 205=RreaOHVxS3N
+1/Y9vQqRFmWOCfU=
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve]
218=CN=sasdemo,OU=People,DC=EXAMPLE,DC=COM
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve] 210=sasdemo
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve] 154=1360867934
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve] 225=3600
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve] 155=1360868551
yyyy-mm-dd 14:02:31,418 DEBUG (tomcat-http--42) [SiteMinderValve] 226=7200
yyyy-mm-dd 14:02:31,422 DEBUG (tomcat-http--42) [SiteMinderValve] SiteMinder
session
for user sasdemo has been verified
```

Configure the Policy Server

Configure the Realm

In the Symantec 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 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 Symantec 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 Symantec 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 17.11 *Resources to Protect*

SAS Application	Resource
SAS BI Web Services	/SASBIWS
SAS Visual Analytics App	/SASVisualAnalyticsTransport/onebi/logon
	/SASVisualAnalyticsTransport/rest/session

Special Considerations for Agent Configuration Parameters

The following table identifies some agent configuration parameters that are known to cause problems in a SAS deployment:

Parameter	Issue
BadUrlChars	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.
RequiredCookies	This parameter can interfere with clients that use single sign-on authentication to SAS web services. Set this parameter to **no** if access to web services is affected.

Revert Configurations to Use SiteMinder after Update in Place to SAS 9.4M7 and Later

After performing an update in place to SAS 9.4M7 and later, the following SiteMinder configuration might need to be reverted from backup files.

1. Open ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServern_m/conf/server.xml** file. Check if following valve definition is still existing. If it is not, revert it using the backup server.xml file that is in the ***SAS-configuration-directory*/Levn/Web/WebAppServer/Backup/SASServern_m.xxx/conf** directory.

```
<Valve
    className="com.sas.svcs.security.vfabrictcsvr.siteminder.SiteMinderValve"
    role="SiteMinderAuthenticated"
    agentName="sasagent"
    validateClientIp="true"
    cookieName="SMSESSION"
    webagentConf="${catalina.base}/conf/WebAgent.conf" />
```

2. Check if the svcs.security.vfabrictcsvr.siteminder.jar exists under the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern__m\lib**. If it does not exist, copy the sas.svcs.security.vfabrictcsvr.siteminder.jar file from the_***SASHOME*\SASWebApplicationServer\9.4\templates\sas\lib** directory.
3. Check if the WebAgent.conf file exists in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf** directory. If it does not exist, restore it from the backup file in the ***SAS-configuration-directory*/Levn/Web/WebAppServer/Backup/SASServern_m.xxx/conf** directory.
4. Check if the SmHost.conf file exists in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer1_1\conf** directory. If it does not exist, restore it with the backup file in the ***SAS-configuration-directory*/Levn/Web/WebAppServer/Backup/SASServer1_1.xxx/conf** directory
5. Check if

```
CLASSPATH="/opt/CA/sdk/java/smagentapi.jar:/opt/CA/sdk/java/cryptoj.jar"
```

exists in the **SASServern_m\bin\setenv.sh** for UNIX deployments and **SASServern_m\conf\wrapper.conf** for Windows deployments. If it does not exist, restore this configuration with the backup setenv.sh or wrapper.conf in the ***SAS-configuration-directory*/Levn/Web/WebAppServer/Backup/SASServern_m.xxx/bin** directory.

Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

6. Ensure that the following configuration SPNEGO authentication in the web.xml file in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\sas_webapps\sas.svcs.logon.war\WEB-INF** directory is not overwritten after updating.

   You can restore this configuration by following the steps in the Web Authentication on page 245 procedure to specify **SiteMinderAuthenticated** for the role-name element in the xml.orig file or back up this file before performing the update in place to SAS 9.4M7.

```
<!-- Enable SPNEGO authentication -->
 <security-constraint>
  <web-resource-collection>
    <web-resource-name>HTMLHostManager and HostManagercommands</web-resource-name>
    <url-pattern>/login</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>SiteMinderAuthenticated</role-name>
  </auth-constraint>
 </security-constraint>
 <login-config>
   <auth_method>BASIC</auth-method>
   <realm-name>Tomcat Host Manager Application</realm-name>
 <login-config>
```

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 unsecure 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.

Note: For the supported web browsers to pass a ticket that was forwarded to the SAS middle-tier machine, the service account in Active Directory that is holding the SPNs must be trusted for delegation.

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 Edge 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 information from your information technology support group, also request the following information:
  - keytab file
  - 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.

CAUTION:
User roles should be assigned before configuring IWA. Otherwise, manual changes to the Shared Services database are required to assign roles to a user or group. Manual changes are not recommended.

Verify Prerequisites

Verify 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:

   setspn -F -Q HTTP/*hostname.example.com*

Output 17.1 *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:

   setspn -L *user-logon-name*

Output 17.2 *Sample Account Query*

```
Registered ServicePrincipalNames for CN=user-logon-name,OU=Service
Accounts,OU=Servers,DC=EXAMPLE,DC=com:
        HTTP/hostname.example.com
        HTTP/hostname
```

   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.

Verify 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 must be copied to the machine that is used for SAS Web Application Server and the file must be readable by the user account that is running SAS Web Application Server. The file should not be readable by other accounts.

The command for verifying that a keytab depends on the operating environment:

Windows Specifics

```
ktab.exe –l –k FILE:keytab-filename-and-path.keytab

Keytab name: keytab-filename-and-path.keytab
KVNO    Principal
-----   ------------------------------------
  1     HTTP/hostname@EXAMPLE.com
```

UNIX Specifics

```
ktutil
rkt keytab-filename-and-path.keytab
list -e
```

```
slot KVNO Principal
---- ---- -------------------------------------------------------------
   1    3    HTTP/hostname@EXAMPLE.com (arcfour-hmac)
```

TIP The principal name, HTTP/*hostname@EXAMPLE*.com, is used in configuring SAS Web Application Server in Step 5 on page 288.

TIP In addition, 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 on Windows and in the **/usr/bin** directory on UNIX systems.

```
kinit -k –t keytab-filename-and-path.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.

Configure SAS Web Application Server

The information in this section is modified from http://tomcat.apache.org/tomcat-8.0-doc/windows-auth-howto.html.

Repeat the following steps for each SASServer*n_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
   ```

   - If your Kerberos configuration file is saved in the ***SAS-configuration-directory*\Lev1\Web\WebAppServer\SASServer*n_m*\conf** directory, you do not have to set the JVM option.

- Java expects forward slashes in the keytab path name, even on Windows systems.
- For Windows deployments, edit the following files:
  - ***SAS-configuration-directory*\Lev1\Web\WebAppServer\SASServern_m\bin\setenv.bat**
  - ***SAS-configuration-directory*\Lev1\Web\WebAppServer\SASServern_m\conf\wrapper.conf**

    Note: Add the JVM option at the end of the Java Additional Parameters section.

    Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.
- For UNIX deployments, edit the ***SAS-configuration-directory*/Lev1/Web/WebAppServer/SASServern_m/bin/setenv.sh** file.

4. If AES-256 encryption ciphers are used, you must use the Java Cryptography Extension. For more information, see "Configure the Java Cryptography Extension" on page 309.

   For information about how to determine whether the machine is using AES-256 encryption, see the *SAS Intelligence Platform: Security Administration Guide*.

5. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\jaas.config** file. Add the following to the end of the file:

```
com.sun.security.jgss.krb5.initiate {
   com.sun.security.auth.module.Krb5LoginModule required
   doNotPrompt=true
   principal="principal-name-in-http-keytab"
   useKeyTab=true
   keyTab="C:/keytab-filename-and-path.keytab"
   storeKey=true;
};

com.sun.security.jgss.krb5.accept {
   com.sun.security.auth.module.Krb5LoginModule required
   doNotPrompt=true
   principal="principal-name-in-keytab"
   useKeyTab=true
   keyTab="C:/keytab-filename-and-path.keytab"
   storeKey=true;
};
```

   Note: Java expects forward slashes in the keytab path name, even on Windows systems.

   Note: The C:/*keytab-filename-and-path*.keytab file should contain only the HTTP principals.

6. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file. For the *Realm className* option, you must replace the following:

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
```

with this:

```
<Realm className="com.sas.vfabrictcsvr.realm.GSSContextEstablishedRealm"
       allRolesMode="authOnly"/>
```

For deployments that include SAS Visual Analytics App, where IWA is implemented for sign-ins, SAS Visual Analytics App can be configured to use BASIC authentication and authenticate credentials with Active Directory using LDAP. For the LDAP authentication, configure JNDIRealm. For more information, see "Configure the Realm for SAS Web Application Server" on page 251.

For single server deployments, the JNDIRealm can be located with the GSSContextEstablishedRealm. For multiple server deployments, the JNDIRealm needs to go in the server.xml file for SASServer12. In both instances, the JNDIRealm is nested inside the LockOutRealm. For more information about realm configuration, see http://tomcat.apache.org/tomcat-8.0-doc/realm-howto.html.

Configure Web Authentication

Follow the steps in the "Web Authentication" on page 245 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.

Configure 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.
5. Confirm the changes by specifying the URL for a SAS web application. You should not be prompted for credentials.

   Do not open a browser from the machine that is used for the SAS Web Server. This does not work. You must use another computer to confirm that the steps were performed correctly.

Configure Google Chrome and Microsoft Edge to Use SPNEGO

Configure Security Settings

To configure the security settings, follow these steps:

1. In the Windows Control Panel, open Internet Options.
2. In the Internet Properties window, select the **Security** tab.

3. Select **Local intranet** and then click **Sites**.
4. 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 the browser recognizes any site with your domain name as the intranet.
   c. Click **Close**, and then click **OK**.
5. 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**.

Configure Connection Settings

If your site uses a proxy server, follow these steps:

1. In the Internet Properties window, 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 window.
6. Click **OK**.

Configure Advanced Settings

To use Integrated Windows Authentication, follow these steps:

1. In the Internet Properties window, select the **Advanced** tab.
2. Scroll to the **Security** section, and verify that **Enable Integrated Windows Authentication** is selected.
3. Click **OK** and restart your computer to activate the changes.

Confirm 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 the SAS Web Server. This does not work. You must use another computer to confirm that the steps were performed correctly.

(Optional) Configure User Delegation

Overview

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*.

Configure 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 after configuring Integrated Windows Authentication:

1. Open the Windows registry editor.
2. Add the following REG_SZ keys:
   - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome \AuthServerAllowlist Specifies which servers should be allowed for integrated authentication. Set the value to the SAS middle-tier host name: *hostname.example.com*.
   - \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome \AuthNegotiateDelegateAllowlist Specifies servers to which Chrome can delegate. Set the value to the SAS middle-tier host name: *hostname.example.com*.

Configure User Delegation for Mozilla Firefox

To configure user delegation for Mozilla Firefox, 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.delegation-uris** Preference Name, enter **`http://`*`hostname.example.com`*`,`** in the **Enter string value** field, and then click **OK**.

Configure User Delegation for Microsoft Edge

To configure user delegation for Microsoft Edge, follow these steps:

1. Open the Windows registry editor.
2. Add the following REG_SZ keys:

   \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge \AuthServerAllowlist
   Specifies which servers to enable for integrated authentication. Set the value to the SAS Web Server host name: *hostname.example*.com.

   \HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge \AuthNegotiateDelegateAllowlist
   Specifies which servers Microsoft Edge can delegate to. Set the value to the SAS Web Server host name: *hostname.example*.com.

Configure the Middle Tier

To enable delegation of credentials to the server tier, configure the JAAS log on module to use Kerberos. The module is used by Platform Foundation Services (PFS). Add the following entries to PFS in the jaas.config file for each SASServer*n_m* instance:

```
PFS {
  com.sas.services.security.login.OMILoginModule  required
       "host"="hostname.example.com"
       "port"="port_number"
       "repository"="Foundation"
       "domain"="DefaultAuth"
       "trusteduser"="sastrust@saspw"
       "trustedpw"="{SAS005}ADD8AB7108595A7D1A69190D78CDFE6145C1EB849CC7A43D
       "aliasdomain"="DefaultAuth"
       "idpropagation"="sspi"
       "sspisecuritypackagelist"="KERBEROS"
       "debug"="false";
};
```

Update 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 window 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 window.

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 falls back to the default authentication, SAS form-based authentication.

SAS Grid Manager does not support fallback. If the user attempts to access SAS Grid Manager using fallback, login is possible. However, the product will not function in the expected way.

Note: If your web browser does not support IWA, authentication falls back to SAS form-based authentication, which is provided by SAS Logon Manager.

Note: Starting with SAS 9.4 M8, SAS uses the enhanced security feature of CAS verion 6.6 on page 118 . To enable fallback authentication:

- Use a web browser that is not configured for IWA.
- For web browsers that are configured for IWA, use the following URL to force fallback authentication:

  ```
  https://<hostname>:<port>/SASLogon/login?service=https%3A%2F%2F<hostname>
  %3A<port>%2F<SASAPPContext>%2Flogin%2Fcas&fallback=true
  ```

Configure 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 284.
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.vfabrictcsvr.authenticator.SasFallbackAuthenticatorValve"
       authMethod="SPNEGO" />
   ```

   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.

   The security module supports the following parameters:

   Table 17.12 *Fallback Security Module Attributes*

Attribute	Default Value	Description
uriPattern	/SASLogon/login.*$	Specifies whether to process requests with a URI, including query string, matching this regular expression.
authMethod	None	Specifies the primary authentication method to use: BASIC, DIGEST, or SPNEGO.

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.

   For SAS 9.4M7 release and prior:

   ```
   <!--
       <error-page>
           <error-code>401</error-code>
           <location>/WEB-INF/view/jsp/default/ui/401Fallback.jsp</location>
       </error-page>
   -->
   ```

   For SAS 9.4M8 release:

```
<!--
    <error-page>
        <error-code>401</error-code>
        <location>/templates/error/401.html</location>
    </error-page>
-->
```

Note: As an alternative to updating the web.xml.orig, you can edit the deployed file, ***SAS-configuration-directory*\Levn\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, comment out 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

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.

Configure Middle-Tier Services for SAS 9.4M2 and Previous Releases

For SAS 9.4M2 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
```

Configure Middle-Tier Services for SAS 9.4M3

For SAS 9.4M3, import the certificate of the CA that signed the client certificate from the trusted CA bundle, using the SAS Deployment Manager. For more information, see *SAS Deployment Wizard and SAS Deployment Manager: User's Guide*, available at http://support.sas.com/documentation/installcenter/en/ikdeploywizug/66034/PDF/default/user.pdf.

Configure TLS for SAS Web Server and SAS Web Application Server

Overview

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 "Configure SAS Web Server Manually for HTTPS" on page 312.

Configure SAS Web Server to Pass the Client Certificate to SAS Web Application Server

Starting with SAS 9.4M8, the Middle-Tier is configured with OpenSSL 3.0 and the Java TLS implementation that supports TLS 1.3 and FIPS. Not all web browsers support post-handshake authentication with TLS 1.3. Currently, only later versions of Mozilla Firefox support it. If you are using Firefox as your web browser, you must enable post-handshake authentication:

1. Open Firefox.
2. In the address bar, type *about:config*.
3. Click **Accept the Risk and Continue**.
4. In the search field, type *handshake*.
5. Set the Boolean value of *security.tls.enable_post_handshake_auth* to true.
6. Close your browser and restart Firefox.

Note: If you want to use Google Chrome or Microsoft Edge and TLS 1.3 with client Certificate Authentication, you need to make changes that are specified in the **Note** in step 2. If the client has a certificate available, it is authenticated during initial handshake rather than post-handshake.

Note: Post-handshake authentication works with Mozilla Firefox, Google Chrome, and Microsoft Edge browsers in TLS 1.1 and 1.2 mode.

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*\Levn\Web\WebServer\ssl** directory.
2. Edit the ***SAS-configuration-directory*\Levn\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.

   Note: Starting with SAS 9.4M8, if your Middle-Tier is configured with TLS 1.3, and you are using Google Chrome or Microsoft Edge, the first three statements must be put outside of the <Location> directive, as shown in the following example:

```
  SSLVerifyClient optional
  SSLVerifyDepth 10
  SSLCACertificateFile "ssl/root-certificate.pem"
<Location /SASLogon/login>
  RequestHeader set SSL_CLIENT_CERT "%{SSL_CLIENT_CERT}s"
  RequestHeader set SSL_CLIENT_VERIFY "%{SSL_CLIENT_VERIFY}s"
</Location>
```

The following table provides details about the options that are set in the httpd-ssl.conf file:

Table 17.13 *SAS Web Server Security Options*

Option	Expected Value	Description
SSLCertificateFile	"/*directory*/*pem-encoded-certificate*.crt"	Specifies the location of the PEM encoded certificate.

Option	Expected Value	Description
SSLCertificateKeyFile	"/*directory*/*key-file*.key"	Specifies the location of the key file, if it is not combined with the certificate.
SSLCertificateChainFile	"/*directory*/*certificate-chain*.pem"	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.
SSLCACertificateFile	"/*directory*/*CA-certificate*.pem"	Specifies the location of the CA certificates for client authentication.
SSLVerifyClient	*verification-type*	Specifies the client certificate verification type. Types are none, optional, require, and optional_no_ca.

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 class=Name="com.sas.vfabrictcsvr.authenticator.SSLAuthenticator"
fallThrough="false" />
```

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: This process is used when Client Certificate Authentication is fully configured.

The security module supports the following parameters:

Table 17.14 *SSLAuthenticator Security Module Attributes*

Attribute	Default Value	Description
uriPattern	/SASLogon/login.*$	Process requests with a URI, including query string, matching this regular expression.

Attribute	Default Value	Description
fallThrough	false	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.
sslClientCert	SSL_CLIENT_CERT	HTTP header that contains the Base64-encoded x509 client certificate.
sslClientVerify	SSL_CLIENT_VERIFY	HTTP header that contains the status of the certificate as NONE, SUCCESS, GENEROUS, or FAILED:reason.

4. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_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.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 "Configure the Realm for SAS Web Application Server" on page 251.

Configure TLS for Stand-Alone SAS Web Application Server

Overview

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 SAS 9.4M2 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. If any of your client or server certificates are signed by a CA that is not already included in the bundle of trusted CAs that are shipped with the JRE, insert that CA certificate into the cacerts file and also into the trusted CA area in the browser.

Starting in SAS 9.4M3, the SAS Deployment Manager handles all tasks associated with trusted CAs. This includes self-signed and site-signed certificates (those signed by a CA managed by your organization's internal IT department). The SAS Deployment Manager adds CAs to and removes CAs from the jssecacerts file. For more information, see "Add a Certificate to the Trusted CA Bundle" in *Encryption in SAS*.

Configure 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*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file to enable one-way TLS. Update the <Connection> definition, based on the following sample:

   - For SAS 9.4M4 and earlier versions:

     ```
     <Connector
         acceptCount="100" connectionTimeout="20000" executor="tomcatThreadPool"
         maxKeepAliveRequests="15"
         port="${bio.https.port}" protocol="org.apache.coyote.http11.Http11Protocol"
         redirectPort="${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" />
     ```

   - For SAS 9.4M5:

     ```
     <Connector acceptCount="100" bindOnInit="false"
     ciphers="TLS_ECDHE_ECDSA_WITH_AES_128
     _GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256
     _GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128
     _GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256
     _GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128
     _CBC_SHA" connectionTimeout="20000" executor="tomcatThreadPool"
     maxHttpHeaderSize="16384" maxKeepAliveRequests="15"
     maxSwallowSize="-1" port="${nio.http.port}"
     protocol="org.apache.coyote.http11.Http11NioProtocol"
     redirectPort="${nio.https.port}" scheme="https"
     useBodyEncodingForURI="true" secure="true" SSLEnabled="true"
     sslProtocol="TLS" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
     keystoreFile="/local/install/certs/serverids.jks"
     keystorePass="password" />
     ```

   - Starting with SAS 9.4M6:

     ```
     <Connector acceptCount="100" bindOnInit="false"
     ciphers="TLS_ECDHE_ECDSA_WITH_AES_128
     _GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256
     _GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128
     _GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256
     _GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128
     _CBC_SHA" connectionTimeout="20000" executor="tomcatThreadPool"
     ```

```
maxHttpHeaderSize="16384" maxKeepAliveRequests="15"
maxSwallowSize="-1" port="${nio.http.port}"
protocol="org.apache.coyote.http11.Http11NioProtocol"
proxyName="hostname.example.com" proxyPort="80"
redirectPort="${nio.https.port}" scheme="http"
sslEnabledProtocols="+TLSv1,+TLSv1.1,+TLSv1.2" sslProtocol="TLS"
useBodyEncodingForURI="true"/>
```

- Starting with SAS 9.4M8 TLSv1.3 is supported:

```
<Connector bindOnInit="false"
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_AES_128_CCM_SHA256,TLS_AES_128_CCM_8_SHA256,
TLS_AES_128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256,
TLS_AES_256_GCM_SHA384" connectionTimeout="20000" executor="tomcatThreadPool"
maxHttpHeaderSize="16384" maxKeepAliveRequests="15"
maxSwallowSize="-1" port="${nio.http.port}"
protocol="org.apache.coyote.http11.Http11NioProtocol"
proxyName="hostname.example.com" proxyPort="80"
redirectPort="${nio.https.port}" scheme="http"
sslEnabledProtocols="+TLSv1,+TLSv1.1,+TLSv1.2,+TLSv1.3" sslProtocol="TLS"
useBodyEncodingForURI="true"/>
```

The following parameters that are needed to support one-way TLS:

Parameter	Description
port	Specifies the HTTPS port of SAS Web Application Server.
scheme	Specifies the communications protocol and should be set to **`https`**.
keystoreFile	Specifies the keystore file that contains the server identity certificate.
keystorePass	Specifies the password for the keystore access.

2. Verify that one-way TLS is working with the default authentication.
3. Follow the steps in the "Web Authentication" on page 245 task, but specify CLIENT-CERT as the auth-method in the web.xml file for SAS Logon Manager.
4. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_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.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 "Configure the Realm for SAS Web Application Server" on page 251.

SAS Web Server Authentication

Overview

Starting with SAS 9.4M1, 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.*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.

Enable Web Authentication

Web authentication must be enabled before configuring authentication in SAS Web Server.

- By default, SAS 9.4M1 does not enable web authentication. To enable web authentication, follow the steps in "Web Authentication" with one exception. Do not edit either of the SAS Logon Manager installation files, **`web.xml.orig`** or **`web.xml`**, which is Step 4 of 'Modify SAS Logon Manager Installation Files' on page 247.
- For SAS 9.4M2 and later releases, to enable web authentication, follow the steps in "Web Authentication" with one exception. Do not "Modify SAS Logon Manager Installation Files" on page 246.

Configure Authentication in SAS Web Server

To configure authentication in SAS Web Server, follow these steps:

1. Edit the ***`SAS-configuration-directory`***`\Levn\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 might include 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 ***SAS-configuration-directory*\Levn\Web\WebServer\conf\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 ***SAS-configuration-directory*\Levn\Web\WebServer\conf\extra\httpd-ssl.conf** file, inside the VirtualHost directive.

3. Depending on the type of authentication that you use, perform whatever additional configuration that is necessary.

   Note: Because the additional configuration is dependent upon the type of authentication that is used in your environment, it cannot be fully described here.

4. Restart the SAS Web Server.

5. Verify that authentication is working by opening a web browser and entering a URL that is similar to the following examples:

   ```
   http://hostname.example.com/SASLogon/login
   ```

   ```
   https://hostname.example.com/SASLogon/login
   ```

   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: If you configure secure requests in the **\extra\httpd-ssl.conf** file, the user name is not displayed by default in the SAS Web Server ssl_request log. To display it, add **%u** to the Log directive.

Set 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 302.

```
<Location /SASLogon/login>
  AuthType xxxxxx
  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>
```

Configure 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 use one-way TLS to the web application server, or set up two-way TLS.

SAS 9.4M2 and Previous Releases

To configure a secret password, follow these steps:

1. Create a Base64-encoded authorization string, using the format *username:password*. For example, *username:password* is encoded as dXNlcm5hbWU6cGFzc3dvcmQ=. For more information, see *Encryption in SAS*.

   Note: You can leave the user name blank and encode only a password. If you do not specify a user name, include the delimiter (:) before the password (for example, *:password*).

   Note: Do not include the end of line character (=) as part of the password. On UNIX, use the following command to verify that the newline character is not appended to the password:

   ```
   echo -n username:password | base64
   ```

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:

   ```
   <Location /SASLogon/login>
     AuthType xxxxxx
     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 authorization string for *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 authorization string, on one line. Here is an example:

```
pw.sas.valve.PrincipalFromRequestHeadersValve=s2enc:
//Qqi4Ba2l9aAAQXYnG0Af8epT9PGWhTgB
```

SAS 9.4M3 and SAS 9.4M4

To configure a secret password, follow these steps:

1. Create a Base64-encoded authorization string, using the format *username:password*. For example, *username:password* is encoded as dXNlcm5hbWU6cGFzc3dvcmQ=. For more information, see *Encryption in SAS*.

   Note: You can leave the user name blank and encode only a password. If you do not specify a user name, include the delimiter (:) before the password (for example, *:password*).

   Note: Do not include the end of line character (=) as part of the password. On UNIX, use the following command to verify that the newline character is not appended to the password:

```
echo -n username:password | base64
```

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:

```
<Location /SASLogon/login>
  AuthType xxxxxx
  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 authorization string for *password* needs to be specified in the application server configuration. It is suggested that you encode the *password*, instead of using plaintext, by following these steps:

   a. Set TCHOME=***SASHOME*\SASWebApplicationServer\9.4** on Windows.

      On UNIX:

```
TCHOME=SASHOME/SASWebApplicationServer/9.4; export TCHOME
```

   b. Run the following command on Windows:

```
java -cp %TCHOME%\lib\com.springsource.org.bouncycastle.jce-1.46.0.jar;
     %TCHOME%\tomcat-7.0.55.A.RELEASE\lib\tcServer.jar;%TCHOME%\
     tomcat-7.0.55.A.RELEASE\bin\tomcat-juli.jar;%TCHOME%\
     tomcat-7.0.55.A.RELEASE\lib\tomcat-coyote.jar
```

```
-Dcom.springsource.tcserver.security.PropertyDecoder.
decoder_prefix=s2enc:// com.springsource.tcserver.security.PropertyDecoder
-encode "app-server-passphrase" password
```

On UNIX:

```
java -cp $TCHOME/lib/com.springsource.org.bouncycastle.jce-1.46.0.jar:
     $TCHOME/tomcat-7.0.55.A.RELEASE/lib/tcServer.jar:$TCHOME/
     tomcat-7.0.55.A.RELEASE/bin/tomcat-juli.jar:$TCHOME/
     tomcat-7.0.55.A.RELEASE/lib/tomcat-coyote.jar
     -Dcom.springsource.tcserver.security.PropertyDecoder.
     decoder_prefix=s2enc:// com.springsource.tcserver.security.PropertyDecoder
     -encode 'app-server-passphrase' password
```

Note: The previous commands must be on one line. They are shown on more than one line for display purposes only.

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 authorization string, on one line. Here is an example:

```
pw.sas.valve.PrincipalFromRequestHeadersValve=s2enc:
//Qqi4Ba2l9aAAQXYnG0Af8epT9PGWhTgB
```

SAS 9.4M5 , SAS 9.4M6, and SAS 9.4M7 Prior to the February 15, 2022 Release

Note: When you update-in-place to SAS 9.4M7 (after February 15, 2022) these secret passwords are not preserved and the following steps must be performed again.

To configure a secret password, follow these steps:

1. Create a Base64-encoded authorization string, using the format *username:password*. For example, *username:password* is encoded as dXNlcm5hbWU6cGFzc3dvcmQ=. For more information, see *Encryption in SAS*.

   Note: You can leave the user name blank and encode only a password. If you do not specify a user name, include the delimiter (:) before the password (for example, *:password*).

   Note: Do not include the end of line character (=) as part of the password. On UNIX, use the following command to verify that the newline character is not appended to the password:

```
echo -n username:password | base64
```

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:

```
<Location /SASLogon/login>
  AuthType xxxxxx
  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 authorization string for *password* needs to be specified in the application server configuration. It is suggested that you encode the *password*, instead of using plaintext, by running one of the following commands:

   On Windows:

   ```
   SASHOME\SASWebApplicationServer\9.4\tcruntime-admin.bat" encode
   value-to-encrypt passphrase
   ```

   On UNIX:

   ```
   SASHOME/SASWebApplicationServer/9.4/tcruntime-admin.sh encode
   value-to-encrypt passphrase
   ```

   The passphrase can be found in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\secure.file** 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 authorization string, on one line.

   - For SAS 9.4M5 and SAS 9.4M6 without HF E3V005/6, here is an example:

     ```
     pw.sas.valve.PrincipalFromRequestHeadersValve= tcEnc://encoded_password
     ```

   - For SAS 9.4M6 with E3V005/6 and subsequent releases, here is an example:

     ```
     pw.sas.valve.PrincipalFromRequestHeadersValve=pbkdf2://encoded_password
     ```

Starting with SAS 9.4M7 February 15, 2022 and Later Releases

Note: When you update-in-place to SAS 9.4M7 (after February 15, 2022) these secret passwords are not preserved and the following steps must be performed again.

To configure a secret password, follow these steps:

1. Create a Base64-encoded authorization string, using the format *username:password*. For example, *username:password* is encoded as dXNlcm5hbWU6cGFzc3dvcmQ=. For more information, see *Encryption in SAS*.

   Note: You can leave the user name blank and encode only a password. If you do not specify a user name, include the delimiter (:) before the password (for example, *:password*).

   Note: Do not include the end of line character (=) as part of the password. On UNIX, use the following command to verify that the newline character is not appended to the password:

   ```
   echo -n username:password | base64
   ```

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:

   ```
   <Location /SASLogon/login>
     AuthType xxxxxx
     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. Starting with SAS 9.4M7 February 15, 2022 release, the SAS Web Application Server is based on Apache Tomcat and because of this, password encryption has changed. A shell script, *decoder.sh* (UNIX), or *decoder.bat* (Windows), is provided to encode and decode the password. To run the shell script, you must first set two environment variables: *JAVA_HOME*, and *CATALINA_HOME*:

   ```
   export JAVA_HOME=SAS-home-directory/SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre
   ```

   ```
   export CATALINA_HOME=SAS-home-directory/SASWebApplicationServer/9.4/apache-tomcat-9.0.65
   ```

   Note: Any Java Runtime works.

   Once the above environment variables are set, navigate to the location of the *decoder* script:

   ```
   cd SAS-config-directory/Levn/Web/WebAppServer/SASServern_m/bin
   ```

   The decoder script encrypts a given value using a passphrase provided by the user. That passphrase can be provided in one of two ways. It is either provided directly on the command line by using the *-pw* option, for example: *-pw passphrase*. Or, it is provided indirectly by specifying a file from which to read the passphrase by using the *-pwf* option, for example: *-pwf passphrase file*. The passphrase can be found in the ***SAS-config-directory*\Levn\Web\WebAppServer\SASServern_m\conf\secure.file** file. You can use the *--help* option to see valid arguments for the script.

   Below is an example of the command that uses a file from which to read the passphrase:

   On Windows:

   ```
   .\decoder.bat -encode -config sasmtr01 -pwf SAS-config-directory\Levn\Web\
   WebAppServer\SASServern_m\conf\secure.file -v value-to-encrypt
   ```

   On UNIX:

   ```
   ./decoder.sh -encode -config sasmtr01 -pwf SAS-configuration-directory/Levn/Web/
   WebAppServer/SASServern_m/conf/secure.file -v value-to-encrypt
   ```

   Note: The command is displayed over multiple lines, but it should be entered on a single line.

4. Edit the ***SAS-config-directory*\Levn\Web\WebAppServer\SASServern_m\conf\catalina.properties** file. At the end of the file, add a new property on one line that contains the encoded authorization string:

   ```
   pw.sas.valve.PrincipalFromRequestHeadersValve=sasmtr01:encoded_password
   ```

Configure the Security Module for SAS Web Application Server

Overview

SAS Web Application Server must be configured to accept the user ID that is sent by SAS Web Server in the HTTP header.

SAS Web Application Server uses Tomcat valve configuration to do this. Valves perform actions that are inserted into the request processing pipeline for the catalina container. Some valves are built into Tomcat and the default SAS configuration uses them. (An example is logging.) SAS has extended the container with a valve to support authentication performed in SAS Web Server. The valve pulls the user ID out of an HTTP request header to create a principal and sets the user ID in the request that is sent to the SAS Logon Manager. This is done by configuring the

com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve class in "Configure the Valve Class" on page 308.

When using this class, the user ID and password of the login user must be present in the request header. See "Set the REMOTE_USER Variable" on page 302. Typically, this is through a pop-up window or a special form.

Note: Depending on valve class implementation, sending a password with the request might not be possible. In this case, you need a second authentication domain for the logins that contain web realm user IDs. Follow the steps in Web Authentication - Confirm User Accounts in SAS Metadata on page 249.

Configure the Valve Class

Add the following valve configuration to the ***SASHOME*\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\META-INF\context.xml** file:

```
<Valve className="com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve"
       secretPassword="${pw.sas.valve.PrincipalFromRequestHeadersValve}"
uriPattern = "/SASLogon/login.*$" fallThrough = "true" />
```

Note: The additional parameters that are used in the above valve configuration, **uriPattern**, **fallThrough**, and **secretPassword**, are explained in the following table. As a best practice, you should define at least **uriPattern** and **fallThrough** in the XML string.

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.

Note: As an alternative to updating the **context.xml** file, you can edit the **SASLogon.xml** deployed file for the SAS Web Application Server that contains the SAS Logon web application. It is located here: ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\Catalina\localhost\SASLogon.xml**. This avoids the need to rebuild and redeploy the application, but you must make sure that your changes are not overwritten if the application is redeployed at a later date.

The security module supports the following parameters:

Table 17.15 *SAS Web Application Server Security Module Attributes*

Attribute	Default Value	Description
uriPattern	/SASLogon/login.*$	Specifies the process requests with a URI, including query string, matching this regular expression.
fallThrough	false	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.

Attribute	Default Value	Description
secretPassword	None	Specifies the secret password to expect in the Basic Authorization header (Optional).
userHeader	X-Remote-User	Specifies the HTTP header containing the authenticated subject name.
roleName	ROLE_USER	Specifies the role to associate with the principal. This value is usually not needed.

Configure the Java Cryptography Extension

Symantec SiteMinder Pure Java API requires that the Java environment that is used by SAS Web Application Server be updated with the JCE. In addition, Integrated Windows Authentication with AES-256 encryption ciphers also requires the JCE.

Note: Check the JRE version that you have installed. If it is 1.7.0_181 or later, you do not have to complete the following procedure.

To configure the extension on systems other than AIX, follow these steps:

1. Download the JCE.
   - For Oracle, the Unlimited Strength Jurisdiction Policy files are available from http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html.
   - For Azul, download the Zulu Cryptography Extension Kit from http://cdn.azul.com/zcek/bin/ZuluJCEPolicies.zip. If you are unable to download the JCE, submit a request for the download from here: https://support.azul.com/hc/en-us/articles/115001122623-Java-Cryptography-Extension-JCE-for-Zulu-Azul-Platform-Core-and-Azul-Platform-Prime.

   Note: Retrieve the files and install them in the SAS Private JRE. These files are used only by SAS applications.
2. Extract the archive. In the **`jce`** directory, extract the files to ***`JAVA_HOME`***`\jre\lib\security`.

   Note: The default JRE is located in ***`SASHOME`***`\SASPrivateJavaRuntimeEnvironment\9.4\jre`.

For information about downloading and installing the unrestricted JCE policy files on AIX systems, see https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.found.admin.common.doc/topics/lmt_scr_downloading_installing_jce_policyfiles.html.

Chapter 18

Middle-Tier Security

Configure SAS Web Server Manually for HTTPS

Use of TLS with SAS Web Applications

Transport Layer Security (TLS) is a successor protocol to Secure Sockets Layer (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 "Use HTTPS" on page 36.

Note: Starting with SAS 9.4M5, it is required that you configure SAS Environment Manager for HTTPS. For more information, see "Configure SAS Environment Manager for HTTPS Starting with SAS 9.4M5" on page 330.

Reconfigure to Use HTTPS

The following table shows when you should revert manual TLS changes to SAS Web Server if you are upgrading.

Table 18.1 *When to Revert Manual TLS Changes to SAS Web Server during an Upgrade*

SAS Release	Upgrade Information
SAS 9.4M4 and later	You do not need to revert manual TLS changes to SAS Web Server if you are upgrading. If you perform an update in place, you might be required to manually reset the TLS settings in the SAS environment file. For more information, see Step 14 on page 316. You must determine which solutions, if any, need to have their manual TLS settings for WebDAV and connection properties reset. This is because some solutions do not preserve manual TLS settings for the WebDAV repository and connection properties. To do this review, complete Step 8 on page 314 and Step 9 on page 315.
SAS 9.4M3 and earlier	You must revert the manual TLS configuration changes to the original non-TLS values before applying any maintenance releases or upgrades to the system. If you did not follow the steps for configuring manual HTTPS as documented in this guide, you must revert those changes as well. For more information, see "Revert Manual HTTPS Changes to SAS Web Server" on page 344. Once maintenance or upgrades have been applied, 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, get a signed certificate, and create a certificate chain. 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*\Levn\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*\Levn\Web\WebServer\conf\httpd.conf** file and make the following changes:

   a. Remove the # from the following line:

   ```
   #Include C:\SAS\Config\Levn\Web\WebServer\conf\extra\httpd-ssl.conf
   ```

   b. To use SAS Environment Manager to monitor SAS Web Server, locate the following line on Windows systems:

   ```
   Listen 80
   ```

Replace the previous line with the following line:

```
Listen localhost:80
```

On UNIX systems, the port number is 7980.

c. Starting with SAS 9.4M6, remove the # from the following line:

```
#LoadModule ssl_module  "SASHOME\SASWebServer\9.4\httpd-2.4\modules\mod_ssl.so"
```

5. Edit the ***SAS-configuration-directory*\Levn\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 file names:

   ```
   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\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file and follow these steps:

   a. Make the following changes to the **Connector** element:

   - Change the **proxyPort** attribute to specify the HTTPS listen port.
   - Change the **scheme** to **https**.

   b. In the **RemoteIpValve** element, if the httpServerPort, httpsServerPort, protocolHeader, and internalProxies options are present, make sure they are correct.

8. Use SAS Management Console to update the protocol and port number for each web application. For more information, see "Specify Connection Properties" on page 79.

   CAUTION:
   Do not modify the connection properties for the DP-SAS-Environment-Manager node. However, the connection properties for the Environment Manager Mid-Tier node should be changed.

   Note the following items:

   - 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 connection information needs to be updated.
   - You can verify the web applications that you need to update by examining the **SAS-configuration-directory/Levn/Web/WebServer/conf/**

sas.conf file to identify the specific web applications to proxy. Each web application that is identified in a pair of ProxyPass and ProxyPassReverse directives must be proxied. Do not edit the **sas.conf** file.

- The connection properties for all **Environment Mgr Mid-Tier** applications should also be changed. Under **Application Management** ⇨ **Configuration Manager**, expand the **Environment Mgr Mid-Tier** node and make connection changes for each application in this node.
- The SAS middle tier must be running for you to use SAS Management Console to modify the Themes Connection information.
- 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 do choose to modify these connections using a SAS script rather than SAS Management Console, the SAS_THEME table in the Shared Services DB is not 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** ⇨ **SAS Themes** ⇨ **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 157.

10. For SAS 9.4M2 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, follow these steps:

    a. Know the location of your self-signed or site-signed certificate.

    b. 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.

For SAS 9.4M3 and later, 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 updates the SAS Private JRE for you. For more information, see "Manage Certificates in the Trusted CA Bundle Using the SAS Deployment Manager" in *Encryption in SAS*.

11. Configure the server tier and client tier.

    UNIX Specifics
    For SAS 9.4M2 and earlier, for the server tier, you can create a base64 encoded certificate chain file that contains all trust certificates in the chain and use the file in the SSLCALISTLOC= SAS system option. Create the chain file by concatenating the individual CA base64 files. For more information, see *Encryption in SAS*.

    For SAS 9.4M3 and later, this step is no longer needed.

    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 protocol and port number. Click **OK**.

       Note: You must perform these steps so that the HTTPS connection information is saved in metadata.

       If the URI does not include the HTTPS protocol and port number, confirm that the LASRAuthorizationService REST and LASRKeyRegistrationService REST Connection protocol and port properties have been changed to HTTPS.

       i. Select **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Visual Analytics version** ⇨ **Visual Analytics Services version** ⇨ **LASR Authenticaion Service version**.

       ii. Right-click **LASRAuthenticationService REST** and select **Properties**.

       iii. Select the **Connection** tab. Change the Communication Protocol and Port Number to the appropriate HTTPS values. Click **OK**.

       iv. Repeat steps 2 and 3 for the **LASRKeyRegistrationService REST** component.

       v. If you made changes to these components, repeat the preceding verification steps for each SAS LASR Analytic Server, to confirm that the URI is set correctly and is stored in metadata.

14. Depending on which products you have installed, you might have to update the ***SAS-configuration-directory*\Levn\Web\WebServer\htdocs\sas**

\sas-environment.xml file. For more information, see "Customize the SAS Environment File" on page 413.

15. To access SAS Environment Manager console with TLS 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*\Levn\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:

    ```
    <constructor-arg
    value="https://server:ssl-port/SASLogon/
    logout?
    sas_svcs_logon_LogonUrl=http%3A%2F%2Fserver%3Aev-server-port%2F"/>
    <property name="loginUrl"
    value="https://server:ssl-port/SASLogon"/>
    <property name="casUrl"
    value="https://server:ssl-port/SASLogon"/>
    <bean id="casTicketValidator"
    class="com.sas.hyperic.security.CasIdentityRetrievingTicketValidator">
                <constructor-arg
    value="https://server:ssl-port/SASLogon"/>
    ```

    b. Edit the ***SAS-configuration-directory*\Levn\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:

    ```
    <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>
    ```

    Note: Depending on the version of SAS Environment Manager that is deployed in your environment, SASWebDoc might not be defined in the web.xml file.

    c. For SAS 9.4M4 and previous releases, edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\classes\identity-service.properties** file. Locate the following line and enter the correct information for your environment:

    ```
    url.base=https\://server/
    ```

    d. 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 cannot 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 140.

For more information about how to update the JVM options, see "Specify JVM Options" on page 44.

See Also

- *Encryption in SAS*
- *SAS Intelligence Platform: Installation and Configuration Guide*

Configure SAS Web Application Server for HTTPS

Overview

Transport Layer Security (TLS) is a successor protocol to Secure Sockets Layer (SSL). This documentation assumes that you have a basic understanding of TLS and that you know how to obtain and use x.509 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.

Reconfigure 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.

The following examples configure SAS Web Application Server with self-signed certificates. However, certificates provided by your IT department or by a public Certificate Authority can also be used.

Starting with SAS 9.4M4, you do not have to revert manual TLS changes to SAS Web Application Server if you are upgrading. When upgrading to SAS 9.4M3 and previous releases, you must revert the manual TLS configuration changes before performing an update or applying maintenance to your system. For more information, see "Revert Manual HTTPS Changes to SAS Web Application Server" on page 347. Also, if you did not follow the steps for configuring manual HTTPS as documented in this guide, or if you did any additional configuration that is not covered in this document, you must revert those changes as well. Once maintenance or upgrades have been applied, the manual TLS configuration steps must be reapplied to the upgraded system.

After adding new products that create an additional server, you are required to enable TLS for any new SAS Web Application Server. The TLS settings for existing servers are preserved.

In order to use HTTPS between SAS Web Server and SAS Web Application Server, follow these steps:

1. Create a keystore in JKS format with the key and certificate. Export the certificate to a file. Convert the exported file to PEM encoding. 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: Make sure the value of **-alias** is unique for all certificates.

   The **keytool** command prompts for a series of identifying characteristics. The identifying information that you provide at each of the prompts is called a Relative Distinguished Name (RDN). The combination of all of the RDNs that you provide is the Distinguished Name (DN). The DN is encoded into the generated certificate as the Subject.

   Once the certificate is created, you can use a tool such as **openssl** to display the contents of the certificate in text format. Here is an example of a subject:

   ```
   Subject: C=US, ST=My State, L=My Town, O=Organization, OU=Business,
   CN=myhost.example.com
   ```

   The subject is what the SAS Deployment Manager uses as the alias when adding the certificate to the list of trusted certificates in the Java keystore.

   Note: When the keytool prompts “What is your first and last name?”, enter the host name that is used on your network to find the computer on which the SAS Web Application Server is located.

   Provide at least one unique RDN when answering the prompts to the **keytool -genkeypair** command. This ensures that every certificate that is created has a unique DN. You must do this because all these certificates must be added to the SAS keystore that contains the bundle of trusted CAs, and duplicate aliases in the keystore are not allowed.

   ```
   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.
   - Make sure the alias is the same value that is used in the first **keytool** command above.

2. For SAS 9.4M2 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.
   - Make sure the alias is the same value that is used in the first **keytool** command above.

- 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.

Starting with SAS 9.4M3, add the self-signed certificate (the Base-64 encoding certificate, myhost.pem) to the trusted CA bundle, using the SAS Deployment Manager. For more information, see *SAS Deployment Wizard and SAS Deployment Manager: User's Guide*, available at http://support.sas.com/documentation/installcenter/en/ikdeploywizug/66034/PDF/default/user.pdf.

3. On the primary SAS middle-tier machine and each middle-tier cluster node, edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer \SASServern_m\conf\server.xml** file. Duplicate the existing **Connector** element and complete the following:

   a. Add the following attributes:

      - For SAS 9.4M5 and previous versions:

```
secure="true"
SSLEnabled="true"
sslProtocol="TLS" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
keystoreFile="/path-to-myhost.jks"
keystorePass="changeit"
```

      - Starting with SAS 9.4M6:

```
secure="true"
SSLEnabled="true"
sslProtocol="TLS" sslEnabledProtocols="+TLSv1,+TLSv1.1,+TLSv1.2"
keystoreFile="/path-to-myhost.jks"
keystorePass="changeit"
```

        Note: If you want to enforce TLS v2 on the middle-tier machine, set **sslEnabledProtocols="+TLSv1.2"**. However, it is recommended that **sslEnabledProtocols="+TLSv1,+TLSv1.1,+TLSv1.2"**.

      - Starting with SAS 9.4M8, TLSv1.3 is supported:

```
secure="true"
SSLEnabled="true"
sslProtocol="TLS" sslEnabledProtocols="+TLSv1,+TLSv1.1,+TLSv1.2,+TLSv1.3"
keystoreFile="/path-to-myhost.jks"
keystorePass="changeit"
```

      Note: Make sure the *keystorePass* value is the same value used in Step 1 on page 319 when generating the JKS file, especially if you changed the default value (which is **changeit**).

   b. Change the port attribute.

      - For SAS 9.4M4 and previous releases, change from **port="${bio.http.port}** to **port="${bio.https.port}"**.
      - Starting with SAS 9.4M5, change from **port="${nio.http.port}** to **port="${nio.https.port}"**.

   c. After you complete your changes and you confirm that SAS Web Application Server is using HTTPS, edit the server.xml file again and remove the **Connector** element that was using HTTP.

Note: This step must be repeated for each server that uses HTTPS.

4. For SAS Web Application Server, modify or add the following JVM options. Make sure that you choose the correct HTTPS port on which the SAS Web Application Server listens.

   For SASServer1_1:

   ```
   -Dsas.scs.port=8443
   -Dsas.scs.scheme=https
   -Dsas.auto.publish.port=8443
   -Dsas.auto.publish.protocol=https
   ```

   For SASServer*n_m*:

   ```
   -Dsas.auto.publish.port=https-port
   ```

   - where the HTTPS port used by SAS Web Application Server can be found in the **SAS-configuration-directory\Levn\Web\WebAppServer\SASServern_m\conf\catalina.properties** file. Locate the **bio.https.port** property for SAS 9.4M4 and previous releases or the **nio.https.port** property starting with SAS 9.4M5.

   ```
   -Dsas.auto.publish.protocol=https
   ```

   Note: Set JVM options in the following files:

   - For Windows deployments, edit the following files:
   - **SAS-configuration-directory\Levn\Web\WebAppServer\SASServern_m\bin\setenv.bat**
   - **SAS-configuration-directory\Levn\Web\WebAppServer\SASServern_m\conf\wrapper.conf**

     Important: After you modify the wrapper.conf file for Feb 15th 2022 and later releases of SAS 9.4M7 and SAS 9.4 M8, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

   - For Linux deployments, edit the **SAS-configuration-directory/Levn/Web/WebAppServer/SASServern_m/bin/setenv.sh** file.

5. For SAS Web Server, make the following changes:

   a. Edit **SAS-configuration-directory\Levn\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\Levn\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"
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
```

If there are multiple SAS Web Application Server machines, choose one of the following options to configure the CA certificates that the SAS Web Application Servers are using:

- Create a chain.pem file with each of the self-signed certificates created for each middle-tier machine to use in the preceding **SSLProxyCACertificateFile** directive.

  Note: The chain.pem file can also be used if FIPS 140–2 compliance is applied on a multiple middle-tier system deployment.

- 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 file names. 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. Set the secure attribute for the session cookies. Configure SAS Web Application Server to return the session ID with the secure attribute by editing the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\web.xml** file:

   a. Locate the existing **session-config** block in the *Default Session Configuration* section. Comment out the block by adding comment tags: **<!--** before the block and **-->** at the end of the block:

```
<!--
    <session-config>
        <session-timeout>30</session-timeout>
        <tracking-mode>COOKIE</tracking-mode>
    </session-config>
-->
```

   b. Add the following block of code directly before the **</web-app>** tag, that is located at the bottom of the file:

```
<session-config>
    <session-timeout>30</session-timeout>
    <cookie-config>
        <secure>true</secure>
```

```
            </cookie-config>
        </session-config>

</web-app>
```

7. Manually update SAS Logon Manager to secure the CAS ticket-granting cookie by performing these steps:
   - Starting with SAS 9.4M8:
     1. Edit the ***SAS-config-directory*/Levn/Web/WebApp Server/SASServer1_1/sas_webapps/sas.svc.logon.war/WEB-INF/classes/application.properties** file.
     2. Change the value of **cas.tgc.secure=false** to **cas.tgc.secure=true** and verify that the property is not commented out.
     3. Save and close the file.
   - In SAS 9.4M7 and prior releases:
     1. Edit the **SASHOME\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\WEB-INF\spring-configuration\ticketGrantingTicketCookieGenerator.xml** file.
     2. Change the value of **p:cookieSecure="false"** to **p:cookieSecure="true"**.

        After you have added the new content, the ticketGrantingTicketCookieGenerator.xml file should resemble the following:

```
<bean id="ticketGrantingTicketCookieGenerator"
class="org.jasig.cas.web.support.CookieRetrievingCookieGenerator"
p:cookieSecure="true"
p:cookieMaxAge="-1"
p:cookieName="CASTGC"
p:cookiePath="/SASLogon"/>
```

     3. Save and close the file.

     Note: Repeat steps' 1-3 on every SAS middle tier machine.

     4. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\sas_webapps\sas.svcs.logon.war\WEB-INF\spring-configuration\ticketGrantingTicketCookieGenerator.xml** file.
     5. Change the value of **p:cookieSecure="false"** to **p:cookieSecure="true"**.

        After you have added the new content, the ticketGrantingTicketCookieGenerator.xml file should resemble the following:

```
<bean id="ticketGrantingTicketCookieGenerator"
class="org.jasig.cas.web.support.CookieRetrievingCookieGenerator"
p:cookieSecure="true"
p:cookieMaxAge="-1"
p:cookieName="CASTGC"
p:cookiePath="/SASLogon"/>
```

     6. Save and close the file.

     Note: You only perform steps' 4-6 on the SAS middle-tier machine where SASServer1_1 web application server is installed.

8. Restart the SAS middle tier.

Update the Properties File

After manually configuring SAS Web Application Server to use HTTPS, complete the following steps:

1. Open the ***SAS-configuration-directory*\Levn\Web\Scripts\AppServer\props\appserver.properties** file.
2. Change the value of the following properties to *https*:
   - member.*n*.protocol
   - server.*n*.protocol
3. Change the value of the following properties to their respective TLS port number:
   - member.*n*.port
   - server.*n*.httpsPort
4. Save the file.

Note: After performing an update in place, you might have to update the ***SAS-configuration-directory*\Levn\Web\Scripts\AppServer\props\appserver.properties** file for both new and existing SAS Web Application Servers. The member.*n*.protocol and member.*n*.port values might be overwritten.

See Also

- "Member Properties" on page 408
- "Server Properties" on page 409

Restart the SAS Middle Tier

To restart the SAS middle tier, follow these steps:

1. Use the method that is appropriate for your operating system:
   - Windows

     Using the Services Snap-in, right-click on each of the services in the list (in the order in which they are listed), and select **Stop**:
     - SAS Environment Manager Agent
     - SAS Environment Manager
     - SAS Web App Server: SASServer2_1
     - SAS Web App Server: SASServer12_1
     - SAS Web App Server: SASServer1_1
     - SAS Web Server
     - SAS Cache Locator Service: ins_41415
     - SAS JMS Broker
   - UNIX

Run ***SAS-configuration-directory*/sas.servers stop**.

2. If the SAS middle tier is on a separate machine, go back to the middle-tier machine, and run the command that is appropriate for your operating system:
   - Windows

     Using the Services Snap-in, right-click on each of the services in the list (in the order in which they are listed), and select **Start**:
     - SAS JMS Broker
     - SAS Cache Locator Service: ins_41415
     - SAS Web Server
     - SAS Web App Server: SASServer1_1
     - SAS Web App Server: SASServer12_1
     - SAS Web App Server: SASServer2_1
     - SAS Environment Manager
     - SAS Environment Manager Agent
   - UNIX

     Run ***SAS-configuration-directory*/sas.servers start**.
3. On any remaining SAS middle-tier machines, run the command appropriate for your operating system:
   - Windows

     Using the Services Snap-in, right-click on each of the services in the list (in the order in which the services are listed), and select **Restart**:
     - SAS JMS Broker
     - SAS Cache Locator Service: ins_41415
     - SAS Web Server
     - SAS Web App Server: SASServer1_1
     - SAS Web App Server: SASServer12_1
     - SAS Web App Server: SASServer2_1
     - SAS Environment Manager
     - SAS Environment Manager Agent
   - UNIX

     Run ***SAS-configuration-directory*/sas.servers restart**.

Configure SAS Environment Manager for HTTPS

Overview

SAS Deployment Wizard must be used to configure SAS Environment Manager to use HTTPS. After the deployment, manual steps must be completed. The following sections provide the manual steps that must be configured post-deployment to enable TLS for

SAS Environment Manager. For information about how to obtain and implement certificates, see "How to Implement Certificates" in *SAS Intelligence Platform: Security Administration Guide*.

Configure SAS Environment Manager Manually for HTTPS for SAS 9.4M4 and Previous Releases

Disable HTTP for SAS Environment Manager

By default, the SAS Environment Manager server listens on both HTTP (7080) and HTTPS (7443) ports. When using site-signed and third-party-signed certificates, you should disable HTTP for the SAS Environment Manager server.

Note: After performing an update in place, check the files in this section to ensure that they specify the correct information.

To disable HTTP for SAS Environment Manager, follow these steps:

Update the HTTP Connector and an Agent Property

1. Log on to the machine hosting SAS Environment Manager as the SAS Installer user.
2. Edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\conf\server.xml** file.
3. Locate the HTTP connector and surround it with comment tags:

```
<!--
<Connector port="${server.webapp.port}" executor="tomcatThreadPool"
maxHttpHeaderSize="8192"
protocol="HTTP/1.1" enableLookups="false"
redirectPort="${server.webapp.secure.port}"
          acceptCount="100"
        connectionTimeout="20000" disableUploadTimeout="true" compression="on"
        compressableMimeType="text/html,text/xml,text/plain,text/css,text/
          javascript" URIEncoding="UTF-8"/>
-->
```

4. Ensure that the *sslEnabledProtocols* attribute is set to is the same value that it is set to in SAS Web Application Server's ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file.

   Note: If the *sslEnabledProtocols* attribute is not set to the same values in the server.xml file for SAS Environment Manager and SAS Web Application Server and single sign-on is enabled, users cannot log on to SAS Environment Manager.

5. Save and close the **server.xml** file.
6. Perform the following steps for every machine where SAS Environment Manager Agent is configured:

   a. Edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\agent-*version*-EE\conf\agent.properties** file.

   b. Locate the **agent.setup.camSecure** property.

   c. Verify that the value is set to *yes*. If the value is not set to *yes*, set it.

4. Save and close the **agent.properties** file.

Integrate Manually with SAS Middle Tier

5. To access SAS Environment Manager console with TLS enabled on SAS Web Server, edit the following files and locate the instances of the URLs listed below that begin with http://*ev-server*:*ev-port*. Modify them to point to https://*ev-server*:*ev-ssl-port*. The default port for HTTPS on Windows and UNIX systems is 7443. The original port for HTTP on Windows and UNIX systems is 7080.

   a. Edit the ***SAS-configuration-directory*\Levn\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:

   ```
   <constructor-arg value="https://server:ssl-port/SASLogon/
   logout?sas_svcs_logon_LogonUrl=https://ev-server:ev-ssl-port" />
   <property name="service" value="https://ev-server:ev-ssl-port/
   j_spring_cas_security_check"/>
   <property name="serviceUrl" value="https://ev-server:ev-ssl-port/
   j_spring_cas_security_check"/>
   <property name="proxyCallbackUrl" value="https://ev-server:ev-ssl-port/
   j_spring_cas_security_proxyreceptor"/>
   ```

   b. Edit the ***SAS-configuration-directory*\Levn\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:

   ```
   <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=https://ev-server:ev-ssl-port/</param-value>
   ```

   Note: Depending on the version of SAS Environment Manager that is deployed in your environment, SASWebDoc might not be defined in the web.xml file.

   c. Edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\classes\identity-service.properties** file.

   Locate the following line and enter the correct information for your environment:

   ```
   url.base=https\://server/
   ```

4. Delete the initial keystore from the database.

   a. Log on to the SAS server-tier machine that contains the SAS Web Infrastructure Platform Data Server as the SAS Installer user.

   b. On the PostgreSQL psql console, connect to the SAS Environment Manager database, using the following command:

   ```
   SAS-installation-directory
   \SASWebInfrastructurePlatformDataServer\9.4\bin\psql -U
   SAS-WIP-data-server-admin-user -d EVManager -h
   SAS-WIP-data-server-name -p SAS-WIP-data-server-port
   ```

   c. On UNIX, run the following commands:

   ```
   export LD_LIBRARY_PATH=SAS-installation-directory/
   SASWebInfrastructurePlatformDataServer/9.4/
   lib:$LD_LIBRARY_PATH
   ```

```
SAS-installation-directory/
SASWebInfrastructurePlatformDataServer/9.4/bin/psql -U
SAS-WIP-data-server-admin-user -d EVManager -h
SAS-WIP-data-server-name -p SAS-WIP-data-server-port
```

d. On the PostgreSQL psql console, remove its default keystore, using the following command:

```
delete from eam_keystore;
```

e. Quit the PostgreSQL psql console by entering **\q**.

Manually Update the hq-server.conf File

To manually update the **hq-server.conf** file for SAS Web Server when you are providing your own certificates, follow these steps:

6. Log on to the machine hosting SAS Environment Manager as the SAS Installer user.
7. Edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\conf\hq-server.conf** file.
8. Make the following property changes in the file:

Table 18.2 *hq-server.conf Properties That Must Be Changed*

Property to Change	Old Value	New Value
server.keystore.path	..\..\conf\hyperic.keystore	Use the value entered in "SAS Deployment Agent Keystore and Truststore Information" in *SAS Intelligence Platform: Security Administration Guide*.
server.keystore.password	hyperic	Use the value entered in "SAS Deployment Agent Keystore and Truststore Information" in *SAS Intelligence Platform: Security Administration Guide*.

Note: The following conditions must exist, in order to request the keystore for SAS Environment Manager.

- The private key entry must use **hq** as its alias name.
- The common name in the certificate must match the machine host name where SAS Environment Manager is installed.
- The keystore password and the private key password must be the same.
- The password must be **hyperic**.

9. Save and close the **hq-server.conf** file.

Manually Update the Protocol and Port Number

10. Log on to SAS Management Console as an unrestricted user.
11. On the **Plug-ins** tab, navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure**.
12. Right-click **DP-SAS-Environment-Manager** ***machine-name*** and select **Properties**.
13. Select the **Internal Connection** tab. Change the protocol to HTTPS and the port number to the secure port that was defined during the SAS Deployment Wizard configuration.
14. Stop the middle tier, using the method that is appropriate for your operating system:
    - Windows

      Using the Services Snap-in, right-click on each of the services in the list (in the order in which they are listed), and select **Stop**:

      Note: The list of services that you see, and need to stop, depends on which managed web application servers are installed in your environment.

      - SAS Environment Manager Agent
      - SAS Environment Manager
      - SAS Web App Server: SASServer2_1
      - SAS Web App Server: SASServer12_1
      - SAS Web App Server: SASServer1_1
      - SAS Web Server
      - SAS Cache Locator Service: ins_41415
      - SAS JMS Broker
    - UNIX

      Run ***SAS-configuration-directory*/Levn/sas.servers stop**.
15. Restart the middle tier.
    - Windows

      Using the Services Snap-in, right-click on each of the services in the list (in the order in which they are listed), and select **Start**:

      Note: The list of services that you see, and need to start, depend on which managed web application servers are installed in your environment.

      - SAS JMS Broker
      - SAS Cache Locator Service: ins_41415
      - SAS Web Server
      - SAS Web App Server: SASServer1_1
      - SAS Web App Server: SASServer12_1
      - SAS Web App Server: SASServer2_1
      - SAS Environment Manager
      - SAS Environment Manager Agent
    - UNIX

      Run ***SAS-configuration-directory*/Levn/sas.servers start**.

Restart SAS Environment Manager Agents on the Server Tier

To ensure that the SAS Environment Manager agents on each of the server-tier machines are still able to communicate with the SAS Environment Manager server, you must stop each SAS Environment Manager Agent, delete its data directory, and then restart it.

16. Log on to a server-tier machine as the SAS Installer user.
17. Use the method appropriate for your operating system to stop the SAS Environment Manager Agent:
    - Windows

      Using the Services Snap-in, right-click **SAS Environment Manager Agent**, and select **Stop**:
    - UNIX

      Enter the following command: **`SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/bin/hq-agent.sh stop`**.
18. Use the command appropriate for your operating system to delete the SAS Environment Manager Agent's data directory:
    - Windows

      ```
      rmdir /S SAS-configuration-directory\Levn\Web
      \SASEnvironmentManager\agent-version-EE\data
      ```
    - UNIX

      ```
      rm -rf SAS-configuration-directory/Levn/Web/
      SASEnvironmentManager/agent-version-EE/data/*
      ```
19. Make sure that SAS Web Application Server and SAS Environment Manager are running.
20. Use the method appropriate for your operating system to start the SAS Environment Manager Agent:
    - Windows

      Using the Services Snap-in, right-click **SAS Environment Manager Agent**, and select **Start**:
    - UNIX

      Enter the following command: **`SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/bin/hq-agent.sh start`**.
21. Perform the steps in this section on every machine where SAS Environment Manager agent is configured.

Configure SAS Environment Manager for HTTPS Starting with SAS 9.4M5

Overview

During an initial configuration, the following configurations are supported:

- Both SAS Web Server and SAS Environment Manager are not configured for TLS.

- Both SAS Web Server and SAS Environment Manager are configured for TLS. If both SAS Web Server and SAS Environment Manager are on the same machine, they use the same certificate. However, if SAS Web Server and SAS Environment Manager are configured on separate machines, you must provide the JKS keystore for SAS Environment Manager.
- If your plan file does not include SAS Web Server, but it does include SAS Environment Manager, you can choose whether to configure SAS Environment Manager for TLS. If you configure SAS Environment Manager for TLS, you must provide the keystore (JKS format), which is needed for a secure connection.

Both SAS Web Server and SAS Environment Manager use the SAS Security Certificate Framework, but they use different certificate formats. If you configure SAS Web Server and SAS Environment Manager on the same machine, you need to configure only the certificate for SAS Web Server. SAS Environment Manager transfers the certificate to the required format and uses the default password for SAS Environment Manager. If SAS Environment Manager is configured on a separate machine, you must provide a keystore for SAS Environment Manager. You must add the certificate for SAS Environment Manager into the JKS format keystore, and you can use a different password. The password must be consistent with the keystore.

During a migration, the configuration properties from SAS Metadata Server are saved. The local certificate file and JKS format keystore are not. Therefore, the configuration process is similar to a new configuration. If you previously configured SAS Environment Manager to use a different certificate than SAS Web Server, during migration, you must use the same certificate for SAS Environment Manager and SAS Web Server.

By default, SAS Environment Manager Agent is configured with the default self-signed certificate. You can customize the configuration by using a site-signed certificate, which is a JKS format keystore with a password. For more information, see "Configure SAS Environment Manager Agents for HTTPS" on page 340.

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 updates the SAS Private JRE for you. For more information, see "Add a Certificate to the Trusted CA Bundle" in *Encryption in SAS*.

Before adding your certificates to the truststore, consider the following information:

- You can add only one certificate at a time with the deployment manager. You must rerun the deployment manager each time you add a certificate to the trusted CA bundle.
- If you have any Windows machines, you must also add the CA root and intermediate certificates to the Windows Certificate stores using Windows MMC. For more information, see "Add Your Certificates to the Windows CA Store" in *Encryption in SAS*.

Supported Scenarios

Table 18.3 *Supported Scenarios for SAS Web Server and SAS Environment Manager*

Scenario	Both are Installed on the Same Machine	SAS Web Server Configured for TLS	SAS Environment Manager Configured for TLS	Required Configuration
1	Yes	Yes	Yes	SAS Environment Manager should use the same customer-supplied, site-signed certificate as SAS Web Server. The keystore is generated from the key and certificate that are used by SAS Web Server. The password is set to **hyperic**.
2	Yes	Yes	No	This scenario is possible only during an update in place. During an update in place, the communication protocol for SAS Environment Manager defaults to the protocol for SAS Environment Manager on the source system. Although SAS Deployment Wizard displays a prompt for the Configured Protocol (which defaults to HTTP), you should not change it. If you do change the protocol to HTTPS, the underlying protocol is still HTTP, matching the SAS Environment Manager protocol on the source system.
3	Yes	No	No	No additional configuration is required.
4	No	Yes	Yes	SAS Deployment Wizard prompts for the customer-supplied, site-signed certificate and for the SAS Environment Manager keystore and password.

Scenario	Both are Installed on the Same Machine	SAS Web Server Configured for TLS	SAS Environment Manager Configured for TLS	Required Configuration
5	No	Yes	No	This scenario is possible only during an update in place. During an update in place, the communication protocol for SAS Environment Manager defaults to the protocol for SAS Environment Manager on the source system. SAS Deployment Wizard displays a prompt for the Configured Protocol, which defaults to HTTPS when SAS Web Server and SAS Environment Manager are installed on different machines. You must change the protocol to HTTP, so that it matches the protocol of SAS Environment Manager on the source system. If you kept the communication protocol as HTTPS (which you are allowed to do), the underlying protocol is still HTTP, matching the SAS Environment Manager protocol on the source system.
6	No	No	No	No additional configuration is required.

Generate the SAS Environment Manager Keystore with a Site-Signed Certificate

If you want to transfer the site-signed certificate, follow these steps:

Note: The following **OpenSSL** and **keytool** commands must be on one line. They are shown on more than one line for display purposes only.

1. If SAS Environment Manager is on the machine where SAS Web Server is installed, complete the following steps:
   a. Create a new directory (for example, **`C:\SSL`**).
   b. Copy the following files to the new directory:
      - SAS Web Server private key: ***`SAS-configuration-directory`***`\Levn\Web\WebServer\ssl\`***`webServerKeyFile`***

- CA .pem: ***SAS-installation-directory*** **\SASSecurityCertificateFramework\1.1\cacerts \trustedcerts.pem**

2. If SAS Environment Manager is not on the machine where SAS Web Server is installed, complete the following steps:

   a. Create a private key, generate a certificate signing request, get a signed certificate, and create a certificate chain for the machine. For more information, see *Encryption in SAS 9.4*.

   b. Use SAS Deployment Manager to add the Certificate Authority (CA) certificate to the trusted CA bundle on the machines in the deployment.

   c. Create a new directory (for example, **C:\SSL**) on the SAS Environment Manager machine and copy the newly created private key, certificate, and certificate chain (***SAS-installation-directory*** **\SASSecurityCertificateFramework\1.1\cacerts \trustedcerts.pem**) to the directory.

3. Create the p12 format keystore, using the information from either step 1 or step 2, depending on your environment.

   ```
   openssl pkcs12 -export -chain -inkey createdServerKeyFile -in
   createdServerCrtFile -name aliasName -CAfile trustedcerts.pem
   -out hyperic.p12
   ```

   Enter *hyperic* when prompted for the password.

   Note: For SAS 9.4M5, to request the keystore for SAS Environment Manager, the private key entry must use **hq** as its alias name and the common name in the certificate must match the machine host name where SAS Environment Manager is installed.

4. If SAS Environment Manager is on a different machine, copy the **hyperic.p12 keystore** file to that machine.

5. Convert the keystore to the JKS format.

   ```
   keytool -importkeystore -deststorepass password -destkeypass password
   -destkeystore hyperic.keystore -srckeystore hyperic.p12 -srcstoretype PKCS12
   -srcstorepass password -alias aliasName
   ```

   Note: By default, the password is set to *hyperic* for the keystore. You can use a different password, but make sure that you enter the same password in the prompt windows in case you need use a customer certificate.

Generate the SAS Environment Manager Keystore with a Self-Signed Certificate

This section applies if, by default, SAS Web Server and SAS Environment Manager are installed on different machines. You must create the certificate for SAS Environment Manager before running the SAS Deployment Wizard, and then load the certificate into the truststore.

Note: The following **OpenSSL** and **keytool** commands must be on one line. They are shown on more than one line for display purposes only.

1. Log on to a machine that has the **OpenSSL** tool.

2. Create an environment variable that points to the OpenSSL install directory.

   ```
   export OPENSSL_HOME=/install/SASServer/SASHome/SASWebServer/9.4/httpd-version
   ```

3. Create an environment variable that forces the **OpenSSL** tool to look for a configuration file in an alternative location.
4. Add the path to the OpenSSL binaries to your computer's path variable.

   ```
   export PATH=$OPENSSL_HOME/bin:$PATH
   ```

   ```
   export OPENSSL_CONF=$OPENSSL_HOME/ssl/openssl.cnf
   ```

5. Create an environment variable that points to the Java JRE install directory.

   ```
   export JAVA_HOME=/install/SASServer/SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre
   ```

6. Add JAVA_HOME to your computer's path variable.

   ```
   export PATH=$JAVA_HOME/bin:$PATH
   ```

7. Generate the certificate and key.

   ```
   openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout
   filename.key -out filename.crt -subj
   "/C=US/ST=state/L=city/O=organization./OU=department/CN=*.computerDomain"
   ```

8. Convert the key to an RSA key.

   ```
   openssl rsa -in filename.key -out filename.key
   ```

9. Create the p12 format keystore.

   ```
   openssl pkcs12 -export -inkey  filename.key -in
   filename.crt -name aliasName -password pass:password -out hyperic.p12
   ```

10. Convert the keystore to the JKS format.

    ```
    keytool -importkeystore -deststorepass password -destkeypass password
    -destkeystore hyperic.keystore -srckeystore hyperic.p12 -srcstoretype PKCS12
    -srcstorepass password -alias aliasName
    ```

    Note: For SAS 9.4M5, the following conditions must exist, in order to request the keystore for SAS Environment Manager.

    - The private key entry must use **hq** as its alias name.
    - The common name in the certificate must match the machine host name where SAS Environment Manager is installed.

    The **keytool** command prompts for a series of identifying characteristics. The identifying information that you provide at each of the prompts is called a Relative Distinguished Name (RDN). The combination of all of the RDNs that you provide is the Distinguished Name (DN). The DN is encoded into the generated certificate as the Subject.

    Once the certificate is created, you can use a tool such as **openssl** to display the contents of the certificate in text format. Here is an example of a subject:

    ```
    Subject: C=US, ST=My State, L=My Town, O=Organization, OU=Business,
    CN=myhost.example.com
    ```

    The subject is what the SAS Deployment Manager uses as the alias when adding the certificate to the list of trusted certificates in the Java keystore.

    Note: When the keytool prompts “What is your first and last name?”, enter the host name that is used on your network to find the computer on which the SAS Web Application Server is located.

    Provide at least one unique RDN when answering the prompts to the **keytool -genkeypair** command. This ensures that every certificate that is created has a unique

DN. You must do this because all these certificates must be added to the SAS keystore that contains the bundle of trusted CAs, and duplicate aliases in the keystore are not allowed.

11. Make a backup of the existing keystore file that is in the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\conf** directory.
12. Copy the newly generated keystore file to the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\conf** directory.
13. Use SAS Deployment Manager to add the self-signed certificate to the trusted CA bundle. For more information, see SAS Deployment Wizard and SAS Deployment Manager: User's Guide

Enable HTTP Strict Transport Security (HSTS) for SAS Environment Manager

HTTP HSTS is a mechanism that allows SAS Environment Manager to declare that it can be accessed only via secure connection (HTTPS). The Strict-Transport-Security header is returned only if you access SAS Environment Manager via HTTPS. Because of this, SAS Environment Manager must be configured with SSL/TLS. See "Configure SAS Environment Manager for HTTPS Starting with SAS 9.4M5" on page 330.

To enable HSTS for SAS Environment Manager, log on to the machine that hosts SAS Environment Manager as the SAS Installer and perform the following steps.

Configure the httpHeaderSecurity Filter and httpHeaderSecurity Filter-Mapping in the web.xml File

1. Edit the **web.xml** file that is located here: ***SAS-configuration-directory*/Levn/Web/SASEnvironmentManager/server-*version*-EE/hq-engine/hq-server/webapps/ROOT/WEB-INF**.
2. Find the **httpHeaderSecurity** filter. Here is an example of the filter contents:

```
<filter>
        <filter-name>httpHeaderSecurity</filter-name>
        <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-class
        <async-supported>true</async-supported>
        <init-param>
            <param-name>hstsEnabled</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>xssProtectionEnabled</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>antiClickJackingOption</param-name>
            <param-value>SAMEORIGIN</param-value>
        </init-param>
</filter>
```

3. Change the parameter value for **hstsEnabled** from **false** to **true**.
4. Verify that the **httpHeaderSecurity** filter-mapping is present in the file. If not, add the filter as follows:

```
    <filter-mapping>
```

```
        <filter-name>httpHeaderSecurity</filter-name>
        <url-pattern>/*</url-pattern>
        <dispatcher>REQUEST</dispatcher>
    </filter-mapping>
```

5. To force HTTP traffic to get redirected to the secure connection, add the following security constraint to end of the file:

```
<!--  for Redirect from HTTP to HTTPS  -->
<security-constraint>
    <web-resource-collection>
        <web-resource-name>Entire Application</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

Configure a Redirect in the server.xml File

6. Review the **`server.xml`** file that is located here:

   ***SAS-config-dir*/Levn/Web/SASEnvironmentManager/server-*version*-EE/hq-engine/hq-server/conf**

7. Ensure that the HTTP connector that is defined in the **`server.xml`** has the **redirectPort** attribute set to point to the correct port. If the connector is not defined, add it. It should look similar to the following:

```
<Connector port="${server.webapp.port}" executor="tomcatThreadPool"
maxHttpHeaderSize="8192"
        protocol="org.apache.coyote.http11.Http11NioProtocol"
        enableLookups="false" redirectPort="${server.webapp.secure.port}"
        acceptCount="100"
        connectionTimeout="20000" disableUploadTimeout="true" compression="on"
        compressableMimeType="text/html,text/xml,text/plain,text/css,text/javascript"
        URIEncoding="UTF-8"/>

    <Connector port="${server.webapp.secure.port}"
          executor="tomcatThreadPool" maxHttpHeaderSize="8192"
         protocol="org.hyperic.bootstrap.Http11NioProtocolExt" SSLEnabled="true"
         scheme="https" secure="true" clientAuth="false"
         keystoreFile="${server.keystore.path}"
         keystorePass="${server.keystore.password}"
         ciphers="TLS_AES_128_CCM_SHA256,TLS_AES_128_CCM_8_SHA256,
                  TLS_AES_128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256,
                  TLS_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
                  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
                  TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
                  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
                  TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA"
         sslProtocol="TLS"
         sslEnabledProtocols="TLSv1.2,TLSv1.3"
         URIEncoding="UTF-8"/>
```

8. Restart SAS Environment Manager.

Update Certificates for SAS Environment Manager

By default, SAS Environment Manager generates a self-signed sample certificate during deployment. This certificate is not trusted. When you disable the HTTP port for SAS Environment Manager, you must replace the sample certificate to allow users to log on to SAS Environment Manager.

In addition, if after performing an update in place and both of the following tasks were completed, you must manually replace the default self-signed certificate:

- Performed an update in place from SAS 9.4M4 and previous releases to SAS 9.4M5.
- Configured TLS using SAS Deployment Wizard without using a client certificate.

To update certificates for SAS Environment Manager, follow these steps.

Manually Update the hq-server.conf File

To manually update the **hq-server.conf** file for SAS Web Server when you are providing your own certificates, follow these steps:

1. Log on to the machine that hosts SAS Environment Manager as the SAS Installer user.
2. Edit the following file:

   ***SAS-config-dir*\Levn\Web\SASEnvironmentManager\ server-*version*-EE\conf\hq-server.conf**

3. Make the following property changes in the file:

   Table 18.4 *hq-server.conf Properties That Must Be Changed*

Property to Change	Old Value	New Value
server.keystore.path	..\..\conf\hyperic.keystore	Use the value entered in "SAS Deployment Agent Keystore and Truststore Information" in *SAS Intelligence Platform: Security Administration Guide*.
server.keystore.password	The password is encrypted.	If you want to change the password, either use plain text or encrypt it using an encryption tool.

   Note: For SAS 9.4M5, the following conditions must exist, in order to request the keystore for SAS Environment Manager.

   - The private key entry must use **hq** as its alias name.
   - The common name in the certificate must match the machine host name where SAS Environment Manager is installed.
   - The keystore password and the private key password must be the same.
   - By default, the password is **hyperic**. If you change the password, it must be consistent with the password entered during the installation and configuration into the SAS Deployment Wizard.

4. Save and close the **hq-server.conf** file.

Delete the Keystore From the Database

5. Log on to the SAS server-tier machine that contains the SAS Web Infrastructure Platform Data Server as the SAS Installer user.
6. Using the PostgreSQL psql console, connect to the SAS Environment Manager database using the following command:

   ```
   SAS-installation-directory
   \SASWebInfrastructurePlatformDataServer\9.4\bin\psql -U
   SAS-WIP-data-server-admin-user -d EVManager -h
   SAS-WIP-data-server-name -p SAS-WIP-data-server-port
   ```

7. Using the PostgreSQL psql console, remove its default keystore using the following command:

   ```
   delete from eam_keystore;
   ```

8. Quit the PostgreSQL psql console by entering **\q**.
9. Restart SAS Environment Manager.

Clear the Cache

10. Log on to a server-tier machine as the SAS Installer user.
11. Use the method appropriate for your operating system to stop the SAS Environment Manager Agent:
    - Windows

      Using the Services Snap-in, right-click **SAS Environment Manager Agent**, and select **Stop**:
    - UNIX

      Enter the following command: **SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/bin/hq-agent.sh stop**.
12. Use the command appropriate for your operating system to delete the SAS Environment Manager Agent's data directory:
    - Windows

      ```
      rmdir /S SAS-configuration-directory\Levn\Web
      \SASEnvironmentManager\agent-version-EE\data
      ```
    - UNIX

      ```
      rm -rf SAS-configuration-directory/Levn/Web/
      SASEnvironmentManager/agent-version-EE/data/*
      ```
13. Use the method appropriate for your operating system to start the SAS Environment Manager Agent:
    - Windows

      Using the Services Snap-in, right-click **SAS Environment Manager Agent**, and select **Start**:
    - UNIX

      Enter the following command:

      ```
      SAS-configuration-directory/Levn/Web/SASEnvironmentManager/
      agent-version-EE/bin/hq-agent.sh start
      ```

14. Clear the cache on every machine where SAS Environment Manager Agent is installed. For more information, see "Clear the Cache" on page 339.

Configure SAS Environment Manager Agents for HTTPS

By default, SAS Environment Manager Agents are configured with a self-signed certificate. You can configure the agents with site-signed certificates using SAS Deployment Wizard if you choose the custom prompting level.

You can also manually configure the SAS Environment Manager Agent with a site-signed certificate by completing the following steps:

1. On the machine where the SAS Environment Manager Server is running, stop the agent:
   - UNIX

     Enter the following command: **SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/bin/hq-agent.sh stop**
   - Windows

     Using the Services Snap-in, right-click **SAS Environment Manager Agent**, and select **Stop**
2. Edit the ***SAS-configuration-directory*/Levn/Web/SASEnvironmentManager/agent-*version*-EE/conf/agent.properties** file.
   - Locate the **agent.setup.camSecure** property and set it to *Yes*.

     `agent.setup.camSecure=Yes`
   - Locate the **agent.keystore.alias=** property. Remove the # symbol from the beginning of the line to uncomment the property and set it to *hq*:

     `agent.keystore.alias=hq`
   - Locate the **agent.keystore.path=** property. Remove the # symbol from the beginning of the line to uncomment the property and set it to ***SAS-configuration-directory*/Levn/Web/SASEnvironmentManager/server-*version*-EE/conf/hyperic.keystore**.

     `agent.keystore.path=`*`SAS-configuration-directory`*`/Levn/Web/SASEnvironmentManager/server-`*`version`*`-EE/conf/hyperic.keystore`
   - Locate the **agent.keystore.password=** property. Remove the # symbol from the beginning of the line to uncomment the property and set it to *hyperic*:

     `agent.keystore.password=hyperic`
3. Use the command appropriate for your operating system to delete the SAS Environment Manager Agent's data directory:
   - UNIX

     rm -rf SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/data/*
   - Windows

     rmdir /S SAS-configuration-directory\Levn\Web\SASEnvironmentManager\agent-version-EE\data

Note: Deleting the **/data** directory causes the agent to read the configuration file when it restarts.

4. Start the SAS Environment Agent:
   - UNIX

     Enter the following command: **SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/bin/hq-agent.sh start**
   - Windows

     Using the Services Snap-in, right-click **SAS Environment Manager Agent**, and select **Start**
5. Verify that the agent is using the TLS port:
   - UNIX

     Enter the following command: **SAS-configuration-directory/Levn/Web/SASEnvironmentManager/agent-version-EE/bin/hq-agent.sh status**

     The output should look similar to the following:

     ```
     HQ Agent is running (PID:<number>).
     Current agent bundle: agent-<version>
     Server IP Address: <hostname>.example.com
     Server (SSL) port: 7443
     Using new trasport; unidirectional=true
     Agent listen port: 2144
     ```

     Note: The default port number is 7443.
   - Windows

     The command to retrieve the status of the SAS Environment Agent is: **SAS-configuration-directory\Levn\Web\SASEnvironmentManager\agent-version-EE\bin\hq-agent.bat query**

     However, the Windows command does not return HTTPS-related configuration values. To find this information, look in the **SAS-configuration-directory\Levn\Web\SASEnvironmentManager\agent-version-EE\log\agent.log** file.
6. For each machine that is running the SAS Environment Manager Agent, do the following:
   a. Copy the ***SAS-configuration-directory*/Levn/Web/SASEnvironmentManager/server-*version*-EE/conf/hyperic.keystore** file from the SAS Environment Manager Server machine to each agent machine.
   b. Complete steps' 1-5 on each agent machine.

Preserve TLS and Existing Customer Reverse Proxy Customizations

Overview

Starting with SAS 9.4M4, you do not have to revert the following list of manual changes before updating SAS software or applying maintenance, if you followed the steps that are documented in this guide:

- HTTPS for SAS Web Server
- HTTPS for SAS Web Application Server
- HTTPS for SAS Environment Manager
- Middle tier to use an existing customer reverse proxy

However, in certain circumstances, you might be required to complete additional steps.

Scope

During a SAS software update and application of maintenance, you can maintain the TLS configuration settings for SAS Web Server and SAS Web Application Server. You can also preserve the existing reverse proxy custom settings. For example, customizations in the following scenarios are now preserved when running a SAS software upgrade:

- Manual HTTPS configurations for SAS Web Server.
- Manual HTTPS configurations for SAS Web Application Server. This includes when SAS Web Server is automatically configured to use TLS and when SAS Web Server is manually configured to use TLS.
- Manual configuration for an existing customer reverse proxy for the following situations:
  - Reverse proxy is configured to use HTTP
  - Reverse proxy is configured to use HTTPS
  - With and without SAS Web Server being configured

Also, when upgrading from SAS 9.4M3 to SAS 9.4M4, your certificates for TLS configurations that are stored in SAS keystore management framework are preserved. However, if you upgrade from SAS 9.4M2 or previous versions to SAS 9.4M4, your certificates for TLS configurations that are stored in the SAS Private JRE are not preserved. You must migrate the certificates. For more information, see "SAS Private JRE" on page 343.

For SAS 9.4M4, if you are adding one or more new SAS 9.4 products that have a web application that trigger an upgrade, follow these steps:

1. Revert the manual TLS changes to SAS Web Application Server before adding the new products. Once the products are added on to the deployment, the manual TLS changes to SAS Web Application Server can be performed.

2. After the products are added on to the deployment, the manual changes to the reverse proxy server files (the server.xml file and the JVM options) need to be performed again.

Starting with SAS 9.4M5, if you are adding one or more new SAS 9.4 products that have a web application that trigger an upgrade and creates a new managed server, follow these steps:

1. Revert the manual TLS changes to SAS Web Application Server before adding the new products. Once the products are added on to the deployment, the manual TLS changes to SAS Web Application Server can be performed. For more information, see "Revert Manual HTTPS Changes to SAS Web Application Server" on page 347 and "Configure SAS Web Application Server for HTTPS" on page 318.
2. Make the manual changes to the reverse proxy server files (the server.xml file and the JVM options) after the products are added on to the deployment. For more information, see steps 1 and 2 of "Revert Manual HTTPS Changes to SAS Web Application Server" on page 347.

Note: The above steps do not have to be performed when adding new SAS 9.4 products to an existing managed server.

For a SAS migration, HTTPS and customer reverse proxy customizations are not migrated to the new system. Instead, they must be configured on the new system.

If you have configured the deployment for FIPS 140-2 compliance:

- In SAS 9.4 M6 and prior releases, you need to revert the FIPS steps before upgrading. Once the upgrade is complete, you can configure your system for FIPS compliance. For more information, see "FIPS 140-2 Compliance" on page 351.
- Starting in SAS 9.4M7 and later releases, you do not need to revert the FIPS steps before upgrading.

SAS Private JRE

If you are migrating from SAS 9.4M2 or earlier releases to SAS 9.4M4, the steps in this section must be performed before starting the migration. Export the certificates to a file outside of the JRE before upgrading to SAS 9.4M4, when the SAS Private JRE is updated to a newer version and changes to the previous version is lost.

Export the certificates into the SAS Private JRE keystore using the Java **keytool -exportcert** command as follows: ***SAS-installation-directory*/SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/keytool.exe -exportcert -alias *aliasName* -storepass *password* -keystore *keystoreLocation* –file *fileName* –rfc**.

For example, on Windows, the command that you run is similar to this:

```
C:\Program Files\SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre\bin\keytool.exe
-exportcert -alias myCertificate -storepass changeit -keystore
C:\Program Files\SASHome\SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security\cacerts
-file exportedCertificate -rfc
```

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.

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.

When running an upgrade, after the installation of the upgrade completes, you need to import the certificates into the SAS certificate management framework using SAS Deployment Manager.

SAS Web Server

If the SAS Web Server was manually configured for TLS after the original configuration, an upgrade to SAS 9.4M4 converts the configuration to an automatic TLS configuration. No additional manual configuration is required to upgrade from any release of SAS 9.4 to SAS 9.4M4.

SAS Web Application Server

After running SAS Deployment Manager, to complete an upgrade from SAS 9.4M2 and earlier to SAS 9.4M4, verify that the following attributes are in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\server.xml** file:

```
sslProtocol="TLS" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
```

If the attributes are not in the file, add them.

Existing Reverse Proxy

No additional manual configuration is required to upgrade from any release of SAS 9.4 to SAS 9.4M4.

Note: When upgrading from SAS 9.4M4 to SAS 9.4M5, if you have two */Connector* elements in the server.xml file, one with TLS-enabled and one without, the attributes in the two */Connector* elements might be merged if the reverse proxy attributes exist only in the non-TLS enabled connector.

Revert Manual HTTPS Changes to SAS Web Server

If you chose to configure your SAS Web Server to use HTTPS during the installation and configuration using the SAS Deployment Wizard, then you can skip this step. Starting with SAS 9.4M4, you do not have to revert manual TLS changes to SAS Web Server if you are upgrading. However, when upgrading to SAS 9.4M3 and previous releases, if you manually configured (initially or afterward) HTTPS, you must revert those manual changes before performing an update or applying maintenance to your system. Also, if you did not follow the steps for configuring manual HTTPS as documented in this guide, you must revert those changes as well. During an update or application of maintenance, automatic configuration overwrites your changes, causing you to lose them.

TIP If your manual changes are not successfully reverted, you receive an error and the upgrade stops. For assistance in determining what might have been missed when reverting the changes, check the SAS Deployment Manager log, SAS Deployment Manager_*YYYY-MM-DD-HH-MM-SS*.log. On Windows, it is located in the

\Documents and Settings*user*\Local Settings\Application Data\SAS \SASDeploymentWizard\9.4 directory. On UNIX, this log typically resides in SASHOME/.SASAppData/SASDeploymentWizard directory.

When upgrading to SAS 9.4M4, you need to determine which solutions, if any, need to have their manual TLS settings for WebDAV and connection properties reset. This is because there are some solutions that do not preserve manual TLS settings for the WebDAV repository and connection properties. To do this review, complete Step 8 on page 314 and Step 9 on page 315.

Note: SAS processes should be running.

1. Edit the ***SAS-configuration-directory*\Levn\Web\WebServer\conf \\httpd.conf** file, and make the following changes:

   a. Add the # to the following line:

   ```
   #Include C:/SAS/Config/Levn/Web/WebServer/conf/extra/httpd-ssl.conf
   ```

   b. To use SAS Environment Manager to monitor SAS Web Server, locate the following line on Windows systems:

   ```
   Listen 80
   ```

   Replace the previous line with the following line:

   ```
   Listen localhost:80
   ```

   On UNIX systems, the port number is 7980.

2. For each instance of SAS Web Application Server, edit the **SAS-configuration-directory\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file and make the following changes to the **Connector** element:

   - Change the **proxyPort** attribute to specify the HTTP listen port.
   - Change the **scheme** to **http**.

3. Use SAS Management Console to update the protocol from HTTPS to HTTP and change the port number for each web application. For more information, see "Specify Connection Properties" on page 79.

   CAUTION:
   Do not modify the connection properties for the DP-SAS-Environment-Manager node. However, the connection properties for the Environment Mgr Mid-Tier node should be changed.

   Note the following items:

   - 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 connection information needs to be updated.
   - The connection properties for all **Environment Mgr Mid-Tier** applications should also be changed. Under **Application Management** ⇨ **Configuration Manager**, expand the **Environment Mgr Mid-Tier** node and make connection changes for each application in this node.
   - 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 do choose to modify these connections using a SAS script rather than SAS Management Console, the SAS_THEME table in the Shared Services 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** ⇨ **SAS Themes** ⇨ **SASTheme_default**. View the **Properties** to determine whether there is connection information that needs to be updated.

4. Use SAS Management Console to update the SAS Content Server connection information. For more information, see "Manual Configuration Tasks" on page 157.

   Note: The "Manual Configuration Tasks" information provides instructions about configuring security. Since you are removing security, for each of the applications using a WebDAV Repository URL, revert to the HTTP port and clear the **Secure** check box. Also, revert the connection parameters for the SAS Content Server.

5. 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 HTTP protocol and port number. Click **OK**.

      Note: You must perform these steps so that the HTTP connection information is saved in metadata.

6. Depending on which products you have installed, you might have to update the ***SAS-configuration-directory*\Levn\Web\WebServer\htdocs\sas\sas-environment.xml** file. For more information, see "Customize the SAS Environment File" on page 413.

7. To access SAS Environment Manager console with TLS enabled on SAS Web Server, edit the following files and locate the instances of the following URLs that begin with https://*server*:*port*. Modify them to point to http://*server*:*ssl-port*.

   a. Edit the ***SAS-configuration-directory*\Levn\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:

      ```
      <constructor-arg value="http://server:port/SASLogon/
      logout?sas_svcs_logon_LogonUrl=http%3A%2F%2Fserver%3Aev-server-port%2F"/>
      <property name="loginUrl" value="http://server:port/SASLogon"/>
      <property name="casUrl" value="http://server:port/SASLogon" />
      ```

   b. Edit the ***SAS-configuration-directory*\Levn\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:

```
<param-value>http://server:port/SASWebDoc</param-value>
<param-value>http://server:port/SASEnvironmentMgrMidTier</param-value>
<param-value>http://server:port/SASLogon/TimedOut.do
?sas_svcs_logon_LogonUrl=http://server:ev-server-port/</param-value>
```

c. Starting with SAS 9.4M3, you must edit the ***SAS-configuration-directory*\Levn\Web\SASEnvironmentManager\server-*version*-EE\hq-engine\hq-server\webapps\ROOT\WEB-INF\classes\identity-service.properties** file. Locate the following line and enter the correct information for your environment:

```
url.base=http\://server/
```

8. Update the SAS Content Server JVM options with the original HTTP URI values that were specified in Step 7.

   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 140.

9. To verify the reversion to non-TLS is complete, restart the SAS Web Server. Then, access the SAS Web Server from your web browser.

Revert Manual HTTPS Changes to SAS Web Application Server

Note: Before performing an update to SAS 9.4M3 or earlier, or if you are applying maintenance to a SAS 9.4M3 or previous system, you must revert any manual TLS configuration changes.

TIP If your manual changes are not successfully reverted, you will receive an error and the upgrade stops. For assistance in determining what might have been missed when reverting the changes, check the SAS Deployment Manager log, SDM_*YYYY-MM-DD-HH-MM-SS*.log. On Windows, it is located in the \Documents and Settings*user*\Local Settings\Application Data\SAS\SASDeploymentWizard\9.4 directory. On UNIX, this log typically resides in SASHOME/.SASAppData/SASDeploymentWizard directory.

Follow these steps:

1. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file. Modify the existing **Connector** element and complete the following:

a. Remove the following attributes:

- For SAS 9.4M5 and previous versions:

```
secure="true"
SSLEnabled="true"
sslProtocol="TLS" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
keystoreFile="/path-to-myhost.jks"
keystorePass="changeit"
```

- For SAS 9.4M6:

```
secure="true"
SSLEnabled="true"
sslProtocol="TLS" sslEnabledProtocols="+TLSv1,+TLSv1.1,+TLSv1.2"
keystoreFile="/path-to-myhost.jks"
keystorePass="changeit"
```

b. Change the port attribute.

- For SAS 9.4M4 and previous versions, change from **port="${bio.https.port}** to **port="${bio.http.port}"**.
- Starting with SAS 9.4M5, change from **port="${nio.https.port}** to **port="${nio.http.port}"**.

Note: This step must be repeated for each server that was configured to use HTTPS.

2. For SAS Web Application Server, set or add the following JVM options, ensuring that you choose the correct HTTP port that SAS Web Application Server is listening on. For SASServer1_1:

```
-Dsas.scs.port=http-port
-Dsas.scs.scheme=http
-Dsas.auto.publish.port=http-port
-Dsas.auto.publish.protocol=http
```

For SASServer*n_m*, set the following option:

```
-Dsas.auto.publish.port=http-port
```

Note: The HTTP port used by SAS Web Application Server can be found in the catalina.properties file by looking for the **bio.http.port** property for SAS 9.4M4 and previous releases or the **nio.http.port** property starting with SAS 9.4M5.

Also, for SASServer*n_m*, add the following option:

```
-Dsas.auto.publish.protocol=http
```

Note: For Windows deployments, edit the following files:

- ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\bin\setenv.bat**
- ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\wrapper.conf**

Note: After you modify the wrapper.conf file for 9.4M7 Feb 16th 2022 and later, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

For UNIX deployments, edit the ***SAS-configuration-directory*/Levn/Web/WebAppServer/SASServern_m/bin/setenv.sh** file.

3. For SAS Web Server, make the following changes:

   a. Edit ***SAS-configuration-directory*\Levn\Web\WebServer\conf\sas.conf** and change the **BalancerMember** directives to use **http** as the protocol and the HTTP port that SAS Web Application Server is listening on. See the following example:

   ```
   BalancerMember http://myhost.example.com:http-port
       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 http://myhost.example.com: http-port
       route=myhost.example.com_SASServer2_1
   ```

   b. Edit ***SAS-configuration-directory*\Levn\Web\WebServer\conf\sas.conf** and remove the following directives from the top of the file:

   ```
   SSLProxyEngine on
   SSLProxyVerify require
   SSLProxyVerifyDepth 10
   SSLProxyCACertificateFile "/path-to/myhost.pem"
   ```

4. To verify the reversion to non-TLS is complete, restart the SAS Web Application Server.

5. The secure attribute for cookies directs a web browser to send the cookie only through an encrypted HTTPS connection. To no longer allow SAS Web Application Server to return the session ID with the secure attribute, follow these steps:

   a. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file. Remove **secure="true"** from the existing **Connector** element.

   b. Edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\web.xml** file. Remove the following from the session-config:

   ```
   <cookie-config>
       <secure>true</secure>
   </cookie-config>
   ```

Note: If you are adding one or more new SAS 9.4 products that trigger the creation of a new web application server instance, you must perform these manual steps on the new server once it is created.

Update the Key and Certificate That Are Used by SAS Web Server

It is assumed that SAS Web Server was already configured for HTTPS, either during the initial installation and configuration using the SAS Deployment Wizard, or manually. For more information about the manual configuration, see "Configure SAS Web Server Manually for HTTPS" on page 312.

To update the key and certificate, 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. Move and update the certificate and key files.
   a. Make a backup of the files in the ***SAS-configuration-directory*\Levn\Web\WebServer\ssl** directory.
   b. Put the certificate file and key file in the **ssl** directory and rename the certificate and key files to the default names of the files that were backed up.

      By default, the files are named after the fully qualified domain name of the host machine.
   c. Change the default permissions of the key file to be read-only for the user that SAS Web Server is configured to run as (the file permissions should be set to **-r--------**).
4. Edit the ***SAS-configuration-directory*\Levn\Web\WebServer\conf\extra\httpd-ssl.conf** file. Locate the uncommented lines for the certificate file and key file and verify that the correct file names are specified:

   ```
   SSLCertificateFile "ssl/myhost.crt"
   SSLCertificateKeyFile "ssl/myhost.key"
   SSLCertificateChainFile "ssl/myhost.crt
   ```
5. (Optional) To verify that security has been configured correctly, start SAS Web Server. Then, access the secure SAS Web Server from your web browser.
6. For SAS 9.4M2 and previous releases, follow these steps on each machine in the deployment:
   a. Know the location of your self-signed or site-signed certificate.
   b. 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.

   For SAS 9.4M3 and later, use the SAS Deployment Manager to import your CA certificates into the trusted CA bundle onto each machine in the deployment. You need to specify the location of your self-signed or site-signed CA certificate to the SAS Deployment Manager, and it updates the SAS Private JRE for you. For more information, see "Manage Certificates in the Trusted CA Bundle Using the SAS Deployment Manager" in *Encryption in SAS*.
7. Configure the server tier and client tier.

UNIX Specifics
For SAS 9.4M2 and earlier, for the server tier, you can create a base64 encoded certificate chain file that contains all trust certificates in the chain and use the file in the SSLCALISTLOC= SAS system option. Create the chain file by concatenating the individual CA base64 files. For more information, see *Encryption in SAS*.

For SAS 9.4M3 and later, this step is no longer needed.

Windows Specifics
For server and client tiers machines, add any required CA certificates to the Windows truststore.

8. Start each SAS Web Application Server instance.

FIPS 140-2 Compliance

Overview

The following sections describe how to configure components in the middle tier to use cryptographic modules that are FIPS 140-2 compliant. Modules certified under FIPS 140-2 are valid for five years or until September 12, 2026. Completing these procedures does not result in middle-tier components that are FIPS 140 compliant, only that the components are using a FIPS 140 compliant cryptographic module.

More information about the Federal Information Processing Standard 140 can be found at https://csrc.nist.gov/publications/detail/fips/140/2/final.

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 (libtcnative)
- OpenSSL libraries

In SAS 9.4M1 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 SAS 9.4M2 or later, contact SAS Technical Support for assistance with getting the native libraries for your platform.

Starting with SAS 9.4M4, if you are upgrading you do not need to revert manual TLS changes made for SAS Web Server or SAS Web Application Server. However, you do need to revert the FIPS steps outlined in this section before upgrading.

Prior to SAS 9.4M7, certain maintenance activities (for example, adding a SAS web application, performing an update in place, and reconfiguring a SAS web application) overwrite the web application server and web server configuration files. Therefore, before performing any of these maintenance activities, you must revert the configuration modifications that are described in the following steps. To assist in this reversion, make a copy of the configuration files before modifying them. After the maintenance is complete, you must reconfigure HTTPS and FIPS manually.

Starting with SAS 9.4M7, you do not need to revert the FIPS settings, and no additional manual steps for configuring HTTPS and FIPS are required.

Starting with SAS 9.4M8, you must perform additional steps to ensure that the SAS Web Server and SAS Web Application Server are using FIPS-compliant TLS communications.

Starting with SAS 9.4M8, you must perform additional steps to ensure that the SAS Web Infrastructure Data Server, a PostgresSQL database, is using FIPS-compliant TLS communications. See ….

Configure SAS Web Server in SAS 9.4M8

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 through the mod_ssl module for Apache HTTP Server. OpenSSL has a FIPS module that is certified as FIPS 140-2 compliant. Follow the steps below to initialize the OpenSSL software in FIPS mode:

1. (OPTIONAL) You can define environment variables for the **SASHome** and **SASConfig** directories:

   ```
   export SASHOME=SAS-home-directory
   echo $SASHOME
   export SASCONFIG=SAS-config-directory
   echo $SASCONFIG
   ```

   Important: If these variables are defined, do not use them in configuration files or other environment variable definitions unless you also plan to make these definitions permanent.

   Note: *$SASHOME* and *$SASCONFIG* variables are referenced in the following steps. If you do not create the variables, you must write out the directory paths in place of the variable names.

2. Verify that the FIPS module, **fips.so**, is present in the ***SAS-home-directory/*** **SASWebServer/9.4/httpd-2.4/lib/ossl-modules** directory:

   ```
   ls -l $SASHOME/SASWebServer/9.4/httpd-2.4/lib/ossl-modules/fips.so
   ```

   Note: On Windows, the FIPS module is **fips.dll** and it is found in the **bin** directory.

3. Add the SAS Web Server's **lib** directory to the LD_LIBRARY_PATH environment variable:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SASHOME/SASWebServer/9.4/httpd-2.4/lib
   echo $LD_LIBRARY_PATH
   ```

4. Generate a new **fipsmodule.cnf** file that corresponds to this instance of the SAS Web Server.

   a. Navigate to the SAS Web Server's **bin** directory:

      ```
      cd $SASHOME/SASWebServer/9.4/httpd-2.4/bin
      ```

   b. Generate and output the new **fipsmodule.cnf** file to the **$SASCONFIG/ Levn/Web/WebServer/conf/extra** directory:

      ```
      ./openssl fipsinstall -out $SASCONFIG/Lev1/Web/WebServer/conf/extra/fipsmodule.cnf
      -module $SASHOME/SASWebServer/9.4/httpd-2.4/lib/ossl-modules/fips.so
      ```

Note: On Windows, the FIPS module is **`fips.dll`** that is referenced on the *-module* flag: ***`SAS-home-directory`*/SASWebServer/9.4/httpd-2.4/bin/fips.dll**

You should see output similar to this:

```
HMAC : (Module_Integrity) : Pass
SHA1 : (KAT_Digest) : Pass
SHA2 : (KAT_Digest) : Pass
SHA3 : (KAT_Digest) : Pass
TDES : (KAT_Cipher) : Pass
AES_GCM : (KAT_Cipher) : Pass
AES_ECB_Decrypt : (KAT_Cipher) : Pass
RSA : (KAT_Signature) : RNG : (Continuous_RNG_Test) : Pass
Pass
ECDSA : (PCT_Signature) : Pass
ECDSA : (PCT_Signature) : Pass
DSA : (PCT_Signature) : Pass
TLS13_KDF_EXTRACT : (KAT_KDF) : Pass
TLS13_KDF_EXPAND : (KAT_KDF) : Pass
TLS12_PRF : (KAT_KDF) : Pass
PBKDF2 : (KAT_KDF) : Pass
SSHKDF : (KAT_KDF) : Pass
KBKDF : (KAT_KDF) : Pass
HKDF : (KAT_KDF) : Pass
SSKDF : (KAT_KDF) : Pass
X963KDF : (KAT_KDF) : Pass
X942KDF : (KAT_KDF) : Pass
HASH : (DRBG) : Pass
CTR : (DRBG) : Pass
HMAC : (DRBG) : Pass
DH : (KAT_KA) : Pass
ECDH : (KAT_KA) : Pass
RSA_Encrypt : (KAT_AsymmetricCipher) : Pass
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
RSA_Decrypt : (KAT_AsymmetricCipher) : Pass
INSTALL PASSED
```

Important: The generated file is NOT transferrable to other machines. A new file must be re-created for each SAS Web Server instance.

Note: There is a default version of **`fipsmodule.cnf`** in ***`SAS-home-directory`*/SASWebServer/9.4/httpd-2.4/ssl** but it is not complete and therefore it cannot be used.

c. Verify the existence of the file:

```
ls -1 $SASCONFIG/Lev1/Web/WebServer/conf/extra/fips*
```

d. Review the contents of the file:

```
cat $SASCONFIG/Lev1/Web/WebServer/conf/extra/fipsmodule.cnf
```

e. You should see output similar to this:

```
[fips_sect]
activate = 1
install-version = 1
conditional-errors = 1
security-checks = 1
```

```
module-mac = 69:1C:F1:BF:D1:6D:E0:34:94:9F:51:A6:71:85:8B:A8:92:5D:15:B1:4C:E8:7D:
install-mac = 41:9C:38:C2:8F:59:09:43:2C:AA:2F:58:36:2D:D9:04:F9:6C:56:8B:09:E0:18
install-status = INSTALL_SELF_TEST_KATS_RUN
```

5. Copy the **openssl.cnf** configuration file from **$SASHOME** directory to **$SASCONFIG** directory:

   ```
   cp $SASHOME/SASWebServer/9.4/httpd-2.4/ssl/openssl.cnf \
   $SASCONFIG/Lev1/Web/WebServer/conf/extra/openssl.cnf
   ```

   Note: You cannot modify files in the *SAS-home-directory*. A best practice is to generate the file to the ***SAS-config-directory*/Lev1/Web/SASWebServer/conf/extra** directory.

6. You must set the OPENSSL_CONF and OPENSSL_MODULES environment variables, which are used by OpenSSL to identify the necessary files and directories.

   - On UNIX:

     ```
     export OPENSSL_CONF=$SASCONFIG/Lev1/Web/WebServer/conf/extra/openssl.cnf
     ```

     ```
     export OPENSSL_MODULES=$SASHOME/SASWebServer/9.4/httpd-2.4/lib/ossl-modules
     ```

     Important: Defining environment variables in an OS shell does not ensure that they will be defined in all subsequent environments. It might be necessary to take steps to ensure that the required environment variables will be defined in the run-time environment for the server in all scenarios. For example, the value might need to be added to the *httpdenv.sh* shell script on UNIX.

   - On Windows:

     The SAS Web Server runs in a Windows Service process, so the environment variables need to be defined in the definition of the service in the Windows registry.

     - To automatically define the environment variables:

       There is typically no need to add these environment variable definitions manually as it is handled automatically by the ***SAS-config-directory*\Levn\Web\WebServer\bin\httpdctl.ps1** PowerShell script that is used to install the Windows Service that hosts the SAS Web Server. The task is handled by a new SAS 9.4M8 PowerShell script named **OpenSSLEnv.ps1** in same location: ***SAS-config-directory*\Levn\Web\WebServer\bin**. When the **httpdctl.ps1** PowerShell script is run with an argument of *install*, the **OpenSSLEnv.ps1** script is invoked. The new script exists specifically to define OPENSSL_CONF= and OPENSSL_MODULES= in the Windows service run-time environment for the SAS Web Server.

     - To manually define the environment variables:

       Add a key within the fields of the service definition named **Environment**, of type **MultiString** (also known as **REG_MULTI_SZ**).

       The Service definition key for the SAS Web Server service is as follows:

       ```
       Registry::HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\$SASServiceName
       ```

       where *$SASServiceName* is similar to SAS[SASConfig-Lev1]httpd-WebServer

       The value that is assigned can be one or more **NAME=value** pairs, with each pair appearing on a separate line in the MultiString editor. The two environment variables that are needed are as follows:

```
set OPENSSL_CONF=%SASConfig%/Lev1/Web/WebServer/conf/extra/openssl.cnf

set OPENSSL_MODULES=%SASHome%/SASWebServer/9.4/httpd-2.4/bin
```

Important: Always use forward slashes in the above path values, even on Windows. The OpenSSL runtime does not handle backward slashes.

7. To configure FIPS support, edit the **SAS-configuration-directory/Levn/Web/WebServer/conf/extra/openssl.cnf** file.

   The OpenSSL configuration file has a content structure that must be considered when making changes. The file is divided into discrete sections. Each section starts with a section label that is within square brackets and appears on its own line. Properties and settings that are appropriate to a specific section are expected to appear after the section label and before the next section label. A property or setting that is not within its expected section will not be found or recognized.

   There is a section that does not have a label at the beginning of the file. It runs from the top of the file until the first section label. Other than a few global settings, new properties and settings should not appear in this section.

   - For FIPS usage, it is recommended that the **config_diagnostics** option is enabled to prevent accidental use of non-FIPS validated algorithms via broken or mistaken configuration. Typically, this is already enabled at the beginning of the file, outside of any section. Verify that this property exists with a value of 1. If it does not exist, add it.

     ```
     # Comment out the next line to ignore configuration errors
     config_diagnostics = 1
     ```

   - Uncomment the **[For FIPS]** section label, and uncomment and update the line that includes the **fipsmodule.cnf** file to refer to the file that was just generated:

     ```
     [For FIPS]
     # Optionally include a file that is generated by the OpenSSL fipsinstall
     # application. This file contains configuration data required by the OpenSSL
     # fips provider. It contains a named section e.g. [fips_sect] which is
     # referenced from the [provider_sect] below.
     # Refer to the OpenSSL security policy for more information.
     .include SAS-configuration-directory/Levn/Web/WebServer/conf/extra/fipsmodule.cnf
     ```

     Important: Always use forward slashes in this path value, even on Windows. The OpenSSL runtime does not handle backward slashes..

   - Verify that the **openssl_init** section exists and it is not commented out. Its value should refer to the provider section:

     ```
     [openssl_init]
     providers = provider_sect
     ```

   - In the **provider_sect** section, the default provider will already be defined. Verify that the section and the default provider both exist and are not commented out.

     Next, uncomment the following property/value pair:

     fips = fips_sect

     Finally, add the following property/value pair to this section:

     ```
     base = base_sect
     ```

     The **provider_sect** section looks like this:

```
# List of providers to load
[provider_sect]
default = default_sect
# The fips section name should match the section name inside the
# included fipsmodule.cnf.
fips = fips_sect
base = base_sect
```

The base provider does not include any cryptographic algorithms, but does include other supporting algorithms that might be required. It is designed to be used in conjunction with the FIPS module. The base provider loads a subset of algorithms that are also available in the default provider, but it specifically does not include any cryptographic algorithms.

- Add the base provider section and within this section, add the following property/value pair:

```
[base_sect]
activate = 1
```

- Verify that the **activate** property in the **[default_sect]** section is commented out.

```
[default_sect]
# activate = 1
```

The default provider contains some of the same cryptographic algorithms that theFIPS provider contains, and having them active would cause a conflict with FIPS.

8. Add the following statement before the VirtualHost directive in the ***SAS-configuration-directory*/Levn/Web/WebServer/conf/extra/httpd-ssl.conf** file:

```
SSLFIPS on
```

Partial file snippet:

```
SSLFIPS on
##
## SSL Virtual Host Context
##

<VirtualHost _default_:8343>
```

9. 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:

```
Operating in SSL FIPS mode
Digest: generating secret for digest ...
[Thu Apr 09 16:43:11 2015] [notice] Digest: done
Operating in SSL FIPS mode
```

Note: In this mode, the server establishes connections only with clients that use the TLSv1.2 or TLSv1.3 protocol and strong encryption.

Configure SAS Web Server in SAS 9.4 M7 and Prior Releases

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 through 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.

Configure 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. Three native components are required: APR library, OpenSSL libraries, and JNI wrappers for APR used by Tomcat (libtcnative).

To modify an existing SAS Web Application Server instance to use the APR, follow these steps:

1. Perform the steps in "Configure SAS Web Application Server for 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 UNIX deployments, on the primary SAS middle-tier machine and each middle-tier cluster node, add the following lines to **tcruntime-ctl.sh** and **setenv.sh** files. They are located in the ***SAS-configuration-directory*/Lev1/Web/WebAppServer/SASServer*n_m*/bin** directory.

  - For UNIX:

    ```
    LD_LIBRARY_PATH="SAS-Home-directory/SASWebServer/9.4/httpd-version/lib"
    export LD_LIBRARY_PATH
    ```

  - For IBM AIX:

    ```
    LD_LIB_PATH="SAS-Home-directory/SASWebServer/9.4/httpd-version/lib"
    export LD_LIB_PATH
    ```

    For SAS 9.4M6 and later releases, *version* is 2.4. For SAS 9.4M5 and previous releases, *version* is 2.2.

  Note: If you obtained the libraries from SAS Technical Support, provide the path to the location where you downloaded the libraries.

  Note: The ***SAS-Home-directory*/SASWebServer/9.4** directory should be available to the cluster system using a shared network or mapped drive to ensure that the SAS Web Server library files are accessible.

- For Windows deployments, edit the ***SAS-configuration-directory*\Lev1\Web\WebAppServer\SASServer*n_m*\conf\wrapper.conf** file on the primary SAS middle-tier machine and each middle-tier cluster node.

  - For the SAS 9.4M7 February 16, 2022 release and for SAS 9.4M8, add the following line to the *wrapper.conf* file:

    ```
    wrapper.java.library.path.1=SAS-Home-directory\SASWebServer\9.4\httpd-2.4\bin
    ```

    Also, add the following lines:

    ```
    # Java Library Path
    wrapper.java.library.path.1=%CATALINA_BASE%\bin\winx86_64
    wrapper.java.library.path.2=SAS-Home-directory\SASWebServer\9.4\httpd-2.4\bin
    ```

    After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

  - For the SAS 9.4M7 February 16, 2022 release, that is SAS 9.4M6 and earlier, or for SAS 9.4M7 without a patch from February 16, 2022 release, add the following line to *wrapper.conf*:

    ```
    set.PATH=SAS-Home-directory\SASWebServer\9.4\httpd-2.4\bin;%PATH%
    ```

    Also, add the following lines:

    ```
    # Java Library Path
    wrapper.java.library.path.1=%CATALINA_BASE%\bin\winx86_64
    wrapper.java.library.path.2=SAS-Home-directory\SASWebServer\9.4\httpd-version\bin
    ```

    For SAS 9.4M6, *version* is 2.4. For SAS 9.4M5 and previous releases, *version* is 2.2.

  Note: The ***SAS-Home-directory*\SASWebServer\9.4** directory should be available to the cluster system using a shared network or mapped drive to ensure that the SAS Web Server library files are accessible.

- Also, for Windows deployments, open the Windows Control Panel and select **System** ⇨ **Advanced system settings** ⇨ **Environment Variables**. Add the APR library path to the PATH environment variable.

3. On the primary SAS middle-tier machine and each middle-tier cluster node, edit the ***SAS-configuration-directory*/Lev1/Web/WebAppServer/SASServer*n_m*\conf\server.xml** file.

   a. Add the following listener to the Server element:

   ```
   <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:

   ```
   <Connector acceptCount="100" connectionTimeout="20000"
     executor="tomcatThreadPool" maxKeepAliveRequests="15"
     port="8443" scheme="https" secure="true"
     protocol="org.apache.coyote.http11.Http11AprProtocol"
     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-8.0-doc/config/http.html#SSL_Support_-_APR/Native.

   c. In addition, for Windows only, add the following parameter to the **Connector** element:

   ```
   address="0.0.0.0"
   ```

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 https://docs.vmware.com/en/VMware-tc-Server/10.1/tc-server/GUID-topics-manual.html?hWord=N4IghgNiBcIGYEsAOBnABARgCwAYQF8g#configuring-fips-140-mode-for-a-tc-runtime-instance/. 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.

5. Starting with SAS 9.4M8, in order for the SAS Web Application Server to communicate using FIPS compliant TLS encryption algorithms, the SAS Web Server must first be configured to use FIPS compliant TLS communications. See "Configure SAS Web Server in SAS 9.4M8" on page 352.

   In SAS 9.4M8, the FIPS configuration for the SAS Web Application Server uses the same configuration files and modules that the SAS Web Server uses.

6. For SAS 9.4 M8, you must set OPENSSL_CONF and OPENSSL_MODULES environment variables because they are used by OpenSSL to identify the files and directories that are needed:

   - On UNIX:

     ```
     export OPENSSL_CONF=$SASCONFIG/Lev1/Web/WebServer/conf/extra/openssl.cnf
     ```

     ```
     export OPENSSL_MODULES=$SASHOME/SASWebServer/9.4/httpd-2.4/lib/ossl-modules
     ```

     Note: The best practice for getting these environment variables defined in the server's run-time environment is to put the definitions into the server's */bin/setenv.sh* file. That ensures that they are always defined when you execute the commands to start the server.

     Important: Be sure to **`export`** the definitions of the OPENSSL_* variables. Starting the server involves creating a new system process for the server to run in, and environment variables that are not exported do not automatically appear in a new process.

   - On Windows:

     1. Add the environment variables to the SAS Web Application Server's configuration file, ***`SAS-config-directory`*\`Levn`\`Web`\`WebAppServer`\`SASServern_m`\`conf`\`wrapper.conf`**:

     ```
     set.OPENSSL_CONF="SAS-config-directory/Levn/Web/WebServer/conf/extra/openssl.cnf"
     ```

     ```
     set.OPENSSL_MODULES="SAS-home-directory/SASWebServer/9.4/httpd-2.4/bin"
     ```

     Important: Always use forward slashes in the above path values, even on Windows. The OpenSSL runtime does not handle backward slashes.

     Note: The dot between the 'set' keyword and the environment variable name is correct syntax for the *wrapper.conf* file.

     Note: The value must be enclosed in quotation marks if the path value contains blanks. Otherwise, quotation marks are not necessary.

     Note: An alternative is to define the environment variables in the definition of the service in the Windows registry.

     2. Once the changes have been made to the *wrapper.conf* file, the Windows service hosting the web application server must be updated to incorporate those changes into the service's run-time environment. Follow the steps located here: "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide*

7. Starting with SAS 9.4M8, in addition to a shared network path or mapped drive to the *SAS-home-directory* directory of the SAS Web Server, the SAS Web Application Server also requires a shared network path or mapped drive to the *SAS-configuration-directory* directory of the SAS Web Server.

   In addition, in a multi-machine configuration where the SAS Web Server resides on a different machine than the SAS Web Application Server, and FIPS is configured for use, you must create a configuration file with the locally mapped paths.

Create a file called **sas-remote.cfg** in the ***SAS-config-directory/Levn/Web/Common*** directory with two properties similar to the following:

```
SASWebServerRemoteHome = /mnt/sas/remote/webserver/home
SASWebServerRemoteConfig = /mnt/sas/remote/webserver/config
```

In this example, the local file system location that the remote directories are mounted to is not a requirement. The two directories can be mounted anywhere in the local file system as long as the mount paths are properly specified in the **sas-remote.cfg** file. The two property names in that file, *SASWebServerRemoteHome* and *SASWebServerRemoteConfig*, are requirements and they are case sensitive.

Important: If the value of a property uses the backslash character (\ ) as a directory separator, each backslash must be escaped by preceding it with the escape character, which is also a backslash. For example, '\' becomes '\\'. As an alternative, the back-slash characters in the path value can be replaced with a forward slash, (/). The Java Virtual Machine always recognizes the forward slash as a directory separator, even on Windows.

Example *sas-remote.cfg* file:

```
# ------------------------------------------------------------------------------
# sas-remote.cfg
# ------------------------------------------------------------------------------
#
# PURPOSE: Record the locally mapped paths to the SAS Web Server installation.
#
# CONFIGURATION FILE FORMAT:
#
# See the Java documentation for the load(Reader reader) method in the
# Properties class for a full description of the properties file format.
#
#   https://docs.oracle.com/javase/7/docs/api/java/util/Properties.html
#
# The basic file formatting rules are:
#
#  - Properties are processed in terms of lines.
#
#  - A line that contains only white space characters is considered blank and is
#    ignored. A comment line has an ASCII '#' or '!' as its first non-white
#    space character.
#
#  - The data of a key-value pair may be spread out across several adjacent
#    lines by escaping the line terminator sequence with a backslash character
#    \. Note that comment line cannot be extended in this manner; every comment
#    line must have its own comment indicator.
#
#  - If a key-value pair is spread across several lines, the backslash escaping
#    the line terminator sequence, the line terminator sequence, and any white
#    space at the start of the following line have no affect on the key or
#    value.
#
#  - The key contains all of the characters in the line starting with the first
#    non-white space character and up to, but not including, the first unescaped
#    '=', ':', or white space character other than a line terminator.
#
# ------------------------------------------------------------------------------
```

```
#
# WARNING: If the value of a property uses the back-slash character ( \ ) as a
# directory separator, each back-slash must be escaped by preceding it with the
# escape character (also a back-slash), thus  \  == becomes ==>  \\       As an
# alternative, the back-slash characters in the path value can be replaced with
# a forward-slash ( / ).  The Java Virtual Machine will recognize the
# forward-slash as a directory separator - yes - even on Windows.
#
# ---------------------------------------------------------------------------

#
# The location on the local system that maps to the SAS Web Server installation
# in SASHome on the remote system.
#
# The mapped location on the remote system will typically be a path something
# like %SASHome%\SASWebServer\9.4 (with %SASHome% resolved).
#
SASWebServerRemoteHome = /mnt/sas/remote/webserver/home


#
# The location on the local system that maps to the SAS Web Server instance
# in SASConfig on the remote system.
#
# The mapped location on the remote system will typically be a path something
# like %SASConfig%\Lev1\Web\WebServer (with %SASConfig resolved).
#
SASWebServerRemoteConfig = /mnt/sas/remote/webserver/config
```

Configure SAS Web Infrastructure Platform Data Server in SAS 9.4M8

The SAS Web Infrastructure Platform Data Server is based on PostgreSQL. Starting in SAS 9.4M8, it can be configured to use a FIPS 140 compliant cryptographic module. Perform the following steps to enable the FIPS module.

Configure PostgreSQL To Use TLS For Secure TCP/IP Connections

1. PostgreSQL reads the system-wide OpenSSL configuration file called **`openssl_conf`**. To find the location of this file, run the following command:

   ```
   openssl version -d
   ```

   In the **`openssl_conf`** file, verify that **`keyUsage = cRLSign, keyCertSign`** is uncommented. This property is found in the **`[ v3_ca ]`** section of the file.

   Note: SAS Web Infrastructure Platform Data Server is deployed on the same machine as Foundation SAS. Starting with SAS 9.4M8, SAS no longer delivers OpenSSL libraries for SAS Foundation servers. See "Using Operating System OpenSSL Libraries with SAS 9.4M8" in *Encryption in SAS* for more information.

2. Uncomment the following TLS parameters in the ***`SAS_configuration_directory`***`/Levn/`**`WebInfrastructurePlatformDataServer/data/postgresql.conf`** file. Set the value of **`ssl`** to **`true`**.

```
ssl = true
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ca_file = 'ca.crt'
```

To start in TLS mode, files containing the server certificate and private key must exist. The following table summarizes the files that are relevant to the TLS setup on the server. By default, these files are expected to be named **server.crt** and **server.key** in the server's data directory, but other names and locations can be specified using the configuration parameters **ssl_cert_file** and **ssl_key_file**.

$PGDATA represents the location of the postgreSQL data directory:

```
PGDATA = SAS_configuration_directory/Levn/WebInfrastructurePlatformDataServer/data
```

Table 18.5 *SSL Server File Usage*

File	Contents	Effect
ssl_cert_file **$PGDATA/server.crt**	server certificate	sent to client to indicate server's identity
ssl_key_file **$PGDATA/server.key**	server private key	proves server certificate was sent by the owner; does not indicate certificate owner is trustworthy
ssl_ca_file	trusted certificate authorities	checks that client certificate is signed by a trusted certificate authority
ssl_crl_file	certificates revoked by certificate authorities	client certificate must not be on this list

3. For instructions on creating certificates, see "Secure TCP/IP Connections with SSL" in PostgreSQL documentation: https://www.postgresql.org/docs/14/ssl-tcp.html.

Configure SAS Web Application Server To Access PostgreSQL in TLS Mode

1. Modify the **server.xml** for each SAS Web Application Server instance. The file is located here: ***SAS_configuration_directory*/Levn/Web/SASServern_n/conf**. Add **?sslmode=require** to the end of each **jdbc:postgresql** URL. The URLs can be found within the **<GlobalNamingResources>** section. Here is an example:

   Change:

   ```
   xaProperties.url="jdbc:postgresql://postgresql_machine:9432/SharedServices"
   ```

   to:

   ```
   xaProperties.url="jdbc:postgresql://postgresql_machine:9432/SharedServices?sslmode=requi
   ```

2. Restart SAS Web application Server.

Verify That The Database Is In FIPS Mode

1. Temporarily set the **ssl_ciphers** parameter in the **postgresql.conf** file to **'RC4-SHA'**. This file is located here: ***SAS_configuration_directory*/Levn/WebInfrastructurePlatformDataServer/Data**
2. Attempt to restart the database service. The database will fail to start because **RCA-SHA** is not approved as a FIPS 140-2 cipher.

   Note: Use the Event Viewer on Windows. The error is found in Event Viewer (Local) - Windows Logs -Application:

   ```
   FATAL: could not set the cipher list (no valid ciphers available)
   ```

3. Revert your change.

Note: If you deployed your SAS 9.4M8 software or downloaded your SAS 9.4M8 software order prior to June 1, 2023, you must apply a hot fix for the SAS Web Infrastructure Platform Data Server 9.4M8 that is available here: https://tshf.sas.com/techsup/download/hotfix/HF2/M1Y.html#M1Y002. This ensures PostgreSQL is FIPS 140 compliant. See "Applying Hot Fixes" in *SAS Guide to Software Updates and Product Changes* for instructions on how to use the SAS Deployment Wizard to retrieve any relevant hot fixes.

Note: If you are installing SAS 9.4M8 for the first time or you are migrating previous SAS deployments to SAS 9.4M8, your version of PostgreSQL is automatically upgraded to PostgreSQL 14 or higher. No additional steps are required. However, if you are performing an upgrade in place, those deployments are not automatically upgraded to PostgreSQL 14 or higher. See "SAS 9.4M8: Upgrading to PostgreSQL 14" in *SAS Guide to Software Updates and Product Changes*.

Allowlist of Websites and Methods Allowed to Link to SAS Web Applications

Overview of the Allowlist

Starting with SAS 9.4M3, websites that link directly to your SAS web applications, via URLs, must be added to an allowlist, 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 Forgery (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 allowlist. 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 allowlist 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 allowlist 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 allowlist can also be manually edited. For more information, see "Modify the Allowlist for URLs and HTTP Request Methods" on page 365.

Note: The port number must be specified if the allow listed site uses port numbers other than the standard 80 for HTTP or 443 for HTTPS.

You can also allowlist 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 allow listed, then all types of HTTP requests are subject to CSRF protection. SAS web applications that use allow listed 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 "Modify the Allowlist for URLs and HTTP Request Methods" on page 365.

Modify the Allowlist for URLs and HTTP Request Methods

CAUTION:

You can choose to disable allowlist 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 allowlist for URLs and HTTP request methods, 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.
5. Complete one of the following:
   - To add a URL or HTTP request method to the allowlist, if one has not been previously defined:
     1. Click **Add** and specify one of the following properties and required values:

Table 18.6 Allowlist Property Descriptions

Property Name	Property Value
sas.web.csrf.referers.allowNull	Specify whether to allow requests that are missing referer headers. When the security filter is set to false, all incoming HTTP requests are required to have a valid origin or referer header. You might set this property to false for specific SAS web applications that do not have API endpoints and for which you know that clients could always send the required header. For example, if a site has an additional proxy server that always added these headers from trusted clients. The default value is true.
sas.web.csrf.referers.blacklist	Specify a list of sites to block, that would otherwise be allowed by a wildcard rule. For example, if you add *.*example*.com to the allowlist but wanted to disallow test.*example*.com, you can add test.*example*.com to the denylist.

Property Name	Property Value
sas.web.csrf.referers.knownHosts	Specify the URL of servers, other than SAS servers, that can perform certain actions within the SAS installation. This allows users to add additional known hosts to the list of known hosts that are automatically calculated by the system. Specify 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. *Note:* You can restrict a value to an application on the allow listed site by including the application's path in the value. Here is an example: http://*hostname.example*.com/*my-application*/. *Note:* By default, during a migration of a product from one version to another version, the value for this property continues to specify the source server.
sas.web.csrf.referers.performCheck	Specify whether the security filter should be run. When set to false, no checking is performed, regardless of the value of any other setting. The default value is true.
sas.web.csrf.referers.skipMethods	Specify the HTTP request method to exclude from security filtering as a comma-delimited string. For example, to skip the GET, OPTIONS, and TRACE methods, enter the following: **`GET,OPTIONS,TRACE`**
sas.web.navigation.performBackUrlCheck	Specifies whether to disable the back URL verification and allow any URL to be passed in.
sas.web.navigation.knownHosts	Specifies entries to be added to the list of allowed URLs.

2. Click **OK** to close the Define New Property dialog box.
3. Click **OK** to close the SAS Application Infrastructure Properties dialog box.

- To change or delete a URL or HTTP request method on the allowlist:
  1. Change or delete the value in the **Property Value** field that corresponds to the property that you want to change or delete.
  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.

Cross Site Request Forgery Token Checking

Overview

Starting with SAS 9.4M6, support has been added to prevent Cross Site Request Forgery (CSRF) attacks using the synchronizer token pattern. CSRF synchronizer token checking uses a filter that handles incoming requests, generating a unique CSRF synchronizer token if necessary and storing the token in the HTTP session. This filter is responsible for CSRF enforcement, verifying that certain requests (for example, HTTP POST requests) contain a CSRF token in the request parameters, and this token matches the value stored in the HTTP session. Requests that fail the check receives a 403 error code.

Important: By default, CSRF synchronizer checking is enabled only on SAS products and solutions that support this functionality. Do not enable this feature, unless it was previously enabled. You can disable this feature by completing the following task.

This feature should eventually be enabled in all SAS web applications. To verify whether CSRF token checking is enabled for a web application, use SAS Management Console to check if the *sas.web.csrf.token.performCheck* property is configured to **`true`**.

Disable Cross Site Request Forgery Token Checking

To disable CSRF token checking, complete the following:

1. Log on to SAS Management Console.
2. On the **Plug-ins** tab, expand **Application Management** ⇨ **Configuration Manager**.
3. Right-click the SAS web application name, and select **Properties**.
4. Click the **Advanced** tab.
5. Set the *sas.web.csrf.token.performCheck* property to **`false`**.

   You can also set the HTTP methods using the *sas.web.csrf.token.allowedMethods* property.

   A description of each property is below:

   Table 18.7 *Properties and Descriptions*

Property Name	Description
sas.web.csrf.token.performCheck	Specifies whether synchronizer token checking is enforced. If this property is set to **`false`**, no synchronizer token checking is performed, regardless of the value of any other setting. The default value is **`true`**.

Property Name	Description
sas.web.csrf.token.allowedMethods	Specify which HTTP methods do not require a synchronizer token on requests, as a comma-delimited string. It is often safe to allow idempotent HTTP methods from CSRF checking (if the implementation of the method is idempotent). For example, to allow the GET, HEAD, TRACE, and OPTIONS methods without a synchronizer token, enter the following: **`GET,HEAD,TRACE,OPTIONS`**

6. Click **OK** to close the SAS Application Infrastructure Properties dialog box.
7. Stop the middle tier, using the method that is appropriate for your operating system:
   - Windows

     Using the Services Snap-in, right-click on each of the services in the list (in the order in which they are listed), and select **Stop**:

     Note: The list of services that you see, and need to stop, depends on which managed web application servers are installed in your environment.

     - SAS Environment Manager Agent
     - SAS Environment Manager
     - SAS Web App Server: SASServer2_1
     - SAS Web App Server: SASServer12_1
     - SAS Web App Server: SASServer1_1
     - SAS Web Server
     - SAS Cache Locator Service: ins_41415
     - SAS JMS Broker
   - UNIX

     Run ***`SAS-configuration-directory`***`/sas.servers stop`.
8. Restart the middle tier.
   - Windows

     Using the Services Snap-in, right-click on each of the services in the list (in the order in which they are listed), and select **Start**:

     Note: The list of services that you see, and need to start, depend on which managed web application servers are installed in your environment.

     - SAS JMS Broker
     - SAS Cache Locator Service: ins_41415
     - SAS Web Server
     - SAS Web App Server: SASServer1_1
     - SAS Web App Server: SASServer12_1
     - SAS Web App Server: SASServer2_1

  - SAS Environment Manager
  - SAS Environment Manager Agent
- UNIX

  Run ***SAS-configuration-directory*/sas.servers start**.

Configure the Cross Domain Proxy Servlet Through an Allowlist

Overview of the Allowlist

Starting with SAS 9.4M3, in scenarios where applications are using the SAS mid-tier as a proxy for accessing external URLs, additional security has been added through an allowlist and logging. If an attempt is made to access a domain that is not on the allowlist, an error message is generated. All external URLs that are accessed are logged at warning level. These logs can be used for security audits.

The allowlist 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.

Modify the Allowlist

To add, change, or delete the allowlist for URLs, 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.
5. Complete one of the following:
   - To add a URL to the allowlist, 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: The comma-separated list should not include a space after the commas.

     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 allowlist:
  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.

Optional Configuration for the Cross Domain Proxy Servlet

In any environment where the internal and external connection information must be different because of access rules, you must specify the **`-Dsas.web.html.cdps.use.internal.urls=true`** JVM option for SAS Web Application Server. The **`-Dsas.web.html.cdps.use.internal.urls`** option is needed for the Cross Domain Proxy Servlet to use internal URIs when external URIs are requested by applications.

Note: The **`-Dsas.web.html.cdps.use.internal.urls`** option works only when the **`-Dsas.retry.internal.url`** platform option is set to true.

See Also

- "Specify JVM Options" on page 44
- "Specify Connection Properties" on page 79

Enable Support for Forward Proxy Authentication

The SAS middle-tier environment can be configured to forward external URL requests through a proxy. Starting with SAS 9.4M3 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 18.8 *JVM Options for Forward Proxy Authentication*

JVM Option	Proxy Configuration	Description
-Dsas.flex.ntlmKerberosSupport	NTLM	When set to true, specifies whether Flex applications are supported.
-Dhttp.auth.ntlm.domain	NTLM	Specifies the domain name.
-Dhttp.auth.ntlm.workstation	NTLM	Specifies the workstation name. This option is optional.

JVM Option	Proxy Configuration	Description
-Dhttp.proxyUser	Basic authentication NTLM	Specifies the user name.
-Dhttp.proxyPassword	Basic authentication NTLM	Specifies the password.
-Dhttp.proxyHost	No authentication Basic authentication NTLM	Specifies the host name, or address, of the proxy server for HTTP connections.
-Dhttps.proxyHost	No authentication Basic authentication NTLM	Specifies the host name, or address, of the proxy server for HTTPS connections.
-Dhttp.nonProxyHosts	No authentication Basic authentication NTLM	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 '|' character. In addition, the wildcard character '*' can be used for pattern matching. For example: `-Dhttp.nonProxyHosts="*.example.com|localhost"`
-Dhttp.proxyPort	No authentication Basic authentication NTLM	Specifies the port number that the proxy server uses for HTTP connections.
-Dhttps.proxyPort	No authentication Basic authentication NTLM	Specifies the port number that the proxy server uses for HTTPS connections.

See Also

"Specify JVM Options" on page 44

SAS Anonymous Web User

Overview

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.

Create the SAS Anonymous Web User

If you did not choose the option to create the SAS Anonymous Web User (webanon) during the initial installation, follow these steps to create the webanon account:

1. Create the web anonymous user as an internal account in the metadata. The SAS Metadata Server must be running.

   Note: Alternatively, you can configure the web anonymous user by using an external account if you need the account to launch a workspace server.

   a. Log on to SAS Management Console as an unrestricted user.

   b. Right-click the **User Manager** plug-in and select **New** ⇨ **User**.

   c. In the New User Properties window, on the **General** tab , provide the name as webanon and the display name as SAS Web Anonymous User.

   d. On the **Accounts** tab, click **Create Internal Account**.

   e. Provide a password. Then, click **OK**.

   f. On the **Groups and Roles** tab, move the Visual Analytics: Basic role from the **Available Groups and Roles** pane to the **Member of** pane.

   g. Click **OK**.

2. Encode the password.

   a. Using Base SAS software, run the following PWENCODE procedure to encode the password. Replace *webanon_password* with the password that you used to create the web anonymous user in SAS Management Console.

   ```
   proc pwencode in="webanon_password";
   run;
   ```

   b. Locate the sas005-encoded password in the SAS log and copy the value (including the {SAS005}). You will use the password in the next step. For more information, see "PWENCODE Procedure" in *Encryption in SAS*.

   c. Update the sas_application_property table in the SAS Shared Services database.

   Note: The SAS Web Infrastructure Platform Data Server must be running. Depending on what database you are using, the method to connect and execute SQL statements varies. Consult your database administrator for more information about updating your SAS Shared Services database.

Complete one of the following procedures, depending on your operating system:

- On Windows, use the pgAdmin3 tool to update the table and complete the following:

  1. Navigate to the ***SASHOME*** **\SASWebInfrastructurePlatformDataServer\9.4\bin** directory.
  2. Double-click the pgAdmin3.exe file.
  3. Click **Add a connection to a server**.
  4. Provide a name for the server registration, a host, a port (9432), a user name (dbmsowner), and a password. Then, click **OK**.
  5. In the Object Browser, double-click **Server Groups** and select **Servers** ⇨ ***hostname*** ⇨ **Databases**.
  6. Click **SharedServices** to connect.
  7. Click **Execute arbitrary SQL queries** and enter the following:

```
INSERT INTO SAS_APPLICATION_PROPERTY (PROPERTYSET_NM, PROPERTY_NM,
PROPERTY_VALUE_TXT) VALUES ('Environment.Properties',
'web.anonymous.userid', 'webanon@saspw');
INSERT INTO SAS_APPLICATION_PROPERTY (PROPERTYSET_NM, PROPERTY_NM,
PROPERTY_VALUE_TXT) VALUES ('Environment.Properties',
'web.anonymous.password' , 'encoded-password');
```

Note: The previous commands must be entered on one line. They are shown on multiple lines for display purposes only.

  8. Click **Execute Query**. You should see the following message on the **Messages** tab in the Output pane:

```
Query returned successfully: 1 row affected, 16 ms execution time.
```

  9. Select **File** ⇨ **Exit** to exit the Query window.
  10. Select **File** ⇨ **Exit** to exit pgAdmin3.

- On UNIX, use the **PSQL** tool to update the table and complete the following:

  1. Navigate to ***SASHOME/*** **SASWebInfrastructurePlatformDataServer/9.4/bin**.
  2. Enter the following code to set up the PSQL environment.

```
POSTGRES_HOME=SASHOME/SASWebInfrastructurePlatformDataServer/9.4 export
PATH=${POSTGRES_HOME}/bin:$PATH export LD_LIBRARY_PATH=${POSTGRES_HOME}
/lib:$LD_LIBRARY_PATH
```

Note: The previous code must be entered on one line. It is shown on multiple lines for display purposes only.

  3. Enter the following command. You are prompted for the Shared Services user's password.

```
psql -h localhost -p 9432 -U SharedServices -c "INSERT INTO
SAS_APPLICATION_PROPERTY (PROPERTYSET_NM, PROPERTY_NM, PROPERTY_VALUE_TXT)
VALUES ('Environment.Properties', 'web.anonymous.userid', 'webanon@saspw');"
psql -h localhost -p 9432 -U SharedServices -c "INSERT INTO
SAS_APPLICATION_PROPERTY (PROPERTYSET_NM, PROPERTY_NM, PROPERTY_VALUE_TXT)
```

```
 VALUES ('Environment.Properties', 'web.anonymous.password',
'encoded-password');"
```

Note: The previous command must be entered on one line. It is shown on multiple lines for display purposes only.

Use 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.

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.

Enable HTTPS Strict Transport Security

HTTP Strict Transport Security (HSTS) is a security feature that authorizes a web server to notify web browsers that only secure connections (HTTPS) are allowed. Connections via the HTTP protocol are not allowed. HSTS is a time-based system. When you enable HSTS, you specify the amount of time during which the web server can be accessed only using secure HTTPS connections.

To enable HSTS, complete the following:

1. Edit the ***SAS-configuration-directory*\Levn\Web\WebServer\conf\extra\httpd-ssl.conf** file.
2. Add the following statement inside the VirtualHost directive:

   ```
   Header always set Strict-Transport-Security "max-age=time-in-seconds;
   includeSubdomains; preload"
   ```

For more information about the directive options, see https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet.

Recommended Security Settings

SAS Web Server

The following server settings are configured by default and harden security for SAS Web Server.

Table 18.9 *Security Settings for SAS Web Server*

Setting	Value	Description
ServerSignature	Off	Configures the footer on server-generated documents. This directive is defined in the ***SAS-configuration-directory*** **\Levn\Web\WebServer\conf\extra\httpd-default.conf** file.
ServerTokens	Prod	Configures the Server *HTTP* response header. This directive is defined in the ***SAS-configuration-directory*** **\Levn\Web\WebServer\conf\extra\httpd-default.conf** file.
TraceEnable	Off	Determines the behavior on *TRACE* requests. This directive is defined in the ***SAS-configuration-directory*****\Levn\Web\WebServer\conf\sas.conf** file.
X-Frame-Options	SAMEORIGIN	Specifies whether the browser should allow the web page to be displayed in a frame within another web page. This HTTP header is defined in the ***SAS-configuration-directory*** **\Levn\Web\WebServer\conf\sas.conf** file.
X-XSS-Protection	1; mode=block	Enables the cross-site scripting (XSS) filter, which prevents the rendering of a web page when an XSS attack is detected. This HTTP header is defined in the ***SAS-configuration-directory*****\Levn\Web\WebServer\conf\sas.conf** file.

Setting	Value	Description
X-Content-Type-Options	nosniff	Prevents the web browser from overriding the response content type. This HTTP header is defined in the ***SAS-configuration-directory*** **\Levn\Web\WebServer\conf\sas.conf** file.
Strict-Transport-Security	max-age=31536000	Protects web pages against protocol downgrade attacks and cookie hijacking. This HTTP header is defined in the ***SAS-configuration-directory*** **\Levn\Web\WebServer\conf\sas.conf** file.
TLS	SAS 9.4M8: V1.3, V1.2 SAS 9.4M7 and earlier: V1.2	Specifies the default protocol setting for data in motion security. For details, see "TLS and OpenSSL Version Support Prior to SAS 9.4M8" in *Encryption in SAS*.

SSLCipherSuite	SAS 9.4M8:
	`TLS_AES_128_GCM_SHA256`
	`TLS_AES_256_GCM_SHA384`
	`TLS_AES_128_CCM_SHA256`
	`TLS_AES_128_CCM_8_SHA256`
	`ECDHE-ECDSA-AES128-GCM-SHA256`
	`ECDHE-RSA-AES128-GCM-SHA256`
	`ECDHE-ECDSA-AES256-GCM-SHA384`
	`ECDHE-RSA-AES256-GCM-SHA384`
	SAS 9.4M7 and earlier:
	`ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256`
		Specifies the OpenSSL cipher specifications that configure the Cipher Suite that the client is permitted to negotiate during SSL handshake. This directive is defined in the ***SAS-configuration-directory*** **\Levn\Web\WebServer\conf\extra\httpd-ssl.conf** file.

CA Certificate Requirements for SAS Visual Analytics

SAS Visual Analytics provides functionality that enables you to print a report to PDF. When this action is performed using the command line, a CURL command is used and requires CA certificates to be installed in the default operating system keystore. Encrypted environments should add all CA certificates to the operating system keystore, **/etc/pki/**.

To add the CA certificates to the keystore on UNIX systems, complete the following steps:

1. Install the ca-certificates package:

   ```
   yum install ca-certificates
   ```

2. Enable the dynamic CA configuration feature:

   ```
   update-ca-trust force-enable
   ```

3. Add the CA files as new files to **/etc/pki/ca-trust/source/anchors/**:

   ```
   cp file-containing-Root-CA /etc/pki/ca-trust/source/anchors/
   cp file-containing-Intermediate-CA /etc/pki/ca-trust/source/anchors/
   ```

   Note: The location of the source files, *file-containing-Root-CA* and *file-containing-Intermediate-CA*, depends on how HTTPS was configured for your SAS Web Server. See "Update the Key and Certificate That Are Used by SAS Web Server" on page 349.

   Note: It might not be necessary to copy the CA files if the certificate for the SAS Web Server was obtained from a well-known 3rd party CA. In that case, the CA certificates will already be present in the operating system keystore that was installed using the *yum install ca-certificates* command.

4. Run the following command:

   ```
   update-ca-trust extract
   ```

Linux Security Hardening

Overview

The following sections provide an introduction to the hardening of Linux environments. This information does not assume that you deploy a specific Linux operating system or have deployed specific applications on the server. This information avoids including specific commands, and instead focuses on applying core security principles to the Linux environment.

Users, Groups, and Permissions

The following is a list of security considerations for users, groups, and permissions:

- Manage access control.
  - Access control to the server should be strictly controlled, enforcing the principle of least privilege.
  - Only authorized individuals, such as system administrators, should be permitted to log in to the server.
  - To further limit the possibility of an attack in the machine, limit the number of users and groups that you create to only those that are necessary for operation.
- Limit the use of the root account.
  - Users, including system administrators, who require elevated privileges, should log in with their own user account and then use **`sudo`**.
  - You can use the audit trail left by **`sudo`** to track privileged actions back to the specific user that executed those actions.
- Application deployments should include their own user and group account on the system, and should not be executed as root. A group is typically created to support an application suite, in which it might be necessary to share files.
- Ensure that a strong password policy is enforced.
  - Password policies should be strong enough to deter brute-force and dictionary-based attacks, but flexible enough to deter users from writing the password down. The Linux PAM module can assist with this.
  - A lockout policy should also be added, so that successive failed login attempts lock the user out the system, and prevent an attacker from gaining access to the machine.

Once users and groups are configured, they should be applied to directories in a manner that enforces the principle of least privilege. For example, application deployments should include only files that can be accessed by the application user. Private key files should be locked down to the fullest extent possible, typically with Read-Only access for the owner and nothing else. Consider making the **`/boot`** directory read-only to prevent unauthorized modifications. Finally, enabling Security-Enhanced Linux (SELinux) can provide additional protection to files through the use of mandatory access control (MAC).

Libraries and Core Services

All operating system components, including services, libraries, and the Linux kernel, should be regularly patched, updated, and upgraded. Adopting a change management policy ensures that these updates are applied on a regular basis. When packages are no longer needed, be sure to uninstall them. Likewise, be sure to disable services and protocols that are not being used in your environment (for example, IPv6, which is not used within the internal networks of most companies).

A firewall should always be enabled. By default, the rules should deny all access. Then, selectively allow access to services needed for your deployed applications (such as HTTPS) and services needed to remotely administer the machine, such as Secure Shell (SSH). SSH access should be permitted only from internal IP addresses. SSH configurations should be locked down to the greatest extend possible (for example, never permit root logins and require users to use public key pairs instead of passwords to log in).

Logging and Auditing

Logs should be sent to a dedicated logging server in order to prevent attackers from easily modifying local log files. Log files should be reviewed and audited regularly to look for signs of unauthorized or suspicious activity.

Additional Resources

For additional information, see the following resources:

CIS Benchmarks
: The Center for Internet Security (CIS) provides benchmarks for several variations of Linux. These benchmarks provide configuration baselines and best practices for securely configuring a system.

Security Enhanced Linux
: SELinux is a patch to the Linux kernel, originally designed by NSA. It provides MAC, allowing administrators to closely control who has access to system resources.

Configure the Same-Site Cookie Attribute for SAS 9.4M7 and Later Releases

Note: HTTPS must be configured prior to setting the same-site attribute.

1. For each instance of SAS Web Application Server, edit the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServern_m\conf\context.xml** file.
2. Add the following code:

```
  <CookieProcessor
class="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
sameSiteCookies="none" />
```

The following table provides the possible values for the same-site attribute and descriptions for the values.

Table 18.10 *Same-Site Cookie Attribute Values and Descriptions*

Value	Description
unset	The same-site cookie attribute is not set. This is the default value.
none	The same-site cookie attribute is set and the cookie is always sent in cross-site requests.
lax	The browser sends the cookie only in same-site requests and cross-site top-level GET requests.
strict	The browser prevents sending the cookie in any cross-site request.

3. Restart SAS Web Application Server.

Part 5

Tools and Utilities

Chapter 19

Use the SAS Web Infrastructure Platform Utilities

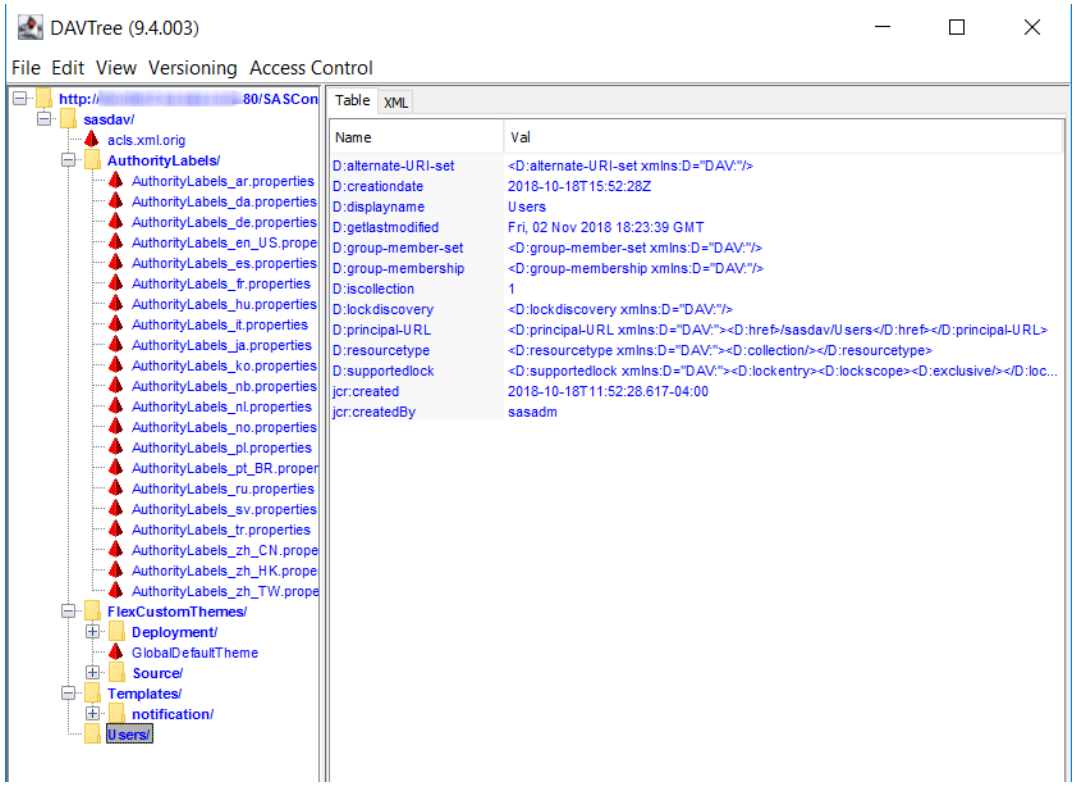
Use the DAVTree Utility to Manage WebDAV Content

Overview

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`\Levn\Web\Utilities\DAVTree.bat`

   On UNIX:

   SAS-configuration-directory`/Levn/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

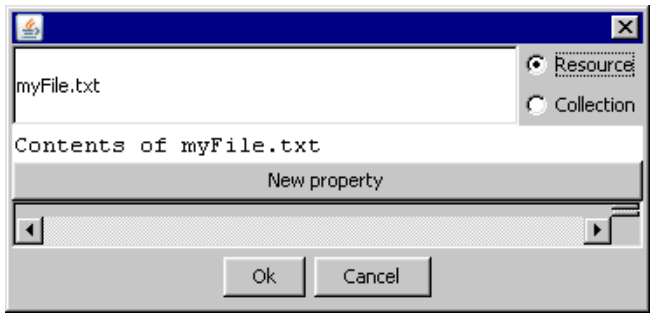
You can copy both text files and binary files to the repository. To copy a file, click and drag the file from the file system to a folder in the DAVTree interface. This action can be performed on Windows systems and on UNIX systems that provide a graphical interface.

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.

Use the Package Cleanup Utility to Remove Packages

Overview

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`\Levn\Web\Utilities\PackageCleanup.bat`

On UNIX:

SAS-configuration-directory`/Levn/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*.

Delete 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 **-promptеach**. 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.

-metapass *Metadata-Server-password*
Specify the password to use when connecting to the SAS Metadata Server.

-domain *authentication-domain*
Specify the authentication domain for the SAS Metadata Server.

-package | **-pkg** *package-name*
Specify the name of a package to delete.

-url *WebDAV-URL*
Specify the WebDAV URL to use to locate packages to delete.

-username *WebDAV-username*
Specify the user name to use to connect to a WebDAV server.

-password *WebDAV-password*
Specify the password to use to connect to a WebDAV server.

-logfile | **-log** *file-name*
Specify the name of a log file to create. If the log file already exists, then the log lines are appended to the current file.

-noprompt
The utility does not prompt for confirmation of deletions.

-deletenodate
The utility lists or deletes packages that do not have an expiration date.

-prompteach
The utility prompts you to confirm each package individually for deletion.

-debug
The utility produces debugging information for all the SAS Foundation Services.

-help
The utility displays this help information. (You must also provide the -metauser, -metapass, and -domain arguments in order to get the Help information.)

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.

```
PackageCleanup -d "2009.10.07 at 12:59 PM"  -metauser userX -metapass passX
     -domain DefaultAuth -url http://myhost.com/Sales/Packages -username davUserX
     -password davPasswordX -pkg 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
```

Use JMX Tools to Manage SAS Resources

Overview of 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.

Access the SAS MBeans

Overview

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 392.
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 392.

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.

How to Use the SAS MBeans

Overview

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 manageServerByName operation.

manageServerByName(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 manageServerByName() 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

Chapter 20

SAS Configuration Scripting Tools

Overview

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*\Levn\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 20.1 *Resource Types, Operations, and Scopes*

Resource Type	Operations	Scopes
server	configure, unconfigure, start, stop, restart	global
mailsession	configure, unconfigure	server
datasource	configure, unconfigure	server
loginmodule	configure, unconfigure	server
application	deploy, undeploy	server
jmserver	configure, unconfigure, start, stop, restart	global
jms	configure, unconfigure	server
balancer	configure, unconfigure	global
member	configure, unconfigure	global
proxypass	configure, unconfigure	global
proxyserver	configure, unconfigure, start, stop, restart	global
cache_locator	configure, unconfigure, start, stop, restart	global
cache_server	configure, unconfigure, start, stop, restart	global

Manage 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.

Rebuild 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.

Execute 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.

Execute 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`*****`\Levn\Web`**.

global.1.configLevWebStagingDir
identifies the path to ***`SAS-configuration-directory`*****`\Levn\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`**, **`unx`**, 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`*****`\Levn\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`*****`\Levn\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*\Levn\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*\Levn\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*\Levn\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.

deleted
is a Boolean value. If set to **`true`**, then this resource has been marked as deleted.

deletedTargets
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.

Part 6

Appendices

Appendix 1

Configure the SAS Environment File

Overview

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.

Configure the SAS Environment File

Customize 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 version="1.0"  encoding="UTF-8">
```

```
<environments xmlns="http://www.sas.com/xml/schema/sas-environments-9.4"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.sas.com/xml/schema/sas-environments-9.4
    http://www.sas.com/xml/schema/sas-environments-9.4/sas-environments-9.4.xsd"
    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>
</environment>
```

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:

```
<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.

Appendix 2

Administer Custom Applications

Overview

Note: The former title of this appendix topic was *Administer Multicast Applications*. It is changed to *Administer Custom Applications* starting with SAS 9.4M8.

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). SAS Remote Services, which enables multicast options, is turned off by default. However, in SAS 9.4 M7 and prior releases of SAS 9.4, you can still take advantage of this communication if you have developed custom applications by starting Remote Services. See *Use of SAS Remote Services to Enable Multicast Options in SAS 9.4M7 and Prior Releases* in this appendix.

Starting with SAS 9.4 M8, SAS Remote Services is no longer supported. Any custom applications that used SAS Remote Services in prior releases need to be re-written to work with SAS 9.4 M8 software.

SAS Remote Services Is No Longer Supported in SAS 9.4M8

SAS Remote Services is not used by SAS 9.4. The service was still installed to support custom applications, but it was not started by default. Starting with SAS 9.4 M8, SAS

Remote Services is no longer supported. Any custom applications that used SAS Remote Services in prior releases need to be re-written to work with SAS 9.4 M8 software.

If your existing SAS 9.4 environment has custom applications that relied on SAS Remote Services, SAS Remote Services would have been enabled and running on your middle-tier machine. To verify whether it is running, you can do the following:

- On UNIX, use the sas.servers script to display the status of your servers:

  ```
  SAS-configuration-directory/Lev1/sas.servers status
  ```

- On Windows, use the Windows Services Manager to verify whether SAS Remote Services is running. (SAS Remote Services might be running as a Windows service even if there are no custom applications using it.)
- You can use the System Evaluation application, which will print information about your system and let you know whether Remote Services is running. The application is included in SAS 9 Content Assessment and you can download it from the Downloads & Hotfixes page. For information about the application, see "Executing the System Evaluation Application" in *SAS Content Assessment*.

Use of SAS Remote Services to Enable Multicast Options in SAS 9.4M7 and Prior Releases

Overview

By default, multicasting is not used in the typical SAS deployment, and SAS Remote Services is turned off. If you created custom applications that use SAS Remote Services, you can use multicasting and enable SAS Remote Services. When installation is performed with the SAS Deployment Wizard, the wizard generates a default multicast address that is based on the 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) and time to live (TTL), 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 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 "Configure Multicast Options" on page 419.

Configure 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

Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

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 "Specify JVM Options" on page 44.

Key Multicast Properties

The following table shows some key multicast properties.

Table A2.1 *Multicast Properties*

Property	Default Value	Unit	Description
multicast.address	239.*X.Y.Z*	Not applicable	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.
multicast.port	8561	Not applicable	This value is provided by the SAS Deployment Wizard prompting mechanism and represents the port on which UDP communication occurs.
multicast_udp_ip_ttl	1	Decimal. 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.

Property	Default Value	Unit	Description
multicast.security	Not applicable	Not applicable	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
multicast.config.file	Not applicable	URL string (file://, http://, and so on)	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.

Configure a Multicast Authentication Token

Understand 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.

Reconfigure to Use a Multicast Authentication Token

To generate a token and set the token for SAS Remote Services, do the following:

1. Use SAS and the PWENCODE procedure to generate an encoded password to use as the multicast authentication token. Here is an example: {SAS005}ADD8AB7108595A7D1A69190D78CDFE6145C1EB849CC7A43D.

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.

   Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

4. Restart SAS Remote Services.

Next, to set the token for the report output generation tool, do the following:

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=-Dsas.app.launch.picklist=picklist;"help\primary.picklist"
   JavaArgs_14=-DMULTICAST_AUTHENTICATION_TOKEN=token
   Classpath=-cp "<VJRHOME>/eclipse/plugins/sas.launcher.jar"
   ```

Configure the JGroups Bind Address

Understand 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:

```
13:39:35,602 ERROR [ContextLoader] Context initialization failed
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 SAS Web Application Server, and the SAS BI Report Services Report Generation tool if the previous error message displays.

Set 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
   ```

   Note: After you modify the wrapper.conf file for the SAS 9.4M7 February 16, 2022 and later release, you need to rebuild the Windows service for each SAS Web Application Server instance. See "Rebuild Windows Service for Each SAS Web Application Server Instance" in *SAS Intelligence Platform: System Administration Guide* for more details.

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%" ^
      -classpath "%CLASSPATH%" ^
      -Dsas.ext.config="C:\Program Files\SASHome\sas.java.ext.config" ^
   ```

```
-Djgroups.bind_addr=ip-address
```

3. Restart SAS Remote Services.

Set 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.

Appendix 3

Validate the Secured Middle-Tier Environment

Overview

This section assumes that you configured SAS Web Server, SAS Web Application Server, or SAS Environment Manager to use HTTPS.

SAS Deployment Wizard can be used to configure SAS Web Server and SAS Environment Manager to use HTTPS. You can also manually configure SAS Web Server for HTTPS. For more information, see "Configure SAS Web Server Manually for HTTPS" on page 312.

After the deployment, additional manual steps are required for SAS Environment Manager. For more information about the post-deployment configuration of SAS Environment Manager, see "Configure SAS Environment Manager for HTTPS" on page 325.

SAS Web Application Server must be configured manually to use HTTPS. For more information, see "Configure SAS Web Application Server for HTTPS" on page 318.

Once these environments have been configured, they must be validated by completing the following sections.

Validate Listening Ports

After configuring HTTPS, validate the listening ports on the SAS Web Server, SAS Web Application Server, or SAS Environment Manager. To confirm that the TCP listening ports on the SAS middle tier are functioning, follow these steps:

1. Log on to the SAS middle-tier machine as the SAS Installer user.

   Note: If there is more than one SAS middle-tier machine, log on to the primary SAS middle-tier machine.

2. Enter the **netstat** command to list the listening ports on the respective server:
   - For SAS Web Server: **netstat -an|grep LISTEN|grep "8343\|7980"**

     You should see the following:

     ```
     tcp 0 0 127.0.0.1:7980 0.0.0.0:* LISTEN
     tcp 0 0 :::8343 :::*
     ```

     Note: If the SAS Web Server has been automatically configured for HTTPS, the listening port on 7980 is disabled, so only port 8343 is shown.
   - For SAS Web Application Server: **netstat -an|grep LISTEN|grep "8443\|8543\|9443\|9543"**

     You should see the following:

     ```
     tcp 0 0 :::8543 :::* LISTEN
     tcp 0 0 :::9443 :::* LISTEN
     tcp 0 0 :::9543 :::* LISTEN
     tcp 0 0 :::8443 :::* LISTEN
     ```
   - For SAS Environment Manager: **netstat -an|grep LISTEN|grep "7080\|7443"**

     You should see the following:

     ```
     tcp 0 0 :::7443 :::* LISTEN
     ```
3. Perform steps 1 and 2 for every machine in your SAS middle tier.

Validate TLS Settings

You can validate the TLS settings for the following SAS middle-tier components:

- SAS Web Server
- SAS Deployment Agent
- SAS Environment Manager
- SAS Web Application Servers

To validate the TLS settings for the servers in the SAS middle tier, follow these steps:

1. Log on to the primary SAS middle-tier machine as the SAS Installer user.
2. Change the directory to the location where the trusted CA bundle resides. Here is an example: **cd /opt/SASHome/SASSecurityCertificateFramework/1.1/cacerts/trustedcerts.pem**
3. Enter the following **openssl** command to validate SAS Web Server:

   path-to-openssl s_client -connect myhost.example.com:secured-port -CAfile trustedcerts.pem

   Here is an example: **/usr/bin/openssl s_client -connect myhost.example.com:8343 -CAfile trustedcerts.pem**

   You should see the following:

   ```
   CONNECTED(00000003)
   depth=1 C = US, ST = My State, L = My Town, O = Example, OU = ACME, CN = BUS Env Root
   CA, emailAddress = nr@.example.com verify return:1 depth=0 C = US,
   ```

```
ST = My State, L = My Town, O = Example, OU = Acme, CN = myhost.example.com
verify return:1
---
Certificate chain
0 s:/C=US/ST=My State/L=My Town/O=Example/OU=BUS/CN=myhost.example.com
  i:/C=US/ST=My State/L=My Town/O=Example/OU=BUS/CN=Int CA/emailAddress=nr@example.com
1 s:/C=US/ST=My State/L=My Town/O=Example/OU=BUS/CN=Int CA/emailAddress=nr@example.com
  i:/C=US/ST=My State/L=My Town/O=Example/OU=BUS/CN=Root CA/emailAddress=nr@example.com
---
Server certificate
-----
BEGIN CERTIFICATE-----
MIIEfjCCAmYCCQDvN3bQlLq36jANBgkqhkiG9w0BAQsFADCBkDELMAkGA1UEBhMC VVMxFzAVBgNVBAgMDk
DYXJ5MQwwCgYD VQQKDANTQVMxDDAKBgNVBAsMA0dFTDEYMBYGA1UEAwwPR0VMIEVudiBSb290IENB MSMw
ZIhvcNAQkBFhRub3JlcGx5QG5vbmUuc2FzLmNvbTAeFw0xNTA4MTMw ODUxNDRaFw0yNTA4MTAwODUxNDRa
AJBgNVBAYTAlVTMRcwFQYDVQQIEw5O b3J0aCBDYXJvbGluYTENMAsGA1UEBxMEQ2FyeTEMMAoGA1UEChMD
wCgYD VQQLEwNHRUwxHjAcBgNVBAMTFWdlbHNhczAzLnJhY2Uuc2FzLmNvbTCCASIwDQYJ KoZIhvcNAQEB
PADCCAQoCggEBAJiOsle9tGwlQIdBaKnkDSLXmZj5uuVc H87IHSuPxcr5bK+XlmaY9Forgd36XwmZFGiAP
kPBTcpAE9M8AEAerePh+ WA3zeZUrd6nLGiN18oASgqN2HYMnl2UA6fRUlpB5d/vKSGd7Bdw3ZIwpTZ/Ay5
3JBy9QAutVP9k2Db0pmL86h5FAc 1SisnsPzlaFw26wY2TY5Kwbw=
-----END CERTIFICATE-----
subject=/C=US/ST=My State/L=My Town/O=Example/OU=BUS/CN=myhost.example.com issuer=
/C=US/ST=My State/L=My Town/O=Example/OU=BUS/CN=Root CA/emailAddress=nr@.example.com
---
No client certificate CA names sent
Server Temp Key: DH, 768 bits
---
SSL handshake has read 3419 bytes and written 431 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES128-SHA
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol : TLSv1.2
    Cipher : DHE-RSA-AES128-SHA
    Session-ID:
55CCB690344AD7746948249F4148748C00B4C3A18886520859A70007B15A65D4
    Session-ID-ctx:
    Master-Key:
8B19E0A0F74AD10EEF69523BDEF8DB70AA7E437F722D2152717ACDEE89ACC540C29DCB206F866
B296EB237E33059BE95
    Key-Arg : None
    Krb5 Principal: None
    PSK identity: None
    PSK identity hint: None
    Start Time: 1439479440
    Timeout : 300 (sec)
    Verify return code: 0 (ok)
---
```

The following line indicates that TLS is properly configured for the SAS middle-tier component whose secured port you entered:

```
Verify return code: 0 (ok)
```

4. Press Ctrl-C to quit the **`openssl`** command.
5. Repeat steps 3 and 4 for each SAS middle-tier component, using the secured ports from the following table:

Table A3.1 *Default Secured Ports for SAS Middle-Tier Components*

SAS Middle-Tier Component	Default Secured Port
SAS Web Server	8343 443 (Windows)
SAS Deployment Agent	5660
SAS Environment Manager	7443
SAS Web Application Server 1	8443
SAS Web Application Server 2	8543
SAS Web Application Server 11	9443
SAS Web Application Server 12	9543

Verify Cookie Settings

After you configure SAS Web Application Server for HTTPS, you can validate the cookie settings by accessing the SAS portal from a Windows client machine.

To validate the cookie settings in the Google Chrome web browser on Windows, follow these steps:

1. Run Google Chrome. Navigate to SAS Web Server (for example, **`https://myhost.example.com:8343/SASPortal`**).
2. Select **Advanced** ⇨ **Proceed to *myhost.example.com* (unsafe)**.
3. Log on to SAS Information Delivery Portal.
4. After SAS Information Delivery Portal has loaded, open a new web browser tab, and enter **`chrome://settings/cookies`**.
5. Select ***myhost.example.com*** ⇨ **CASTGC**, and make sure that **Send for: Secure connections only** is selected.
6. Log off from SAS Information Delivery Portal, and close Google Chrome.

Appendix 4

Troubleshooting the Middle-Tier Environment

Memory Error in Web Application Server Log in SAS 9.4M8 with an EBI Deployment on IBM AIX

You might encounter a memory error that impedes the start up of the middle-tier when you install SAS 9.4M8 on AIX. The hosts of the middle-tier need at least 32G of physical memory, which should alleviate this issue.

The following error in the ***SAS-config-directory*/Levn/Web/WebAppServer/SASServer1_1/logs/server.log** file indicates that there is not enough memory:

```
ERROR (Catalina-utility-2) [org.apache.geode.internal.cache.control.HeapMemoryMonitor]
No tenured pools found.  Known pools are: [(Name=class storage;Type=Non-heap
memory;UsageThresholdSupported=false), (Name=JIT code cache;
Type=Non-heap memory;UsageThresholdSupported=false), (Name=tenured-LOA;Type=Heap
memory;UsageThresholdSupported=true), (Name=nursery-survivor;Type=Heap memory;
UsageThresholdSupported=false),
(Name=JIT data cache;Type=Non-heap memory;UsageThresholdSupported=false),
(Name=tenured-SOA;Type=Heap memory;UsageThresholdSupported=true),
(Name=nursery-allocate;Type=Heap memory;UsageThresholdSupported=false),
(Name=miscellaneous non-heap storage;Type=Non-heap memory;UsageThresholdSupported=false)]
```

To remove this error message from the log, do the following:

1. Add the **XX:+HeapManagementMXBeanCompatibility** option to the ***SAS-config-directory*/Levn/Web/WebAppServer/SASServer1_1/bin/setenv.sh** file.
2. Restart the SAS Web Application Server.

Beginning with IBM Java 8 (IBM J9 virtual machine), there are important enhancements made to the IBM Garbage Collector and Memory Pool MXBeans to provide more detailed information about Garbage collection activity and associated memory pools. Because VMware GemFire expects only one heap memory pool and relies on the memory pool name "Java heap", you need to use the XX:+HeapManagementMXBeanCompatibility option so that IBM J9 virtual machine is compatible with earlier versions of the VM.

Error While Performing an Update in Place in a High-Availability Environment

The high-availability deployment might require an adjustment to the default pool size value for database connections. In the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file, you can increase

or decrease the *maxPoolSize* property for each SAS Web Application Server instance as needed for your environment. The file should look similar to the following:

```
<Resource auth="Container" driverClassName="org.postgresql.Driver"
factory="com.sas.vfabrictcsvr.atomikos.BeanFactory" maxPoolSize="100"
minPoolSize="10" name="sas/jdbc/SharedServices"
password="${pw.sas.jdbc.SharedServices}" testQuery="select 1"
type="com.atomikos.jdbc.nonxa.AtomikosNonXADataSourceBean"
uniqueResourceName="sas/jdbc/SharedServices"
url="jdbc:postgresql://rdcesx04001.race.sas.com:9432/SharedServices"
user="SharedServices"/>
```

In the ***SAS-configuration-directory*\Levn\WebInfrastructurePlatformDataServer\data\postgresql.conf** file, increase the *max_connections* property for the SAS Web Infrastructure Data Server. The value should be less than or equal to the total value of all the *maxPoolSize* properties for all the SAS Web Application Servers in the deployment (horizontal and vertical). The file should look similar to the following:

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = 'localhost'           # what IP address(es) to listen on;
                                  # comma-separated list of addresses;
                                  # defaults to 'localhost'; use '*' for all
                                  # (change requires restart)
#port = 5432                       # (change requires restart)
max_connections = 256              # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directories = '/tmp'  # comma-separated list of directories
                                  # (change requires restart)
#unix_socket_group = ''                   # (change requires restart)
#unix_socket_permissions = 0777           # begin with 0 to use octal notation
                                  # (change requires restart)
#bonjour = off                     # advertise server via Bonjour
                                  # (change requires restart)
#bonjour_name = ''                 # defaults to the computer name
                                  # (change requires restart)
```

During an update in place, the SAS Deployment Wizard will not preserve the values of the *maxPoolSize* and *max_connections* properties. The values are reverted to the default values. In additional, an error might occur at Stage 7, Step 2 when each SAS Web Application Server instance is started. If you changed the value of the *maxPoolSize* and *max_connections* properties, then change the values back for each SAS Web Application Server instance and restart the SAS Web Infrastructure Data Server. Then, retry Stage 7, Step 2 in the SAS Deployment Wizard.

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 support.sas.com/en/books.html. 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: support.sas.com/en/books.html

Glossary

authentication
the process of verifying the identity of a person or process for security purposes. Authentication is commonly used in providing access to software, and to data that contains sensitive information.

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-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.

credential
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). Credentials are used to authenticate a user.

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, client-side pooling, server-side pooling.

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, pool.

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. *See also* SAS Metadata Repository, WebDAV repository.

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

R

S

T

X