



THE
POWER
TO KNOW.

SAS[®] 9.2

Intelligence Platform

Web Application

Administration Guide,

Third Edition



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® 9.2 Intelligence Platform: Web Application Administration Guide, Third Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.2 Intelligence Platform: Web Application Administration Guide, Third Edition

Copyright © 2010, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227–19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, May 2010

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/publishing or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

<i>What's New</i>	ix
Overview	ix
Common Web Administration	ix
Better Integration with Third-Party Security Products	xi
SAS Content Server	xi
SAS Web Application Themes	xi
SAS Information Delivery Portal Administration	xi
SAS BI Portlets Administration	xii
SAS Web Report Studio Administration	xii
SAS BI Dashboard Administration	xiv

PART 1 Getting Started 1

Chapter 1 △ Before You Begin 3
Introduction to This Guide 3
Accessibility Features in the SAS Intelligence Platform Products 3
Prerequisites for Administering the Web Applications 3
High-Level Overview of Administrative Tasks 4
Chapter 2 △ Working in the Middle-Tier Environment 7
Understanding the Middle-Tier Environment 7
Third-Party Software Components 9
SAS Web Infrastructure Platform 10
SAS Content Server 13
SAS Shared Services 14
SAS Foundation Services 14
SAS Web Applications 14
Starting the Web Applications 17

PART 2 Middle-Tier Environment and Topologies 21

Chapter 3 △ Best Practices for Configuring Your Middle Tier 23
Best Practices for Middle-Tier Configuration 23
Sample Middle-Tier Deployment Scenarios 24
Tuning the Web Application Server 34
Configuring a Cluster of Web Application Servers 34
Configuring HTTP Sessions in Environments With Proxy Configurations 34
Using an HTTP Server to Serve Static Content 35
Using a Proxy Plug-in between the Web Application Server and the HTTP Server 36
Using Apache Cache Control for Static Content 37
Chapter 4 △ Middle-Tier Security 39

Middle-Tier Security	39
Using the SAS Anonymous Web User With SAS Authentication	40
Multicast Security	40
Using Single Sign-On Among Web Applications	41
Using Secure Sockets Layer (SSL) for Web Applications	41
Configuring and Deploying Restrictive Policy Files	45

Chapter 5 △ Interacting with the Server Tier 53

Configuration Shared between the Middle Tier and the Server Tier	53
Configuring an SMTP Mail Server for Use with the SAS Middle Tier	53
Client-Side Pooling and Server-Side Pooling Options	54
Configuring Data Sources Used by the SAS Middle Tier	54
Configuring Application Response Measurement (ARM) Capabilities	56

PART 3 Middle-Tier Administration 59

Chapter 6 △ Administering the SAS Web Infrastructure Platform 61

SAS Web Infrastructure Platform	61
SAS Management Console	63
Using Configuration Manager	64
Setting Global Properties for SAS Applications Using SAS Application Infrastructure Properties	67
Specifying Connection Parameters for HTTP and HTTPS Sessions	70
Using the SAS Web Administration Console	72

Chapter 7 △ Using the SAS Web Infrastructure Platform Utilities 79

Using the DAVTree Utility to Manage WebDAV Content	79
Using the Package Cleanup Utility to Remove Packages	82
Using JMX Tools to Manage SAS Resources	87

Chapter 8 △ Administering SAS Web Applications 93

Using the SAS Deployment Manager	94
Rebuilding the SAS Web Applications	94
Redeploying the SAS Web Applications	97
Reconfiguring the Web Application Server	101
Deploying SAS OnlineDoc Manually for the Web	102
Working with Exploded EAR Files in a Development Environment	103
Administering Logging for SAS Web Applications	105
Configuring Auditing for SAS Web Applications	109
Configuring a Custom Logoff Message for Web Application Users	111
Configuring the HTTP Session Time-out Interval	112
Configuring the Display of a Warning Message for Inactive User Sessions	114

Chapter 9 △ Administering SAS Shared Services 117

About SAS Shared Services	117
Setting the Default Alert Notification Delivery Type	117

Administering the SAS Comment Manager Web Application 118

Chapter 10 △ **Administering the SAS Content Server** 121

About the SAS Content Server 121

Backing Up the SAS Content Server 122

Using the SAS Content Server Administration Console 122

Implementing Authorization for the SAS Content Server 127

Reconfiguring the SAS Content Server to Share the Database Used by SAS Shared Services 129

Chapter 11 △ **Administering SAS BI Web Services** 135

Managing Generated Web Services 135

Configuring SAS BI Web Services for .NET 136

Configuring SAS BI Web Services for Java 137

Overview of Security for Web Services 142

Securing SAS BI Web Services for .NET 143

Securing SAS BI Web Services for Java 144

Additional Administrative Tasks for SAS BI Web Services for Java 150

Chapter 12 △ **Administering SAS Web Application Themes** 153

Overview 153

Steps for Defining and Deploying a New Theme 156

Deploying SAS Web Application Themes on a Different Web Application Server 165

Deleting a Custom Theme from the Metadata 167

Migrating Custom Themes 167

Chapter 13 △ **Administering Multicast Options** 169

Overview of Multicasting 169

Configuring Multicast Options 169

Configuring Multicast Security 172

Multicasting with JGroups 173

PART 4 **SAS Web Report Studio Administration** 175

Chapter 14 △ **Introduction to SAS Web Report Studio Administration** 177

Introduction to SAS Web Report Studio 177

About SAS Web Report Viewer 178

SAS Web Report Studio and the SAS Intelligence Platform 178

New Features in the Administration of SAS Web Report Studio 4.2 178

Prerequisites for Administering SAS Web Report Studio 180

Main Tasks for Administering SAS Web Report Studio 180

Additional Documentation for SAS Web Report Studio 182

Chapter 15 △ **Configuring SAS Web Report Studio** 183

Configuring SAS Web Report Studio 183

Configuring Logging for SAS Web Report Studio 188

Improving the Performance of SAS Web Report Studio 192

Redeploy SAS Web Report Studio 196

Chapter 16 △ **Managing SAS Web Report Studio Content and Users** 197

SAS Web Report Studio Folders 197

Predefined Roles 200

Adding Content for Use by Report Creators 207

Managing Access to Reports 214

Chapter 17 △ **Customizing SAS Web Report Studio Report Styles** 221

SAS Web Application Themes and Custom Report Styles 221

Customizing Report Styles for SAS Web Report Studio 221

Add Disclaimer Text to Graphs and Tables 232

Specify Fonts for PDF Reports Generated by SAS Web Report Studio 233

Chapter 18 △ **Pre-generated Reports From SAS Web Report Studio** 235

Overview of Pre-generated Reports from SAS Web Report Studio 235

Configuring a Scheduling Server 239

Verifying Permissions for the Distribution Library 241

Setting Up a Recipient List for Report Distribution 241

Processing Reports Outside of SAS Web Report Studio 248

PART 5 **SAS Information Delivery Portal Administration** 259

Chapter 19 △ **Overview of the SAS Information Delivery Portal** 261

Introduction to the SAS Information Delivery Portal 261

Understanding the SAS Information Delivery Portal 262

Understanding the Portal Components 263

Chapter 20 △ **Introduction to SAS Information Delivery Portal Administration** 267

Prerequisites for Administering the Portal 267

Who Can Administer the Portal 269

Main Tasks for Administering the Portal 270

Suggestions for Verifying Portal Operation 273

Important Portal Administrative Files 274

Logging for SAS Information Delivery Portal 274

Additional Documentation for the Portal 275

Chapter 21 △ **Administering Portal Authorization** 277

Overview of Portal Authorization Tasks 277

Planning for Portal Users and Groups 278

Understanding Portal Authorization 280

Configure a Group Content Administrator 282

Sharing Content in the Portal 283

Setting Up Authorization for SAS Publication Channels 287

Managing Portal Permission Trees in Metadata 288

Chapter 22 △ **Adding Content to the Portal** 291

Overview of Adding Content	293
Summary of Content That Can Be Added to the Portal	294
Understanding Pages and Page Templates	296
Adding, Editing, and Removing Pages	302
Adding, Editing, and Removing Page Templates	304
Understanding Portlets	312
Main Steps to Add a Portlet	316
Adding WebDAV Graph Portlets	317
Understanding Portlet Deployment	321
Hiding Portlets from Users	322
Adding Custom-Developed Portlets	324
Removing Portlet Configurations	326
Adding Links	327
Adding Files	328
Adding Custom Web Applications	329
Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java	333
Adding Syndication Channels	337
Adding SAS Packages	340
Adding SAS Publication Channels	342
Executing SAS Stored Processes from the SAS Information Delivery Portal	345
Adding SAS Information Maps	349
Adding SAS Reports	350

PART 6 SAS BI Portlets Administration 351

Chapter 23 △ Administering SAS BI Portlets	353
Introduction to SAS BI Portlets	353
Using the SAS BI Portlets from a Portal	354
Configuring SAS BI Portlets for the WebSphere Portal	355
Removing SAS BI Portlets from the WebSphere Portal Server	359

PART 7 SAS Business Intelligence Dashboard Administration 361

Chapter 24 △ Administering SAS BI Dashboard	363
Overview of SAS BI Dashboard	363
Accessing SAS BI Dashboard	364
Main Tasks for Administering SAS BI Dashboard	364
Setting Up Indicator Alerts for SAS BI Dashboard Users	365
Understanding the Data Source XML (DSX) Files	366
Specifying the Location of SAS Data Sets for SAS BI Dashboard	367
Improving the Performance of SAS BI Dashboard	367
Configuring Alert Latency with Event Generation Framework	372
Seamless Access to SAS BI Dashboard From SAS Information Delivery Portal	374
Enabling the Display of Custom Repository Folders in SAS BI Dashboard	374
Managing Security for SAS BI Dashboard	375

Removing the SAS BI Dashboard Configuration **379**

PART 8 SAS Web OLAP Viewer Administration 381

Chapter 25 △ Configuring SAS Web OLAP Viewer for Java 383

Introduction to SAS Web OLAP Viewer for Java **383**

Main Tasks for Administering SAS Web OLAP Viewer for Java **383**

Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java **384**

Configure Logging for SAS Web OLAP Viewer for Java **384**

Redeploy SAS Web OLAP Viewer for Java **384**

Additional Documentation for SAS Web OLAP Viewer for Java **385**

Chapter 26 △ Customizing SAS Web OLAP Viewer for Java 387

Customizing SAS Web OLAP Viewer for Java **387**

Changes That Can Be Made to WebOLAPViewerConfig.xml **388**

PART 9 Appendixes 395

Appendix 1 △ Configuring the ESRI Map Component 397

About the ESRI Map Component **397**

Create an ESRI ArcGIS Server Definition **398**

Facilitate Authentication to the ESRI ArcGIS Server **398**

Creating Geographic Map Services **399**

Including Geographic Information in Cubes **401**

Appendix 2 △ Configuring the SAS Environment File 403

About the SAS Environment File **403**

Configuring the SAS Environment File **403**

Appendix 3 △ Recommended Reading 405

Recommended Reading **405**

Glossary 407

Index 421

What's New

Overview

The middle tier of the SAS Intelligence Platform provides an execution environment for business intelligence Web applications such as SAS Web Report Studio and SAS Information Delivery Portal.

The middle tier has the following changes and enhancements:

- common Web administration
- better integration with third-party security products
- SAS Content Server
- SAS Web Application Themes
- SAS Information Delivery Portal administration
- SAS Web Report Studio administration
- SAS BI Dashboard administration

The middle tier and its applications are supported on Windows and UNIX. JBoss, WebLogic, and WebSphere application servers are supported on Windows and UNIX. Beginning with the third maintenance release for SAS 9.2, the WebSphere application server is supported on the z/OS platform.

Common Web Administration

A common Web infrastructure provides consistent and cohesive Web applications that are well integrated. This infrastructure layer results in the following enhancements:

- A new SAS Web Infrastructure Platform provides basic services and applications that are used by all Web applications that run in the middle tier of the intelligence platform. The SAS Web Infrastructure Platform contains some of the technology that the SAS Web Infrastructure Kit contained in previous releases, such as the SAS Services Application and the SAS Stored Processes Web Application. However, the SAS Web Infrastructure Platform technologies are used not only by the SAS Information Delivery Portal, but also by other SAS Web applications. The SAS Web Infrastructure Kit no longer exists.

- You can customize SAS themes in a single place. The Web applications can then be configured to use the theme that you customize.
- All Web authentication occurs through a central authentication service. This change simplifies the task of configuring Web authentication.
- SAS Web Administration Console enables you to log on from a remote site and view information that is normally available in the SAS Management Console. You can view authenticated users and system users that are logged on to a SAS Web applications, as well as the current configuration of Web applications deployed at your site.
- In the second maintenance release for SAS 9.2 and later, the Restart Maintenance Wizard provides the following features:
 - Enables you to send e-mail to users to log off from their sessions within a specified deadline.
 - Logs off users after the specified deadline.
 - Prohibits new users from logging into their user applications.
- The second maintenance release for SAS 9.2 and later also includes the Quiesce System. The Quiesce System feature is useful when you want to enable existing users to stay logged on to their user sessions, but you want to quiesce the system by preventing new users from logging on to SAS Web applications. You can also access the SAS Content Server page to manage folders and permissions for content in the SAS Content Server.
- With the SAS Deployment Manager, you can either remove an existing configuration or rebuild SAS Web applications. Beginning with the second maintenance release for SAS 9.2 and later, when you rebuild Web applications by using the SAS Deployment Manager, the EAR files for the Web applications are automatically exploded and placed in two different directories.
- The SAS Configuration Manager, which is available in the SAS Management Console, enables you change settings or specify property names and values for several applications, including SAS Information Delivery Portal, SAS Web Report Studio, SAS BI Dashboard, and SAS Web OLAP Viewer for Java.
- Logging for all SAS applications is accomplished by using the Logging Service Configuration dialog box within the Configuration Manager in SAS Management Console.
- System folders are not accessible to SAS Web application users, and can be accessed only through SAS Management Console. For example, **System** and **Users** folders are both system folders, and are not visible from within SAS Web Report Studio 4.2.
- Beginning with the second maintenance release for SAS 9.2 and later, the **sas-environment.xml** file is used to define the available set of SAS environments for SAS client applications. You use this file to configure multiple environments by customizing and deploying this file to an HTTP server.
- Beginning with the third maintenance release for SAS 9.2, you can alert users by displaying a warning message before they are logged out of their inactive sessions. This feature is supported for SAS Web Report Studio, SAS Package Viewer, SAS Shared applications, SAS Preferences, SAS Web Infrastructure Platform administration, and SAS Stored Processes. However, if the SAS Information Delivery Portal 4.2 is configured at your site, this feature should not be enabled. In the future, SAS Information Delivery Portal 4.3 will support this feature.
- Beginning with the third maintenance release for SAS 9.2, the JVM command for the bind address is enabled by default to ensure that the Web application server and the JGroups software use the same bind address.

Better Integration with Third-Party Security Products

The following enhancements have been added:

- Web authentication is supported for JBoss, Oracle WebLogic, and IBM WebSphere application servers.
- The middle-tier software supports interaction with WebSeal and SiteMinder.
- Administrators can configure client certification for one-way and two-way SSL authentication.

SAS Content Server

The SAS Content Server is a content repository that stores digital content (such as documents, reports, and images) that is created and used by SAS client applications. The Web-based Distributed Authoring and Versioning (WebDAV) protocol is used to access the SAS Content Server. The SAS Web Administration Console enables you to access the SAS Content Server to view directories, change permissions to directories, and create and delete directories.

Beginning with the second maintenance release for SAS 9.2 and later, multiple folders in the SAS Content Server can be deleted concurrently within the SAS Web Administration Console.

Beginning with the third maintenance release for SAS 9.2, the SAS Content Server supports database persistence. By default, the SAS Content Server is configured to use the file system for persistence and need not be changed. In special cases, the SAS Content Server can be reconfigured to share and use the database that is used by SAS Shared Services. By default, SAS Shared Services uses the SAS Table Server, but it can be configured to use a different database such as Oracle, MySQL, PostgreSQL, DB/2, or SQL Server.

Beginning with the third maintenance release for SAS 9.2, the manual setting of JAVA_HOME variable to point to Java 5 is not required.

SAS Web Application Themes

SAS Web Application Themes contain definitions for themes that are used by several SAS Web applications. Themes enable you to create and apply consistent, visual customization and company branding that will be applied to all theme-enabled SAS Web Applications.

SAS Web applications such as SAS Web Report Studio, SAS Information Delivery Portal, and SAS BI Dashboard can be configured to facilitate a common look and feel across SAS applications. In SAS Web Report Studio, themes apply to the user interface, including the dialog boxes that are used to view, create, edit, and share reports. Themes contain images, HTML templates, and cascading style sheets (CSS).

SAS Information Delivery Portal Administration

The following enhancements pertain to SAS Information Delivery Portal administration:

- The portal uses the SAS Web Infrastructure Platform for authentication, security, and other common Web infrastructure services. The portal uses a common

framework that provides consistency among all SAS Web applications for configuration:

- message logging within Configuration Manager
- WebDAV content access
- logon and authentication methods
- themes and branding specifications
- For managing portal content, a group content administrator is recommended. The group content administrator can share personal content with the group, and can edit or remove content that has been shared with the group.
- STICKY pages are referred to as PERSISTENT pages.
- The Portal Admins group, which existed in SAS Information Delivery Portal 2.0, is not created in the current version. Instead, the Portal ACT is created and used to set permissions on the Permissions trees.
- The SAS Trusted User, who is also the portal administrator, is responsible for administering the portal, and is a member of the Portal ACT.
- The public kiosk does not exist anymore. The content administrator shares content with the PUBLIC group to ensure that all users have access to the content.
- By default, PUBLIC users have restricted access. In order to enable PUBLIC users to access content in the SAS Information Delivery Portal, you will need to enable permissions.
- In SAS 9.1.3, the portal created permission trees for identity groups that defined the roles. In SAS 9.2, the portal does not create the permission trees associated with the roles.

SAS BI Portlets Administration

The October 2009 release provides SAS BI Portlets that are based on JSR 168 and available in the SAS Enterprise BI Server offering. These portlets are seamlessly integrated into the SAS Information Delivery Portal. SAS BI Portlets are also compatible with the WebSphere Portal 6.1.0.

SAS Web Report Studio Administration

The following enhancements pertain to SAS Web Report Studio administration:

- In SAS Web Report Studio 4.2, you have the flexibility to choose the location of the SAS Web Report Studio user folders anywhere below the SAS Folders directory in the **Folders** tab window in SAS Management Console. Previously, in SAS Web Report Studio 3.1, you were required to use predefined storage folders. This is no longer the case.
- The **/BIP Tree/ReportStudio/Shared/Reports** folder path does not exist in SAS Web Report Studio 4.2. However, if your organization migrated from SAS 9.1.3 to SAS 9.2, the legacy path is automatically enabled in the Web Report Studio 4.2 Properties dialog box within SAS Management Console. As a result, the legacy folder paths are available to users on the SAS Web Report Studio **Location** drop-down menu.
- SAS Web Report Studio 4.2 offers three predefined roles with certain capabilities that are assigned to these roles initially. These predefined roles include Report Viewing, Report Creation, and Advanced. You are not required to use predefined

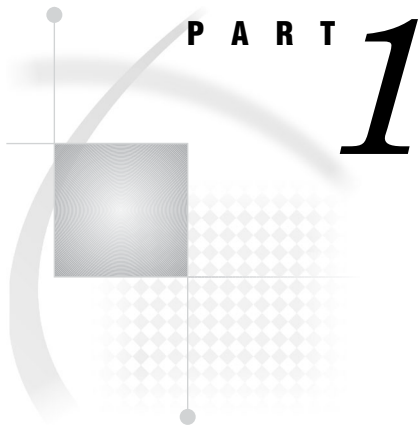
roles. You have the ability to create roles and capabilities that meet the needs of your organization. You can determine the number of roles to create, which features are available for each role, and control other aspects of role-based behavior.

- The Report Distribution Wizard has been significantly enhanced to enable you to create and edit recipient lists by specifying recipient names and e-mail addresses, and channel information within the wizard's dialog boxes.
- When users need to view a report, and are transferred by an external application such as the SAS Information Delivery Portal, those users are transferred to SAS Web Report Studio (if the application is installed). The functionality offered by the SAS Web Report Studio is determined by the role capability assigned to these users.
- You can set security measures to limit SAS Web Report Studio to interact only with information maps that are in designated locations. For example, you might limit the availability of all relational information maps because some of those information maps include row-level permissions.
- The **DefaultLoggerProperties.xml** file is not used. Logging for all applications is now accomplished by using the Logging Service Configuration dialog box within Configuration Manager in SAS Management Console.
- Previously, if client-side pooling was configured, SAS Web Report Studio 3.1 required that the pooling administrator's user name and password be stored in the metadata. This requirement has been waived in SAS Web Report Studio 4.2.
- In previous versions of SAS Web Report Studio, the **LocalProperties.xml** file offered the only practical method to override properties and their values. Although this file is available and supported in SAS Web Report Studio 4.2, it is recommended that you use the Configuration Manager in SAS Management Console to configure and set properties for SAS Web Report Studio. The Configuration Manager offers a consistent interface to set properties for all SAS applications.
- SAS Web Report Studio 4.2 enables you to add disclaimer text to graphs and tables.
- The report output generation tool enables you to create pre-generated, static versions of reports from the command line. This tool has been updated with the **rptbatch.bat** file, which calls upon the **outputgen.exe** file. In addition, new parameters have been added to the report output generation tool.
- Beginning with the third maintenance release for SAS 9.2, the log file, **SASBIReportServices4.2.log**, is created when you first run the report output generation tool with default permissions.
- In the third maintenance release for SAS 9.2, if you run the commands for the report output generation tool on a z/OS system with WebSphere, retrieve the fully qualified path name to the generated file. Then, locate the generated .in file or .jcl file, and copy the relevant contents into the command line text used to run the distribution job.
- The **WebReportStudioProperties.xml** file is no longer used in SAS Web Report Studio 4.2. Instead, the **Advanced** tab in Web Report Studio Properties 4.2 is used to specify property names and property values.
- Previous users of SAS Web Report Studio will find that their home folders have been moved into new SAS 9.2 home folders. These new folders have restricted metadata permissions.
- SAS Web Report Studio 4.2 maintains a working area that is hidden from users. This working area, which is located at **/System/Applications/SAS Web Report Studio/Web Report Studio 4.2**, is accessed by using the SAS Management Console. This location might store shared content such as images.

- Banner images are stored in the **/Web Report Studio 4.2/BannerImages** folder. Sample conditional highlighting image files are deployed in the **/Web Report Studio 4.2/ConditionalHighlightingImages** folder.
- In SAS Web Report Studio 4.2, PUBLIC users do not have personal home folders. In addition, PUBLIC users do not have a location to store their history and preferences. As a result, PUBLIC users' report history is not retained.
- By default, PUBLIC users have restricted access. In order to enable PUBLIC users to access SAS Web Report Studio, you will need to enable permissions.
- In the third maintenance release for SAS 9.2, you can change the location of the temporary workspace for SAS Web Report Studio and SAS Web Report Viewer.

SAS BI Dashboard Administration

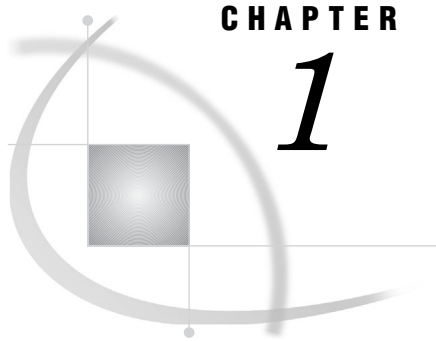
In SAS BI Dashboard 4.2, an alert can be set up to enable Event Generation Framework to regularly poll BI Dashboard indicators, determine whether an event qualifies for an alert, and generate an alert for the user. In order to minimize the impact of constant polling on BI Dashboard's performance, you can customize and set parameters for alert latency.



Getting Started

Chapter 1 **Before You Begin** 3

Chapter 2 **Working in the Middle-Tier Environment** 7



CHAPTER

1

Before You Begin

<i>Introduction to This Guide</i>	3
<i>Accessibility Features in the SAS Intelligence Platform Products</i>	3
<i>Prerequisites for Administering the Web Applications</i>	3
<i>What You Should Know</i>	4
<i>What You Should Do</i>	4
<i>High-Level Overview of Administrative Tasks</i>	4

Introduction to This Guide

This guide covers the administration of the SAS Web applications that run in the middle tier of the SAS Intelligence Platform.

The middle tier provides an execution environment for business intelligence Web applications such as SAS Web Report Studio and SAS Information Delivery Portal. These applications communicate with the user by sending data to and receiving data from the user's Web browser. Users in your organization work with the Web applications in order to query data, to generate reports, and to share and deliver information across the entire enterprise.

As an administrator, you can create a custom middle-tier environment for your users that meets your organization's security, availability, scalability, performance, and maintainability requirements. This guide provides post-installation instructions for carrying out the administrative tasks that you might need to perform.

This guide assumes that you are familiar with the concepts and terminology that are introduced in the *SAS Intelligence Platform: Overview* document. For a list of all of the documents that SAS publishes to support administration of the SAS Intelligence Platform, see <http://support.sas.com/92administration>.

Accessibility Features in the SAS Intelligence Platform Products

For information about accessibility for any of the products mentioned in this book, see the documentation for that product. If you have questions or concerns about the accessibility of SAS products, send e-mail to accessibility@sas.com.

Prerequisites for Administering the Web Applications

What You Should Know

Before you administer the Web applications, familiarize yourself with the following:

- basic concepts and components of the SAS Intelligence Platform, as described in the *SAS Intelligence Platform: Overview*.
- the SAS environment, as described in the *SAS Intelligence Platform: System Administration Guide*.
- the SAS applications servers. You should understand how the servers are started and which servers are required for different types of content.

For the start-up order for servers, see “Starting the Web Applications” on page 17. For a summary of the servers that are required for particular content, see the *SAS Intelligence Platform: Application Server Administration Guide*.

- security concepts, as described in the *SAS Intelligence Platform: Security Administration Guide*. You should understand authentication and authorization, and know how to manage access in the metadata layer. You should also know how to create and manage user and group definitions in metadata.
 - the middle-tier environment, as described in “Understanding the Middle-Tier Environment” on page 7.
 - basic procedures for using the applications that you plan to administer. For example, if you are responsible for administering SAS Web Report Studio, then you should know how to log on, navigate, and create reports in SAS Web Report Studio.
-

What You Should Do

The Web applications must be functional before they can be administered. Therefore, before you administer the Web applications, do the following:

- Perform a planned installation and initial configuration, as described in the *SAS Intelligence Platform: Installation and Configuration Guide*.
 - If you are upgrading from SAS 9.1.3 to SAS 9.2, then refer to the *SAS Intelligence Platform: 9.1.3 to 9.2 Migration Guide*.
 - If you are upgrading from SAS 9.2 to a maintenance release of SAS 9.2, see *Maintenance Planning for SAS 9.2*.
 - Your installation should include the standard, required SAS user accounts that are described in the *SAS Intelligence Platform: Installation and Configuration Guide*.
 - Verify that your Web applications operate correctly. You should be able to start the Web applications, log on, and perform basic tasks in those applications.
-

High-Level Overview of Administrative Tasks

After you have installed the middle-tier software, you can administer the Web applications in the middle tier. Some of the tasks you might perform include the following:

- Make resources and content items available to the Web applications.

For example, you can make fonts and graphics available to report creators who work in SAS Web Report Studio. If your deployment includes the SAS Information Delivery Portal, then you can add reports, files, links, and other items to the portal environment.
- Ensure that users see only the information that they are authorized to access.

In order to implement security, you register users in metadata, assign users to groups, and set up authorization for those groups. In this way, you can control access to all content.

Note: Most of the tasks related to user management and authorization are described in the *SAS Intelligence Platform: Security Administration Guide*. △

- Change the method of authentication.

Instead of using the SAS Metadata Server for authentication, you can use a Web application server (JBoss, IBM WebSphere, or Oracle WebLogic) to authenticate users. You can also implement single sign-on, so that users are not repeatedly prompted for their user IDs when they access different Web applications.

For a detailed discussion of different types of authentication and configuration guidelines, see “Authentication Mechanisms” in the *SAS Intelligence Platform: Security Administration Guide*. For information about configuring Web authentication for JBoss, IBM WebSphere, or Oracle WebLogic, go to <http://support.sas.com/resources/thirdpartysupport/v92/>.

- Customize the environment for your users.

The Web applications enable you to customize the interface in different ways:

- SAS Web Report Studio enables you to customize reports for your organization.
- You can customize the display for SAS Web OLAP Viewer for Java.
- The SAS Information Delivery Portal enables you to create different views for different types of users. In addition, your developers can create the content, custom portlets, logos, company colors, and page themes that best suit your organization.

- Optimize performance.

One way to improve performance is to set up workspace server pooling, as described in the *SAS Intelligence Platform: Application Server Administration Guide*. You can also make configuration changes that are specified in Chapter 3, “Best Practices for Configuring Your Middle Tier,” on page 23.

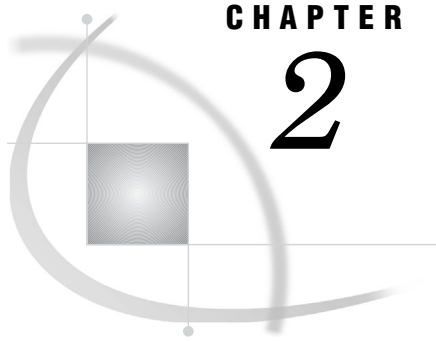
In addition, the Web applications have their own specific tasks:

“Main Tasks for Administering the Portal” on page 270

“Main Tasks for Administering SAS Web Report Studio” on page 180

“Main Tasks for Administering SAS BI Dashboard” on page 364

“Main Tasks for Administering SAS Web OLAP Viewer for Java” on page 383



CHAPTER

2

Working in the Middle-Tier Environment

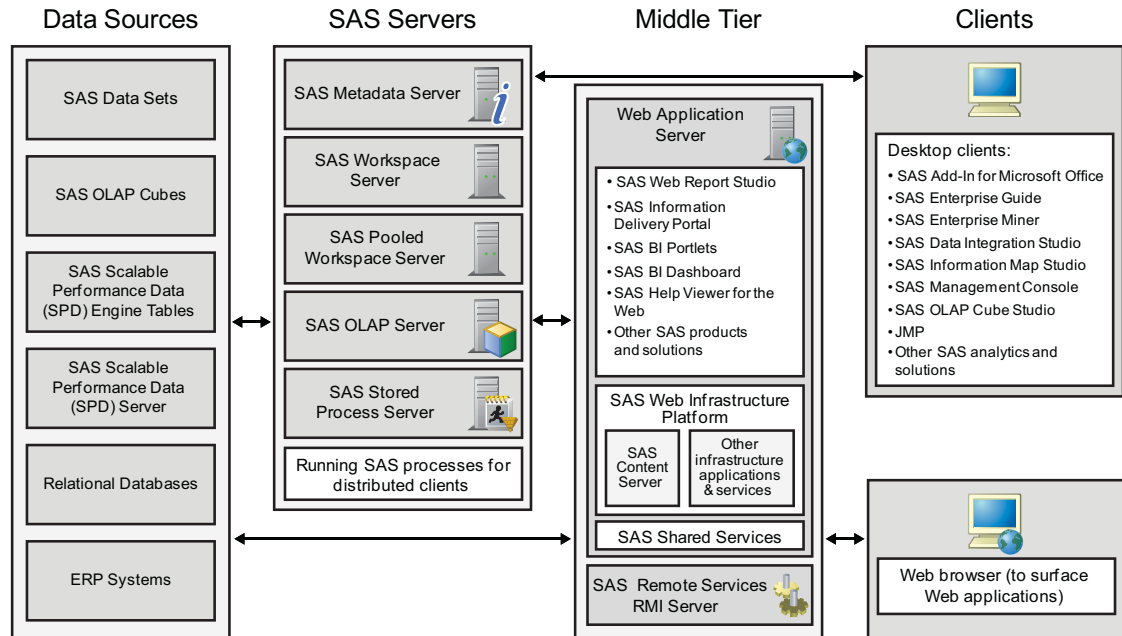
<i>Understanding the Middle-Tier Environment</i>	7
<i>Third-Party Software Components</i>	9
<i>Web Application Server</i>	9
<i>Java Development Kit</i>	10
<i>SAS Web Infrastructure Platform</i>	10
<i>SAS Foundation Services</i>	12
<i>SAS Web Infrastructure Platform Services</i>	13
<i>SAS Content Server</i>	13
<i>SAS Shared Services</i>	14
<i>SAS Foundation Services</i>	14
<i>SAS Web Applications</i>	14
<i>SAS Web Report Studio</i>	15
<i>SAS Web OLAP Viewer for Java</i>	15
<i>SAS Information Delivery Portal</i>	16
<i>SAS BI Dashboard</i>	16
<i>SAS Documentation for the Web</i>	16
<i>SAS BI Portlets</i>	17
<i>Starting the Web Applications</i>	17
<i>Main Steps for Starting the Web Applications</i>	17
<i>Required Servers</i>	17
<i>Starting Servers in the Correct Order</i>	18
<i>Deploying EAR Files in the Correct Order</i>	19

Understanding the Middle-Tier Environment

The middle tier of the SAS Intelligence Platform enables users to access intelligence data and functionality via 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 in which the business intelligence Web applications, such as SAS Web Report Studio and the SAS Information Delivery Portal, can execute. These products run in a Web application server and communicate with the user by sending data to and receiving data from the user's Web browser. The middle tier 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 other tiers of the SAS Intelligence Platform. For a description of these components, see *SAS 9.2 Intelligence Platform: Overview*.

Display 2.1 Architecture of the SAS Intelligence Platform

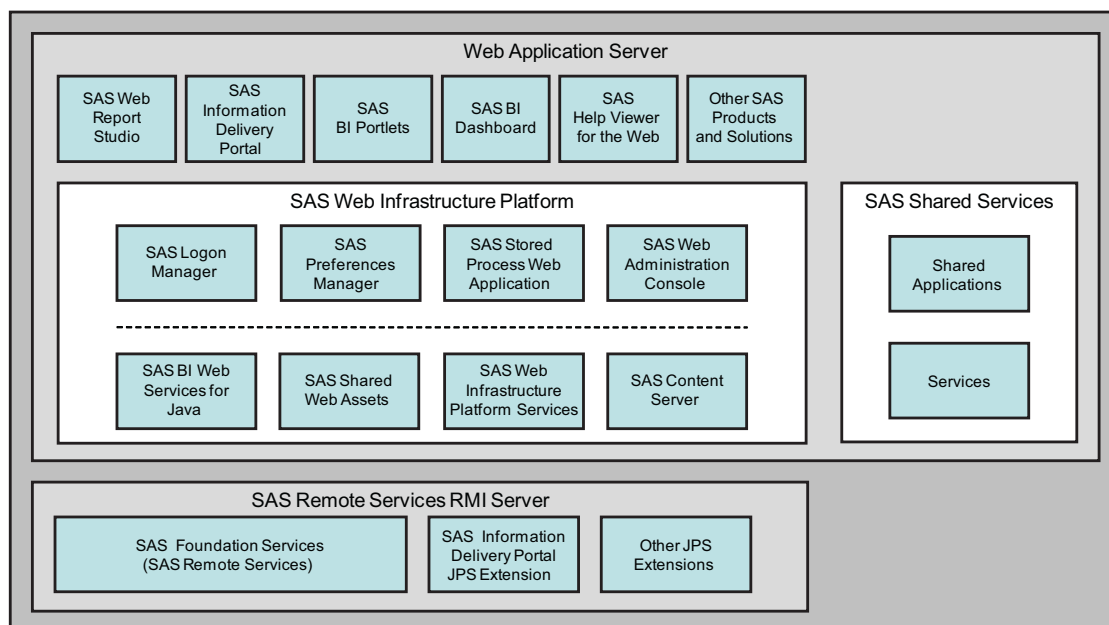
The middle tier includes the following third-party software and SAS software elements:

- a Web application server
- a Java Development Kit
- SAS Web applications, which can include SAS Web Report Studio, the SAS Information Delivery Portal, the SAS BI Dashboard, the SAS OLAP Viewer for Java, SAS Help Viewer for the Web, and other SAS products and solutions
- the SAS Web Infrastructure Platform, which includes the SAS Content Server and other infrastructure applications and services
- SAS Shared Services
- a Java remote method invocation (RMI) server, which enables access to SAS Foundation Services and associated extension services

The SAS Intelligence Platform architecture gives you 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. In addition, the components of the different tiers, such as Web applications that run in a Web application server, might be distributed on separate machines.

SAS software components in the middle-tier include the SAS Foundation Services and the SAS Web Infrastructure Platform.

The following figure illustrates the middle-tier components:

Display 2.2 Middle-Tier Components

Third-Party Software Components

Web Application Server

The Web application server provides the execution environment for Web applications that run on the middle tier. The following third-party servers are supported:

- JBoss Application Server
- IBM WebSphere Application Server
- Oracle WebLogic Server

For information about the currently supported versions of these products, see the SAS third-party Web page at <http://support.sas.com/resources/thirdpartysupport/v92/>. The Web applications in the middle-tier are supported on Windows and UNIX operating systems. Beginning with the third maintenance release for SAS 9.2, Web applications in the middle-tier are supported on the z/OS operating system.

The following applications and services run in the Web application server environment:

- applications and services that are part of the SAS Web Infrastructure Platform
- SAS Shared Services
- the SAS Web Report Studio, SAS Web OLAP Viewer for Java, 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 Development Kit

If you are using JBoss or WebLogic application server, a Java Development Kit (JDK) must be installed. (WebSphere application server ships with its own version of the JDK which is installed when you install WebSphere software). For information about the currently supported versions of the JDK, see the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.

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

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

Table 2.1 Services and Applications in the SAS Web Infrastructure Platform

Application or Service	Features
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. The most common protocol is the Simple Object Access Protocol (SOAP). The SAS BI Web Services for Java interface is based on the XML For Analysis (XMLA) Version 1.1 specification.
SAS Shared Web Assets	Contains graph applet JARs that are shared across SAS Web applications. They display graphs in stored processes and in the SAS Stored Process Web application.
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, e-mail, theme management, and data sharing across SAS Web applications.

Application or Service	Features
SAS Logon Manager	<p>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.</p> <p>You can configure SAS Logon Manager to display custom messages and to specify whether a logon dialog box is displayed when users log off.</p> <p>In addition, you can use third-party products in conjunction with SAS Logon Manager to enable users to access multiple Web applications within the same browser session.</p>
SAS Preferences Manager	<p>Provides a common mechanism for managing preferences for SAS Web applications. The feature enables administrators to set default preferences for locale, theme, alert notification, and time, date, and currency display. Within each Web application, users can view the default settings and update their individual preferences.</p>
SAS Stored Process Web Application	<p>Executes stored processes on behalf of a Web client and returns results to a Web browser. The SAS Stored Process Web application is similar to the SAS/IntrNet Application Broker, and has the same general 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:</p> <ol style="list-style-type: none"> 1 Users enter information in an HTML form using their Web browser and then submit it. The information is passed to the 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 that information. 5 The results of the SAS program are sent back through the Web application and Web server to the Web browser and the awaiting users.

Application or Service	Features
SAS Web Administration Console	<p>Provides features for monitoring and administering middle-tier components. This browser-based interface enables administrators to do the following:</p> <ul style="list-style-type: none"> □ Monitor authenticated users and system users who are logged into SAS Web applications, and send e-mail to authenticated users. □ Before system maintenance, use the Restart Maintenance Wizard to send e-mail to users to log off from their sessions within a specified deadline, log off users after the deadline, and prohibit new users from logging into their applications. □ Before minor system maintenance, use the Quiesce System feature to allow existing users to stay logged into their user sessions, and quiesce the system by preventing new users from logging into SAS Web applications. □ Create, delete, and manage permissions for multiple folders on the SAS Content Server □ View configuration information for each middle-tier component.
SAS Content Server	Stores digital content (such as documents, reports, and images) that is created and used by the SAS Web applications.

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

In the Web application server, two services are available to all SAS Web applications:

- SAS Foundation Services
- SAS Web Infrastructure Platform Services

SAS Foundation Services

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

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

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

and configured the Web applications, the platform services will be 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. The addition of SAS Shared Services extends the functionality.
- 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.
- Themes provide access to theme definitions for presentation assets used in Web applications.
- Registry provides access to services for desktop clients; a client needs to know only a single endpoint to determine other required locations.

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-based 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 Shared Services

SAS Shared Services provides standard features that are used by the SAS BI Dashboard and which can also be used by the SAS Information Delivery Portal, SAS Web Report Studio, and certain SAS solutions. The features include the following:

alert registration and notification	enables users to register to receive time-sensitive, action-oriented messages when a specified combination of events and conditions occurs. Alerts empower users by allowing them to control the type of notifications that they receive and when the notifications are delivered. Alerts can be sent to the user's e-mail address or displayed in the SAS Information Delivery Portal.
comment management	enables users to create comments related to business intelligence objects. Users can then reply to, search for, retire, delete, or add attachments to existing comments. In addition, users can find comments that were created by a particular user, with a specified date range, or with specific text. This feature enables the capture of human intelligence and supports collaborative decision making related to business data.

SAS Foundation Services

SAS Foundation Services is a set of core middleware infrastructure services that integrate distributed applications on the middle tier with other components of the SAS Intelligence Platform. This suite of Java APIs provides the following services:

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

Extension services for information publishing, event management, and SAS Stored Process execution are also provided. All of the SAS Web applications that are described in this document use SAS Foundation Services. For more information about SAS Foundation Services, see the *SAS Foundation Services: Administrator's Guide*.

SAS Web Applications

SAS Web applications reside and execute on the middle tier. These applications require a Web browser on each client machine and a Web application server on the middle-tier machine where the application will run. These applications communicate with the user by sending data to and receiving data from the user's Web browser. For example, an application of this type displays its user interface by sending an HTML

document to the user's browser. The user can submit input to the application by sending it an HTTP response, usually by clicking a link or submitting an HTML form.

SAS Web Report Studio

SAS Web Report Studio 4.2 enables you to view, interact, create, and distribute both public and private reports. Users can interactively get the information that they need without having to understand a programming language. In addition, SAS predictive analytical results can be used by business professionals across the enterprise via their Web browsers.

You can use the SAS Web Report Studio for the following tasks:

- Creating reports. Beginning with a simple intuitive view of your data provided by SAS Information Maps (created in SAS Information Map Studio), you can create reports based on either relational or multidimensional data sources. You can use the Report Wizard to quickly create simple reports or the Edit Report view to create sophisticated reports that have multiple data sources, each of which can be filtered. These reports can include various combinations of list tables, crosstabulation tables, graphs, images, and text. Using the Edit Report view, you can adjust the style to globally change colors and fonts. You can also insert stored processes that take the results from a block of SAS code and embed those results directly into a report.
- Viewing and working with reports. While viewing reports using a Web browser, you can filter, sort, and rank the data that is shown in list tables, crosstabulation tables, and graphs. With multidimensional data, you can drill down on data in crosstabulation tables and graphs, and drill through to the underlying data.
- Organizing reports. You can create folders and subfolders for organizing your reports. Information consumers can use keywords to find the reports that they need. Reports can be shared with others or kept private. You can schedule reports to refresh on a recurring basis. Specific report pages can be distributed via e-mail or subscription channels.
- Printing and exporting reports. You can preview a report in PDF and print the report, or save and e-mail it later. You have control over many printing options, including page orientation, page range, and size of the tables and graphs. You can also export data as a spreadsheet and export graphs as images. You can also export data for list tables, crosstabulation tables, and graphs. The output can be viewed in a Microsoft Excel spreadsheet.

SAS Web Report Studio runs within the Web application server, and 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 Web Report Studio can be invoked from the SAS Information Delivery Portal.

SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java is a Web-based application for viewing SAS OLAP data. It provides an easy-to-use interface from which you can select a data source, view the data, and customize your view with features such as sorting and filtering. SAS Web OLAP Viewer for Java can be run separately, or it can be launched from the SAS Information Delivery Portal

The SAS Web OLAP Viewer enables you to perform the following tasks:

- subset your data by drilling down and filtering

- save and restore bookmarked views of your data
- search for data values
- calculate new data items
- view ESRI maps
- export your data view as a SAS report, Microsoft Excel file, Microsoft document, or PDF document.

Note: You cannot use the SAS Web OLAP Viewer to make changes to information maps or to physical data. △

For more information, see the SAS Web OLAP Viewer Help, which is available from within the product.

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 publish content to SAS publication channels or WebDAV repositories, to subscribe to publication channels, and to 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

The SAS BI Dashboard enables users to create, maintain, and view dashboards to monitor key performance indicators that convey how well an organization is performing. The application is Web-based and can be accessed from within the SAS Information Delivery Portal.

The SAS BI Dashboard includes an easy-to-use interface for creating dashboards that include graphics, text, colors, and hyperlinks. Dashboards can link to SAS reports and analytical results, SAS Strategic Performance Management scorecards and objects, externally generated data, and virtually anything that is addressable by a Uniform Resource Identifier (URI).

All content is displayed in a role-based, secure, customizable, and extensible environment. Users can customize how information appears on their personal dashboards.

For more information, see the SAS BI Dashboard Help, which is available from within the product, and the *SAS BI Dashboard: User's Guide*, available at <http://support.sas.com>.

SAS Documentation for the Web

Your installation can include the following documentation software:

- SAS Help Viewer for the Web is an application that enables users to view and navigate SAS online Help in the various SAS Web applications.

Note: SAS Help Viewer Metadata Configuration is another name that is associated with this application. Whereas SAS Help Viewer for the Web is the name for the delivery system software, SAS Help Viewer Metadata Configuration is the name for the configured component. This component combines SAS Help Viewer for the Web software with various Help content and creates an EAR file that can be deployed on a Web application server. △

- SAS OnlineDoc for the Web is an application that contains an online library of reference documentation for the SAS System.

Note: You must manually deploy SAS OnlineDoc for the Web in your Web application server. Manual deployment is required even if you selected the automatic deployment option in SAS Deployment Wizard. For more information, see “Deploying SAS OnlineDoc Manually for the Web” on page 102. △

You can install SAS Help Viewer for the Web without installing SAS OnlineDoc for the Web. However, the reverse is not true. In order to install and configure SAS OnlineDoc for the Web, in the SAS Deployment Wizard you must choose to install both SAS Help Viewer for the Web and SAS OnlineDoc for the Web.

SAS BI Portlets

The October 2009 Release provides SAS BI Portlets that are based on JSR 168 and available in the SAS Enterprise BI Server offering. These portlets are seamlessly integrated into the SAS Information Delivery Portal that runs on JBoss, WebLogic, or WebSphere Web application servers. 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.

Starting the Web Applications

Main Steps for Starting the Web Applications

To start the Web applications, follow these steps:

- 1 Start the necessary servers and services in the correct order. For the correct start-up order, see “Starting Servers in the Correct Order” on page 18.
- 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 with the Web application server that you are using and the configuration that you have defined for your environment.
- 3 Log on to the Web application. For instructions about logging on to a Web application, see the online Help that is provided with the application.

Required Servers

In order for clients to access the SAS Intelligence Platform, the following components must be running on network-accessible machines:

- SAS Metadata Server
- SAS Object Spawner, which acts as a listener for SAS Workspace Server, SAS Pooled Workspace Servers, and SAS Stored Process Servers

Depending on which SAS products you have installed, one or more instances of the following additional components might also be required to be running on network-accessible machines:

- SAS OLAP Server
- SAS Table Server
- SAS Services Application
- SAS/CONNECT Spawner
- SAS/SHARE Server
- SAS Content Server
- Web Application Server (JBoss, WebLogic, or WebSphere)

For example, the following table documents some SAS servers that can be involved when a user accesses a stored process or a report.

Table 2.2 Examples of Server Interactions for Stored Processes and Reports

Object Type	SAS Servers
Stored process	<ul style="list-style-type: none"> □ Metadata server □ Workspace server (for package results) or stored process server (for streaming results)¹
Report	<ul style="list-style-type: none"> □ Metadata server □ Workspace server¹ (for relational data) □ OLAP server (for multidimensional data) □ SAS Content Server (for retrieval from WebDAV)

¹ Workspace servers and stored process servers have a dependency on the object spawner.

Starting Servers in the Correct Order

Because of dependencies, it is important to start the servers in the correct order. Server dependencies are explained in the *SAS Intelligence Platform: System Administration Guide*. The following is an example of a startup order that meets the dependency requirements for servers:

- 1 Start the SAS Metadata Server
- 2 Start the SAS OLAP Server
- 3 Start SAS object spawner
- 4 Start the SAS/SHARE server
- 5 Start the SAS/CONNECT server
- 6 Start the SAS Table Server
- 7 Start the SAS Services Application (Remote Services)
- 8 Start the Web application server. (The SAS Content Server starts automatically when the application server is started).

The SAS OLAP Server, SAS/CONNECT Spawner, and SAS/SHARE Server can be started anytime after the SAS Metadata Server has been started.

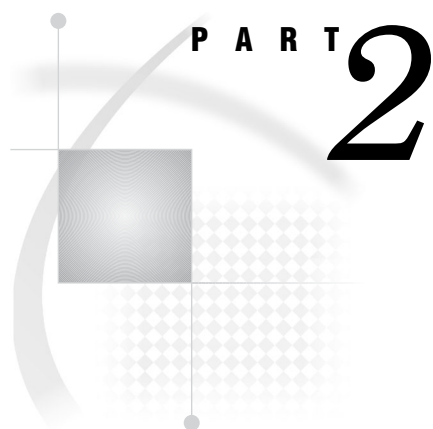
Servers should be stopped in the inverse order, with the metadata server stopped last.

Deploying EAR Files in the Correct Order

The SAS Deployment Wizard deploys SAS Web applications to the Web application server. However, you can also deploy EAR files manually from the Web application server. EAR files reside in the *SAS-configuration-directory\Lev1\Web\Staging* directory.

There is no required start-up order for deploying the EAR files to JBoss or WebLogic. Although, you can deploy the EAR files in any order of your choice, it is highly recommended that you follow this sequence for WebSphere.

- 1 SAS Web Application Themes (**sas.themes.ear**)
- 2 SAS Web Infrastructure Platform Services (**sas.wip.services9.2.ear**)
- 3 SAS Web Infrastructure Platform Applications (**sas.wip.apps9.2.ear**)
- 4 SAS Content Server (**sas.wip.scs9.2.ear**)
- 5 SAS Stored Process Application (**sas.wip.stp9.2.ear**)
- 6 SAS Information Delivery Portal 4.2 (**sas.portal4.2.ear**)
- 7 SAS Package Viewer 4.2 (**sas.packageviewer4.2.ear**)
- 8 SAS Web Report Studio 4.2
- 9 SAS Shared Services Application (**sas.shared9.2.ear**)
- 10 SAS BI Dashboard 4.2 (**sas.bidashboard4.2.ear**)
- 11 SAS BI Portlets (**sas.biportlets4.2.ear**)
- 12 SAS Help Viewer Metadata Configuration (**webdocmd9.2.ear**)
- 13 SAS Web OLAP Viewer for Java (**sas.webolapviewer4.2.ear**)

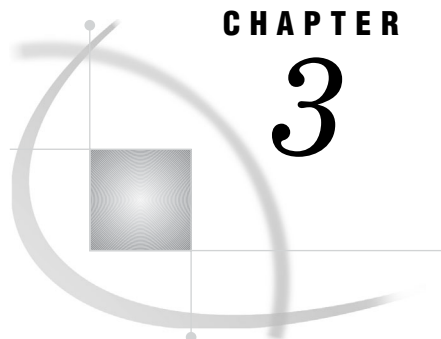


Middle-Tier Environment and Topologies

Chapter 3.....**Best Practices for Configuring Your Middle Tier** 23

Chapter 4.....**Middle-Tier Security** 39

Chapter 5.....**Interacting with the Server Tier** 53



CHAPTER

3

Best Practices for Configuring Your Middle Tier

<i>Best Practices for Middle-Tier Configuration</i>	23
<i>Sample Middle-Tier Deployment Scenarios</i>	24
<i>Overview of Middle-Tier Deployment Scenarios</i>	24
<i>Scenario 1: Web Applications Deployed in a Single Web Application Server</i>	24
<i>Further Considerations for Scenario 1</i>	26
<i>Scenario 2: Static Content Deployed in a Reverse Proxy</i>	27
<i>Scenario 3: Web Applications Deployed across a Web Application Server Cluster</i>	29
<i>Understanding Clusters</i>	30
<i>Requirement for Session Affinity</i>	31
<i>Understanding Demilitarized Zones</i>	31
<i>Additional Considerations for a Deployment</i>	32
<i>Load-Balancing Software and Hardware for the HTTP Servers</i>	32
<i>Secure Sockets Layer</i>	32
<i>Web Authentication</i>	33
<i>Heap Size for SAS Remote Services Application</i>	33
<i>Tuning the Web Application Server</i>	34
<i>Configuring a Cluster of Web Application Servers</i>	34
<i>Configuring HTTP Sessions in Environments With Proxy Configurations</i>	34
<i>Using an HTTP Server to Serve Static Content</i>	35
<i>Overview of Using an HTTP Server to Serve SAS Themes Web Application Static Content</i>	35
<i>Example: Use Apache HTTP Server to Serve SAS Themes Web Application Static Content</i>	36
<i>Using a Proxy Plug-in between the Web Application Server and the HTTP Server</i>	36
<i>Using Apache Cache Control for Static Content</i>	37

Best Practices for Middle-Tier Configuration

This chapter provides sample middle-tier topologies and guidelines for achieving better efficiency and performance with the middle-tier components in the SAS Intelligence Platform. The middle tier provides an environment for running the following SAS Web clients:

- ❑ SAS Information Delivery Portal
- ❑ SAS Web Report Studio
- ❑ SAS BI Dashboard
- ❑ SAS Web OLAP Viewer for Java

Configuration instructions vary depending on the Web application server installed at your site. For configuration instructions that pertain to the topics discussed in this chapter, see the following third-party vendor Web sites:

- ❑ JBoss Application Server: <http://www.jboss.org/docs>

- IBM WebSphere Application Server: <http://www.ibm.com/support/documentation/us/en>
- Oracle WebLogic Server: <http://www.oracle.com/technology/documentation/index.html>

Sample Middle-Tier Deployment Scenarios

Overview of Middle-Tier Deployment Scenarios

This section describes sample topologies for the middle-tier components. These sample topologies can help you design a middle-tier configuration that meets the needs of your organization with regard to performance, security, maintenance, and other factors.

As with all tiers in the SAS Intelligence Platform, deployment of the middle tier involves careful planning. When you design and plan the middle tier, you must balance performance requirements against a number of other criteria. To understand these criteria and to evaluate sample deployment scenarios, see the following subsections:

- “Scenario 1: Web Applications Deployed in a Single Web Application Server” on page 24
- “Scenario 2: Static Content Deployed in a Reverse Proxy” on page 27
- “Scenario 3: Web Applications Deployed across a Web Application Server Cluster” on page 29
- “Additional Considerations for a Deployment” on page 32

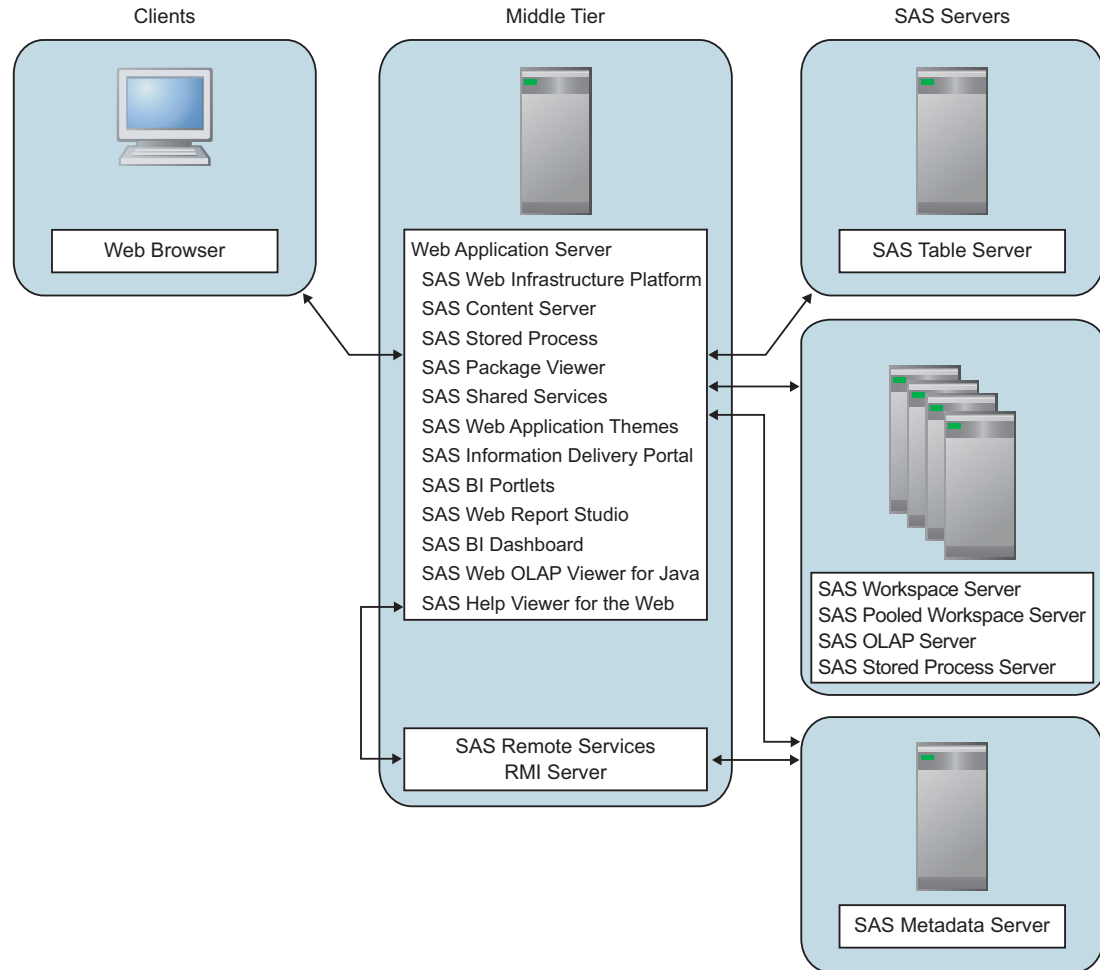
The topologies that are presented here range from simple to complex. Scenario 1 represents the deployment that results from using the SAS Deployment Wizard to configure the Web application server and deploy the SAS Web applications. Scenarios 2 and 3 provide advanced features, such as greater security and efficiency, but 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

Scenario 1 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 Web application server. The SAS Remote Services application is also installed on the same middle-tier server, but runs as a server application outside the Web application server.

The following figure illustrates the topology for Scenario 1.

Figure 3.1 Scenario 1: Middle Tier on a Single System

After installation, the system contains the following middle-tier software:

- ❑ Web application server (WebLogic Server, WebSphere Application Server, or JBoss)
- ❑ the following SAS Web applications, which run in the Web application server:
 - ❑ SAS Web Infrastructure Platform
 - ❑ SAS Content Server
 - ❑ SAS Stored Process
 - ❑ SAS Package Viewer
 - ❑ SAS Shared Services
 - ❑ SAS Web Application Themes
 - ❑ SAS Information Delivery Portal
 - ❑ SAS BI Portlets (available in the October 2009 Release)
 - ❑ SAS Web Report Studio
 - ❑ SAS BI Dashboard
 - ❑ SAS Help Viewer for the Web
 - ❑ SAS Web OLAP Viewer for Java
- ❑ the SAS Remote Services application, which runs in a separate Java Virtual Machine process

Here are the advantages and disadvantages of this topology:

Table 3.1 Scenario 1 Advantages and Disadvantages

Topic	Advantages	Disadvantages
Security	None	The SAS Web applications are exposed to attacks from Web clients. If SSL is enabled, the middle-tier server has the computational load of encrypting data, in addition to the load of hosting the SAS Web applications.
Performance	None	In 32-bit computing environments, the Java virtual machine might reach memory addressing limits.
Scalability	None	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. SAS provides another topology that can be completely installed and configured by the SAS Deployment Wizard, yet provides better scalability and performance.

In 32-bit computing environments, the scenario 1 topology can reach a performance limit due to the memory constraints of 32-bit addressing and the fact that all of the SAS Web applications are running in a single Java virtual machine that is provided by the Web application server. A variation of this scenario is to use the SAS Deployment Wizard to distribute the SAS Web applications across two Web application server instances (managed servers) on the same middle-tier server. This distribution of Web applications is different from clustering in that there is still only one instance of each application. By distributing the applications to two managed servers, this alternative configuration allows 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 and some Web application servers that use 32-bit Java environments 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 the custom prompting level and selecting this feature.

Scenario 2: Static Content Deployed in a Reverse Proxy

This sample topology delivers static HTML content to clients from an HTTP server that is configured as a reverse proxy. This strategy reduces the work load on the Web application server. Examples of HTTP servers that can be configured as reverse proxies are Apache HTTP Server and Microsoft Internet Information Services (IIS).

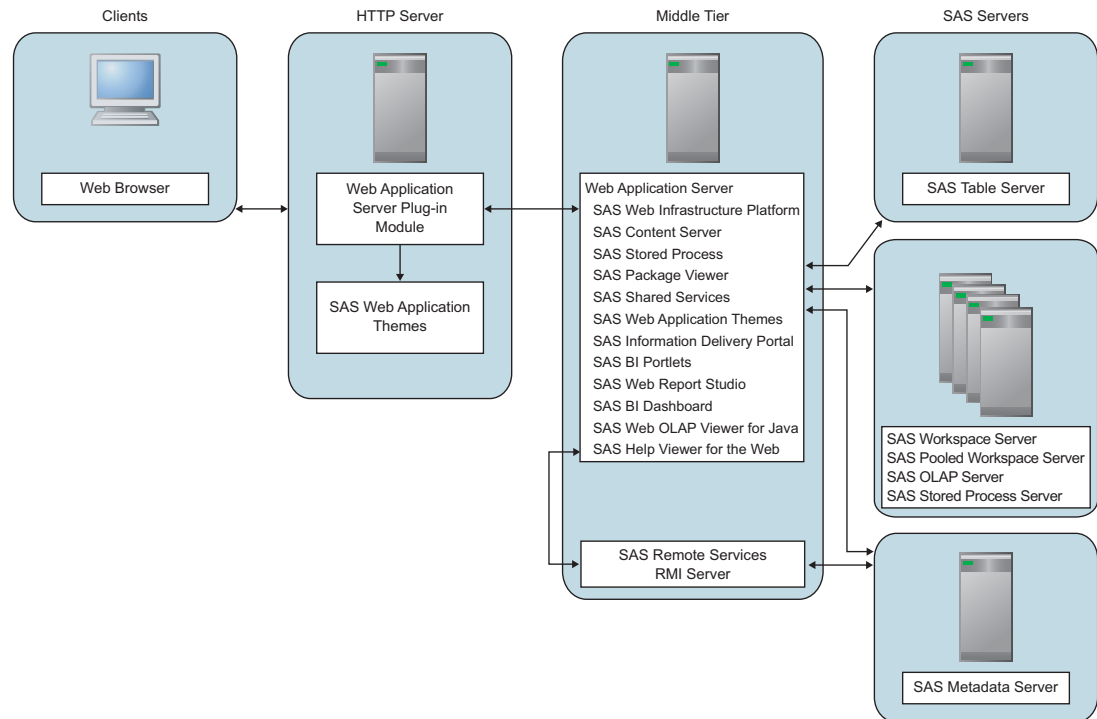
When a browser makes a request for a SAS Web application, a part of the request is for static content such as HTML files, images, cascading style sheets, and JavaScript scripts. The SAS Themes Web application provides this static content. In this scenario, the SAS Themes content is unpacked and delivered by the reverse proxy. The reverse proxy simply returns the requested content to the browser, and the browser displays the document.

Note: If you unpack and deploy the SAS Themes static content on the reverse proxy, then you must redeploy this content if you later install a SAS software upgrade or apply maintenance that includes new files for the static content. △

If the reverse proxy can be configured to cache content, then the performance improvement is even greater. The portion of the request that is for dynamic content still requires some type of data manipulation by the SAS Web applications and the Web application server must perform that work before returning the requested page.

The following figure illustrates the topology for scenario 2.

Figure 3.2 Scenario 2: Using a Reverse Proxy



In a typical configuration, the HTTP server is configured with a module or plug-in that enables the reverse proxy function of communicating with the Web application server. By having the reverse proxy as the single point of contact for browser requests, the Web application server is not directly exposed to clients. The reverse proxy provides a layer of security for the SAS Web applications.

Although this topology must be manually configured and maintained, here are the advantages and disadvantages of this topology:

Table 3.2 Scenario 2 Advantages and Disadvantages

Topic	Advantages	Disadvantages
Security	<p>The reverse proxy provides a layer of security.</p> <p>The network on the middle-tier server can be configured to reject HTTP packets that do not originate from the reverse proxy.</p> <p>SSL can be enabled on the client side of the reverse proxy without affecting the work load on the Web application server or the performance of the SAS Web applications.</p> <p>The Web application server and SAS Web applications can be configured to perform Web authentication for single sign-on to SAS Web applications and other Web resources in the network.</p>	<p>Adding firewalls to the network is a good next step.</p>
Performance	<p>Response time is improved because processing static content is offloaded from the Web application server to the reverse proxy.</p>	<p>As with scenario 1, in 32-bit computing environments, the Java virtual machine might reach memory addressing limits. However, a second managed server instance can be configured, as mentioned in the scenario 1 section.</p>
Scalability	<p>There are no advantages in this scenario, but the topology provides an upward path to clustering Web application servers.</p>	<p>This topology does not support hundreds of concurrent users.</p>
Availability	<p>None</p>	<p>This topology has no provision for planned or unplanned down time.</p>
Maintainability	<p>The SAS Deployment Wizard can still automate the configuration and deployment of the Web application server and the SAS Web applications.</p>	<p>After manual or automatic installation and configuration with the SAS Deployment Wizard, there are manual steps to perform.</p> <p>The reverse proxy must be configured with the connection information for the SAS Web applications.</p>

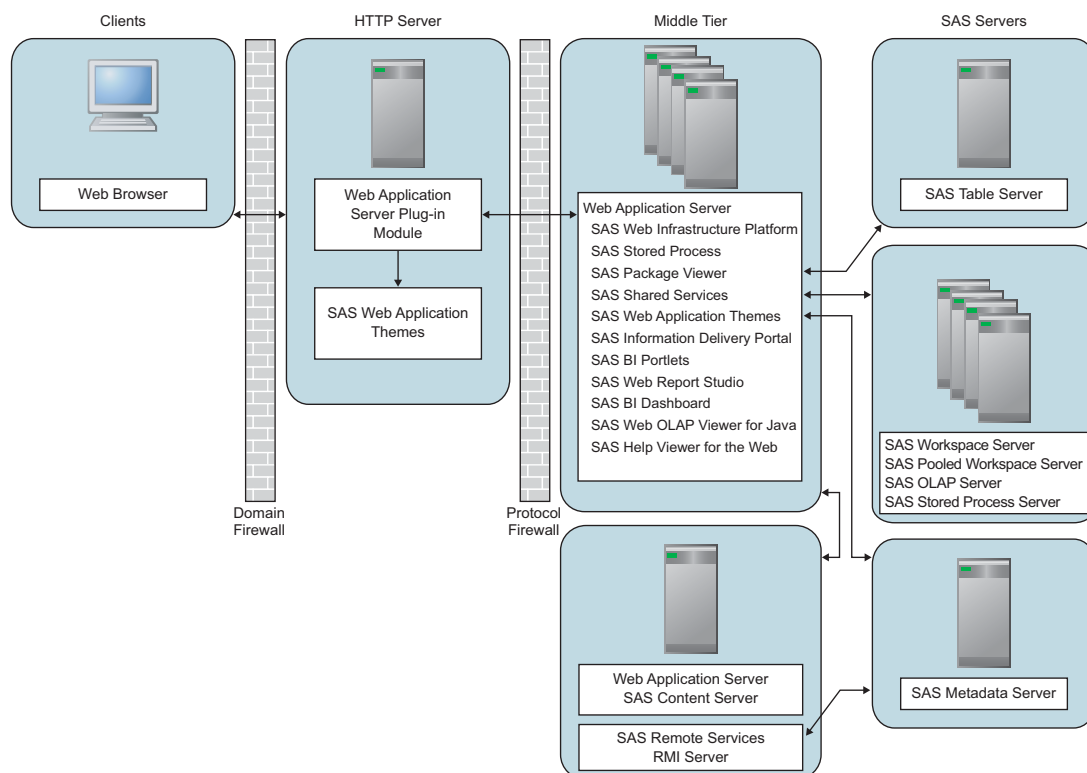
For instructions about how to configure an HTTP server as a reverse proxy for SAS Web applications deployed on JBoss, WebSphere Application Server, or WebLogic Server, see the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.

Scenario 3: Web Applications Deployed across a Web Application Server Cluster

The sample topology in scenario 3 includes a cluster of Web application servers in a network that implements a secure demilitarized zone (DMZ).

The following figure illustrates the topology for scenario 3. Note that the Web application servers and SAS Web applications are distributed across multiple middle-tier machines.

Figure 3.3 Scenario 3: Clustered Web Application Servers and a Demilitarized Zone



Note: As indicated in the figure, if you configure a cluster of Web application servers, then you must deploy all the SAS Web applications to each node in the cluster. Each node must be configured identically. △

In the figure, note that the SAS Remote Services Application and SAS Content Server Web application reside on a host that is separate from the cluster of Web application servers. This separation serves to illustrate that the SAS Remote Services Application is a server application that does not participate in clustering. The SAS Remote Services Application could just as well reside on any one of the hosts in the cluster. The separation in the figure also shows that the SAS Content Server Web application is a Web application that is deployed on a Web application server.

In this scenario, the SAS Content Server is not in a cluster.

Although this topology requires manual configuration and greater maintenance than the topologies in the previous scenarios, here are the advantages and disadvantages of this topology:

Table 3.3 Scenario 3 Advantages and Disadvantages

Topic	Advantages	Disadvantages
Security	<p>The SAS Web applications and the Web application server cluster are protected by the DMZ.</p> <p>The Web application server and SAS Web applications can be configured to perform Web authentication for single sign-on to SAS Web applications and other Web resources in the network.</p>	None
Performance	Response time is improved because processing static content performed by the reverse proxy and because of the greater computing capacity of the Web application server cluster.	None
Scalability	Once the cluster of Web application servers is established, additional managed servers can be added to the cluster to support larger numbers of concurrent users.	None
Availability	<p>Clustering provides fault isolation that is not possible with a single Web application server. If a node in the cluster fails, then only the users with active sessions on that node are affected.</p> <p>You can plan downtime for maintenance by taking managed servers offline. New requests are then directed to the SAS Web applications deployed on the remaining nodes while maintenance is performed.</p>	None
Maintainability	Configuration and deployment of the first Web application server and the SAS Web applications can still be automated with the SAS Deployment Wizard. This first Web application server can be cloned to speed the creation of the cluster.	<p>The reverse proxy must be configured with the connection information for the SAS Web applications.</p> <p>Creating the Web application server cluster requires additional configuration.</p>

Understanding Clusters

In order to provide greater scalability, availability, and robustness, WebLogic Server, WebSphere Application Server, and JBoss support some form of clustering. With clustering, multiple Web application server instances participate in a load-balancing scheme to handle client requests. Workload distribution is usually managed by the same application server plug-in module that enables the use of a reverse proxy for static content. See “Scenario 2: Static Content Deployed in a Reverse Proxy” on page 27.

The Web application server instances (managed servers) in a cluster can coexist on the same machine (vertical clustering), or the managed servers can run on a group of

middle-tier server machines (horizontal clustering). The SAS Web applications can be deployed on both vertical and horizontal clusters.

A different approach to load distribution involves merely deploying individual SAS Web applications on separate, non-clustered Web application servers. Though this approach reduces the memory load for any given server, a clustering strategy is preferable. Deployment is easier to manage with a cluster because all machines and server instances are identically configured. Furthermore, Web application servers provide deployment management services that facilitate management of a cluster. It is relatively easy to add additional nodes and increase the size of the cluster.

Requirement for Session Affinity

For SAS Web applications to be deployed into a clustered environment, the Web application servers must implement 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 in WebSphere Application Server and WebLogic Server.

Why session affinity is required: Although WebSphere Application Server, WebLogic Server, and JBoss provide the ability to migrate HTTP sessions from one server to another, the SAS Web applications do not support this capability. Business intelligence sessions often contain large data elements, such as results sets from ad hoc queries, reporting, and analytical tasks, that cannot be migrated easily among Web application servers.

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 opened. 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 Web application servers 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.

Additional Considerations for a Deployment

This section presents a few more things that you might want to consider when you plan your middle-tier deployment.

Load-Balancing Software and Hardware for the HTTP Servers

In scenario 3, the Web application servers are clustered to balance the load and to provide increased availability. While this scenario provides redundancy for the application servers, the HTTP server that is deployed as a reverse proxy remains a potential bottleneck and single point of failure. To improve availability and increase capacity, you can distribute HTTP traffic across multiple reverse proxies by placing load-balancing software or hardware in front of those servers. A single load-balancing component can accept client HTTP requests and distribute those requests across a cluster of reverse proxies.

A number of vendors sell load-balancing software and hardware products for HTTP servers, including IBM, Cisco, and Nortel. If you are interested in this type of load-balancing, you can explore the product lines for these and other vendors.

Secure Sockets Layer

If you are moving sensitive information across the Internet, then you might want to use HTTPS and Secure Socket Layer (SSL) to encrypt your communication links. SSL uses Public Key Cryptography, which is based on the implementation of a public and private key pair. Each of your servers that handles encrypted communications manages certificates that contain both the private key and the public key. A description of how Public Key Cryptography and SSL work is beyond the scope of this document. However, there are many good sources for that information.

Here are some factors to consider when determining whether and how to use SSL:

- Which links do you want to encrypt? In the figures shown for the various scenarios, each arrow represents a potential communications link that might be encrypted. You should consider encrypting the following:
 - Encrypt any data that is capable of moving across the public Internet. If connections to your site go through a virtual private network (VPN), then those connections are already encrypted. Otherwise, traffic to and from your site is open to packet analysis by Internet users.
 - Encrypt all traffic that moves between the client and your HTTP server that resides in the DMZ.
 - Always encrypt traffic that is used to transmit credit card numbers, Social Security numbers, and any other sensitive information.

To achieve strong security, encrypt links all the way to the Web application server. If you are concerned about internal packet analysis, you can encrypt everything. However, total encryption comes with a cost, as explained in the remaining considerations.

- Some load-balancing schemes might rely on packet content for routing. When that is the case, encryption can impede the work that is performed by load-balancing software or hardware because encryption renders the packet content undecipherable.
- Cryptography requires resource-intensive computation, and this resource requirement typically reduces the amount of traffic that your servers are able to handle.

- The certificates that are used with SSL expire at fixed intervals. When a user's certificate expires, the user must obtain a new certificate before logging on to your applications. If you want a highly available system, then you should prepare for certificate renewal in advance to avoid unexpected downtime.
- You must decide whether to use certificates that are generated by a Certification Authority (CA), or whether self-signed certificates are adequate for your application. Self-signed certificates can save you money, but you are responsible for managing their distribution to clients.

Web Authentication

By default, SAS Web applications use the form-based authentication that is provided by the SAS Logon Manager Web application. When credentials are provided to the SAS Logon Manager Web application, 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, the Web application server handles the initial authentication. In this configuration, the Web application server's JAAS login module authentication provider verifies the user's identity. Then, the SAS Logon Manager Web application makes a trusted user connection to the metadata server to check that the authenticated user has a SAS identity in metadata.

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 the following topics:

- For a detailed explanation of different types of authentication, see "Authentication Mechanisms" in the *SAS Intelligence Platform: Security Administration Guide*.
- For information about setting up the middle-tier applications to use Web authentication, see the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.
- For information about achieving a single sign-on approach to authentication, see "Using Single Sign-On Among Web Applications" on page 41.

Heap Size for SAS Remote Services Application

Middle-tier applications use the SAS Remote Services Application to pass session and user context between Web applications. The SAS Remote Services application enables the user to pass seamlessly through to the target without the requirement for a separate logon.

During installation, the SAS Deployment Wizard enables you to specify the desired initial and maximum heap size for the Remote Services application by using the JVM option format.

JVM options of the SAS Remote Services Application are set to handle a moderately high number of concurrent users. For a very large number of concurrent users and a distributed SAS 9.2 topology, you should tune the JVM options to accommodate the deployment.

If you use the Windows service, you can increase the minimum and maximum heap size of the SAS Remote Services Application. Edit the **wrapper.conf** file located in the *SAS-configuration-directory\Lev1\Web\Applications\RemoteServices* directory.

Alternatively, you can add the recommended JVM options to one of the following scripts:

- On Windows:
`SAS-configuration-directory\Lev1\Web\Applications\RemoteServices\RemoteServices.bat`
- On UNIX:
`SAS-configuration-directory/Lev1/Web/Applications/RemoteServices/RemoteServices.sh`
- On z/OS:
- `SAS-configuration-directory/Lev1/Web/Applications/RemoteServices/RemoteServices.sh`

Tuning the Web Application Server

In addition to specifying Java Virtual Machine options, you can improve the performance of SAS Web applications by configuring other aspects of your 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 Java Virtual Machine 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. In addition, SAS provides additional tuning information in “SAS 9.2 Web Applications: Tuning for Performance and Scalability” that is available with the Web application server documentation at

<http://support.sas.com/resources/thirdpartysupport/v92>.

Configuring a Cluster of Web Application Servers

Cluster configuration varies widely between Web application server vendors. Consult your vendor's documentation for configuration instructions. Note, however, that you must deploy all the SAS Web applications to all nodes of the cluster. For a visual representation, see “Scenario 3: Web Applications Deployed across a Web Application Server Cluster” on page 29.

It is possible to configure a cluster that consists of just one node. You might set up a single-node cluster when your sole objective is to route browser requests to an HTTP server instead of to the Web application server. For this configuration, you set the address of the single-node cluster equal to the address of the HTTP server.

Configuring HTTP Sessions in Environments With Proxy Configurations

SAS Web Report Studio 4.2 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, 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 your Web application server.

Using an HTTP Server to Serve Static Content

Overview of Using an HTTP Server to Serve SAS Themes Web Application Static Content

Your middle-tier deployment can use an HTTP server to handle requests for the static content in the SAS Themes Web application. This HTTP server can be configured as a reverse proxy to forward requests for dynamic content to your Web application server, or the content can be deployed on a standard HTTP server. This strategy makes efficient use of the HTTP server, and enables the Web application server to devote its resources to dynamic content. The performance benefits are particularly notable for large-scale deployments that include a cluster of Web application servers. For an overview of this configuration, see “Sample Middle-Tier Deployment Scenarios” on page 24.

Example: Use Apache HTTP Server to Serve SAS Themes Web Application Static Content

This example describes how to configure Apache HTTP Server to server the SAS Themes Web application as static content on a Windows system. There is more than way one to configure Apache HTTP Server, and your configuration might differ from what is presented here.

- 1 Locate the EAR file for SAS Web Application Themes. The EAR file is named **sas.themes.ear** and is located in the *SAS-configuration-directory\Levn\Web\Staging* directory.
- 2 Unpack **sas.themes.ear**. On a Windows system, you can use the unzip utility. The EAR archive contains several files, including **sas.theme.default.war** and **application.xml**.
- 3 Open the **application.xml** file in an editor. In **application.xml**, the URL for the application is shown as **sas.theme.default.war**. The mapping shows that the name of the application (the context root) is **SASTheme_default**.
- 4 In the Apache document directory, create a directory that has the same name as the application (**SASTheme_default**). Here is an example:
D:\Apache2.2\htdocs\SASTheme_default
- 5 Extract the contents of **sas.theme.default.war** into the directory that you created in the previous step.
- 6 Stop and restart Apache HTTP Server.
- 7 On the Web application server, stop and uninstall the SAS Themes Web application.
- 8 Use SAS Management Console to change the connection properties for SAS Themes.
 - a Select **Application Management > Configuration Management**.
 - b Right-click **SASTheme_default** and select **Properties**.
 - c Click the **Connection** tab, set **Host Name** and **Port Number** to the host name and port number of the Apache HTTP Server, and then click **OK**.
- 9 Restart each of the SAS Web applications, or restart the Web application server.

When you access a SAS Web application, the SAS Themes Web application content is served by Apache HTTP Server rather than the Web application Server. In the Apache HTTP Server access log, you can verify that the files from the **SASTheme_default** directory are being served by Apache HTTP Server.

Using a Proxy Plug-in between the Web Application Server and the HTTP Server

WebLogic Server, WebSphere Application Server, and JBoss provide plug-in modules that enable integration with an HTTP server, such as Apache HTTP Server or Microsoft Internet Information Services (IIS).

The plug-ins are useful for either or both of the following:

- to forward requests for dynamic content to the Web application server or servlet container. In this scenario, the HTTP server handles all the static content and relies on the Web application server for dynamic content.

- to forward requests and distribute those requests among a cluster of Web application servers using a load-balancing algorithm.

The plug-in enables the HTTP server to behave as a reverse proxy, which typically filters requests and passes requests that meet the filter requirements to the Web application server. Many reverse proxies use a local cache of Web pages to respond to requests.

The configuration instructions vary greatly depending on your particular architecture. For the best information, see the vendor's documentation.

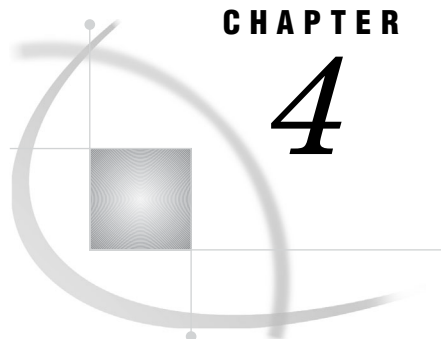
Using Apache Cache Control for Static Content

To avoid sending unnecessary requests to the server each time a client requests a static content item, you can configure Apache HTTP Server to set cache time-out values for static content.

Typically, after a browser initially downloads a static resource from the HTTP server, the browser sends a conditional HTTP GET request each time the browser encounters that resource again. For example, when a browser first downloads a SAS Web Report Studio logo image, the browser stores a local copy of the image. For each subsequent page that references the logo, the browser requests that the image be sent again if the image has been modified since the previous download. This sequence occurs for every static element and can result in large numbers of HTTP requests. Because the static content for is not modified often, most of these requests are unnecessary.

When you specify a cache time-out for each static element, clients (browser, proxy, or server cache) can avoid sending unnecessary requests to the HTTP server in order to check the validity of the content. When the browser first accesses a static element, the browser stores that element locally for the duration of the time-out value that is configured. During this time, subsequent queries to the HTTP server are suppressed for that element. The browser resumes queries as appropriate when the time-out period elapses within the session.

You can configure Apache HTTP Server to set cache time-out values for static content. This is true whether Apache HTTP Server is configured to serve that static content or is merely a reverse proxy to your Web application server.



CHAPTER

4

Middle-Tier Security

<i>Middle-Tier Security</i>	39
<i>Using the SAS Anonymous Web User With SAS Authentication</i>	40
<i>Multicast Security</i>	40
<i>Using Single Sign-On Among Web Applications</i>	41
<i>Using Secure Sockets Layer (SSL) for Web Applications</i>	41
<i>Overview of SSL</i>	41
<i>Set Up the SSL Environment for Your Web Application Server</i>	42
<i>SSL for SAS Web Applications</i>	42
<i>One-Way SSL for SAS Information Delivery Portal</i>	43
<i>Step 1: Specify https Protocol and Port Number</i>	43
<i>Step 2: Specify https Protocol and Port Number for the SAS Content Server</i>	44
<i>Step 3: Update Remote Portlets for SSL</i>	44
<i>Step 4: Restart the SAS Remote Services Application and the Web Application Server</i>	44
<i>Step 5: Verify the SSL Connection</i>	44
<i>Two-Way SSL for SAS Information Delivery Portal</i>	44
<i>Configuring and Deploying Restrictive Policy Files</i>	45
<i>About Restrictive Policy Files</i>	45
<i>Example Policy Files for JBoss and WebSphere</i>	46
<i>Create Restrictive Policies for JBoss</i>	47
<i>Create Restrictive Policies for WebSphere</i>	48
<i>Restore Your SAS Environment to Use Default Policies</i>	48
<i>Disable Restrictive Policy Handling for JBoss</i>	49
<i>Disable Restrictive Policy Handling for WebSphere</i>	49
<i>Customize Permissions for Socket Access</i>	49
<i>Access Permissions for Custom Portlets and Web Applications</i>	50
<i>About Access Permissions for Custom Portlets and Web Applications</i>	50
<i>CodeBase: <Remote Portlet or Web Application></i>	50
<i>CodeBase: Portal</i>	51
<i>CodeBase: SASServices</i>	51

Middle-Tier Security

To determine how to implement middle-tier security, you should consider your organization's internal security policies, the security mechanisms that are in place in your environment, the types of users who will need to access the Web applications, and the types of content that will be made available.

Important concepts and tasks concerning middle-tier security are as follows:

- **Authentication.** For a detailed discussion of different types of authentication and configuration guidelines, see "Authentication Mechanisms" in the *SAS Intelligence Platform: Security Administration Guide*. For information about configuring Web

authentication for JBoss, IBM WebSphere, or Oracle WebLogic, go to <http://support.sas.com/resources/thirdpartysupport/v92/>.

- SAS Anonymous Web User. See “Using the SAS Anonymous Web User With SAS Authentication” on page 40.
- Multicast Security. See “Multicast Security” on page 40.
- Single Sign-On. See “Using Single Sign-On Among Web Applications” on page 41.
- Secure Sockets Layer (SSL). See “Using Secure Sockets Layer (SSL) for Web Applications” on page 41.
- Restrictive Policy Files. See “Configuring and Deploying Restrictive Policy Files” on page 45.

Using the SAS Anonymous Web User With SAS Authentication

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, which is configured with the SAS Deployment Wizard, 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.

If the webanon account is configured, it will be used when a Web service is configured for SAS authentication, and credentials are not supplied. If the webanon account is not configured, there will be no credentials for authentication, and the request will fail.

In a default SAS 9.2 installation, 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 installations of SAS 9.2, 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.

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 Web application server’s startup script (for example, **SASServer1.bat**) and the **environment.properties** file located in the **SAS-configuration-directory\Lev1\Web\Applications\RemoteServices** directory.

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 your Web application server.

For more information, see Chapter 13, “Administering Multicast Options,” on page 169.

Using Single Sign-On Among Web Applications

Single Sign-On (SSO) is an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. SSO 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. SSO 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 in.

SAS provides these SSO features:

- To bypass the logon prompt when launching a desktop application (such as SAS Information Map Studio, SAS Enterprise Guide, SAS Data Integration Studio, SAS OLAP Cube Studio, or SAS Management Console), use Integrated Windows authentication. The client and the metadata server must be in the same Windows domain or in domains that trust each other.
- To bypass the logon prompt when launching a SAS Web application (such as SAS Web Report Studio or SAS Information Delivery Portal), use Web authentication.
- Seamless access to data servers and processing servers is provided by mechanisms including SAS token authentication, Integrated Windows authentication, credential reuse, and credential retrieval.

For more information about SSO, see the *SAS Intelligence Platform: Security Administration Guide*.

Using Secure Sockets Layer (SSL) for Web Applications

Overview of SSL

Secure Sockets Layer (SSL) is a protocol that provides network security and privacy. Developed by Netscape Communications, SSL uses encryption algorithms that include RC2, RC4, DES, TripleDES, IDEA, MD5, and others. In addition to providing encryption services, SSL uses trusted certificates to perform client and server authentication, and it uses message authentication codes to ensure data integrity. SSL is supported by both Firefox and Internet Explorer. Many Web sites use the protocol to protect confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection begin with HTTPS instead of HTTP. The SSL protocol is application independent and allows protocols such as HTTP, FTP, and Telnet to be transparently layered above it. SSL is optimized for HTTP. SSL includes software

that was developed by the OpenSSL Project for use in the OpenSSL Toolkit. For more information, see <http://www.openssl.org>.

This documentation assumes that you have a basic understanding of SSL, and that you know how to obtain and use trusted certificates. See your Web application server's documentation for SSL implementation details at the following Web sites:

- <http://www.jboss.org/docs>
- <http://www.oracle.com/technology/documentation/index.html>
- <http://www.ibm.com/support/documentation/us/en/>

Also, see <http://support.sas.com/resources/thirdpartysupport/v92>.

Note: Transport Layer Security (TLS) is the successor to SSL V3.0. The Internet Engineering Task Force (IETF) adopted SSL V3.0 as the de facto standard and renamed it TLS. Throughout this document, any reference to SSL also applies to TLS. △

Set Up the SSL Environment for Your Web Application Server

After you have configured the Web application server for SSL, the Java Run Time Environment (JRE) in which the Web applications run will be ready to provide certificates in response to client requests.

If Web applications that communicate with each other are distributed across different machines, then the JRE that is used by each application requires a certificate. For example, suppose that a user logs on to the SAS Information Delivery Portal. If the user clicks on a report from within the portal, then the portal invokes SAS Web Report Studio in order to display the report (if the user has the appropriate role capability assigned for viewing the report with SAS Web Report Studio).

If the portal and SAS Web Report Studio run on different machines, then the certificate must reside in the JRE for each machine. Because the portal communicates directly with SAS Web Report Studio, and SAS Web Report Studio sends the requested page back to the portal, the portal's JRE must have the certificate that is used by the JRE for SAS Web Report Studio.

If the applications run on different Web application servers within the same host, you should configure the JRE for each Web application server.

SSL for SAS Web Applications

After you have copied the keystore file to the appropriate directory in your Web application server, and edited the applicable configuration file or settings to enable SSL, you specify the https protocol and ports for SAS Web applications. This step ensures that the appropriate SAS Web applications will use the https protocol. See "Step 1: Specify https Protocol and Port Number" on page 43.

One-way SSL enables the application operating as the SSL client to verify the identity of the Web application server that is operating as the SSL server. In two-way SSL, the SSL client application verifies the identity of the Web application server, and then the Web application server verifies the identity of the SSL-client application. In two-way SSL, the application presents its certificate to the Web application server after the Web application server authenticates itself to the SSL client application.

You can configure one-way or two-way SSL. For more information, see:

- "One-Way SSL for SAS Information Delivery Portal" on page 43.
- "Two-Way SSL for SAS Information Delivery Portal" on page 44.

One-Way SSL for SAS Information Delivery Portal

You can configure one-way SSL for SAS Information Delivery Portal by specifying the https protocol and ports in the SAS Management Console Configuration Manager for the portal and other SAS applications that are used by the portal. SAS Information Delivery Portal uses the SAS Content Server and the following SAS applications:

- ☐ SAS Logon Manager
- ☐ SAS Package Viewer
- ☐ SAS Preferences Manager
- ☐ SAS Help Viewer Metadata Configuration
- ☐ SAS Web Application Themes
- ☐ SAS Stored Process Web Application

When SSL is configured one way, all communications from all applications to the Web application server are encrypted and protected by SSL. However, encrypting and decrypting all communications might consume more resources and impact performance. For efficient use of resources, you can encrypt some of the communications by applying the https protocol to specific SAS applications. For example, SAS Web Application Themes and Help Viewer Meta Config use static content. Therefore, you can allow the SAS Web Application Themes and SAS Help Viewer Meta Config applications to use the regular http protocol and port numbers, and configure all of the other applications shown above to use the https protocol and port number.

Mixing the http and https protocols could enable an efficient environment where all communications are not being encrypted and protected by SSL. If you mix the http and https protocols, make sure that the SAS Logon Manager uses the https protocol. In addition, there are other considerations that might apply to your Web application server. For example, when you mix the protocols and your SAS Web applications are deployed in the Web application server, both the HTTP port and the HTTPS port must remain open and be available for use.

Step 1: Specify https Protocol and Port Number

To specify the https protocol and port number for the SAS Information Delivery Portal and other applications used by the portal, follow these steps:

- 1 In SAS Management Console, navigate to **Plug-ins ► Application Management ► Configuration Manager ► Information Delivery Portal 4.2** and right-click to display the Information Portal Delivery 4.2 Properties dialog box.
- 2 Click the **Connection** tab.
- 3 In the **Connection** tab, modify the **Communication Protocol** field to display **HTTPS**.
- 4 On the **Port Number** field, enter the appropriate port number for the https protocol. For example, the default port numbers for the Web application servers are as follows: 8443 for JBoss, 7002 for WebLogic, and 9443 for WebSphere.
- 5 Click **OK** to exit the Information Delivery Portal 4.2 Properties dialog box.
- 6 To specify the https protocol and port number for other SAS applications, repeat the preceding steps for these applications:
 - ☐ SAS Logon Manager
 - ☐ SAS Package Viewer
 - ☐ SAS Preferences Manager
 - ☐ SAS Help Viewer

- SAS Web Application Themes
- SAS Stored Process

Next, specify the https protocol and port number for the SAS Content Server. See “Step 2: Specify https Protocol and Port Number for the SAS Content Server” on page 44.

Step 2: Specify https Protocol and Port Number for the SAS Content Server

When you configure SSL for SAS Information Delivery Portal, specify the https protocol and port number for the SAS Content Server. To specify the https protocol and ports for the SAS Content Server, follow these steps:

- 1 In SAS Management Console, navigate to **Plug-ins ► Server Manager ► SAS Content Server** and select it to display **Connection:SAS Content Server** in the right-side panel of SAS Management Console.
- 2 Select the **Connection:SAS Content Server** and press the right-mouse button to display and select **Properties**.
- 3 On the **Options** tab, modify the **Application protocol** field to display **https**.
- 4 In the **Port number** field, enter the correct port number. The default port numbers for the Web application servers are as follows: 8443 for JBoss, 7002 for WebLogic, and 9443 for WebSphere.
- 5 Click **OK** to save your settings and exit the Connection: SAS Content Server Properties dialog box.

Note that these changes will go into effect when you start your Web application server.

Step 3: Update Remote Portlets for SSL

If you have remote portlets, update the protocol and port numbers for those portlets. Update the URL within the **portlet.xml** file, recreate the PAR file, and redeploy it.

Step 4: Restart the SAS Remote Services Application and the Web Application Server

Stop and restart the SAS Remote Services and the Web application server.

Step 5: Verify the SSL Connection

Log on to the appropriate URL that is configured with SSL. For example, to verify the SSL connection, log on to the appropriate URL:

https://yourmachine.company.com:8443/SASPortal

Two-Way SSL for SAS Information Delivery Portal

If you want to configure two-way SSL for SAS Information Delivery Portal, there are additional steps you must complete in addition to the procedures that apply to one-way SSL. With two-way SSL, you create a new, unique client certificate, install it in the client (the Web browser), and configure your Web application server to ask for and accept this certificate. Here is a brief summary of steps to be followed when you configure two-way SSL:

- Create new client certificate and sign it with your authority.
- Import the client certificate into the browser.

- Configure your Web application server to use the trust store, and to request and accept a client certificate
- Modify your JVM parameters for SAS Remote Services and for your Web application server by specifying the locations of the server keystore and trust store.

For information about configuration, see your vendor documentation. Also, see <http://support.sas.com/resources/thirdpartysupport/v92/>.

Configuring and Deploying Restrictive Policy Files

About Restrictive Policy Files

An express or typical installation completed with the SAS Deployment Wizard creates a SAS environment that does not use restrictive policy files to limit the access given to SAS Web applications. By default, the **sas.all.permissions.policy** file is used to allow access to the SAS Web applications. As a result, SAS Web applications can access the necessary content.

Java 2 Security provides a policy-based, fine-grain access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. By default, Java 2 Security is turned off. If your site requires Web applications to use Java 2 Security, the custom installation option in the SAS Deployment Wizard enables you to configure your SAS environment with restrictive policy files.

A custom installation of SAS 9.2 software gives you the opportunity to select the use of restrictive policy files for JBoss or IBM WebSphere application servers. Although WebLogic provides restrictive policy files, implementation of these policy files is problematic, and they cannot be used in the SAS 9.2 environment. Therefore, SAS 9.2 does not support restrictive policy files for WebLogic.

Your **Instructions.html** file provides basic guidelines for creating policy files from existing sample files, saving those files, and rebuilding the applications. If you chose not to enforce restrictive policy files at the time of initial installation, choose from one of the following methods for configuring restrictive policy files:

- Use the SAS Deployment Manager to remove the existing configuration of your SAS environment. Then, reconfigure the environment by choosing the custom installation option in SAS Deployment Wizard. The custom installation option enables you to configure restrictive policy files. This method, which is highly recommended, offers the most dependable and thorough approach to ensure that your SAS environment is set up correctly to use the Java 2 Security and restrictive policy files.
- Manually configure and enforce the use of restrictive policy files. Follow this method if your site has significantly large amounts of custom content, and the previously described method is not feasible at your site.

CAUTION:

SAS strongly discourages the use of restrictive policy files on SAS middle-tier applications because they provide no end-user security, they are difficult to maintain, and they can be very detrimental to application performance. △

The SAS Deployment Wizard implements the following restrictive policies by using different methods for JBoss and WebSphere:

- JBoss application server. When **policy** files are edited and the SAS Web applications are rebuilt by using the SAS Deployment Manager, the edits made to the **policy** files are united into a single policy file (**sas.restrictive.permissions.policy**) that is applied to JBoss.
- WebSphere. Policy files for WebSphere are applied to each EAR file. Each policy file's inputs are placed into the corresponding EAR file as a **was.policy** file.

Example Policy Files for JBoss and WebSphere

SAS applications provide policy files (**example.policy**) for JBoss and WebSphere in the *SAS-configuration-directory\Lev1\Web\Common\SASServer1\Application-name\PolicyFileInputs\ears* directory. These **example.policy** files contain default restrictive policy settings. You do not edit policy files directly. Instead, you make a copy of the **example.policy** file, rename the copied file as **policy**, and edit the **policy** file. If the **policy** file exists, it is used to implement restrictive policies. See “Create Restrictive Policies for JBoss” on page 47.

Note: The united **example.policy** file for JBoss is located in the *SAS-configuration-directory\Lev1\Web\Commons\SASServer1\JBoss\PolicyFileInputs\ears* directory. △

The following table shows the directory paths for the JBoss and WebSphere policy files with security restrictions for SAS applications.

Table 4.1 Policy Files with Security Restrictions for JBoss and WebSphere

Application	Location of example.policy below \Lev1\Web\Common\SASServer1 Directory
SAS Information Delivery Portal	SASPortal4.2\PolicyFileInputs\ears\sas.portal\
SAS Web Report Studio	SASWebReportStudio4.2\PolicyFileInputs\ears\sas.webreportstudio\
SAS Content Server	SASContentServer9.2\PolicyFileInputs\ears\sas.wip.scs\
SAS Shared Services	SASSharedServices9.2\PolicyFileInputs\ears\sas.shared\
SAS Stored Process Application	SASStoredProcessApplication9.2\PolicyFileInputs\ears\sas.storedprocess\
SAS WebInfrastructure Platform	SASWebInfrastructurePlatformApplications9.2\PolicyFileInputs\ears\sas.wip.apps\
SAS Web OLAP Viewer	SASWebOLAPViewer4.2\PolicyFileInputs\ears\sas.webolapviewer\
SAS BI Dashboard	SASBIDashboard4.2\PolicyFileInputs\ears\sas.bidashboard\
SAS BI Portlets ¹	SASBIPortlets4.2\PolicyFileInputs\ears\sas.biportlets\
SAS Package Viewer	SASPackageViewer4.2\PolicyFileInputs\ears\sas.packageviewer\

Application	Location of example.policy below \Lev1\Web\Common\SASServer1 Directory
SAS Preferences	SASPreferences9.2\CustomContent\wars\ sas.preferences\
SAS Help Viewer for the Web	SASWebDoc9.2\PolicyFileInputs\ears\sas.webdocmd\
SAS OnlineDoc for the Web	<i>SAS-installation-</i> <i>directory\Documentation\9.2\onlinedocweb\</i>
SAS Stored Process	SASStoredProcessApplication9.2\CustomContent\wars\ sas.storedprocess\

1 Available in the October 2009 Release.

Create Restrictive Policies for JBoss

To create a restrictive policy file for JBoss, follow these steps for each applicable SAS application's **policy** file:

- 1 Make a copy of the **example.policy** file in the same directory and name the copied file as **policy**. If you need to edit the restrictive policy settings for JBoss, make a copy of the **example.policy**, rename it as **policy**, and save the renamed file in the *SAS-configuration-directory\Lev1\Web\Common\jboss\PolicyFileInputs* directory.

Note: SAS OnlineDoc for the Web is not delivered with the SAS Intelligence Platform, and is deployed separately. To apply Java permissions to SAS OnlineDoc for the Web, make a copy of the **example.policy** in the *SAS-installation-directory\Documentation\9.2\onlinedocweb* directory, name the copied file **policy**, and place the renamed file within the same directory. Then, copy the contents of the **policy** file for SAS Online Doc for the Web directly to the JBoss policy file. △

- 2 Edit the **policy** file that you created from the original **example.policy** file. Policy files must use UTF-8 character encoding.
- 3 Run the SAS Deployment Manager to rebuild SAS Web applications. Select **JBoss** and any applications for which you have edited the restrictive policy file. Rebuilding for JBoss will recreate the **Java 2 security policy** file, and the **sas.restrictive.permissions.policy**. For information about how to rebuild Web applications, see “Rebuilding the SAS Web Applications” on page 94. If you are using the second maintenance release for SAS 9.2 and rebuilding Web applications, the EAR files are automatically exploded. Previously, in SAS 9.2, this was not the case.

Note: SAS Online Doc for the Web is not rebuilt or redeployed via SAS Deployment Manager. You must manually rebuild and redeploy SAS Online Doc for the Web. For information about manual deployment of the application, see “Deploying SAS OnlineDoc Manually for the Web” on page 102. △

- 4 If you perform an auto-configuration of JBoss, restart the JBoss application server. If you want to follow a manual process, copy the **sas.restrictive.permissions.policy** file located in the *SAS-configuration-directory\Lev1\Web\Common\jboss* directory to the *JBoss-installation-directory\server\SASServer1\conf* directory. Then restart JBoss.

Create Restrictive Policies for WebSphere

To convert a SAS 9.2 environment that does not use restrictive policies to an environment where restrictive policies are applied, you modify the **policy** file for each SAS application that has a EAR file associated with it.

Note: SAS OnlineDoc for the Web is not delivered with the SAS Intelligence Platform, and it is deployed separately. The **was.policy** file for SAS OnlineDoc for the Web already contains the appropriate restrictive policies, and is included in the EAR file for WebSphere. Therefore, you do not edit any policies for SAS OnlineDoc for the Web. △

Although the following procedure applies to the policy file for SAS Information Delivery Portal, you can follow the same steps by substituting the appropriate directories for the policy file that applies to each SAS application. To convert from all permissions to restrictive permissions for SAS applications, follow these steps:

- 1 In the WebSphere Admin Console, navigate to **Security ► Secure administration, applications, and infrastructure**. Enable Java 2 Security by selecting the check box **Use Java 2 Security to restrict application access to local resources**. Save your changes.
- 2 Make a copy of the **example.policy** file located in the *SAS-configuration-directory\Lev1\Web\Common\SASServer1\SASPortal4.2\PolicyFileInputs\ears\sas.portal* directory, name the copied file as **policy**, and save this file in the same directory where **example.policy** file resides.
- 3 For all SAS applications for which you want to implement restrictive policies (with the exception of SAS OnlineDoc for the Web), edit the **policy** file that you created from the original **example.policy** file, and make any changes that should apply to your site. Save the modified **policy** file in the *SAS-configuration-directory\Lev1\Web\Common\SASServer1\SASPortal4.2\PolicyFileInputs\ears\sas.portal* directory. Policy files must use UTF-8 character encoding.
- 4 Run the SAS Deployment Manager to rebuild the SAS Web applications (select the applications for which the policy files were modified). For information about how to rebuild Web applications, see “Rebuilding the SAS Web Applications” on page 94. The edited **policy** files are stripped of all comments, and their contents are inserted into the appropriate EAR file as a **was.policy** file. When you rebuild the Web applications, SAS Deployment Manager rebuilds a complete EAR file that includes any new content that was added to the **policy** files.

Note: SAS OnlineDoc for the Web is not rebuilt or redeployed via SAS Deployment Manager. You must manually rebuild and redeploy SAS OnlineDoc for the Web. For information about manual deployment of this application, see “Deploying SAS OnlineDoc Manually for the Web” on page 102. △

- 5 Using the WebSphere Admin Console, redeploy each SAS Web application that was modified previously.
- 6 Using the WebSphere Admin Console, restart the Web application server.

Restore Your SAS Environment to Use Default Policies

If you customized your SAS environment by implementing the use of restrictive policy files, and you determined that the policy restrictions are unnecessary or that the performance impact is debilitating, you can restore your SAS environment to use default policies. To turn off restrictive policies and the use of Java 2 Security in your SAS environment, follow these steps:

- 1 Use the SAS Deployment Manager to remove the current configuration of your SAS environment.
- 2 Use the SAS Deployment Wizard to configure your SAS environment by not selecting the option to use restrictive policy files.

It is highly recommended that you use the SAS Deployment Manager and the SAS Deployment Wizard to complete the process of disabling restrictive policy files. However, if your site contains large amounts of custom content, or there are other reasons that require you to manually disable restrictive policy handling, see the following topics:

- “Disable Restrictive Policy Handling for JBoss” on page 49.
- “Disable Restrictive Policy Handling for WebSphere” on page 49.

Disable Restrictive Policy Handling for JBoss

To manually disable the use of SAS restrictive policy files for JBoss, follow these steps:

- 1 On Windows, access the **SASServer1.bat** file located in the *JBoss-home-directory\bin* directory. On UNIX, access the **SASServer1.sh** file located in the *JBoss-home-directory\bin* directory.
- 2 In the section **JAVA_OPTS** line located within the **start_as_script** section, remove the following parameters:

```
---Djava.security.manager=Djava.security.policy=
```

```
JBoss-home-  
directory\server\SASServer1\sas.restrictive.permissions.policy
```

- 3 Restart the JBoss application server.

If JBoss is running as a Windows service, follow these steps to remove restrictive policy files:

- 1 On Windows, access the **wrapper.conf** file located in the *JBoss-home-directory\server\SASServer1* directory.
- 2 Remove the following parameters in the **wrapper.conf** file:

```
wrapper.java.additional.##=-Djava.security.manager  
wrapper.java.additional.##=-Djava.security.policy=
```

```
JBoss-home-  
directory\server\SASServer\conf\sas.restrictive.permissions.policy
```

- 3 Restart the JBoss application server.

Disable Restrictive Policy Handling for WebSphere

To manually disable SAS restrictive policy handling for WebSphere, follow these steps:

- 1 Using the WebSphere Admin Console, navigate to **Security ► Secure administration, applications, and infrastructure**.
- 2 To disable Java 2 security deselect the check box for **Use Java 2 security to restrict application access to local resources**.
- 3 Restart the WebSphere application server.

Customize Permissions for Socket Access

For each application (Web or stand-alone) that needs to communicate with a SAS server, the Java policy files for the calling application include a permission to

communicate with the SAS Server. By default, the **example.policy** files for each SAS Web application contain wildcard permission for socket access:

```
permission java.net.SocketPermission "*",
"accept,connect,listen,resolve";
```

This wildcard permission enables the Java code in the applications to connect to any host or port that is accessible to your site's network topology. If you want to provide strong protection with custom access, you can create specific socket permissions for the hosts and ports that are accessed by an individual SAS Web application.

Access Permissions for Custom Portlets and Web Applications

About Access Permissions for Custom Portlets and Web Applications

If you implement a remote portlet or foundation service-enabled Web application, you must add additional permissions to each Web application component's codebase and define a codebase and permissions for the remote portlet or foundation service-enabled Web application.

The following sections show the permission statements you must specify in each application or portlet's policy file in order to enable communication with its required servers and services.

CodeBase: <Remote Portlet or Web Application>

The localhost is the machine where the Web application server resides along with the metadata server and SAS Remote Services. When using a localhost, specify the permissions for the remote portlet or Web application's CodeBase:

- access to the SAS Metadata Server:

When running on localhost, create an entry that contains the fully qualified host name.

```
// permission java.net.SocketPermission
// "localhost:8561", "listen, connect, accept, resolve";
```

```
permission java.net.SocketPermission
<SAS Metadata Server's machine>:8561,
"listen, connect, accept, resolve";
```

- access to the Java RMI server and remote SAS Foundation Services:

When running on localhost, create an entry that contains the fully qualified host name.

```
// permission java.net.SocketPermission
// "localhost:1024-", "listen, connect, accept, resolve";
```

```
permission java.net.SocketPermission
<SAS Services application's machine name>:1024-,
"listen, connect, accept, resolve";
```

- Access to the remote portlet or Web application's local SAS Foundation Services:
Always create an entry for both the localhost and fully qualified host name.

```
permission java.net.SocketPermission
"localhost:1024-", "listen, connect, accept, resolve";
```



```

permission java.net.SocketPermission
  <remote portlet or Web application's machine name>:1024-,
  "listen, connect, accept, resolve";

```

- Access for foundation service-enabled applications that call this application to pass objects (via RMI to this application):

Create one entry per machine.

```

permission java.net.SocketPermission
  <portal Web application's machine name>:1024-,
  "listen, connect, accept, resolve";

```

- Access to a SAS Stored Process, Workspace, or OLAP server:

Create one entry per machine.

```

permission java.net.SocketPermission
  <SAS Workspace Server's machine name>:1024-,
  "connect, resolve";
permission java.net.SocketPermission
  <SAS Stored Process Server's machine name>:1024-,
  "connect, resolve";
permission java.net.SocketPermission
  <SAS OLAP Server's machine name>:1024-,
  "connect, resolve";

```

- Access to the host and port where the SAS Web Application Themes is running:

```

// ----- Socket Access to Themes -----
permission java.net.SocketPermission
  Theme_host:Theme_Port:,
  "connect, resolve";

```

CodeBase: Portal

Access for foundation service-enabled applications that are called by this application to pass objects (via RMI) (for example, remote portlets, Web applications, and applications):

Create one entry per machine.

```

permission java.net.SocketPermission
  <remote portlet/Web application's machine name>:1024-,
  "listen, connect, accept, resolve";

```

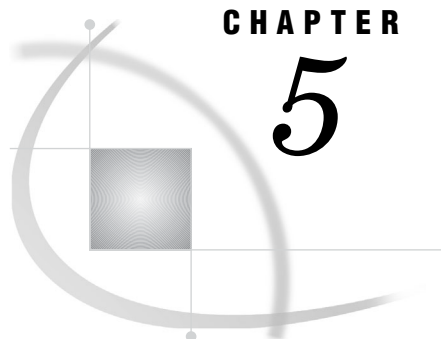
CodeBase: SAServices

The **remoteservices.policy** file is located in the *SAS-configuration-directory* \Lev1\web\applications\remoteservices directory. The following applies to connections with applications that use SAS Foundation Service session sharing:

```

permission java.net.SocketPermission
  <remote portlet/Web application's machine name>:1024-,
  "listen, connect, accept, resolve";

```

CHAPTER

5

Interacting with the Server Tier

<i>Configuration Shared between the Middle Tier and the Server Tier</i>	53
<i>Configuring an SMTP Mail Server for Use with the SAS Middle Tier</i>	53
<i>Client-Side Pooling and Server-Side Pooling Options</i>	54
<i>Configuring Data Sources Used by the SAS Middle Tier</i>	54
<i>About the Data Sources Used by the Middle Tier</i>	54
<i>Using the SAS Table Server with the SAS Middle Tier</i>	55
<i>About the SAS Table Server</i>	55
<i>Configuration Information for the SAS Shared Services Data Source</i>	55
<i>The SAS Shared Services Database on SAS Table Server</i>	56
<i>Back Up and Restore the Shared Services Database</i>	56
<i>Using Other Relational Databases with the SAS Middle Tier</i>	56
<i>Configuring Application Response Measurement (ARM) Capabilities</i>	56
<i>About ARM</i>	57
<i>Enable ARM Capabilities for SAS Logon Manager</i>	57

Configuration Shared between the Middle Tier and the Server Tier

The Web applications and services that form the SAS middle tier require specific configured connections to back-end servers. You might want to modify the connections and settings in the following ways:

- ❑ Change the connection to an SMTP Mail Server.
- ❑ Modify the connection to a relational database via a data source that is defined in the Web application server (for those deployments that include SAS Shared Services, SAS solutions, or both).
- ❑ Define pooling options for connections to SAS Workspace Servers.
- ❑ Integrate Application Response Measurement (ARM) capabilities between the SAS middle tier and SAS servers.

Configuring an SMTP Mail Server for Use with the SAS Middle Tier

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

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

- The `mail.transport.protocol` property must be set to `smtp`.
- The `mail.smtp.host` property must be set to the name of the SMTP mail server that is used by SAS applications to send mail.
- The `mail.smtp.port` property must be set to the corresponding port (typically 25 for SMTP servers).

During typical production deployments, the value of the `mail.debug` property is set to `false`. You can set this value to `true` if you need to debug mail transactions.

In a standard installation of SAS middle-tier components, the configuration of the Java Mail Session is typically automated using prompted values that are provided by the installer. To modify the settings for the Java Mail Session (for example, if the host name of the SMTP mail server changes), see the appropriate documentation for your Web application server.

Client-Side Pooling and Server-Side Pooling Options

A collection of reusable workspace server and stored process server processes is referred to as a pool. By reusing server processes, pooling avoids the cost that is associated with creating a new process for each connection. If your client application uses frequent, short-duration connections to SAS, pooling might greatly improve your server performance.

SAS supports the following types of pooling:

server-side pooling	is the process by which the SAS Object Spawner maintains a collection of workspace servers 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.
client-side pooling	is the process by which the client application maintains a collection of reusable workspace server processes.

For a comparison of client-side pooling and server-side pooling, see “Choices in Workspace Server Pooling” in the “Server Configuration, Data Retrieval, and Risk” chapter in the *SAS Intelligence Platform: Security Administration Guide*.

For more detailed information about pooling, see “Understanding Server Pooling” in the *SAS Intelligence Platform: Application Server Administration Guide*.

For instructions on configuring client-side pooling properties, see “Configuring Client-Side Pooling” in the *SAS Intelligence Platform: Application Server Administration Guide*.

Configuring Data Sources Used by the SAS Middle Tier

About the Data Sources Used by the Middle Tier

Several services in the middle tier, including SAS Shared Services as well as 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 shared platform relational tables are intrinsic to or used primarily for the operations of a particular application, product, or a specific service.

Applications and services access data from the database server via a JDBC data source that is defined for the Web application server. In a deployment that includes SAS Shared Services, a data source is created exclusively for use by the Shared Services. Other SAS applications and solutions might specify additional data sources.

Using the SAS Table Server with the SAS Middle Tier

About the SAS Table Server

SAS Table Server stores audit records for enhanced auditing functionality supported by SAS Shared Services.

During an Enterprise BI Server installation, a database and a data source name (DSN) definition are created automatically on the SAS Table Server for the exclusive use of SAS Shared Services. Both the database and the DSN are named *SharedServices*.

Configuration Information for the SAS Shared Services Data Source

When using SAS Table Server, use the following configuration information for the SharedServices DSN:

Configuration Parameter	Setting
JNDI name:	<code>sas/jdbc/SharedServices</code>
connection URL:	<code>jdbc:sastkts://serverName:port?stmtpooling=0&constring=(DSN=SharedServices)</code> In the URL, substitute the server name and port number of the SAS Table Server at your site. The default port for SAS Table Server is 2171.
JDBC driver class:	<code>com.sas.tkts.TKTSDriver</code>

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

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

The following JAR files must reside on the Web application server:

- **`sas.core.jar`**
- **`sas.core.nls.jar`**
- **`sas.icons.jar`**
- **`sas.icons.nls.jar`**
- **`sas.intrnet.javatools.jar`**
- **`sas.intrnet.javatools.nls.jar`**

- ☐ `sas.nls.collator.jar`
- ☐ `sas.oda.tkts.jar`
- ☐ `sas.oda.tkts.nls.jar`
- ☐ `sas.rutil.jar`
- ☐ `sas.rutil.nls.jar`
- ☐ `sas.security.ssapi.jar`
- ☐ `sas.svc.connection.jar`
- ☐ `sas.svc.connection.nls.jar`

To modify the settings for a data source, see the documentation for your Web application server.

The SAS Shared Services Database on SAS Table Server

On Windows, the **SharedServices.fdb** database is located in the *SAS-configuration-directory\Lev1\SASTS\Content* directory. On UNIX and z/OS, the **SharedServices.fdb** database is located in the *SAS-configuration-directory/Lev1/SASTS/Content* directory. The Web application server points to this database. The DSN definition is a metadata object that contains connection information that allows SAS Shared Services to connect to the Shared Services database. The connection information is configured by default to be optimum for SAS Shared Services.

CAUTION:

Do not change the name or contents of the DSN. Doing so prevents SAS Shared Services from functioning. \triangle

Back Up and Restore the Shared Services Database

To back up the database, use the following command:

```
sastback -b SharedServices.fdb -mo read_only [-verbose] SharedServices.fbk
```

To restore the database, use the following command:

```
sastback -r SharedServices.fbk [-verbose] SharedServices.fdb
```

On Windows, UNIX, and z/OS, the SASTBACK executable is in the *SAS-installation-directory/SASFoundation/9.2/vulcan/bin* directory.

Using Other Relational Databases with the SAS Middle Tier

SAS Shared Services can be configured to use a relational database other than SAS Table Server to handle its storage requirements. In addition, some SAS solutions and applications might require a database other than SAS Table Server (such as MySQL). The other relational databases that can be used vary depending on the set of SAS applications that your site has installed. Contact your on-site SAS support personnel for more information.

Configuring Application Response Measurement (ARM) Capabilities

About ARM

The SAS server environment provides instrumentation for standard Application Response Measurement (ARM) processing. ARM is an industry standard application programming interface (API) for measuring end-to-end application response time. With SAS servers, these ARM capabilities enable you to measure and monitor the time required for specific SAS activities. The SAS Web Infrastructure Platform provides additional ARM capabilities that enable the measurement and monitoring of activities that originate in the SAS middle tier.

In the SAS Web Infrastructure Platform, the SAS Logon Manager application has been instrumented for ARM processing. Depending on the software that is installed at your site, other SAS Web applications and solutions might implement ARM processing as well. For more information, see the documentation for particular SAS applications and solutions.

Enable ARM Capabilities for SAS Logon Manager

ARM processing for SAS Logon Manager is disabled by default in a standard deployment. To modify configuration information to enable ARM processing, follow these steps:

- 1 Ensure that the **`sas.arm.log4j.jar`**, **`sas.entities.jar`**, and **`sas.core.jar`** files are available on the CLASSPATH that is used by log4j logging.

The JAR files can be obtained from the SAS Versioned Jar Repository, which is typically located under
SAS-installation-directory\SASVersionedJarRepository\9.2\eclipse\plugins.

- 2 In a text editor, open the **`SASLogon-log4j.xml`** file, which is typically located in *SAS-configuration-directory\Levn\Web\Common\LogConfig.*

The directory where this file is located is defined by the `com.sas.log.config.url` system property, which can be found in the start-up script or configuration file for the Web application server.

- 3 Add the following lines to **`SASLogon-log4j.xml`** file:

```
<appender name="ArmAppender" class="com.sas.arm.log4jappender.ArmAppender">
  <param name="AppName" value="IOM.APP"/>
  <param name="GroupName" value="SAS"/>
</appender>
<logger name="com.sas.arm.log4j.logger">
  <level value="debug"/>
  <appender-ref ref="ArmAppender"/>
</logger>
```

Note: In the XML code, the appender element comes before the logger element. When you edit a log4j configuration XML file, all appenders must precede all loggers and categories. Otherwise, the configuration fails. △

- 4 In the Web application deployment, edit the **`aop-config.xml`** file, which is found at *exploded-apps-location\sas.wip.apps9.2.ear\sas.svcs.logon.war\WEB-INF\spring-config*. Remove the XML comments around the definition and reference of the `armProcessor` bean.

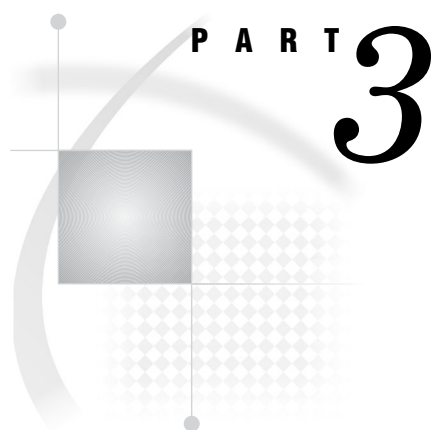
Note: If the EAR files in your environment are deployed unexploded, you must explode **`sas.wip.apps9.2.ear`** in order to edit the **`aop-config.xml`** file. To explode the file, unarchive both the EAR file and any WAR files within the EAR file. After you edit the **`aop-config.xml`** file, re-archive the WAR and EAR files. △

- 5 In the Web application deployment, edit the **services-remote-config.xml** file, which is found at *exploded-apps-location\sas.wip.services9.2.ear\sas.wip.services.war\WEB-INF\spring-config*. Remove the XML comments around the definition and reference of the `armProcessor` bean.

Note: If the EAR files in your environment are deployed unexploded, you must explode **sas.wip.services9.2.ear** in order to edit the **services-remote-config.xml** file. To explode the file, unarchive both the EAR file and any WAR files within the EAR file. After you edit the **services-remote-config.xml** file, re-archive the WAR and EAR files. △

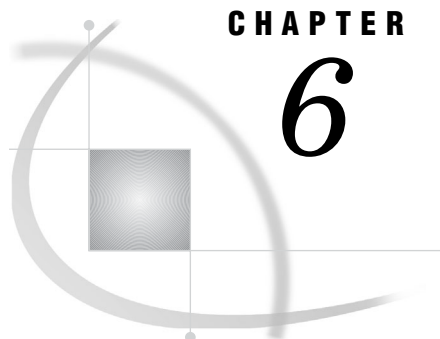
- 6 Restart the SAS Services Application and the Web application server.

Upon successful start of the application server, the ARM monitoring of logon and logoff activities is enabled.



Middle-Tier Administration

<i>Chapter 6</i>	Administering the SAS Web Infrastructure Platform	61
<i>Chapter 7</i>	Using the SAS Web Infrastructure Platform Utilities	79
<i>Chapter 8</i>	Administering SAS Web Applications	93
<i>Chapter 9</i>	Administering SAS Shared Services	117
<i>Chapter 10</i>	Administering the SAS Content Server	121
<i>Chapter 11</i>	Administering SAS BI Web Services	135
<i>Chapter 12</i>	Administering SAS Web Application Themes	153
<i>Chapter 13</i>	Administering Multicast Options	169



CHAPTER

6

Administering the SAS Web Infrastructure Platform

<i>SAS Web Infrastructure Platform</i>	61
<i>About the SAS Web Infrastructure Platform</i>	61
<i>SAS Preferences Manager</i>	62
<i>SAS Logon Manager</i>	62
<i>About SAS Logon Manager</i>	63
<i>Example: Enable the Log On Button on Time-out Pages</i>	63
<i>SAS Management Console</i>	63
<i>Overview of SAS Management Console</i>	64
<i>Modify Folders</i>	64
<i>Using Configuration Manager</i>	64
<i>Overview of Configuration Manager</i>	64
<i>Summary of Steps for Using Configuration Manager</i>	66
<i>Example: Configure a Property for SAS Web Report Studio</i>	66
<i>Setting Global Properties for SAS Applications Using SAS Application Infrastructure Properties</i>	67
<i>Specifying Connection Parameters for HTTP and HTTPS Sessions</i>	70
<i>Using the SAS Web Administration Console</i>	72
<i>About the SAS Web Administration Console</i>	72
<i>Access the SAS Web Administration Console</i>	73
<i>Monitor Users</i>	73
<i>About the Users That Appear in the SAS Web Administration Console</i>	73
<i>Send E-Mail to One or More Users</i>	73
<i>Force Users to Log Off</i>	74
<i>System Maintenance Tools for Managing User Login Sessions</i>	74
<i>Maintenance Restart Wizard</i>	75
<i>Quiesce the System</i>	76
<i>View Information about Web Applications</i>	76

SAS Web Infrastructure Platform

About 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 10.

The following sections describe two of the applications.

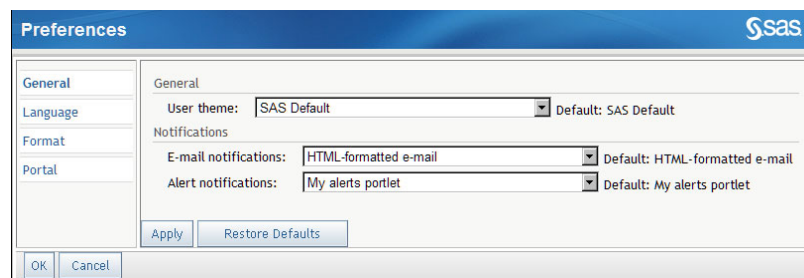
SAS Preferences Manager

The SAS Preferences Manager is a Web application that provides a central facility for users to manage their preferences and settings. When a user sets preferences, those preferences apply to all SAS Web applications that use the SAS Preferences Manager.

Users can invoke the SAS Preferences Manager in the Web application that they are using. The method used to invoke the SAS Preferences Manager varies with the application that is being used. 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.

Display 6.1 SAS Preferences Manager Console



Here are the generic settings:

General

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

Users can also specify the format for notifications that are generated by SAS applications and solutions.

Language

Select the locale (language and country) that you prefer.

Format

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

Portal

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

SAS Logon Manager

About SAS Logon Manager

The SAS Logon Manager is a Web application that handles all authentication requests for SAS Web applications that build on the business intelligence platform. As a result, users see the same logon page when they access the SAS Web applications.



The purpose of the SAS Logon Manager is to authenticate and direct a successful logon to the appropriate page. 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.

You can configure settings for the Logon Manager in the **Application Management ► Configuration Manager** section in SAS Management Console. For example, you can specify the default theme. You can also set policies that affect logon and logoff, such as whether to display a **Log On** button on the Web time-out page.

Example: Enable the Log On Button on Time-out Pages

To enable the display of a **Log On** button on users' Web time-out pages, follow these steps:

- 1 Log on to SAS Management Console.
- 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 In the left pane, select **Policies**.

- 6 In the right pane, click the locked icon  next to **Allow user login from web timeout page**. The icon changes to unlocked . Click **OK**.

When a field has a locked icon, the value or setting for that particular field cannot be modified on the **Settings** tab for other SAS applications that inherit their settings from this application. By default, all SAS applications inherit their settings from SAS Application Infrastructure. For more information about locking and unlocking settings, see “Setting Global Properties for SAS Applications Using SAS Application Infrastructure Properties” on page 67.

- 7 On the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Logon Manager 9.2**.
- 8 Right-click **Logon Manager 9.2** and select **Properties**.
- 9 Click the **Settings** tab.
- 10 In the left pane, select **Policies**.
- 11 In the right pane, select **Yes** in the list box for **Allow user login from web timeout page**, and click **OK**.

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 66.

Overview of SAS Management Console

The SAS Management Console is the primary tool available to administer the SAS Intelligence Platform. It is a framework in which a variety of plug-ins are installed to expand the capability of the SAS Management Console. A plug-in is an application module that is designed to create and maintain metadata for a specific type of resource.

Only certain users can view and use plug-ins. A user's access to plug-ins depends on which roles the user is assigned to and which capabilities are assigned to those roles.

For information about SAS Management Console and plug-ins, see "Understanding the State of Your System" in the *SAS Intelligence Platform: System Administration Guide*. Also, see the online Help available with the SAS Management Console.

Modify Folders

If you have installed SAS Web applications such as SAS Web Report Studio, then some SAS folders (for examples, folders that contain metadata for reports) have associated physical content that resides on the SAS Content Server. In these instances, a procedure called *content mapping* maps the metadata folder structure to corresponding physical folders that have the same organization. Content mapping is automatically configured when you install your system. For more information about content mapping, see "Protect Report Content in the WebDAV Server" on page 217.

Do not move, rename, or delete folders within SAS Folders that are mapped to a physical folder structure on the SAS Content Server. If you modify a folder that has content mapping specified, then the physical content to which the folder is mapped might become inaccessible.

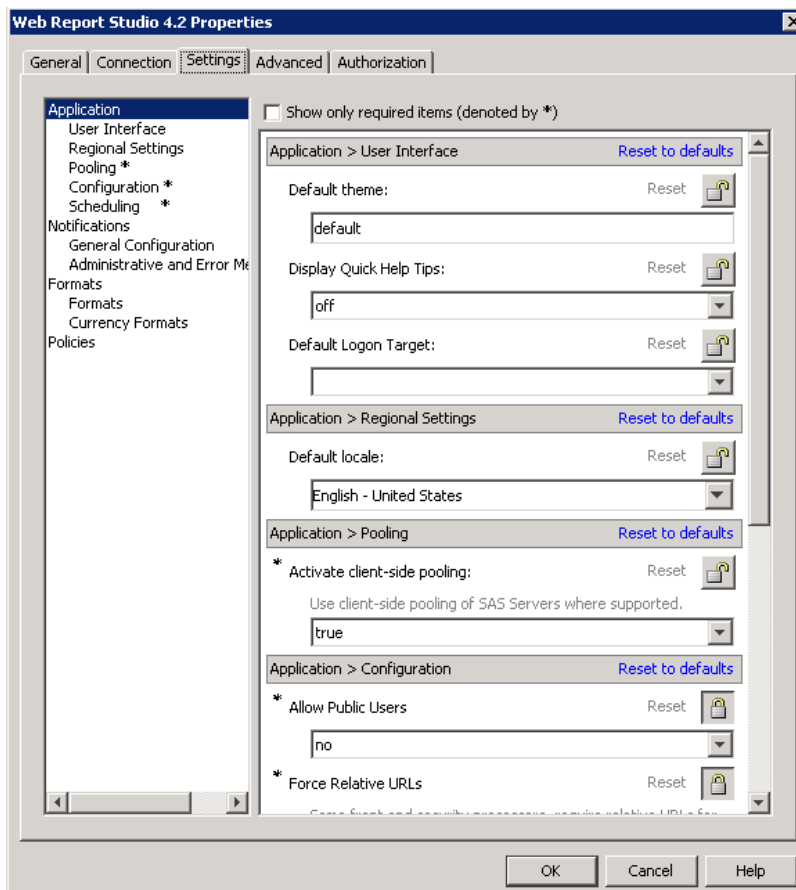
For information about the SAS Folders in the SAS Management Console, see the *SAS Intelligence Platform: System Administration Guide*.

Using Configuration Manager

Overview of Configuration Manager

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

Configuration Manager offers a consistent interface to set properties for all SAS Web applications. Each SAS Web application has its own properties window with tabs. For example, the following display shows the **Settings** tab of the Web Report Studio 4.2 Properties dialog box.

Display 6.2 Settings Tab for SAS Web Report Studio Properties

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

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

- The **General** tab provides basic information about the application.
- The **Connection** tab enables you to modify the parameters for connections to SAS Web applications. For more information, see “Specifying Connection Parameters for HTTP and HTTPS Sessions” on page 70.
- The **Settings** tab offers default values for settings that can be modified. For modifying values in the **Settings** tab, and to understand how the lock and unlock icons function, see “Setting Global Properties for SAS Applications Using SAS Application Infrastructure Properties” on page 67.
- 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, **LocalProperties.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**.
 - 2 To access the properties for an application, right-click the application's node and select **Properties**.
 - 3 Add or modify properties as needed. You might need to unlock particular properties before you can change them. See “Setting Global Properties for SAS Applications Using SAS Application Infrastructure Properties” on page 67.
 - 4 Changes to properties do not take effect immediately on the run-time system. To apply these changes, you must perform one of the following tasks:
 - Stop and then restart the Web applications whose properties you changed.
 - Use the application's JMX management bean to reload the configuration (if the application supports JMX beans). For more information about JMX, see “Using JMX Tools to Manage SAS Resources” on page 87.
 - Alternatively, stop and then restart SAS Services Application and the Web application server.
-

Example: Configure a Property for SAS Web Report Studio

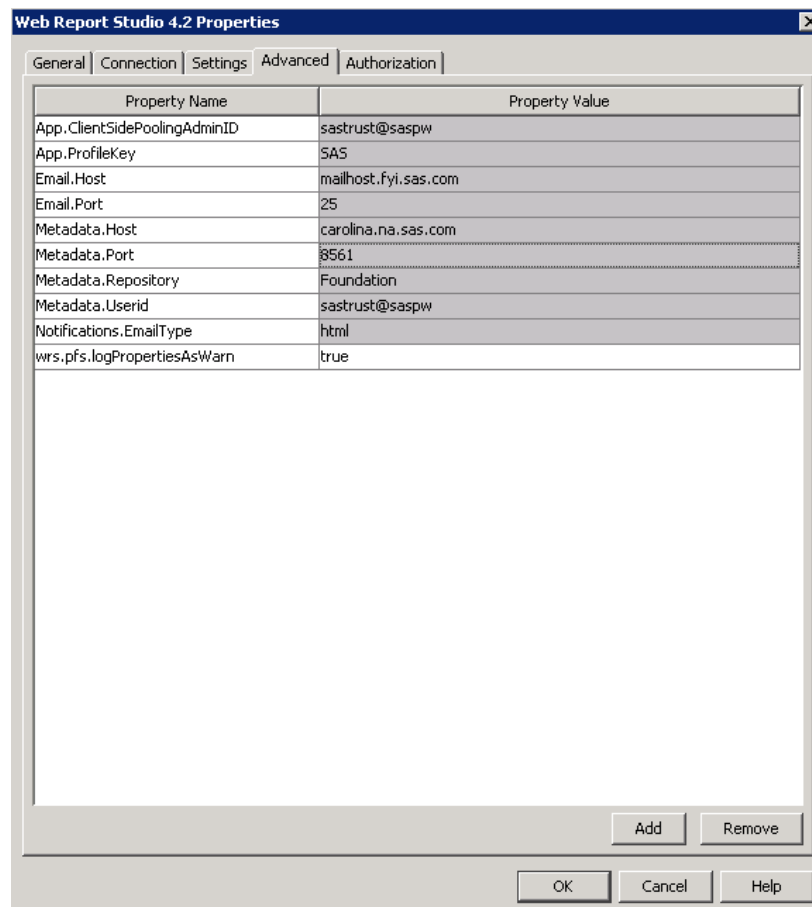
Suppose that you want to add the property, `wrs.pfs.logPropertiesAsWarn` for SAS Web Report Studio, and specify the value to be true for this property. To configure this property and its value, follow these steps:

- 1 Log on to SAS Management Console.
- 2 In SAS Management Console, navigate to **Plug-ins ► Application Management ► Configuration Manager ► Web Report Studio 4.2**. Right-click and select **Properties** to display the Web Report Studio 4.2 Properties window.
- 3 Click on the **Advanced** Tab.
- 4 Click **Add** to display the Define New Property window.
- 5 Enter the property name as shown and specify the property value:

Property Name: wrs.pfs.logPropertiesAsWarn
Property Value: true
- 6 Click **OK** to exit the Define New Property window.
- 7 Click **OK** to exit the Web Report Studio 4.2 Properties window.

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 66.

The following display shows the property name, `wrs.pfs.logPropertiesAsWarn`, and its property value specified on the **Advanced** tab.

Display 6.3 Advanced Tab for SAS Web Report Studio Properties

For more information about the properties and values that apply to SAS Web Report Studio 4.2, see “Configuring SAS Web Report Studio” on page 183.

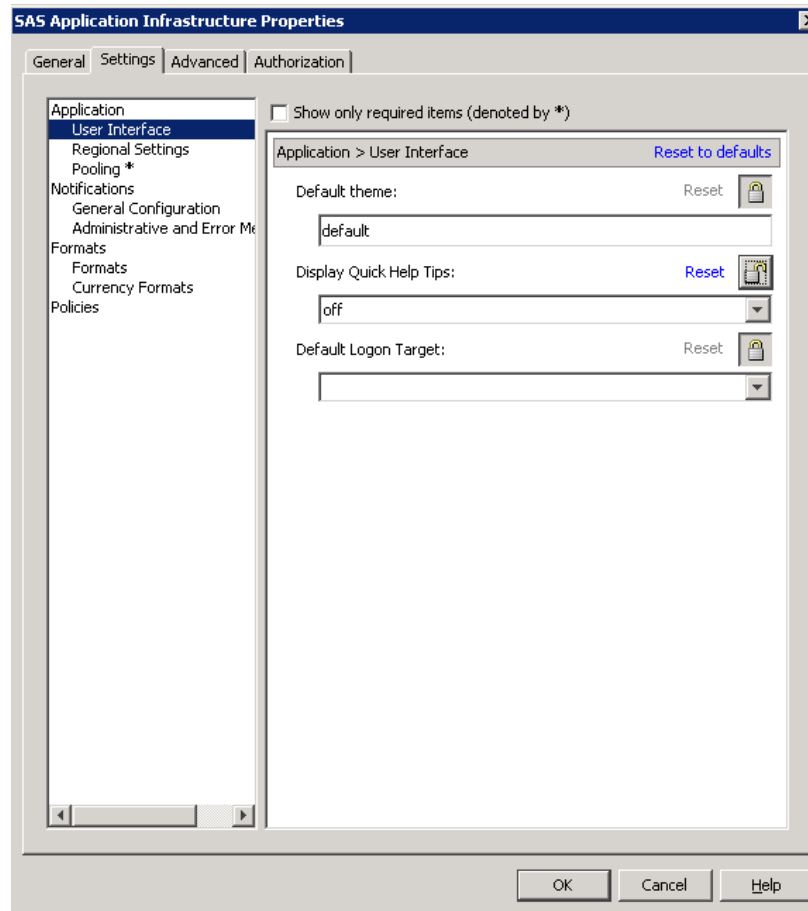
Setting Global Properties for SAS Applications Using SAS Application Infrastructure Properties


The Configuration Manager plug-ins within SAS Management Console enable you to configure properties and specify values and settings individually for each SAS application.

By default, all SAS applications inherit their settings from SAS Application Infrastructure. Settings that are changed in the SAS Application Infrastructure Properties dialog box apply to all SAS applications that inherit their settings from SAS Application Infrastructure.


By default, most settings are locked at the SAS Application Infrastructure level, and these settings cannot be changed for individual SAS applications. When you unlock a setting at the SAS Application Infrastructure level, that setting can be overridden for individual applications. When you lock a setting at the SAS Application Infrastructure level again, all applications inherit that setting from the SAS Application Infrastructure.

The following display shows the options for **User Interface** on the **Settings** tab for SAS Application Infrastructure.

Display 6.4 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 modified on the **Settings** tab for other SAS applications that inherit their settings from this application.

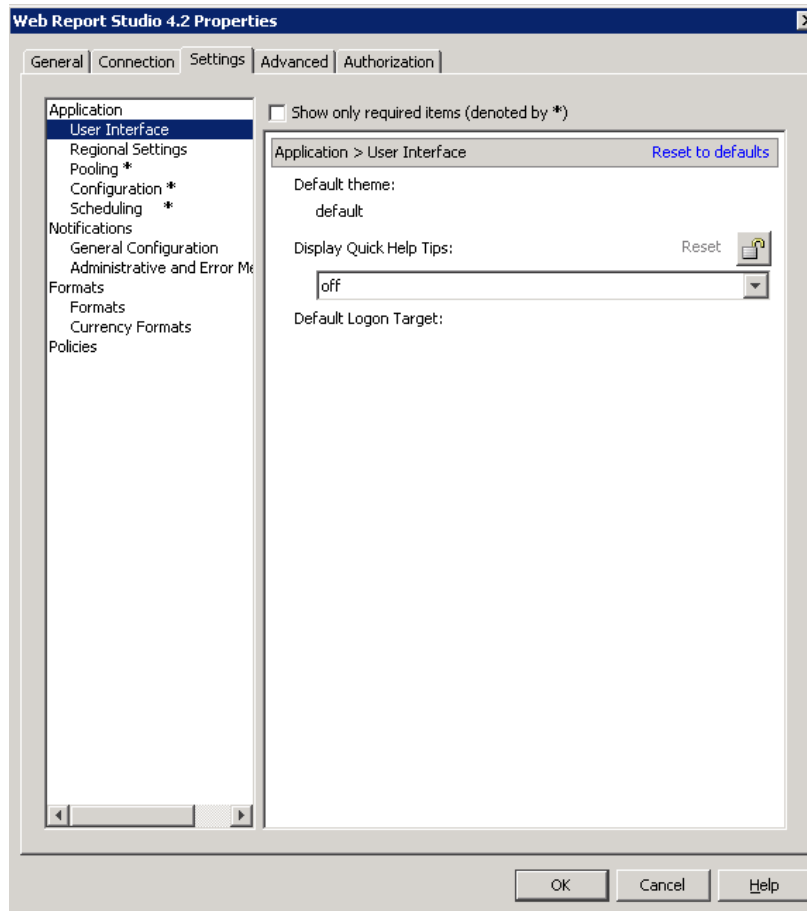
To unlock a field, follow these steps:

- 1 Click the locked icon. The icon changes to an unlocked icon  .
- 2 Click **OK** to save the unlocked field.

In the example, the **Display Quick Help Tips** field has been unlocked. The unlocked icon indicates that the values for **Display Quick Help Tips** can be modified individually for any SAS application in Configuration Manager. The other fields remain locked. Therefore, they cannot be modified individually for any SAS application.

Note: By default, all fields on the **Settings** tab of the SAS Application Infrastructure Properties dialog box are locked. △

The following display shows the effect of these locked states on the **Settings** tab for the Web Report Studio 4.2 Properties dialog box.

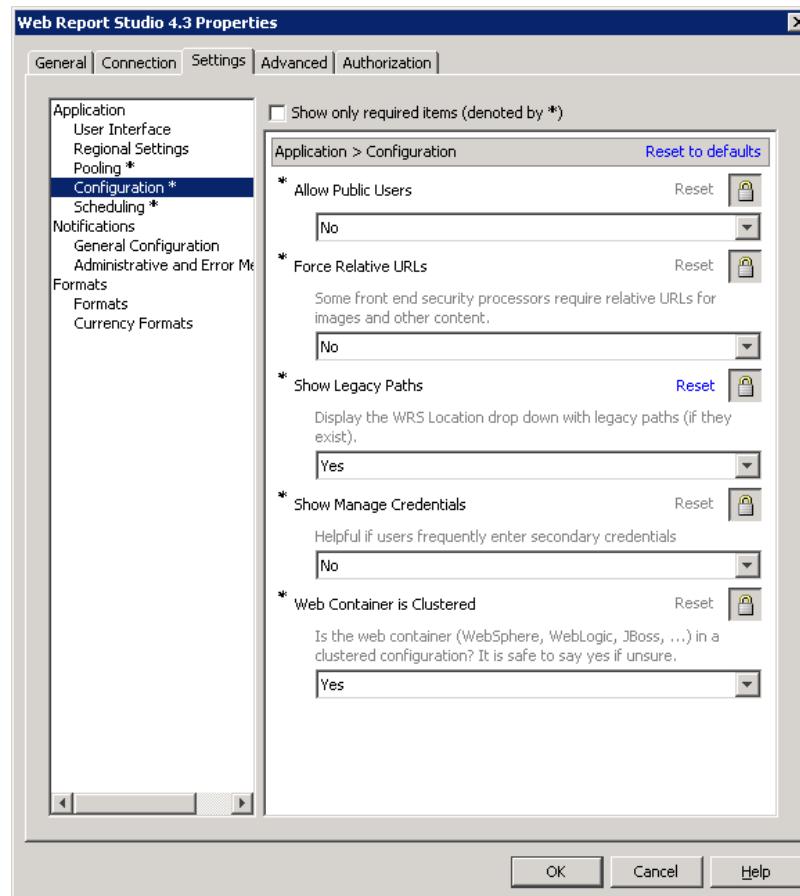
Display 6.5 Settings Tab for SAS Web Report Studio Properties

In this display, the unlocked icon shows that **Display Quick Help Tips** can be modified individually for SAS Web Report Studio. However, the other fields cannot be modified individually for SAS Web Report Studio for the following reasons:

- Those fields are locked in the SAS Application Infrastructure Properties dialog box.
- The locked states in the SAS Application Infrastructure dialog box apply globally to all SAS application plug-ins in Configuration Manager.

Certain applications (for example, SAS Web Report Studio) have settings that are unique to that application. Those settings are not inherited from the SAS Application Infrastructure. Therefore, those settings can be modified on the **Settings** tab for the Web Report Studio 4.2 Properties dialog box without any need to reset the lock for these settings at the SAS Application Infrastructure level.

The following display shows settings that are unique to SAS Web Report Studio. All of these configuration settings, which are not inherited from the SAS Application Infrastructure level, can be modified and saved individually in this dialog box.

Display 6.6 Unique Settings for SAS Web Report Studio Properties

Specifying Connection Parameters for HTTP and HTTPS Sessions

The **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 **Connection** tab determine the URL that is used to access the application's resources or services.

The following display shows the **Connection** tab for SAS BI Dashboard properties.

Display 6.7 Connection Tab for SAS BI Dashboard Properties

The screenshot shows the 'BI Dashboard 4.2 Properties' dialog box with the 'Connection' tab selected. The 'Communication Protocol' is set to 'HTTP'. The 'Host Name' is 'pubhdc1.na.sas.com', the 'Port Number' is '8080', and the 'Service' is '/SASBIDashboard'. The 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

Connection to the Application	
Communication Protocol:	HTTP
Host Name:	pubhdc1.na.sas.com
Port Number:	8080
Service:	/SASBIDashboard

If your site changes its configuration after initial deployment, you might need to edit the connection information parameters. Here are some situations where the connection parameters are updated on the **Connection** tab:

- If a SAS Web application is moved to a different machine, you must modify the host name property for its connection.
- If you configure Secure Sockets Layer (SSL) for improved security, you must edit the Protocol property to modify the connection protocol to HTTPS for each affected application.
- If clustering or load balancing is configured, the connection parameters should be updated.
- If you deploy SAS Web Application Themes to a different Web application server, you should modify the theme metadata by specifying the name of the theme, and update other parameters such as host name and port number.

Changing the values for the **Host Name**, **Port**, or **Service** fields on the **Connection** tab enables the SAS Web Application Infrastructure to seamlessly redirect clients to the proper locations in a custom environment.

Note: 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] △

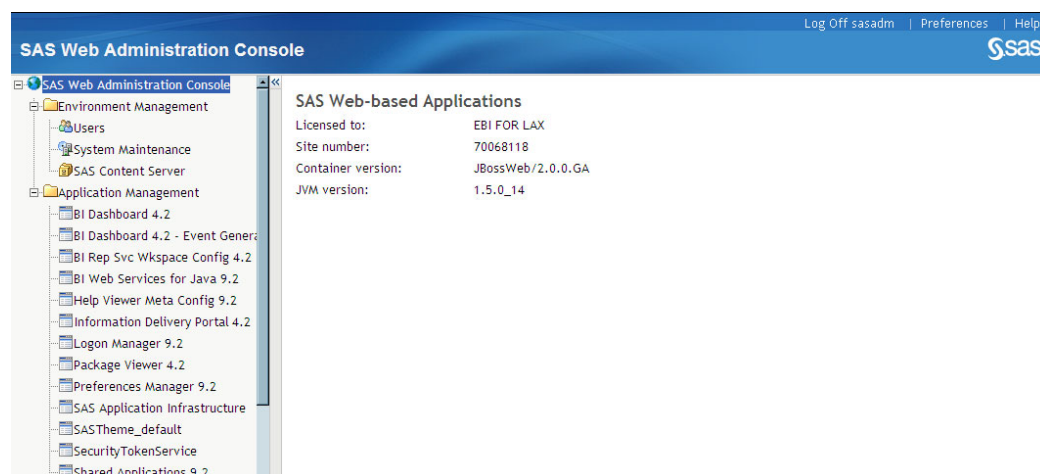
Using the SAS Web Administration Console

About the SAS Web Administration Console

The SAS Web Administration Console provides a central location for monitoring users, enabling an environment for system maintenance tasks, managing folders and permissions for the SAS Content Server, and managing SAS Web applications.

The following display shows an expanded view of a main page for the SAS Web Administration Console.

Display 6.8 Main Page in SAS Web Administration Console



Here is a description of what you can accomplish with the SAS Web Administration Console:

- The Users page enables you to view and monitor authenticated users and system users that are currently logged on to a SAS Web application. See “Monitor Users” on page 73.
- The System Maintenance page provides the Restart Maintenance Wizard and the Quiesce System feature. When you want to perform system maintenance, the Restart Maintenance Wizard enables you to send e-mail to users to log off from their sessions within a specified deadline, to log off users after the deadline, and to prohibit new users from logging on to their applications. The Quiesce System feature is useful when you want to allow existing users to stay logged on to their user sessions, but you want to quiesce the system by preventing new users from logging on to SAS Web applications. See “System Maintenance Tools for Managing User Login Sessions” on page 74.
- The SAS Content Server page enables you to manage folders and permissions for content in the SAS Content Server. You manage content by using either the SAS Content Server Administration Console (within the SAS Web Administration Console) or by using a stand-alone SAS Content Server Administration Console. You must be an unrestricted user in order to access the SAS Content Server Administration Console.

To access the SAS Content Server feature in the SAS Web Administration Console, select **Environment Management ► SAS Content Server** in the navigation pane.

For instructions on administering the SAS Content Server, see “Using the SAS Content Server Administration Console” on page 122.

- The Application Management page enables you to view the current configuration for Web applications that have been deployed at your site. For more information, see “View Information about Web Applications” on page 76.

Note: The SAS Web Administration Console can be extended by other SAS applications. Depending on the software that is installed at your site, your SAS Web Administration Console might be different from the one shown here. 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 server name and port number of your Web application server:

http(s)://server:port/SASAdmin

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

Note: The SAS Content Server Administration Console has its own logon requirements. For more information, see “Using the SAS Content Server Administration Console” on page 122. △

Monitor Users

About the Users That Appear in the SAS Web Administration Console

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. The information provided on the Users page is for informational purposes only. You cannot manage these users from the SAS Web Administration Console.


Send E-Mail to One or More Users

You can send e-mail to any of the users who are currently logged on to a SAS Web application. This feature is useful if you want to notify users of an impending system operation or a system outage.

To send e-mail to a user, 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 send e-mail to several users. To select all of the check boxes, select the check box in the heading of the last column.

- 3 Click the action menu  in the heading of the last column and select **Send E-mail**.
- 4 If necessary, enter the e-mail address of the recipient. If you enter more than one address, separate the addresses with a semicolon.
The addresses are already listed for users who have an e-mail address defined in SAS metadata.
- 5 Enter the subject and text of the message.
- 6 If you have more than one recipient, specify whether you want to send a single message to all recipients or to send a separate message to each recipient.
- 7 Click **Send**.

Force Users to Log Off

In some cases, users might not be actively working with a SAS Web application, and yet their sessions remain active in the system. You can force the termination of these user sessions by using the SAS Web Administration Console.

To force users to log off, follow these steps:

- 1 Select **Environment Management ► Users** in the navigation pane.
- 2 In the Users pane, select the check box next to an authenticated user's name.
You can select multiple check boxes in order to force off several users. To select all of the check boxes, select the check box in the heading of the last column.
- 3 Click the action menu  in the heading of the last column and select **Force Log Off**.
A confirmation page displays the user ID, e-mail 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.

System Maintenance Tools for Managing User Login Sessions

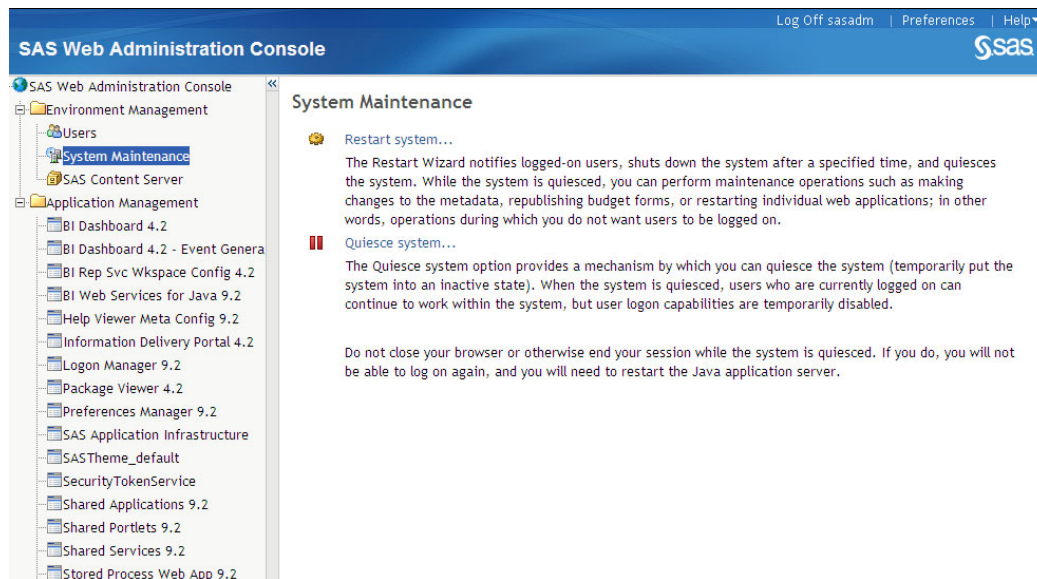
Tasks such as making changes to the metadata, restarting a metadata server, restarting the object spawner, or restarting a Web application can be performed safely only when users are not logged on to applications or when new users are prohibited from logging on to the applications. The Maintenance Restart Wizard enables you to perform a sequence of 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*.

Note: Do not close the Web browser during a quiesced state or when you are completing the steps in the Restart Maintenance Wizard. If the Web browser is closed during these sessions, restart your Web application server. △

The following display shows the System Maintenance page in the SAS Web Administration Console.

Display 6.9 System Maintenance Page



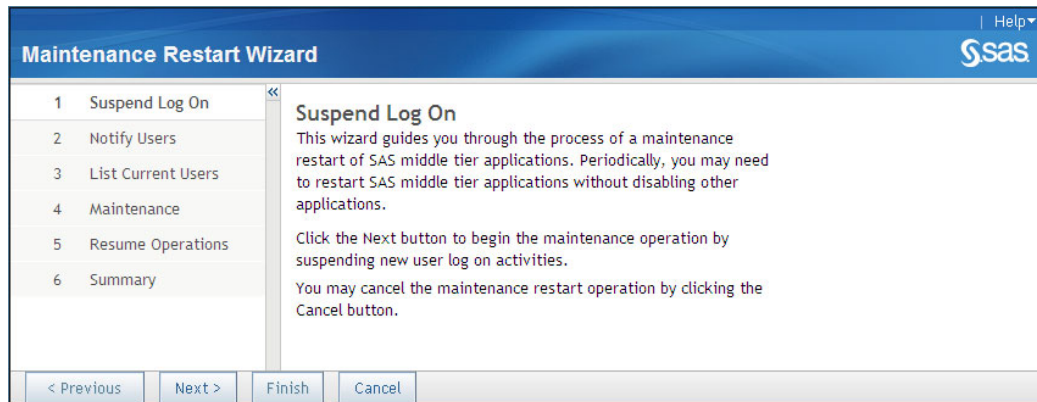
Maintenance Restart Wizard

Use the Maintenance Restart Wizard to prepare a system for maintenance and resume system operations as described in the following list:

- ☐ Notify authenticated users who are logged on to applications that system maintenance is planned, and specify a deadline by which they need to log off from their applications.
- ☐ Enable the shutdown of the system after a specified deadline.
- ☐ Enable the system to prohibit new users from logging on to their applications.
- ☐ If the notification deadline has passed, and users have not terminated their sessions, the system forces authenticated users to exit and terminate their sessions. All users are logged off.
- ☐ Quiesce the system by temporarily putting the system into an inactive state. When the system is quiesced, users' logon capabilities are disabled.
- ☐ Begin maintenance operations such as restarting the metadata server, the object spawner, or a Web application.
- ☐ Resume system operation by removing the quiesced state from the system, and enabling users to log on to the system and their applications.

To use the Maintenance Restart Wizard, log on to the SAS Web Administration Console. Navigate to **Environment Management** ► **System Maintenance**. Click **Restart System** and follow the Wizard's instructions.

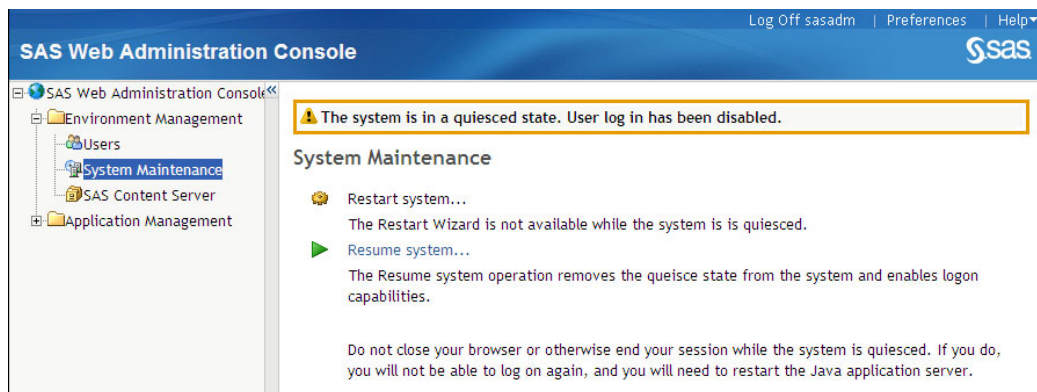
The following display shows the main page for the Maintenance Restart Wizard.

Display 6.10 Maintenance Restart Wizard

Quiesce the System

You can quiesce a system by allowing existing users to stay logged on to their applications, and prohibiting new users from logging on to their applications.

To quiesce the system, log on to the SAS Web Administration Console. Navigate to **Environment Management ► System Maintenance**. Click **Quiesce System**. When you are finished with your maintenance operations, click **Resume system** to remove the quiesced state and enable users to log on to their applications. The following display shows the message that is displayed when a system is quiesced.



View Information about Web Applications

The SAS Web Administration Console provides read-only information about the SAS Web applications that were installed and configured at your site. This feature is useful for viewing application information from any machine without the need to have SAS Management Console installed on the machine.

To display the applications, expand the **Application Management** node in the navigation pane.

The tree view on the left side of the page displays a hierarchical list of configured applications. The list varies depending on the software that has been installed.

When you click the name of an application, the right side of the page displays the following types of information:

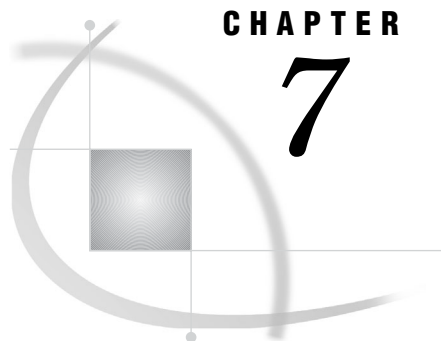
Application settings

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 e-mail 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.



CHAPTER

7

Using the SAS Web Infrastructure Platform Utilities

<i>Using the DAVTree Utility to Manage WebDAV Content</i>	79
<i>About the DAVTree Utility</i>	79
<i>Start the Utility and Connect to a WebDAV Location</i>	80
<i>Add Resources to WebDAV</i>	80
<i>Edit a Text File in WebDAV</i>	82
<i>Copy or Move a File in WebDAV</i>	82
<i>Advanced Features</i>	82
<i>Using the Package Cleanup Utility to Remove Packages</i>	82
<i>Overview of the Package Cleanup Utility</i>	82
<i>Deleting Packages</i>	83
<i>Delete Packages</i>	83
<i>Minimal Syntax for Deleting Packages</i>	83
<i>Delete Specific Packages</i>	84
<i>Change Prompt Behavior</i>	84
<i>List Packages</i>	84
<i>Arguments</i>	85
<i>Utility Logging and Debugging</i>	86
<i>Examples</i>	86
<i>Using JMX Tools to Manage SAS Resources</i>	87
<i>About JMX and MBeans</i>	87
<i>Accessing the SAS MBeans</i>	87
<i>About Accessing the SAS MBeans</i>	87
<i>Configure the Web Application Server to Enable JMX Client Access</i>	88
<i>Set Authentication Credentials for SAS MBean Access</i>	88
<i>Manage SAS Resources Using JConsole</i>	89
<i>Understanding How to Use the SAS MBeans</i>	90
<i>About the SAS MBeans</i>	90
<i>ServerFactory MBean</i>	90
<i>Spawner MBean</i>	90
<i>Server MBean</i>	91

Using the DAVTree Utility to Manage WebDAV Content

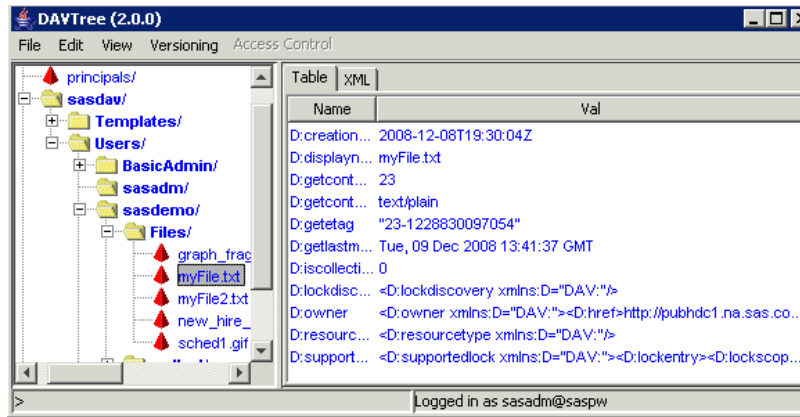
About the DAVTree Utility

The DAVTree utility is a stand-alone Java application that provides a tree view of WebDAV resources.

The utility enables you to manipulate content by copying files to a WebDAV repository or by creating text files such as forms and templates.

The utility presents information in a tree view. When you select a resource item in the tree on the left side of the window, the WebDAV properties for the resource are displayed on the right side.

Here is an example DAVTree interface:



In the interface, you see only the content that you are authorized to see.

The following guidelines apply to the use of the DAVTree utility:

- ☐ If you have not yet installed the third maintenance release for SAS 9.2, then you must manually configure the JAVA_HOME system variable to point to Java 5.
- ☐ If you have installed the third maintenance release for SAS 9.2, there is no need to manually configure the JAVA_HOME system variable to point to Java 5.

Start the Utility and Connect to a WebDAV Location

To use this utility, follow these steps:

- 1 Run the following command:

SAS-configuration-directory\Levn\Web\Utilities\DAVTree.bat

On UNIX and z/OS, the utility command is **DAVTree.sh**.

The DAVTree utility opens.

- 2 Select **File ► Open**.

The DAV Location dialog box opens.

- 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 2–6 and providing the URL for the new location.

Add Resources to WebDAV

You have the following options for adding resources to the WebDAV repository.

- Copy files to DAVTree. You can copy both text files and binary files to the repository.

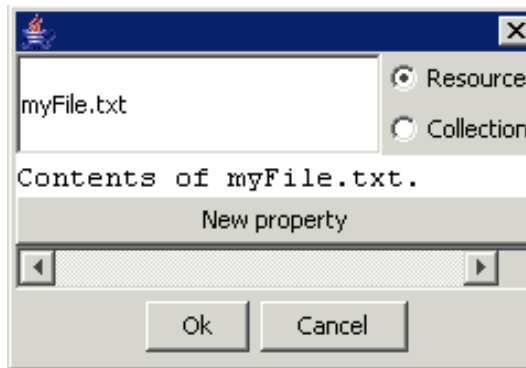
To copy a file, click and drag the file from the file system to a folder in the DAVTree interface.

This action can be performed on Windows systems and on UNIX systems that provide a graphical interface.

- 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 opens. 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 opens.

- 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**.

Note: To delete a resource, select the resource in the tree and then select **Edit ► Delete**. You are prompted to confirm the deletion. 

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 opens 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.

Note: You must refresh the view in order to see the change. △

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 advanced WebDAV functions, which are not covered in this documentation, should be performed only by someone who has WebDAV expertise. △

Using the Package Cleanup Utility to Remove Packages

Overview of the Package Cleanup Utility

The Package Cleanup utility provides a simple, command-line interface for deleting or listing packages that have been published in a publication channel or in a WebDAV repository.

The SAS Publishing Framework supports channels that you define in the SAS Metadata Repository. Once channels have been defined, users can publish packages to the channels. For example, portal users can subscribe to available channels, view the persisted packages, and publish content (files, links, stored processes, and information maps).

Channels can be defined with archive or WebDAV persistent stores. When a package is published to a channel that is defined with a persistent store, the package is first persisted to that location and then it is published to all subscribers of that channel. All persisted packages have an expiration date. However, expired packages are not deleted automatically; you must explicitly delete them. You can use the Package Cleanup utility for this purpose.


Here is the path to the utility:

SAS-configuration-directory\Levn\Web\Utilities\PackageCleanup.bat

On UNIX and z/OS, the utility command is **PackageCleanup.sh**.

The Package Cleanup utility enables you to review basic information about a persisted package and delete both the metadata and the actual package. Deletions are based on the expiration date of the package. This utility supports the deletion of packages from either type of persistent store (archive or WebDAV). The utility also supports the deletion of packages that are not defined in any channel.

The Package Cleanup utility also supports a listing feature. The utility can be used to display information about packages that are published in a particular channel, packages that are not defined in any channel, and packages that exist on a WebDAV server.


Note: You must have the appropriate permissions on a channel in order to delete packages from the channel. See the “Authorization Model” chapter in the *SAS Intelligence Platform: Security Administration Guide*. 

Deleting Packages

Delete Packages

To delete packages, follow these steps:

- 1 Run the command and specify the deletion date. You can also provide one of the following arguments:
 - ☐ a channel name in order to delete packages that are defined in a specific channel
 - ☐ a WebDAV URL in order to delete packages that are in the specified WebDAV location

Note: If you do not provide the channel or WebDAV URL, then the utility deletes only orphaned packages that are not defined for any channel or WebDAV URL. 

After you run the command, the utility displays a list of packages that match your deletion criteria and prompts you to confirm deletion.

- 2 Respond to the prompt to confirm deletion of the packages or to exit without deleting any packages.

Minimal Syntax for Deleting Packages

Here is the minimal syntax for deleting packages that are defined in a channel:

```
PackageCleanup
  -d expiration-date
  -ch channel-name
  -metauser Metadata-Server-username
  -metapass Metadata-Server-password
  -domain authentication-domain
```

The utility deletes all packages in the specified channel that expire before the date and time specified.

Here is the minimal syntax for deleting packages that are not defined in a channel:

```
PackageCleanup
  -d expiration-date
  -metauser Metadata-Server-username
  -metapass Metadata-Server-password
  -domain authentication-domain
```

Here is the minimal syntax for deleting packages that are defined in a WebDAV server:

```
PackageCleanup
    -url WebDAV-URL
    -username WebDAV-Server-username
    -password WebDAV-Server-password
    -d expiration-date
    -metauser Metadata-Server-username
    -metapass Metadata-Server-password
    -domain authentication-domain
```

Delete Specific Packages

To delete a specific package, specify **-package *package-name*** (or **-pkg *package-name***) along with the date. The **PACKAGE** option enables you to specify the name of the package to delete.

Change Prompt Behavior

When you run the utility command, the utility displays a list of packages that match your deletion criteria and prompts you to confirm deletion of all the packages that are listed.

You can override this default behavior in order to be prompted for each package individually.

To override the default, specify **-prompteach**. You are then prompted to delete each package that meets the deletion criteria. After each package is processed, the utility displays a final list of all packages that were selected. You can then choose to delete all of those packages or exit without deleting any packages.

You can also turn off prompting altogether by specifying **-noprompt**. When you run the utility in batch mode, you must use the **-noprompt** option (unless shell programming is provided to respond to the prompts). It is best to run with prompts when you are learning how to use the application. With prompts, you can review proper date formatting and correct package deletion candidates with the option to exit without deleting any packages.

List Packages

To obtain a list of packages, run the command and specify the **-list** option. You can also provide one of the following arguments:

- a channel name in order to list packages that are defined in a specific channel
- a WebDAV URL in order to list packages that are in the specified WebDAV location

Note: If you do not provide the channel or WebDAV URL, then the utility displays only orphaned packages that are not defined for any channel or WebDAV URL. △

The **LIST** option lists the following information for each package:

- package name
- date and time that the package was created
- date and time that the package expires

Here is the minimal syntax for listing packages that are defined in a channel:

```
PackageCleanup
    -list
    -ch channel-name
    -metauser Metadata-Server-username
    -metapass Metadata-Server-password
```

```
-domain authentication-domain
```

Here is the minimal syntax for listing packages that are not defined in a channel:

```
PackageCleanup
  -list
  -metauser Metadata-Server-username
  -metapass Metadata-Server-password
  -domain authentication-domain
```

Here is the minimal syntax for listing packages that are defined in a WebDAV server:

```
PackageCleanup
  -list
  -url WebDAV-URL
  -username WebDAV-Server-username
  -password WebDAV-Server-password
  -metauser Metadata-Server-username
  -metapass Metadata-Server-password
  -domain authentication-domain
```

Arguments

The utility supports the following arguments:

-channel | **-ch** *channel-name*

Specify the channel that contains the packages that you want to list or delete.

-deletionDate | **-d** "*expiration-date*"

Specify the expiration date and time for the packages to be deleted. You can also use this argument when you list packages. The utility deletes or lists packages that have an expiration date before the date and time that you specify. The date and time should be enclosed in quotation marks. Format: "yyyy.MM.dd at hh:mm"

-list

The utility displays a list of packages (no deletion occurs).

-metauser *Metadata-Server-username*

Specify the user name to use when connecting to the SAS Metadata Server.

-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 have no expiration date defined.

-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
```

Using JMX Tools to Manage SAS Resources

About JMX and MBeans

SAS servers implement common administrative interfaces. These interfaces enable you to perform basic administrative functions such as stopping, pausing, and resuming servers. You can also use the interfaces to monitor the health of the servers via real-time and historical metrics. Java Management Extensions (JMX) is a Java technology that supplies tools for managing and monitoring applications, system objects, devices (such as printers), and service-oriented networks. JMX managed beans, known as MBeans, have been implemented to provide a standard way of managing SAS resources.

Accessing the SAS MBeans

About Accessing the SAS MBeans

You can use any of the standard JMX monitoring tools to access the MBeans that manage SAS resources. To use these tools, you must do the following:

- 1 Enable access to the MBeans from the Web application server. See “Configure the Web Application Server to Enable JMX Client Access” on page 88.
- 2 Make sure that the appropriate authentication credentials are set. See “Set Authentication Credentials for SAS MBean Access” on page 88.

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 89.

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.

For more information about the recommended JVM arguments in the different application server environments, see the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.

Set Authentication Credentials for SAS MBean Access

The primary MBean for SAS resources is the ServerFactory MBean. This MBean is initialized with credentials that are used to connect to the SAS Metadata Server. The values of `server.admin.userid` and `server.admin.password` properties are used to initialize the MBean. These properties must identify an unrestricted user so that the SAS servers can be managed properly.

To verify that these property settings are properly defined, follow these steps:

- 1 In a SAS session, enter *metabrowse* in the command prompt and then press ENTER. The Metadata Server Configuration dialog box opens.
- 2 Enter your server connection information as follows:
 - ☐ Enter the fully qualified server name of the system on which the SAS Metadata Server resides.
 - ☐ The user name and password combination must be an unrestricted user to access the required metadata.

The Metadata Browser opens.

- 3 In SAS, select **Tools ► Options ► Explorer**.
- 4 Select the **General** tab and clear the **Metadata Browse Mode** check box.
- 5 Click **OK**.
- 6 In the Metadata Browser tree view, navigate to **Foundation ► SoftwareComponent ► Foundation Services 9.2 ► PropertySets ► Environment.Properties ► SetPropertyies ► server.admin.userid**.

- 7 In the right pane, right-click **DefaultValue** and select **Modify**. The Modify Value dialog box opens and displays a user name.
- 8 Make sure that the user name that is displayed is an unrestricted user (for example, sasadm@saspw). If not, enter a user name that is unrestricted and click **OK**.
- 9 If you changed the user name in the previous step, then you must also change the password. Follow these steps:
 - a In the tree view (under **SetProperties**), select **server.admin.password**.
 - b In the right pane, right-click **DefaultValue** and select **Modify**. The Modify Value dialog box opens and displays the password in encoded format.
 - c Change the value and click **OK**.
 For security reasons, you should enter a password that you have encoded using SAS. To obtain this encoded password, use PROC PWENCODE. For more information, open SAS Help and Documentation from the Help menu in SAS, and then search on PWENCODE.
- 10 Restart the SAS Remote Services and the Web application server.

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:
 - a Select the **Remote** tab.
 - b Enter the host on which the JVM is running, along with the port where the RMI connector was registered.
 - c You might need to specify credentials if authentication to the MBean server is required.
 - d 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 that is connected to the specified, running SAS server. The newly registered MBean can then be used to manage and monitor that particular SAS server.

Understanding How to Use the SAS MBeans

About the SAS MBeans

There are three primary MBeans provided by the SAS Web Infrastructure Platform for managing and monitoring SAS resources:

- ServerFactory MBean
- Spawner MBean
- Server MBean

The following sections describe these MBeans.

ServerFactory MBean

The ServerFactory MBean is the starting point for managing SAS servers. This MBean is registered during deployment of the SAS Web Infrastructure Platform and is named as follows:

```
com.sas.services:type=ServerFactory
```

During initialization, the ServerFactory MBean connects to the SAS Metadata Server. This enables the MBean to list all SAS servers defined in the metadata. The MBean can then be used to register additional MBeans that enable the running servers to be managed and monitored directly. The ServerFactory MBean does not have any attributes, but supports three operations:

`listDefinedServers()`

provides a list of SAS IOM servers that are defined in the Metadata Server.

Information that is returned for each defined server includes the server name, host, port, and server type. To begin actively managing a server, specify the name of the server on the `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 Table 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"

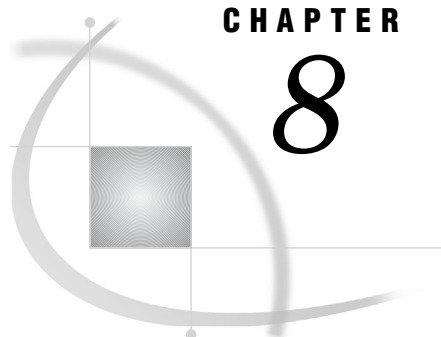
com.sas.services:type=Server, serverType=StoredProcess, logicalServer=
    "LogicalServerName", name="Server Name",
    instanceid="Unique instance ID"

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

8

Administering SAS Web Applications

<i>Using the SAS Deployment Manager</i>	94
<i>Rebuilding the SAS Web Applications</i>	94
<i>When to Rebuild the SAS Web Applications</i>	95
<i>Rebuild One or More Web Applications</i>	95
<i>Names of the EAR Files</i>	97
<i>Redeploying the SAS Web Applications</i>	97
<i>Redeploying EAR Files to a Single Server or Multiple Servers</i>	97
<i>JBoss</i>	98
<i>WebLogic</i>	98
<i>Stop and Delete All SAS Applications</i>	99
<i>Shut Down the Managed Servers</i>	99
<i>Reinstall the SAS Applications</i>	99
<i>Restart the Managed Servers</i>	100
<i>Start the SAS Applications</i>	100
<i>WebSphere</i>	101
<i>Reconfiguring the Web Application Server</i>	101
<i>Deploying SAS OnlineDoc Manually for the Web</i>	102
<i>Manually Deploy SAS OnlineDoc for the Web</i>	102
<i>WebSphere Class Loader Settings</i>	103
<i>Access SAS OnlineDoc for the Web</i>	103
<i>Working with Exploded EAR Files in a Development Environment</i>	103
<i>Administering Logging for SAS Web Applications</i>	105
<i>Administering Logging Service Settings for Web Applications</i>	105
<i>Logging for SAS Web Applications</i>	105
<i>Change the Location of the Log Files</i>	107
<i>Change the Logging Levels</i>	107
<i>Understanding How the Web Applications Provide a Logging Context</i>	108
<i>Configuring Auditing for SAS Web Applications</i>	109
<i>Overview of Auditing</i>	109
<i>Configure the Audit Log File</i>	109
<i>Auditing That Uses Relational Tables</i>	110
<i>Enable Auditing for User Authentication Actions</i>	110
<i>Configuring a Custom Logoff Message for Web Application Users</i>	111
<i>Configuring the HTTP Session Time-out Interval</i>	112
<i>Modifying a Session Time-out Interval</i>	112
<i>Configuring the Display of a Warning Message for Inactive User Sessions</i>	114
<i>Configuring the Display of a Warning Message for Inactive User Sessions</i>	115

Using the SAS Deployment Manager

The SAS Deployment Manager enables you to accomplish the following tasks:

- Update passwords for the service accounts that were configured when you ran the SAS Deployment Wizard. For information about how to update passwords by using the SAS Deployment Manager, see the *SAS Intelligence Platform: Security Administration Guide*.
- Rebuild Web applications. You can rebuild Web applications that have previously been configured but whose configuration has changed. This option rebuilds the Web application based on the current configuration. See “Rebuilding the SAS Web Applications” on page 94.
- Remove one or more components of a SAS Intelligence Platform configuration from your environment. This option enables you to remove the configuration for an application that you are no longer using or that you are moving to another machine. You can then use the SAS Deployment Wizard to reinstall or reconfigure the application. For details, see “Removing a SAS Configuration” in the *SAS Intelligence Platform: Installation and Configuration Guide*.

Note the following about removing a configuration:

- Installed products are not removed.
- If you remove the configuration for the SAS Information Delivery Portal, do not select the **Remove all User Content** option unless you have made a backup copy of the content repository. If you choose this option, you must re-create the content later from your backup. When you choose to remove portal content, all pages, portlets, and other items created by the users are removed.
- If you remove the configuration for the Web Infrastructure Platform, the contents of the SAS Content Server repository (located in the *SAS-configuration-directory*\Levn\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.
- Renew your software license for some SAS solutions that depend on a SAS middle tier. For details, see “Renewing Your Software License for SAS Solutions” in the *SAS Intelligence Platform: Installation and Configuration Guide*.
- Change the host names (including the network domains to which they belong) of server machines in your deployment. For details, see “Changing Host Names” in the *SAS Intelligence Platform: System Administration Guide*.
- Configure Updated SAS Products. For details, see “Configure Updated SAS Products” in the *SAS Intelligence Platform: Installation and Configuration Guide*.


Access the SAS Deployment Manager by running the **config.exe** file in the *SAS-installation-directory*\SASDeploymentManager\9.2 directory.

Rebuilding the SAS Web Applications

When to Rebuild the SAS Web Applications

The Rebuild Web Applications feature of 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 application's Java security configuration.
- 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, you place the file for the EAR or the WAR in the *SAS-configuration-directory*
`\Lev1\Web\Common\SASServer1\SASStoredProcess9.2\CustomContent\ears\sas.storedprocess\input` directory. Then, use the SAS Deployment Manager to rebuild the SAS Stored Process application.
- If custom portal content, such as a custom portlet, is created for your organization, rebuild the SAS Information Delivery Portal. For more information, see [LINK TBD](#).
- Rebuild SAS Help Viewer Metadata Configuration after your initial deployment if you later install or upgrade a SAS Web application that uses SAS Help Viewer Metadata Configuration. (SAS Help Viewer Metadata Configuration combines SAS Help Viewer for the Web software with various Help content into its EAR file.)
 The following Web applications use SAS Help Viewer Metadata Configuration:
 - 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 the second maintenance release for SAS 9.2 or later, you must use the SAS Deployment Manager to rebuild the EAR files for all Web applications that were updated at your site. Because the EAR files are rebuilt, you might lose any customizations that you added to the EAR files after initial deployment. For more information, see *Maintenance Planning for SAS 9.2*.

Rebuild One or More Web Applications

In SAS 9.2, the **Rebuild Web Applications** option in the SAS Deployment Manager enables you to rebuild a subset of the EAR files. In the second maintenance release for SAS 9.2 and later, when you rebuild Web applications by using the SAS Deployment Manager, the EAR files for the Web applications are automatically placed in two directories:

- *SAS-configuration-directory*\Lev1\Web\Staging. The approximate size of the collection of EAR files for EBI is 2 GB.

- *SAS-configuration-directory\Lev1\Web\Staging\exploded*. The approximate size of the Exploded directory is 2 GB. The size of the Exploded directory is similar to the size of the EAR files in the *SAS-configuration-directory\Lev1\Web\Staging* directory.

In the second maintenance release for SAS 9.2 and later, the exploded EAR files are always recreated whenever you rebuild Web applications in the SAS Deployment Manager. Previously, in SAS 9.2, this was not the case.

Note: If you are using the second maintenance release for SAS 9.2 or later, and you want to save disk space, you can delete the unwanted directories below the *SAS-configuration-directory\Lev1\Web\Staging\exploded* directory. However, if you are using WebLogic as your Web application server, do not delete the exploded copy of the **sas.wip.services9.2** EAR file. △

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

- 1 Verify the status of the Web application server.
 - Shut down the JBoss application server. This is required because the SAS Deployment Manager modifies the existing configuration for JBoss when it rebuilds the Web applications.
 - The WebLogic Administration Server and Node Manager must be running when you rebuild Web applications.
 - Shut down the WebSphere application server, but leave the dmgr and nodeagent running.
- 2 Make sure that the SAS Metadata Server is running.
- 3 In Windows, access the SAS Deployment Manager by running the **config.exe** file in the *SAS-installation-directory\SASDeploymentManager\9.2* directory. In a UNIX or z/OS environment, run the **config.sh** file.
- 4 Select the **Rebuild Web Applications** option and click **Next**.
- 5 In the next dialog box, specify the configuration directory and the level (for example, Lev1) in which you want to rebuild the applications.
- 6 In the next dialog box, enter the user ID and password for an unrestricted administrative user (for example, sasadm@saspw).
- 7 In the next dialog box, select the check boxes for the applications that you want to rebuild.
- 8 In the next dialog box, click **Start** to start the rebuild. The SAS Deployment Manager builds the EAR files for the selected applications. For the names and location of the EAR files, see “Names of the EAR Files” on page 97.
- 9 Deploy the EAR files to the Web application server. See “Redeploying the SAS Web Applications” on page 97.
- 10 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 as an EAR in a 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.
 - If SAS Web Application Themes is exploded and deployed in an HTTP server (such as Apache HTTP Server), then the Web application server does not need to be restarted based on any theme changes.

For all other Web applications, you do not need to stop and restart the Web application server.

Names of the EAR Files

When the Web applications are rebuilt, the SAS Deployment Manager places the EAR files in the following directories: *SAS-configuration-directory\Lev1\Web\Staging* and *SAS-configuration-directory\Lev1\Web\Staging\exploded* directory.

Depending on the software that you have installed, the following EAR files are available:

Table 8.1 List of EAR Files

Application	EAR File
SAS BI Dashboard	sas.bidashboard4.3.ear
SAS BI Portlets ¹	sas.biportlets4.3.ear
SAS Package Viewer (ships with SAS Information Delivery Portal)	sas.packageviewer4.3.ear
SAS Information Delivery Portal	sas.portal4.3.ear
SAS Shared Services	sas.shared9.2.ear
SAS Stored Process	sas.storedprocess9.2.ear
SAS Web Application Themes	sas.themes.ear
Flex Themes	sas.flexthemes.ear?
SAS Help Viewer Metadata Configuration	sas.webdocmd9.2.ear
SAS Web Report Studio	sas.webreportstudio4.3.ear
SAS Web Infrastructure Platform Applications	sas.wip.apps9.2.ear
SAS Content Server	sas.wip.scs9.2.ear
SAS Web Infrastructure Platform Services	sas.wip.services9.2.ear

¹ Available in the October 2009 Release and later.

Redeploying the SAS Web Applications

Redeploying EAR Files to a Single Server or Multiple Servers

When the SAS Deployment Manager rebuilds SAS Web applications, the generated 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 SAS 9.2 installation included multiple servers (for example, SASServer1 and SASServer2) that were configured with different Web Applications.

If you have multiple managed servers that were installed and configured by the SAS Deployment Wizard in your environment, make a note of the names of the servers and the Web applications that reside on those servers. For example, if you have six applications located on SASServer1 and three Web applications located on SASServer 2, make a list of the applications residing on each of these servers. Alternatively, you can refer to your **Instructions.html** file, which specifies the following:

- the list of SAS Web applications to be deployed
- the location of the applications
- the target server where each application should be deployed

When you manually redeploy the SAS Web applications, you can refer to your list or the **Instructions.html** file, to ensure that you redeploy the EAR files to the correct server.

JBoss

To redeploy a SAS Web application to JBoss, follow these steps:

- 1 Create a directory where you can store unused EAR files. Do not create this directory below the deployment directory or below the *JBoss-installation-directory\server\SASServer1\deploy_sas*. Instead, choose a different location to store these unused EAR files.
- 2 Remove the unused application EAR files from the current deployment directory, and place them in the directory you created for unused EAR files. Typically, the current deployment directory is *JBoss-installation-directory\server\SASServer1\deploy_sas*.
- 3 If applicable, repeat the previous step for each JBoss server.
- 4 Copy all of the new EAR files from the *SAS-configuration-directory\Lev1\Web\Staging* directory to the *JBoss-installation-directory\server\SASServer1\deploy_sas* directory. For a list of EAR files, see “Names of the EAR Files” on page 97. If restrictive policies were implemented at your site, you must deploy exploded EAR files instead of unexploded EAR archives. The exploded EAR files reside in the exploded location within the *SAS-configuration-directory \Lev1\Web\Staging\exploded* directory. For instructions about exploding EAR files, see “Working with Exploded EAR Files in a Development Environment” on page 103. If you are using the second maintenance release for SAS 9.2 or later, the process of rebuilding Web applications in the SAS Deployment Manager automatically results in exploding of the EAR files.
- 5 Repeat Step 4 for any additional JBoss application servers (for example, SASServer2) that have SAS applications deployed.
- 6 If necessary, restart JBoss. If the JBoss server has been configured for hot deployment, then you do not need to restart JBoss. Step 1 undeploys the application and step 2 redeploys it. However, if hot deployment is not enabled, then you must restart JBoss.

For complete deployment instructions, see the JBoss documentation at <http://www.jboss.org/docs>.

WebLogic

There are a number of ways to redeploy applications in WebLogic. To redeploy SAS applications to WebLogic by using the WebLogic Administration Console, follow these steps:

- 1 Stop and delete all SAS applications. See “Stop and Delete All SAS Applications” on page 99.
- 2 Shut down the SAS managed servers. See “Shut Down the Managed Servers” on page 99.
- 3 Reinstall the SAS applications. See “Reinstall the SAS Applications” on page 99.

- 4 Restart the managed servers. See “Restart the Managed Servers” on page 100.
- 5 Start the SAS applications. See “Start the SAS Applications” on page 100.

For complete deployment instructions about WebLogic, see the WebLogic documentation at <http://www.oracle.com/technology/documentation/index.html>.

Stop and Delete All SAS Applications

To stop and delete all SAS applications, follow these steps:

- 1 In the WebLogic Administration Console, select **Deployments** in the **Domain Structure** panel.
- 2 In the **Deployments** panel, select all applications by selecting the check box next to **Name**.
- 3 On the menu for **Stop**, select **Force Stop Now**.
- 4 In the **Summary of Deployments** tab, select **Yes**.
- 5 Wait until all applications display in Prepared state. Refresh the view as needed until all applications reach the Prepared state.
- 6 When the managed servers are running, delete all applications by selecting **Lock and Edit** in the **Change Center** panel.
- 7 In the **Deployments** panel, select all applications by selecting the check box next to **Name**.
- 8 Click **Delete**.
- 9 In the **Delete Application Assistant** panel, select **Yes**.
- 10 When the message “Selected deployments were deleted,” is displayed, select **Activate Changes** in the **Delete Application Assistant** panel.

Shut Down the Managed Servers

It is recommended that you shut down the SAS Managed Servers while the WebLogic Admin Server is running.

To shut down the SAS managed servers while the WebLogic Admin Server running (recommended), follow these steps:

- 1 In the **Domain Structure** panel within the WebLogic Administration Console, select **Environment ► Servers in the Domain Structure**.
- 2 Leave the Admin server running; do not stop it. Then, for each SAS server, complete the following steps:
 - a In the Summary of Servers table, select the server (for example, SASServer1).
 - b Select the **Control** tab.
 - c From the menu for **Shutdown**, select **Force Shutdown Now**.
 - d Click **Yes** to the prompt **Forcibly Shutdown Servers**.
 - e Verify that the server has been shut down.

Reinstall the SAS Applications

With the exception of the **sas.wip.services9.2** EAR file, all EAR files should be deployed from the **SAS-configuration-directory\Lev1\Web\Staging** directory. The **sas.wip.services9.2** EAR file should be deployed from the **SAS-configuration-directory\Lev1\Web\Staging\exploded** directory.

To redeploy EAR files and install SAS applications, follow these steps:

- 1 Locate the **Instructions.html** file in the **SAS-configuration-directory\Lev1\Documents** directory, and make a note of the

list of SAS applications and their associated servers. This information is available in the Web Application Server section. You will need this information when you redeploy and install the EAR files.

- 2 In the WebLogic Administration Console, within the **Domain Structure** panel, select **Deployments**.
- 3 Click **Lock and Edit** in the Change Center panel.
- 4 In the **Summary of Deployments** panel, click **Install**.
- 5 In the **Install Application Assistant** panel, browse and navigate to the *SAS-configuration-directory\Lev1\Web\Staging* directory.
- 6 In the **Install Application Assistant** panel, under Locate deployment to install and prepare for deployment, select an EAR file and click **Next**.
- 7 In the options available for Choose targeting style, retain the default (Install this deployment as an application), and click **Next**.
- 8 See the **Instructions.html** file to identify the server associated with the EAR file that you are deploying. Typically, for most SAS applications, the target server is SASServer1.
- 9 In the **Install Application Assistant** panel, under **Select deployment targets**, select the target server and click **Next**. Typically, SAS applications are deployed to SASServer1.
- 10 Under **Optional Settings, General**, enter a name for the EAR file or the directory for this deployment.
- 11 If the Admin server and the managed server are on the same machine, under **Source accessibility**, select **I will make the deployment accessible from the following location** and click **Next**. Note that this is not a staged mode.
- 12 Under **Review your choices and click Finish**, select **No, I will review the configuration later**, and click **Finish**.
- 13 In the **Change Center** panel, select **Activate Changes**. The application should display in a New state.
- 14 Repeat these steps to redeploy the other EAR files and install the SAS applications.

Restart the Managed Servers

To restart the managed servers in WebLogic, follow these steps:

- 1 In the **Domain Structure** panel within the WebLogic Administration Console, select **Environment ► Servers**.
- 2 In the **Change Center** panel, select **Activate Changes**.
- 3 On the Settings page, select the **Control** tab.
- 4 In the Servers table under **Summary of Servers**, click on the server name (for example, SASServer1).
- 5 In the Server Status table, click **Start**.
- 6 In the **Server Life Cycle Assistant** panel, click **Yes**.
- 7 In the Server Status table, verify that the task has been completed.
- 8 If applicable, repeat these steps for other managed servers.

Start the SAS Applications

To start the SAS applications in WebLogic, follow these steps:

- 1 In the **Domain Structure** panel within the WebLogic Administration Console, select **Deployments**. All SAS applications should display in a Prepared state.

- 2 In the **Deployments** panel, select the check box next to **Name**. All applications are selected.
- 3 From the **Start** menu, select **Servicing All Requests**.
- 4 In the **SAS Application Assistant**, select **Yes**.

WebSphere

There are two ways to redeploy a SAS Web application to WebSphere. In the first method, you can update an installed application and select **Replace the entire application**. With this method, you can maintain all of the application settings, such as the class loader policy and mode for the EAR and WAR modules. In the second method, you undeploy and redeploy each application individually until all of the rebuilt applications have been redeployed.

Although you can redeploy the EAR files in any order of your choice, it is highly recommended that you follow the sequence of EAR files specified for WebSphere. See “Deploying EAR Files in the Correct Order” on page 19.

To redeploy a SAS Web application to WebSphere by undeploying and redeploying each application individually, follow these steps:

- 1 Uninstall and then reinstall the application. For instructions, see the WebSphere documentation.
- 2 In the WebSphere administrative console, select **Applications ► Enterprise Applications**. Then select the SAS Web application that you are redeploying.
- 3 Click **Class loading and update detection**.
- 4 In the **Polling interval for updated files** field, enter 3.
- 5 For the class loader order, select the **Classes loaded with application class loader first** radio button.
- 6 Leave the WAR class loader policy set to **Class loader for each WAR file in application**.
- 7 Click **OK**.
- 8 Click **Manage Modules**.
- 9 For each module (WAR file), click the WAR file link. Then select **Classes loaded with application class loader first** from the **Class loader order** list box.
- 10 After you have performed the previous step for each WAR file, click **OK**.
- 11 Save your changes.
- 12 Perform a full resynchronization of the dmgr and nodeagent WebSphere servers. This action ensures the WAS Master Repository and Node Repository are updated and synchronized. Follow these steps:
 - a In the WebSphere administrative console, select **Administration ► Nodes**.
 - b Select the check box for the application server node.
 - c Click **Full Resynchronize**.

When you have completed these instructions, you can restart the application. (For the proper start-up order of the SAS Web applications, see “Deploying EAR Files in the Correct Order” on page 19.)

For complete deployment instructions, see the WebSphere documentation at <http://www.ibm.com/support/documentation/us/en>.

Reconfiguring the Web Application Server

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

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

It is important to reconfigure your Web application server in the same manner that it was configured initially. If you manually configured the Web application server when you deployed SAS 9.2, then configure it again manually. If the SAS Deployment Wizard automatically configured your Web application server, then choose the automatic configuration option.

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

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

Deploying SAS OnlineDoc Manually for the Web

Manually Deploy SAS OnlineDoc for the Web

Your installation might include SAS OnlineDoc for the Web, which is an online library of reference documentation for the SAS System. You must manually deploy SAS OnlineDoc for the Web in your Web application server. Manual deployment is required even if you selected the automatic deployment option in SAS Deployment Wizard.

When you installed SAS OnlineDoc for the Web using the SAS Deployment Wizard, you encountered a dialog box that asked whether to create the application’s Enterprise Archive (EAR) file. The EAR file that is created is located here:

SAS-installation-
directory\Documentation\9.2\onlinedocweb\sas.onlinedocweb9.2.ear

To manually deploy SAS OnlineDoc for the Web, follow these steps:

- 1 Make sure that you have configured your Web application server as follows:
 - If you performed a planned installation and chose automatic configuration of the Web application server, then the application server is already configured.
 - If you performed a planned installation and chose manual configuration of the Web application server, then follow the instructions in your **Instructions.html** file (located in the SAS configuration directory).
 - If you performed a nonplanned installation, then see the instructions for your Web application server on the SAS third-party Web site at **<http://support.sas.com/resources/thirdpartysupport/v92>**.
- 2 Deploy the **sas.onlinedocweb9.2.ear** file to your Web application server. For deployment instructions, see the documentation that is provided for your Web application server.

Note: If you are deploying to WebSphere, then see “WebSphere Class Loader Settings” on page 103. △

If you later need to remove SAS OnlineDoc for the Web, undeploy the application from your Web application server like you would any application. Then, uninstall the application (for example, by using the Add or Remove utility on Windows systems).

WebSphere Class Loader Settings

If you are deploying to WebSphere, then you need to change the class loader settings. After you have added the SAS OnlineDoc for the Web application to WebSphere, follow these steps:

- 1 In the WebSphere administrative console, select **Applications ► Enterprise Applications**. Then select **SAS Online Doc for the Web**.
- 2 Click **Class loading and update detection**.
- 3 In the **Polling interval for updated files** field, enter 3.
- 4 For the class loader order, select the **Classes loaded with application class loader first** radio button.
- 5 Leave the WAR class loader policy set to **Class loader for each WAR file in application**.
- 6 Click **OK**.
- 7 Click **Manage Modules**.
- 8 For each module (WAR file) click the WAR file link. Then select **Classes loaded with application class loader first** from the **Class loader order** list box.
- 9 After you have performed the previous step for each WAR file, click **OK**.
- 10 Save your changes.
- 11 Perform a full resynchronization of the dmgr and nodeagent WebSphere servers. This action ensures the WAS Master Repository and Node Repository are updated and synchronized. Follow these steps:
 - a In the WebSphere administrative console, select **Administration ► Nodes**.
 - b Select the check box for the application server node.
 - c Click **Full Resynchronize**.

Access SAS OnlineDoc for the Web

After you deploy SAS OnlineDoc for the Web, users can access the application by pointing their browser to the following URL:

`http://server:port/SASOnlineDoc/oldoc/ui/contents?selectedProduct=ONLINEDOCWEB¯oVersion=9.2&selectedTopic=doccommon.hlp/online_doc_main.htm`

In the URL, substitute the server name and port number of your Web application server.

Working with Exploded EAR Files in a Development Environment

In the second maintenance release for SAS 9.2 and later, the **Rebuild Web Applications** option in the SAS Deployment Manager automatically explodes all EAR

files and places them in the *SAS-configuration-directory\Lev1\Web\Staging\exploded* directory. See “When to Rebuild the SAS Web Applications” on page 95. In most cases, it is the EAR file itself that is deployed to the Web application server. The exploded copies are typically provided for your convenience.

It can be useful to work with exploded EAR files when you want to debug or develop new JavaServer Pages (JSP) in a Web application. If you have deleted the exploded copy, or need to explode an EAR file manually, you can extract the contents for the EAR file into a directory. Second, recursively extract the contents of each WAR file found in the EAR file. Each WAR file needs to be exploded into the same parent directory where it resides.

There are multiple ways to explode EAR files into their constituent parts. EAR files can be exploded manually using command line tools or by using a script. The following task outlines the general steps involved in exploding EAR files.

- 1 Change the working directory to the location where the EAR file resides.
- 2 Create a subdirectory called **exploded**.
- 3 Change the working directory to the new subdirectory **exploded**.
- 4 Create a subdirectory that matches the name of the EAR file. For example:
filename.ear.
- 5 Change the working directory to the *filename.ear* subdirectory.
- 6 Use the **jar** command to explode the EAR file located in the working directory that is two levels above the current directory.
- 7 Create a subdirectory called **temp_dir**.
- 8 For each WAR file that is extracted into the *filename.ear* subdirectory, complete the following steps:
 - a Move the WAR file to the **temp_dir** subdirectory.
 - b Create a subdirectory in the *filename.ear* directory, and make sure that the name of the subdirectory matches the name of the WAR file (for example, *filename.war*).
 - c Change the working directory to the *filename.war* directory.
 - d Use the **jar** command to explode the WAR file from the **temp_dir** directory.
 - e Change the working directory to the *filename.war* directory.
 - f Repeat these steps until you have completed extracting all WAR files.

The following example shows how to explode EAR files manually in a UNIX and z/OS environment by using pseudocode that you replace with the applicable syntax:

```
cd [location of ear]
mkdir exploded
cd exploded
mkdir [earname]
cd [earname]
jar -xvf ../../[earname]
mkdir temp_dir
for each [warname]=*.war file
    move [warname] temp_dir
    mkdir [warname]
    cd [warname]
    jar -xvf ../temp_dir/[warname]
    cd ..
done
delete temp_dir
cd ../../
```

Deployment of modified EAR files from an exploded directory varies with the Web application server as follows:

JBoss

JBoss has direct support for deployment of exploded EAR files. To deploy an exploded EAR file, move the exploded directory to the deployment directory (**deploy_sas** under the JBoss server). Because the exploded directory must be named the same as the original EAR file, the original EAR file must be removed from the deployment directory.

WebLogic

WebLogic has direct support for deployment of exploded EAR files. To deploy an exploded EAR file using the administrative console, select the full path to the exploded EAR.

WebSphere

WebSphere explodes deployed EAR files on its own. Deployed files must be either EAR files or WAR files.

Starting with version 6.1, WebSphere has the ability to update an existing deployed application with individual files or modules. By selecting the full path to the JSP or WAR directory, individual components of a modified, exploded EAR file can be used in a deployed application. You can use the administrative console to update a deployed application.

When you are ready to deploy your changes, use the SAS Deployment Manager to rebuild the EAR file. See “Rebuilding the SAS Web Applications” on page 94.

Administering Logging for SAS Web Applications

Administering Logging Service Settings for Web Applications

Logging for SAS Web Applications

The SAS 9.2 Intelligence Platform uses a standard logging facility to perform logging for SAS servers. In SAS Management Console, the Logging Service Configuration dialog box enables you to accomplish several tasks for SAS Web applications:

- ☐ Edit existing output types or create a new output type.
- ☐ Save an output type with a new name.
- ☐ Modify the layout pattern for the log message.
- ☐ Specify log event output to Console, File, Socket, or ARM.
- ☐ Define new outputs.
- ☐ Track user logons. You can monitor usage patterns by logging activity for SAS Web application logons.
- ☐ Change the logging levels.

For an overview and guidelines about logging, see “Administering Logging for SAS Servers” in the *SAS Intelligence Platform: System Administration Guide*.

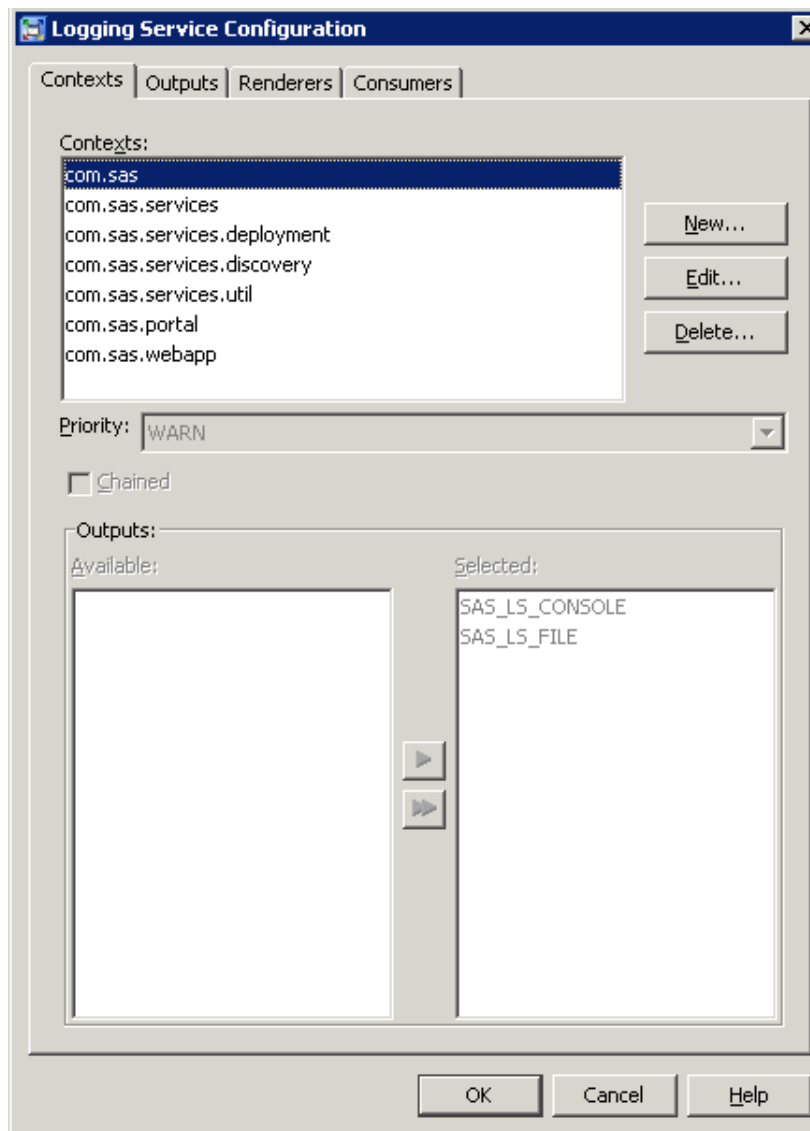
The logging configuration for each application is controlled independently. For example, you can choose an application, such as SAS Web Report Studio or SAS Information Delivery Portal, and view or set logging levels for the application.

To see the logging configuration for a SAS application, follow these steps:

- 1 Log on to SAS Management Console.
- 2 On the **Plug-ins** tab, navigate to **Environment Management ► Foundation Services Manager ► <Application> ► Core ► Logging Service**.
- 3 Right-click **Logging Service** and select **Properties**. The Logging Service Properties dialog box opens.
- 4 Select the **Service Configuration** tab and then click **Configuration**. The Logging Service Configuration dialog box opens for the application.

The following display shows an example Logging Service Configuration dialog box.

Display 8.1 Example Logging Service Configuration Dialog Box



For information about how to modify and customize the logging, see “Modifying Service Configurations” in the *SAS Foundation Services: Administrator’s Guide*.

By default, the Logging Service Configuration dialog box in SAS Management Console contains two output types:

- ☐ SAS_LS_CONSOLE
- ☐ SAS_LS_FILE

One is a console output and the other is a file output.

Message logging is accomplished with a logging context. A logging context is usually the fully qualified class name of the class where the logging message originated. In the SAS Management Console, logging contexts are created or edited in the Logging Service Context window (accessed through the Logging Service Configuration dialog box).

The following logging contexts are common to all SAS applications:

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

The five logging contexts are five packages that represent different locations in SAS software. These logging contexts should not be edited, but you can remove them or add additional logging contexts.

For detailed information about the logging facility, see *SAS Logging: Configuration and Programming Reference*.

Change the Location of the Log Files

By default, log files are stored in the *SAS-configuration-directory\Lev\Web\Logs* directory. Typically, you change the directory path for the location of the log files, if you have a clustered environment.


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

- 1 Log on to SAS Management Console.
- 2 On the **Plug-ins** tab, navigate to **Environment Management ► Foundation Services Manager ► <Application> ► Core ► Logging Service**.
- 3 Right-click **Logging Service** and select **Properties**. The Logging Service Properties dialog box opens.
- 4 Select the **Service Configuration** tab and then click **Configuration**. The Logging Service Configuration dialog box opens for the application.

Change the Logging Levels

In the SAS Management Console, the Logging Service Configuration window enables you to configure the settings for logging. The log has five levels of detail: DEBUG, INFO, WARN, ERROR, and FATAL. Enabling a level also enables the less detailed levels above the selected level. By default, the level is set to WARN, which means that only 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 logging 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.

For an explanation of all warnings, see “Administering Logging for SAS Servers” in the *SAS Intelligence Platform: System Administration Guide*.

To change the types of messages that are stored in the log, choose the priority level for the appropriate logging context in the Logging Context dialog box. Here is a brief description of each type of logging:

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 will probably cause the application to end abnormally.

To enable a different level of log messages for an application, follow these steps:

- 1 Log on to SAS Management Console.
- 2 On the **Plug-ins** tab, navigate to **Environment Management ► Foundation Services Manager ► <Application> ► Core ► Logging Service**.
- 3 Right-click **Logging Service** and select **Properties**. The Logging Service Properties dialog box opens.
- 4 Select the **Service Configuration** tab and then click **Configuration**. The Logging Service Configuration dialog box opens for the application.
- 5 On the **Contexts** tab, select a context and click **Edit**. The Edit Logging Service Context dialog box opens.
- 6 Select the desired level of warning from the **Priority** drop-down list box.
- 7 Click **OK** to exit from each window.
- 8 To enforce the changes you have made, restart the SAS Web application.

Understanding How the Web Applications Provide a Logging Context

The SAS Web Infrastructure Platform has implemented a framework for providing separate logging contexts for the individual SAS Web applications. It is useful to understand how SAS implements individual logging contexts in case you need to modify any of the related configuration files. In addition, you might want to create a logging context for Web applications that are developed at your site.

The SAS Web applications implement this capability as follows:

- The Web application’s **web.xml** file contains the following tag element:


```
<listener>
  <listener-class>com.sas.svcs.logging.LoggingContextListener
</listener-class>
</listener>
```

The listener element must directly follow the filter and filter-mapping elements and directly precede the servlet element. This listener is added as the first listener within the <web-app> element if there are multiple listeners.

The **web.xml** file also contains this tag:

```
<context-param>
  <param-name>log4j-config-name-prefix</param-name>
  <param-value>YourWebAppName</param-value>
</context-param>
```

In the tag, *YourWebAppName* should be a name that contain no spaces or special characters (for example, SASWebReportStudio). This name is prepended to **-log4j.xml** to form the filename of the application-specific log4j configuration file.

- Web application servers have a property defined on their command line (com.sas.log.config.url) that points to the directory where the log4j config files are located. The framework within the SAS Web Infrastructure Platform uses this property to locate a particular Web applications's log4j configuration. The framework then loads the configuration into a private logging context for the Web application. Any loggers that the Web application obtains from a locally deployed Logging Service will also share this same private logging context. Check your Web application server start-up script or configuration file to determine the appropriate directory location.

Here is an example directory for a default deployment:

SAS-configuration-directory\Levn\Web\Common\LogConfig

The name of the log4j file has this form:

YourWebAppName-log4j.xml

where *YourWebAppName* is the string that was provided in the <context-param> element of the **web.xml** file.

- A copy of the **sas.svcs.common.jar**, which contains the com.sas.svcs.logging.LoggingContextListener class, must be in the Web application's CLASSPATH. This is accomplished by placing the JAR file in the Web application WAR file's **WEB_INF/lib** directory.

Configuring Auditing for SAS Web Applications

Overview of Auditing

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

There are two primary ways in which auditing occurs:

- basic auditing with records that are logged to a file
- more detailed auditing with records that are written to a relational database

All SAS environments can be configured to use the basic auditing via logging that is available through services in the SAS Web Infrastructure Platform. Alternatively, those SAS environments that include SAS Shared Services are configured by default to use the detailed auditing via a relational database.

Configure the Audit Log File

The services in the SAS Web Infrastructure Platform provide default auditing using Apache log4j standards. The context com.sas.svcs.audit can be configured to use valid appenders. The context can also apply standard logging configuration rules that are

defined for the SAS Web Infrastructure Platform Services application (sas.wip.services9.2.ear/sas.wip.services.war).

You can change the logging configuration for the SASWIPServices9.2 Local Services definition in the Foundation Services Manager plug-in in SAS Management Console. For more information about setting up logging contexts for Web applications, see “Administering Logging for SAS Web Applications” on page 105.

To set up the auditing context, follow these steps:

- 1 Log on to SAS Management Console.
- 2 On the **Plug-ins** tab, navigate to **Environment Management ► Foundation Services Manager ► SASWIPServices9.2 Local Services ► Core ► Logging Service**.
- 3 Right-click **Logging Service** and select **Properties**. The Logging Service Properties dialog box opens.
- 4 Select the **Service Configuration** tab and then click **Configuration**. The Logging Service Configuration dialog box opens for the application.
- 5 On the **Contexts** tab, click **New** and specify a logging context with the name *com.sas.svcs.audit*.
- 6 From the **Priority** list box, specify the logging level that you want. The log4j auditing occurs at the INFO level. Therefore, you must specify either INFO or DEBUG in order to get audit messages.
- 7 Specify the outputs that you want for the audit messages by moving one or more outputs from the **Available** list box to the **Selected** list box.

Note: You can create a new output by selecting the **Outputs** tab in the Logging Service Configuration dialog box and clicking **New**. Then enter the information for the new output. Help is available from the dialog box. △

- 8 Click **OK** to exit from each window.
- 9 To enforce the changes that you have made, restart the SAS Web Infrastructure Platform Services application.

Auditing That Uses Relational Tables

SAS environments that include SAS Shared Services provide more sophisticated audit retrieval and reporting capabilities, particularly for those applications with more regulatory or compliance requirements. In the relational database configuration, audit records are stored in a database named SharedServices. The records are written in three relational tables: SharedServices.SAS_AUDIT, SharedServices.ACTION_EXECUTOR, and SharedServices.SAS_AUDIT_ENTRY.

Enable Auditing for User Authentication Actions

The core auditing features in the middle tier can be configured to record information about successful user logon actions, failed logon attempts such as those attributed to incorrect credentials, and user logoff actions. By default, auditing of user authentication and session termination is disabled in a standard configuration.

To enable auditing of user authentication, follow these steps:

- 1 Open the following file in a text editor:

SAS-installation-
directory\SASWebInfrastructurePlatform\9.2\Static\wars
sas.wip.services\WEB-INF\spring-config\aop-config.xml

- 2 Remove the comments from the following lines:<!--

```
<bean class="com.sas.svcs.authentication.impl.aop.FailedLogonAuditAdvice">
<property name="auditRecorder" ref="auditRecorder" />
</bean>
<bean class="com.sas.svcs.authentication.impl.aop.SuccessfulLogoffAuditAdvice">
<property name="auditRecorder" ref="auditRecorder" />
</bean>
<bean class="com.sas.svcs.authentication.impl.aop.SuccessfulLogonAuditAdvice">
<property name="auditRecorder" ref="auditRecorder" />
</bean>
-->
```
- 3 Save your changes and rebuild the EAR file for SAS Web Infrastructure Platform Services (**sas.wip.services9.2.ear**). See “Rebuilding the SAS Web Applications” on page 94.

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. △

Configuring a Custom Logoff Message for Web Application Users

You can configure a customized message that will display when users of SAS Web applications log off.

Edit, customize, and save the **logoff_custom.jsp** file located in the **C:\Program Files\SAS\SASWebInfrastructurePlatform\9.2\Static\wars\sas.vcs.logon** directory. On UNIX, the **logoff_custom.jsp** file is located in the **SAS_HOME/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.logon** directory. This file is included as part of an HTML page. Therefore, it should contain valid HTML code.

To enable the display of a custom message when users of a SAS Web application log off from their application, follow these steps:

- 1 Log on to SAS Management Console.
- 2 On the **Plug-ins** tab, select **Application Management ► SAS Application Infrastructure**, and right-click on **Properties**.
- 3 In the SAS Application Infrastructure dialog box, select the **Settings** tab.
- 4 In the **Display custom logoff message** field, select **Yes** and click **OK**.
- 5 Exit from SAS Management Console.
- 6 Use the SAS Deployment Manager to rebuild the SAS Web applications for the SAS Web Infrastructure Platform. For instructions, see “When to Rebuild the SAS Web Applications” on page 95.
- 7 Redeploy the EAR files that were rebuilt in the SAS Deployment Manager. The **sas.wip.apps9.2** EAR file must be redeployed. In addition, it is recommended that you redeploy **sas.wip.services9.2**, **sas.wip.scs9.2.ear**, and **sas.storedprocess9.2.ear** files. For instructions, see “Redeploying EAR Files to a Single Server or Multiple Servers” on page 97.
- 8 Verify that the custom logoff message is displayed when you log on and log off the Web application.

Configuring the HTTP Session Time-out Interval

A session time-out interval logs off users' inactive sessions after a specific period of time that is defined in the Web application server configuration. The default value for a session time-out interval is 30 minutes.

You can customize the session time-out interval for your environment by modifying one or more of the **web.xml** files, and specifying a different time-out interval. For more information, see “Modifying a Session Time-out Interval” on page 112.

Modifying a Session Time-out Interval

You can specify a session time-out interval for inactive user sessions with Web applications. To specify a custom session time-out interval, you should manually modify the time-out interval in the appropriate **web.xml** files. If an application has multiple **web.*xml** files, then you should modify all the files that are listed.

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

- 1 Modify the following code in the appropriate files:


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

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

When you are finished, save and close the file.

- 2 Use the SAS Deployment Manager to rebuild the EAR files that contain the SAS Web applications.
- 3 If the Web application server is running, stop it.
- 4 Redeploy the Web applications whose files you modified. See “Redeploying the SAS Web Applications” on page 97.
- 5 Restart the Web application server.

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

Table 8.2 Web Application Files to Modify for the Time-out Interval

Web Application	Files to Modify	Location
SAS Information Delivery Portal	web.xml.orig	<i>SAS-installation-directory</i> \SASInformationDeliveryPortal\ 4.3\Configurable\wars\sas.portal\ WEB-INF
SAS Package Viewer (ships with SAS Information Delivery Portal)	web.xml.orig	<i>SAS-installation-directory</i> \SASInformationDeliveryPortal\ 4.3\Configurable\wars\ sas.packageviewer\WEB-INF
SAS Web Report Studio	web.jboss.xml.orig web.weblogic.xml.orig web.websphere.xml.orig	<i>SAS-installation-directory</i> \SASWebReportStudio\4.3\ Configurables\wars\ sas.webreportstudio\WEB-INF

Web Application	Files to Modify	Location
SAS BI Portlets (Available in the October 2009 Release and later)	web.xml-idp.orig	<i>SAS-installation-directory</i>
	web.xml-thirdparty.orig	SASBIPortlets\4.3\Configurable \wars\sas.biportlets\WEB-INF
SAS Help Viewer Metadata Configuration	web.xml.orig	<i>SAS-installation-directory</i>
		\Documentation\9.2\Static\wars\sas.webdoc\WEB-INF

The following table lists the file or files that should be modified to specify a different time-out interval for SAS shared applications or SAS Web Infrastructure Platform files.

Table 8.3 SAS Shared Application and Web Infrastructure Platform Files to Modify for the Time-out Interval

SAS Shared Applications and Web Infrastructure Platform Files		
Files	Files to Modify	Location
SAS Shared Applications	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable\wars\sas.shared.apps\WEB-INF
SAS Shared Services	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable\wars\sas.shared.services\
SAS Shared Portlets	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable\wars\sas.svcs.portlets\WEB-INF
SAS Workflow	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable\wars\sas.workflow\WEB-INF
SAS Workflow Web Service	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable\wars\sas.workflow.webservice\WEB-INF
SAS Preferences	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable\wars\sas.preferences\WEB-INF

SAS Shared Applications and Web Infrastructure Platform		
Files	Files to Modify	Location
SAS Stored Process	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.storedprocess\WEB-INF
SAS Logon Manager	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.svcs.logon\WEB-INF
SAS Content Server	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.svcs.scs\WEB-INF
SAS Web Infrastructure Platform Client Access	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.wip.access\WEB-INF
SAS Web Infrastructure Platform Administration	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.wip.admin\WEB-INF
SAS Web Infrastructure Platform Services	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.wip.services\WEB-INF
SAS SOAP Services	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.wip.soapservices\WEB-INF
SAS BI Web Services (JBoss)	jboss-web.xml.orig web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.biws\jboss\WEB-INF
SAS BI Web Services (WebLogic)	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.biws\weblogic\WEB-INF
SAS BI Web Services (WebSphere)	web.xml.orig	<i>SAS-installation-directory</i> \SASSharedServices\9.2\Configurable \wars\sas.biws\websphere\WEB-INF


Configuring the Display of a Warning Message for Inactive User Sessions

In SAS 9.2 and previous maintenance releases for SAS 9.2, users with inactive user sessions were logged out and automatically redirected to the time-out Web page displayed by the SAS Logon Manager. Beginning with the third maintenance release

for SAS 9.2, you can alert users by displaying a warning message before they are logged out of their inactive sessions. By default, this warning message displays for 5 minutes, but you can customize it by specifying a different value in minutes.

This feature is supported for:

- ☐ SAS Web Report Studio
- ☐ SAS Package Viewer
- ☐ SAS Shared applications
- ☐ SAS Preferences
- ☐ SAS Web Infrastructure Platform administration
- ☐ SAS Stored Processes

Note: If SAS Information Delivery Portal 4.2 is configured at your site, this feature should not be enabled. Beginning with the August 2010 release, SAS Information Delivery Portal 4.3 enables you to alert users by displaying a warning message before they are logged out of their inactive sessions. 

The following table summarizes the features of the warning message.

Table 8.4 Display Warning Message for Inactive User Sessions

Feature	Third Maintenance Release for SAS 9.2
For inactive user sessions associated with SAS applications (SAS Information Delivery Portal, SAS Web Report Studio, SAS Package Viewer, SAS Shared applications, SAS Preferences, SAS Web Infrastructure Platform administration, and SAS Stored Process), display a warning message to users before logging them off their applications.	(Optional) Specify the Policy.DisplaySessionTimeoutWarning property and set the value to true . By default, this warning message displays for 5 minutes.
Specify the length of time for the display of a warning message that is enabled when the Policy.DisplaySession TimeoutWarning property is set to true .	(Optional) Specify the App.SessionTimeoutWarningInterval property and provide a value in minutes. Specifying the Policy.DisplaySession TimeoutWarning property is a prerequisite for using this property. The default value is 5 minutes. The value specified for this property cannot exceed the value defined for the session time-out interval in the web.xml file.

Configuring the Display of a Warning Message for Inactive User Sessions

Inactive users are logged off their Web applications when their sessions are inactive for 30 minutes or for the amount of time specified by the administrator in the **web.xml** files. In the third maintenance release for SAS 9.2, before logging out inactive sessions, you can alert users about the impending logoff by displaying a warning message. When the warning message is displayed, users can click the **Continue** button to activate and extend their sessions. In the third maintenance release for SAS 9.2, the following applications support the display of a warning message:

- SAS Web Report Studio
- SAS Package Viewer
- SAS Shared applications
- SAS Preferences
- SAS Web Infrastructure Platform administration
- SAS Stored Process

Beginning with the August 2010 release, SAS Information Delivery Portal 4.3 supports the display of a warning message.

To configure a warning message for inactive user sessions associated with SAS applications and specify the number of minutes that the warning message should be displayed, follow these steps:

- 1 On the **Plug-ins** tab, under **Application Management ► Configuration Manager**, right-click **SAS Application Infrastructure** and select **Properties**.

In the SAS Application Infrastructure Properties dialog box, click the **Advanced** tab.

- 2 Click **Add**.

- 3 In the Define New Property dialog box, enter each property name and property value as follows and click **OK**. Note that specifying the **App.SessionTimeoutWarningInterval** is optional. If the **App.SessionTimeoutWarningInterval** property is not specified with a custom value, the default value of 5 minutes applies to the **Policy.DisplaySessionTimeoutWarning** property. The value specified for the **App.SessionTimeoutWarningInterval** must be smaller than the value or values specified for session time-out intervals in the **web.xml** files. For information about **web.xml** files, see “Configuring the HTTP Session Time-out Interval” on page 112.

Property Name **Policy.DisplaySessionTimeoutWarning**

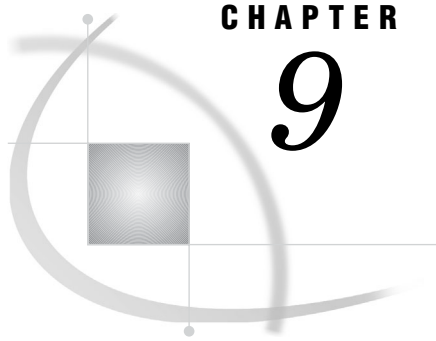
Property Value **true**

Property Name **App.SessionTimeoutWarningInterval**

Property Value *Value for Session Time-out Warning Interval*

- 4 Click **OK** to save your changes.
- 5 To enable these properties to take effect, restart the Web application server.

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.



CHAPTER

9

Administering SAS Shared Services

<i>About SAS Shared Services</i>	117
<i>Setting the Default Alert Notification Delivery Type</i>	117
<i>Administering the SAS Comment Manager Web Application</i>	118

About SAS Shared Services

SAS Shared Services provides standard features that are used by the SAS BI Dashboard and which can also be used by the SAS Information Delivery Portal, SAS Web Report Studio, and certain SAS solutions. The features include the following:

alert registration and notification	enables users to register to receive time-sensitive, action-oriented messages when a specified combination of events and conditions occurs. Alerts empower users by allowing them to control the type of notifications that they receive, and when they are delivered. Alerts can be sent to the user's e-mail address or displayed in the SAS Information Delivery Portal.
comment management	enables users to create comments related to business intelligence objects. Users can then reply to, search for, retire, delete, or add attachments to existing comments. In addition, users can find comments that were created by a particular user, with a specified date range or with specific text. This feature enables the capture of human intelligence and supports collaborative decision making related to business data.

During an Enterprise BI Server installation, a database and a data source name definition are created automatically on the SAS Table Server for the exclusive use of SAS Shared Services. For more information, see "Using the SAS Table Server with the SAS Middle Tier" on page 55.

Setting the Default Alert Notification Delivery Type

Alert notifications can be sent to users either via e-mail or via display in a portlet that users view in the SAS Information Delivery Portal. The default notification delivery type is specified in the properties for SAS Application Infrastructure using the Configuration Manager plug-in to SAS Management Console.

To change the alert notification setting, 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**.

- 3 Right-click **SAS Application Infrastructure** and select **Properties**.
- 4 Click the **Settings** tab.
- 5 Select **Notifications** in the left pane.
- 6 In the right pane, select a value from the **Alert notifications type** list box. Here are the choices:
 - ☐ My alerts portlet (default)
 - ☐ Via e-mail
 - ☐ Both e-mail and alerts portlet
- 7 Click **OK**.
- 8 To apply this setting and make it available, restart the SAS Web Infrastructure Platform Services, SAS Shared Services, and SAS Web Report Studio applications. (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 66.)

This setting becomes the default for your site. Users can specify an individual preference that overrides this value so that they receive notifications via a specific delivery type that meets their needs.

Administering the SAS Comment Manager Web Application

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 configure and enable a special administrative role in SAS metadata.

To enable the editing and deletion of comments, follow these steps:

- 1 Enable the administration of SAS Comment Manager as follows:
 - a Log on to SAS Management Console as an administrator.
 - b On the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► SAS Application Infrastructure**.
 - c Right-click **SAS Application Infrastructure** and select **Properties**.
 - d Select the **Advanced** tab.
 - e Add the property *Policy.CommentAdministrationEnabled* and set its value to *true*.
 - f Click **OK**.
- 2 Configure a Comment Administrator role and assign users to the role. In SAS Management Console, right-click **User Manager** and create a new role that has the following properties:
 - ☐ For the name, enter *Comment Administrator*. You must specify the name exactly as it appears here.
 - ☐ For the display name, enter *Comment Manager: Advanced*. This name follows the naming convention for roles.
 - ☐ For the description, enter *Members can edit and delete comments [implicit]*. Include [implicit] to signify that the role has implicit capabilities.
 - ☐ For capabilities, you do not need to grant capabilities for the role. This role has implicit capabilities.

- For members, add the appropriate users. Users who are assigned to this role have the ability to edit and delete comments.

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. △

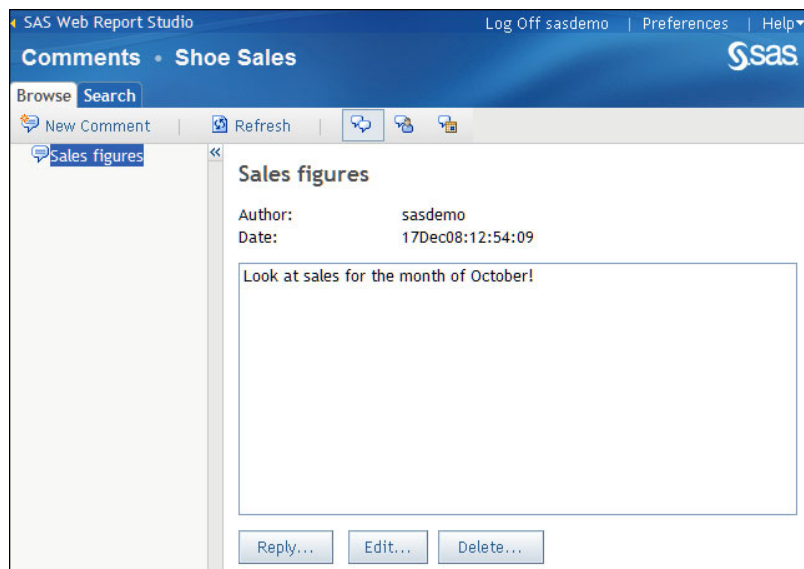
For instructions about creating roles, see the User Manager Help in SAS Management Console.

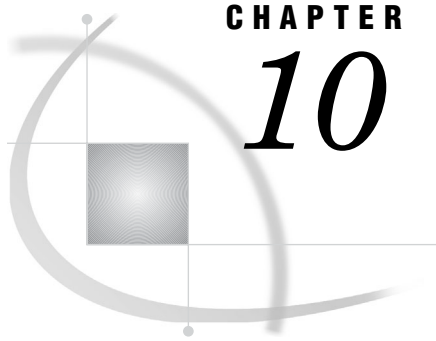
3 Restart the Web application server.

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 window 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.

Here is an example of SAS Comment Manager with a comment displayed.





CHAPTER

10

Administering the SAS Content Server

<i>About the SAS Content Server</i>	121
<i>Backing Up the SAS Content Server</i>	122
<i>Using the SAS Content Server Administration Console</i>	122
<i>About the SAS Content Server Administration Console</i>	122
<i>Access the SAS Content Server Administration Console</i>	122
<i>A Brief Tour of the Console Interface</i>	123
<i>Modify Permissions for WebDAV Folders and Files</i>	124
<i>Create a New Folder</i>	126
<i>Add Files to the SAS Content Server</i>	126
<i>Delete Folders or Files</i>	126
<i>Implementing Authorization for the SAS Content Server</i>	127
<i>Overview of SAS Content Server Authorization</i>	127
<i>Example Scenario: SAS Content Server Authorization</i>	127
<i>Reconfiguring the SAS Content Server to Share the Database Used by SAS Shared Services</i>	129
<i>Create the JCRCopyRepository File</i>	129
<i>Reconfigure SAS Content Server on JBoss to Share the Database Used by SAS Shared Services</i>	131
<i>Reconfigure SAS Content Server on WebSphere to Share the Database Used by SAS Shared Services</i>	132
<i>Reconfigure SAS Content Server on WebLogic to Share the Database Used by SAS Shared Services</i>	133

About the SAS Content Server

The SAS Content Server is a content repository that stores digital content (such as documents, reports, and images) created and used by SAS client applications. Examples of such content include reports and documents created by users of SAS Web Report Studio and the SAS Information Delivery Portal.

The Web-based 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 starts automatically when the Web application server is started and depends on the SAS Services Application. The SAS Services Application deploys a set of services called Remote Services that are used by SAS Information Delivery Portal, the SAS Stored Process Web application, and other Web applications. The SAS Services Application must be started before you start your Web application server.

Backing Up the SAS Content Server

The SAS Content Server should be backed up whenever the metadata server is backed up. For instructions about how to back up the SAS Content Server, see “Best Practices for Backing Up Your SAS System” in the *SAS Intelligence Platform: System Administration Guide*.

Use the WebDAVDump and WebDAVRestore utilities to:

- ☐ 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.

For instructions about using the WebDAVDump and the WebDAVRestore utilities, see SAS Note 38667.

Using the SAS Content Server Administration Console

About the SAS Content Server Administration Console

The SAS Content Server Administration Console enables you to manage files and WebDAV folders in the SAS Content Server. Using the console, you can perform the following management tasks:

- ☐ view folders
- ☐ control access to WebDAV folders and files by setting permissions
- ☐ create folders
- ☐ delete folders

Access the SAS Content Server Administration Console

To access the console, enter the following URL in your Web browser and substitute the server name and port number of your SAS Content Server:

`http://server:port/SASContentServer/dircontents.jsp`

Note: This console is also part of the SAS Web Administration Console. You can administer the SAS Content Server by using either interface. For more information about accessing the SAS Web Administration Console, see “Using the SAS Web Administration Console” on page 72. △

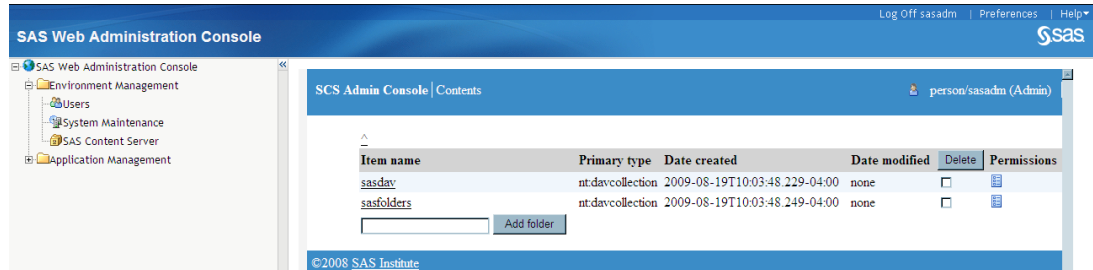
Log on to the console as someone who has user administration capabilities (for example, sasadm@saspw). You must be an unrestricted user to administer content in the SAS Content Server. The term "(Admin)" after your name at the top of the page indicates that you are logged on as an unrestricted user and that you have 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.

Display 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:

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 e-mail notification in SAS solutions.

sasfolders

contains content that has been defined in the SAS Folders tree in the SAS Metadata Server. You see a folder only if the folder contains content.

CAUTION:

Administrators should not manage folders and content here. The content within this folder and subfolders is mapped to SAS Folders in the SAS Metadata Server. It is recommended that you use the SAS Management Console to add and manage folders. △

Depending on the software that is installed at your site, your console might contain additional folders.



To navigate in the console, follow these steps:

- 1 Click an item in the list to display information about that item.
- 2 Use the breadcrumb trail above the list to return to a parent folder. For example,

in the breadcrumb trail, click **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 when the object was created.
Date modified	is the date when the object was modified.

- Delete** when the delete icon  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.

Display 10.2 Folder Permissions in the SAS Content Server

SCS Admin Console | Permissions

 person/sasadm (Admin) | [Logout](#)

^ / sasdav / Users / sales / northeast


Principal	READ	WRITE	DELETE	ADMIN	INHERIT	INHERIT	INHERIT	INHERIT	Remove
	READ	WRITE	DELETE	ADMIN	READ	WRITE	DELETE	ADMIN	
Owner(Nobody)	Yes	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	Yes	Yes	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	Yes	
jcr:authenticated	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="checkbox"/>
Add principal: <input type="text"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	

☒ Subfolders and files
☐ This folder only
☐ Overwrite permissions for all

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

Table 10.1 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).
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 10.2 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:8080/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 and z/OS, the utility command is **DAVTree.sh**.

For more information about using DAVTree, see “Using the DAVTree Utility to Manage WebDAV Content” on page 79.

- 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**.
- 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 be followed 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 **Delete** check box next to the name of the folder or file that you want to delete. The item is deleted; you are not prompted to confirm the deletion. To delete multiple folders or files, select multiple check boxes for **Delete**.

Implementing Authorization for the SAS Content Server

Overview of SAS Content Server Authorization

SAS users and groups are defined in a SAS Metadata Repository. The SAS Web Administration Console enables you to specify which users or groups are authorized to access specific folders in the SAS Content Server repository, and 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 10.3 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 login credentials that will access the SAS Content Server. When you define login credentials, you must specify the same authentication domain name that you specified for the SAS Content server during installation.

For this example, the following users, groups, and logins are defined:

Table 10.4 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 login with a user ID of salesusr1 and an authentication domain of DefaultAuth. The America Sales group might include other members as well.

- 2 In the SAS Web Administration Console, create your new directory under the sasdav directory. For this example, navigate to the **sasdav** directory, and then create these three subdirectories: **Catalog Sales**, **Field Sales**, and **Sales Execs**.
- 3 In the SAS Web Administration Console, configure the access permissions for the folders that you created. For this example, set the access permissions for each subdirectory, using the following tables as guides:

Table 10.5 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 10.6 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 10.7 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

Reconfiguring the SAS Content Server to Share the Database Used by SAS Shared Services

Beginning with the third maintenance release for SAS 9.2, SAS Content Server supports database persistence. By default, the SAS Content Server uses the file system for persistence, and you need not change this setup. In some environments though, there might be a compelling need to use a database back end. In such cases, you can reconfigure the SAS Content Server to share the database that is used by SAS Shared Services.

By default, SAS Shared Services uses the SAS Table Server. However, SAS Shared Services can be configured to use a different database such as Oracle, MySQL, PostgreSQL, DB/2, or SQL Server.

Create the JCRCopyRepository File

To migrate the contents of the current WebDAV repository to the database-based repository, create the **JCRCopyRepository.bat** or the **JCRCopyRepository.sh** file, and place it in the *SAS-configuration-directory/Web/Utilities* directory. You will later reconfigure the SAS Content Server to share the database used by SAS Shared Services and customize this file for your environment.

Here is an example of the **JCRCopyRepository.bat** file in Windows:

```
@echo on
:Script for executing the JCRCopyRepository utility

setlocal

REM Define needed environment variables
call "%-dp0..\..\level_envbat"

set LAUNCHERJAR=%SASVJR_HOME%\eclipse\plugins\sas.launcher.jar
set UTILITIESDIR=%LEVEL_ROOT%\Web\Utilities
set PICKLISTS=%SAS_HOME%\SASWebInfrastructurePlatform\9.2\Picklists\wars\
sas.svcs.scs\picklist
set DRIVER=path-to-jdbc-driver-JAR-file
set CLASSPATH=%UTILITIESDIR%;%LAUNCHERJAR%

"%JAVA_JRE_COMMAND" ^
  -classpath "%CLASSPATH%" ^
  -Djava.system.class.loader=com.sas.app.AppClassLoader ^
  -Dsas.app.launch.config="%PICKLISTS%" ^
  -Dsas.app.repository.path="%SASVJR_REPOSITORYPATH%" ^
  -Dsas.app.class.path="%UTILITIESDIR%;%DRIVER%" ^
  -Djava.security.auth.login.config=%LEVEL_ROOT%\Web\Common\login.config^
  -Xmx256m ^
  -Dscs.jndi.jndiName=sas/jdbc/SharedServices ^
  -Dscs.jndi.jdbcUrl=jdbc-url ^
  -Dscs.jndi.driver=jdbc-driver-class^
  -Dscs.jndi.user=database-user ^
  -Dscs.jndi.pwd=password ^
  org.apache.jackrabbit.core.JCRCopyRepository %1 %2
endlocal
if [%2] EQU [exit] exit %ERRORLEVEL%
```

Here is an example of the **JCRCopyRepository.sh** file in UNIX:

```
#!/bin/sh
#
# JCRCopyRepository.sh
#
. `dirname $0`/../../level_env.sh
LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar

UTILITIES=$LEVEL_ROOT/Web/Utilities
PICKLISTS=/OPT/SAS_92/SASWebInfrastructurePlatform/9.2/Picklists
/wars/sas.svcs.scs/picklist
DRIVER=path-to-jdbc-driver-JAR-file
CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR

"$JAVA_JRE_COMMAND" \
  -CLASSPATH "$CLASSPATH" \
  -Djava.system.class.loader=com.sas.app.AppClassLoader
  -Dsas.app.launch.config="$PICKLISTS" \
  -Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
  -Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
  -Djava.security.auth.login.config=../Common/login.config
```



```

-Xmx256m \
-Dscs.jndi.jndiName=sas/jdbc/SharedServices \
-Dscs.jndi.jdbcUrl=jdbc-url \
-Dscs.jndi.driver=jdbc-driver-class
-Dscs.jndi.user=database-user
-Dscs.jndi.pwd=password \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2

exit 0

```

Reconfigure SAS Content Server on JBoss to Share the Database Used by SAS Shared Services

To reconfigure the SAS Content Server on JBoss to share and use the same database that is used by SAS Shared Services, follow these steps.

- 1 Open the **SharedServices-ds.xml** file located in the *SAS-configuration-directory/Lev1/Web/Common/jboss* directory.
- 2 Make a note of the values specified in that file for the following parameters: host, port, dname, user name, and password. You will need this information later.
- 3 Stop the JBoss application server.
- 4 Rename the directory for the existing repository.

Here is an example for UNIX:

```
mv SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryFS
```

- 5 Create a directory in the original repository location.

Here is an example for UNIX:

```
mkdir SAS-configuration-directory/Lev1/AppData/SASContentServer/
Repository
```

- 6 Copy the **repository.xml** file from the *SAS-installation-directory/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates* directory to the directory that you created in the previous step.

Here is an example for UNIX:

```
cp /opt/SAS_92/SAS/SASWebInfrastructurePlatform/9.2/Static/wars/
sas.svcs.scs/WEB-INF/templates/repository.oracle.xml Repository/
repository.xml
```

The contents of the **repository.xml** file should identify the database that is used for SAS Shared Services.

- 7 Edit the **repository.xml** file, and specify the URL value for **java: namespace**. Be sure to make this modification in six locations within the file. Here is an example:

```
<param name="url" value="java:sas/jdbc/SharedServices"/>
```

- 8 In the **JCRCopyRepository** file that you created and placed in the *SAS-configuration-directory/Web/Utilities* directory, modify the value of the **DRIVER** parameter to indicate the path to the JDBC driver for the database.

Here is the syntax for the parameter:

```
DRIVER=path-to-jdbc-driver-JAR-file
```

Earlier, you opened the **SharedServices-ds.xml** file and retrieved the values for these parameters: **driver-class**, **connection-url**, **user-name**, and **password**. Specify these values for the **-Dscs.jndi** parameters in the **JCRCopyRepository** file and save the file.

```
-Dscs.jndi.jndiName=sas/jdbc/SharedServices ^
-Dscs.jndi.jdbcUrl=jdbc-url ^
-Dscs.jndi.driver=jdbc-driver-class ^
-Dscs.jndi.user=database-user ^
-Dscs.jndi.pwd=password ^
```

Although the **repository.xml** file contains the JNDI name for the database connection that is specified in the **java: namespace** parameter, do not add that prefix to this script.

9 Verify that the JBoss application server is stopped.

10 Run the **JCRCopyRepository** script once.

Here is an example on UNIX:

```
./JCRCopyRepository.sh SAS-configuration-directory/Lev1/AppData/
SASContentServer/RepositoryFS SAS-configuration-directory/Lev1/AppData/
SASContentServer/Repository
```

In this command syntax, the first argument is the old repository directory, and the second argument is the new repository directory.

11 To enable the changes to go into effect, restart the JBoss application server.

Reconfigure SAS Content Server on WebSphere to Share the Database Used by SAS Shared Services

To reconfigure the SAS Content Server on JBoss to share and use the same database that is used by SAS Shared Services, follow these steps.

1 In the WebSphere Admin Console, navigate to **Resources ► JDBC ► Data Sources**.

2 Click **SharedServices**.

3 Make a note of the value displayed in the **Implementation class name** field.

Here is an example:

```
com.sas.tkts.TKTSConnectionPoolDataSource.
```

4 Navigate to **Resources ► JDBC ► JDBC Providers**.

5 Click **Custom Properties**.

6 Make a note of the value displayed for **serverUrl**.

Here is an example:

```
jdbc:sastkts://redwood.na.sas.com:2171
```

7 Stop the WebSphere application server.

8 Rename the directory for the existing repository.

Here is an example for UNIX:

```
mv SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryFS
```

9 Create a directory in the original repository location.

Here is an example for UNIX:

```
mkdir SAS-configuration-directory/Lev1/AppData/SASContentServer/
Repository
```

10 Copy the **repository.xml** file from the **SAS-installation-directory/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates** directory to the directory that you created in the previous step.

Here is an example for UNIX:

```
cp /opt/SAS_92/SAS/SASWebInfrastructurePlatform/9.2/Static/wars/
sas.svcs.scs/WEB-INF/templates/repository.oracle.xml Repository/
repository.xml
```

The contents of the **repository.xml** file should identify the database that is used for SAS Shared Services.

- 11 In the **JCRCopyRepository** file that you created and placed in the *SAS-configuration-directory/Web/Utilities* directory, modify the value of the **DRIVER** parameter to indicate the path to the JDBC driver for the database.

Here is the syntax for the parameter:

```
DRIVER=path-to-jdbc-driver-JAR-file
```

Earlier, you retrieved the values for the **Implementation class name** and **serverUrl** parameters in the WebSphere Admin Console. Specify these values for the **-Dscs.jndi** parameters in the **JCRCopyRepository** file and save the file.

```
-Dscs.jndi.jndiName=sas/jdbc/SharedServices ^
-Dscs.jndi.jdbcUrl=jdbc-url ^
-Dscs.jndi.driver=jdbc-driver-class ^
-Dscs.jndi.user=database-user ^
-Dscs.jndi.pwd=password ^
```

Although the **repository.xml** file contains the JNDI name for the database connection that is specified in the **java: namespace** parameter, do not add that prefix to this script.

- 12 Verify that the WebSphere application server is stopped.
- 13 Run the **JCRCopyRepository.sh** or the **JCRCopyRepository.bat** file once.

Here is an example for UNIX:

```
./JCRCopyRepository.sh SAS-configuration-directory/Lev1/AppData/
SASContentServer/RepositoryFS SAS-configuration-directory/Lev1/AppData/
SASContentServer/Repository
```

In this command syntax, the first argument is the old repository directory, and the second argument is the new repository directory.

- 14 To enable the changes to take effect, restart the WebSphere application server.

Reconfigure SAS Content Server on WebLogic to Share the Database Used by SAS Shared Services

To reconfigure the SAS Content Server on JBoss to share and use the same database that is used by SAS Shared Services, follow these steps.

- 1 In the WebLogic Admin Console, navigate to **SASDomain ► Services ► JDBC ► Data Sources**.
- 2 Click **SharedServices**.
- 3 Click the **Connection Pool** tab.
- 4 Make a note of the values for these parameters: **URL:** and **Driver Class Name**. You will also need the values for the user and password.
- 5 Stop the WebLogic application server.
- 6 Rename the directory for the existing repository.

Here is an example for UNIX:

```
mv SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryFS
```

- 7 Create a directory in the original repository location.

Here is an example for UNIX:

```
mkdirSAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
```

- 8 Copy the **repository.xml** file from the *SAS-installation-directory / SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates* directory to the directory that you created in the previous step.

Here is an example for UNIX:

```
cp /opt/SAS_92/SAS/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates/repository.oracle.xml Repository/repository.xml
```

The contents of the **repository.xml** file should identify the database that is used for SAS Shared Services.

- 9 In the **JCRCopyRepository** file that you created and placed in the *SAS-configuration-directory/Web/Utilities* directory, modify the value of the **DRIVER** parameter to indicate the path to the JDBC driver for the database.

Here is the syntax for the parameter:

```
DRIVER=path-to-jdbc-driver-JAR-file
```

Earlier, you retrieved the values for **URL:** and **Driver Class Name** parameters from the WebLogic Admin Console. Specify these values for the **-Dscs.jndi** parameters in the **JCRCopyRepository** file and save the file.

```
-Dscs.jndi.jndiName=sas/jdbc/SharedServices ^
-Dscs.jndi.jdbcUrl=jdbc-url ^
-Dscs.jndi.driver=jdbc-driver-class ^
-Dscs.jndi.user=database-user ^
-Dscs.jndi.pwd=password ^
```

Although the **repository.xml** file contains the JNDI name for the database connection that is specified in the **java: namespace** parameter, do not add that prefix to this script.

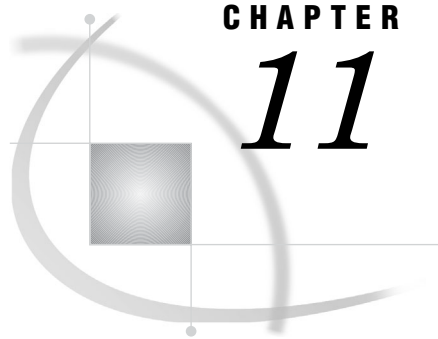
- 10 Verify that the WebLogic application server is stopped.
- 11 Run the **JCRCopyRepository.sh** or the **JCRCopyRepository.bat** file once.

Here is an example on UNIX:

```
./JCRCopyRepository.sh SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryFS SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
```

In this command syntax, the first argument is the old repository directory, and the second argument is the new repository directory.

- 12 To enable the changes to take effect, restart the WebLogic application server.



CHAPTER

11

Administering SAS BI Web Services

<i>Managing Generated Web Services</i>	135
<i>Configuring SAS BI Web Services for .NET</i>	136
<i>Configuring SAS BI Web Services for Java</i>	137
<i>Overview of Security for Web Services</i>	142
<i>Securing SAS BI Web Services for .NET</i>	143
<i>Securing SAS BI Web Services for Java</i>	144
<i>SAS Authentication</i>	144
<i>Web Authentication</i>	144
<i>Overview of Web Authentication</i>	144
<i>Message-level Security</i>	144
<i>Transport-level Security</i>	145
<i>Update the JBoss Application Server Configuration</i>	146
<i>Update the WebSphere Application Server Configuration</i>	147
<i>Update the WebLogic Application Server Configuration</i>	148
<i>Update Remote Services Files</i>	149
<i>Additional Administrative Tasks for SAS BI Web Services for Java</i>	150
<i>Enabling Java2 Security</i>	150
<i>Running Exploded in the WebLogic Application Server</i>	150
<i>Java Development Kit (JDK) Requirement</i>	150
<i>Setting Up Clustering</i>	150
<i>Enabling Application Scope</i>	152
<i>Using the Deploy As Web Service Wizard to Overwrite an Existing Web Service</i>	152

Managing Generated Web Services

Starting with SAS 9.2, 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 more information about developing Web services, see the *SAS BI Web Services: Developer's Guide*. For more 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 also generates metadata about the new Web service and stores it on the SAS Metadata Server. The Web Service Maker stores information about the URL of the Web service, keywords that are associated with the Web service, and which stored processes are used by the Web service. You can view and update this information, perform impact analysis, and migrate these Web services at a later time by using SAS Management Console and the Configuration Manager in SAS Management Console.

To delete a Web service that was generated by the Web Service Maker, use the Configuration Manager in SAS Management Console. In the Configuration Manager,

find and expand **SAS BI Web Services for Java** or **SAS BI Web Services for .NET**, depending on your platform. Expand the **WebServiceMaker** node, right-click the generated Web service, and select **Delete**. Deleting a generated Web service removes the artifacts on the server and also removes the metadata that is associated with the generated Web service. This operation cannot be undone. For more information about the Configuration Manager, see the product Help.

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 default group named **BI Web Services Users** is created, which has these permissions. You can add users to this group to allow them to create and delete Web services, or use your own groups and permission settings. △

Configuring SAS BI Web Services for .NET

If you are using SAS BI Web Services for .NET, then you can change the configuration settings for generated Web services by editing a web.config file. The following configuration settings can be edited manually in the web.config file:

NamespaceForGeneratedServices

specifies the namespace that a generated Web service uses.

AuthenticationType

specifies one of the following two values:

Trusted

The Web service is configured (in the **SystemMetadataFile** or the **UserMetadataFile**) with a trusted connection to the metadata server. The Web service examines the HTTP context (which means that the caller authenticates using the standard container capabilities) in order to get the current user ID, and uses that ID when connecting to the metadata server. Trusted authentication is also known as Web authentication.

Host

The middle tier authenticates against the metadata server as a non-trusted user. Host authentication is also known as SAS authentication. Using this type of authentication means:

- If you configure the **MetadataAuthenticator SecurityTokenManager** (in the **microsoft.web.services3** section of this configuration file), then clients might pass in credentials that get used to connect to the metadata server. This configuration is enabled by default.
- If a client does not pass in credentials, or the **MetadataAuthenticator** is not configured, then a default set of credentials (which might be SSPI) needs to be set in the **SystemMetadataFile** or **UserMetadataFile**. These credentials enable callers to anonymously connect to a server (if the container is also configured to allow this connection).
- If a **Username Token** is used, the user name and password are passed as plain text in the WS-Security Username Token. It is strongly recommended that you use secure HTTPs when using a plain text password.

This setting is not specific to Web services, but applies to the middle tier in general.

AcceptableSyscc

indicates which error codes do not cause a SOAP fault. After a stored process runs, the Web service checks the SYSCC macro value in SAS. If that value is not in the comma-separated list for AcceptableSyscc, then a SOAP fault is generated and the message is obtained from SYSMSG.

SystemMetadataFile

specifies the location of the system metadata file. The system metadata file contains the location (host, port, and encryption) of the metadata server, and might contain credentials.

UserMetadataFile

specifies credentials that override what is in the SystemMetadataFile setting. To force callers of the service to specify credentials on each call, you should not provide default credentials for either the SystemMetadataFile or the UserMetadataFile settings.

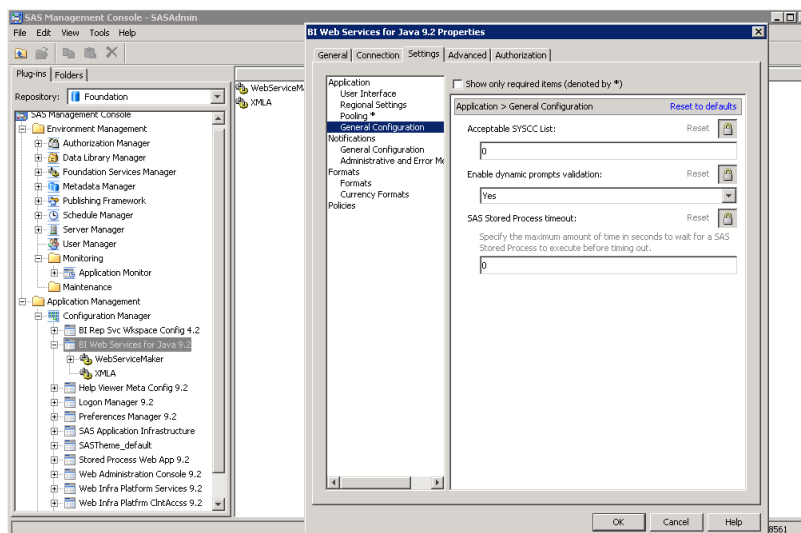
ClientName

is passed to the stored process at run time at the start of the `_CLIENT` macro.

Configuring SAS BI Web Services for Java

SAS BI Web Services for Java is initially configured during installation using the SAS Deployment Wizard. To modify this initial configuration, use the Configuration Manager plug-in for SAS Management Console.

To modify common configuration properties that apply to XMLA, WebServiceMaker, and generated Web services, navigate to the **BI Web Services for Java 9.2** application folder. Then navigate to the **Settings** tab within the Properties dialog box.



In the **Application ► General Configuration** section, you have the ability to 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 conveniently 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, so see the appropriate SAS documentation to determine possible values that might be returned and whether these values are errors or just warnings in your case. 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.

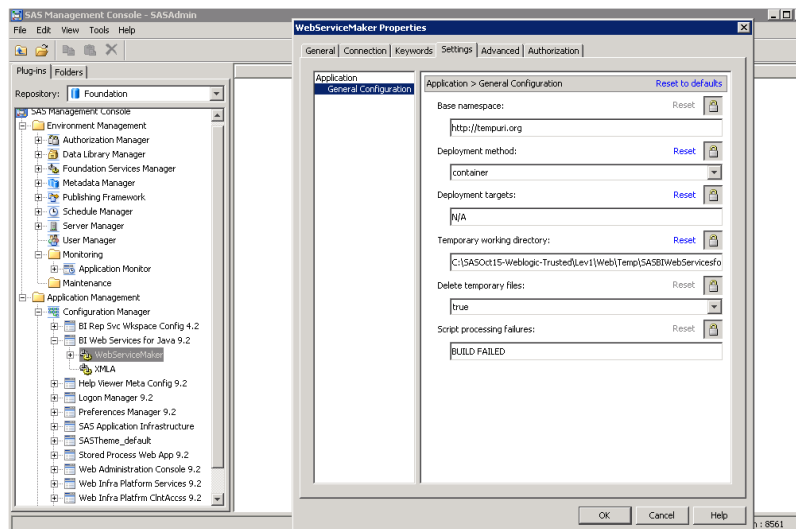
Enable dynamic prompts validation

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 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 will be canceled and a SOAP fault will be returned to the invoking client. A value of zero indicates no time-out period.

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.



In the **Application ► General Configuration** section, you have the ability to modify the following configuration properties:

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 this Web service becomes **http://tempuri.org/test**.

Deployment method

The method used to deploy generated Web services. Select **Container** to deploy a generated Web service directly to the application server (container) and make it immediately available for execution. Select **Directory** to copy the generated Web service artifact to a designated directory so that it can be deployed to the container at a later time.

Deployment directory

When the deployment method is **Directory**, this property specifies the file location to copy the generated Web service artifact.

Temporary working directory

This property specifies the directory that is used to generate Web service artifacts.

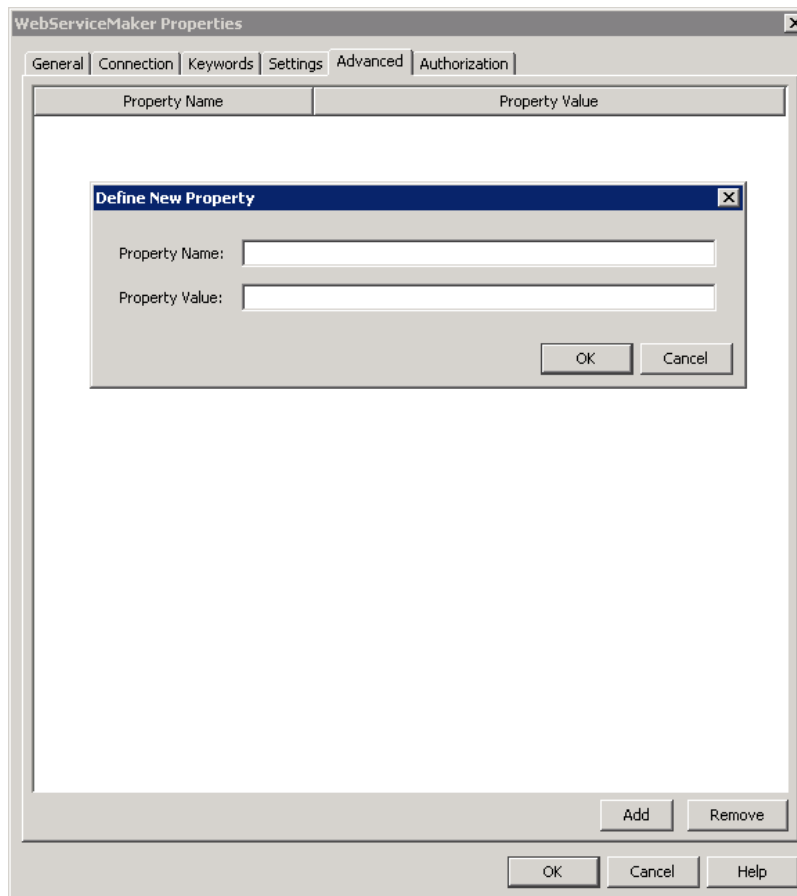
Delete temporary files

This property specifies whether to delete temporary files after a successful deployment. In most cases you should clean up temporary files after a successful deployment.

Script processing failures

Ant scripts are used to perform specific tasks. The output streams that are produced from running such tasks are analyzed for failures by searching for well-known error messages such as **BUILD FAILED**. More than one comma-separated value can be specified. If any of the keywords specified in this property are encountered, the generation process is ended, and a SOAP fault is returned to the invoking client.

To modify more advanced configuration properties that are specific to the Web Service Maker, navigate to the **Advanced** tab within the Properties dialog box.



The following advanced configuration properties are available:

AddWSSecurityAuthConstraint

If Web authentication is enabled and this property is set to **true**, then generated Web services will be configured to require WS-Security Username Token authentication. However, to fully configure application server authentication using Java Authentication and Authorization Service (JAAS), additional configuration is necessary. For more information, see “Securing SAS BI Web Services for Java” on page 144. The default value for this property is **true**. Set this property to **false** if you do not want to automatically configure WS-Security for generated Web services. An example would be if you wanted to configure HTTP transport-level security instead.

JAASLoginConfigName

This property enables you to specify an alternate JAAS login configuration when using WS-Security Web authentication within the WebSphere application server. By default, the **WSLogin** JAAS login configuration is used.

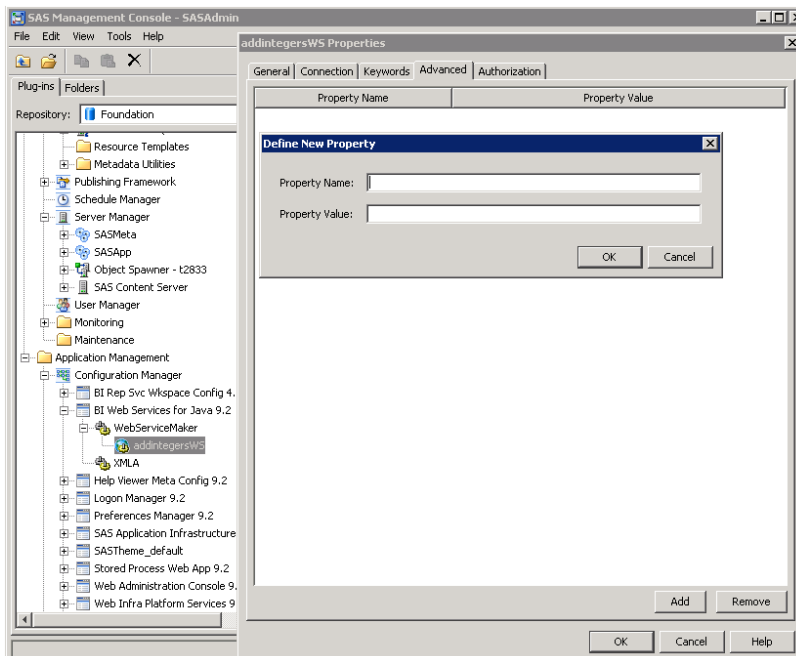
AttachmentConformance

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.

JavaHome

This property specifies the JDK to use when compiling generated code. The default is to use the java.home system property, which resolves to the JDK that is being used by the application server JVM.

To modify configuration properties that are specific to a generated Web service, navigate to the folder for that service. Then navigate to the **Advanced** tab within the Properties dialog box.



In this example, **addIntegersWS** is the Web service that is being modified. The following advanced configuration properties are available:

AcceptSysccList

See Acceptable SYSCC List. This property overrides its analogous common configuration property.

DynamicPromptsSupport

See Enable dynamic prompts validation. This property overrides its analogous common configuration property.

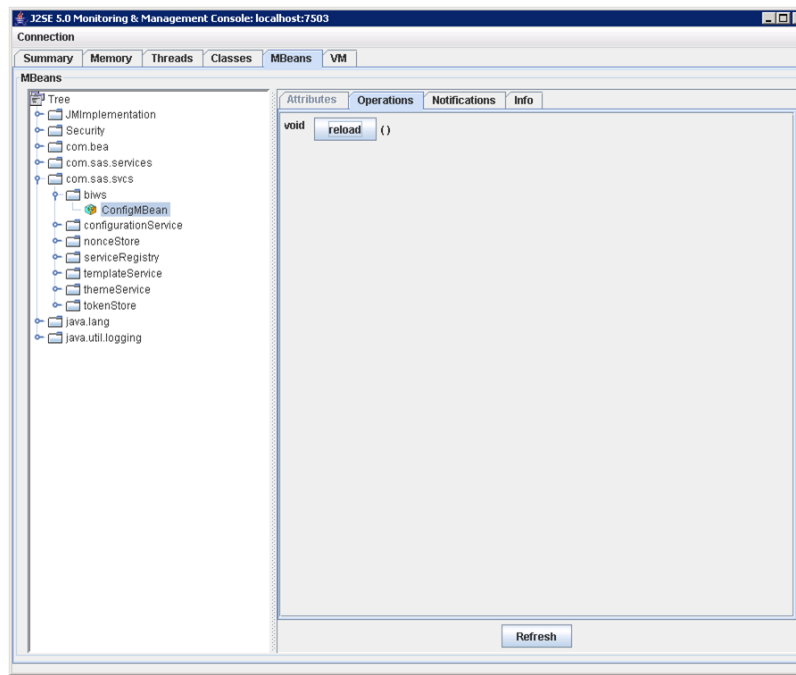
MaxSTPExecTime

See SAS Stored Process timeout. This property overrides its analogous common configuration property.

Changes to properties do not take effect immediately. To apply these changes you must perform one of the following tasks:

- Either stop and restart the application server, or stop and restart the SAS BI Web Services for Java Web application (sas.wip.services9.2.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 or .NET is not highly secure. The default security mechanism 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. Authentication is performed by authenticating client credentials at the SAS Metadata Server. Whenever user names and passwords must be sent as clear-Text, SSL should be enabled to provide transport layer security.

If you want to use SSL and the Deploy as Web Service Wizard to communicate with a server, then you need to enable SSL. In order to enable SSL, 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://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html>.
- 2 Pass the keystore location and password into SAS Management Console using Java JVM arguments. The arguments 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/SAS/SASManagementConsole/9.2:**

```
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 be secured by using Web authentication. This provides a way for SAS BI Web Services to identify the calling user by using basic Web authentication. The following two types of Web authentication can be configured:

- WS-Security message-level security
- HTTP transport-level security

Note: Web authentication can be used with both XMLA Web services and generated Web services but cannot be used with the Web Service Maker Web service. The Web Service Maker must be able to authenticate one-time-passwords that are generated by SAS Management Console clients. △

SAS BI Web Services for .NET can be secured by using Web Services Enhancements 3.0 for Microsoft .NET, which enables support for the latest security and interoperability standards for Web services. For detailed information about using Web Services Enhancements 3.0, WS-Security, and WS-Policy to secure SAS BI Web Services, see “Securing SAS BI Web Services for .NET” on page 143. The current release of SAS BI Web Services for Java does not support WS-Policy.

Securing SAS BI Web Services for .NET

If you choose to enable the anonymous user during SAS Deployment Wizard configuration, SAS BI Web Services for .NET is configured with a default user to use for clients who do not provide credentials when calling generated Web services. This is not highly secure because it allows anyone to call your generated Web services. You can change the default credentials to be a different user, to use SSPI, or you can remove the default credentials to disable anonymous access. These credentials are stored in the file pointed to by the SystemMetadataFile setting in your Web.config file. You can use this anonymous user technique only when SAS BI Web Services for .NET is configured in Host (SAS) authentication mode. When in Trusted (Web) authentication mode, the trusted user's credentials should be stored in the SystemMetadataFile.

Web Services Enhancements for Microsoft .NET (WSE) is a Microsoft add-on to .NET that implements advanced Web services specifications, such as WS-Security. SAS BI Web Services for .NET is secured with WSE by default in 9.2 using UsernameToken authentication in clear text. Additional security measures can be configured by performing the following tasks:

- modifying the wse3polycache.config policy file that indicates the security policy required by SAS BI Web Services to enable more advanced forms of security including Kerberos, X.509, and encryption
- modifying client applications to use WSE, and to also use a security policy that matches the Web service

You need to modify the GeneratedServicesPolicy WSE policy in the wse3polycache.config file to enable these advanced options. For information, see <http://msdn.microsoft.com/en-us/library/aa480582.aspx>.

Note: Use caution when using the Microsoft WSE Configuration Tool (WseConfigEditor.exe) to edit a SAS BI Web Services for .NET configuration. The WSE Configuration Tool does not support custom policy assertions and using the tool

completely removes them. Instead, edit the `wse3policyCache.config` file manually and add the desired security elements to the existent SAS configuration. △

Securing SAS BI Web Services for Java

SAS Authentication

The default security configuration out-of-the box for SAS BI Web Services for Java is *SAS authentication*. In this mode the 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 configured 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.
- 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

Overview of Web Authentication

Alternatively to SAS authentication, the 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*. To configure Web authentication for XMLA Web services and generated Web services, select the **Custom** prompting level in the SAS Deployment Wizard and choose Web authentication during configuration. By default, when Web authentication is configured, WS-Security constraints are applied to the XMLA Web service as well as to generated Web services. However, transport-level security can be configured instead of message-level security (WS-Security) if desired. The following information describes the manual configuration steps that are necessary to enable Web authentication.

Message-level Security

By default, WS-Security security constraints are automatically configured when Web authentication is configured. In this case, Rampart security is engaged and configured in the `services.xml` deployment descriptor as follows:

```
<module ref="rampart"/>
<parameter name="InflowSecurity">
  <action>
    <items>UsernameToken</items>
```

```

        <passwordCallbackClass>
            com.sas.web.services.axis2.PwcbHandler
        </passwordCallbackClass>
    </action>
</parameter>

```

For more information about Axis2/Rampart configuration, see <http://ws.apache.org/axis2/>.

Currently, the Rampart security module is configured to require a Username Token. The Rampart module parses the Username Token credentials. Ultimately, the implementation consumes these credentials and uses them to programmatically invoke the application server's authentication provider configured for the current security realm, thereby allowing seamless integration into the container's security subsystem. The following list shows how this authentication is accomplished for the different Web containers:

JBoss	Uses the <code>org.jboss.web.tomcat.login.WebAuthentication</code> class to integrate with Java Authentication and Authorization Service (JAAS).
IBM WebSphere	Invokes WSSLogin JAAS login configuration. To invoke a different JAAS login configuration, set the <code>JAASLoginConfigName</code> SAS BI Web Services for Java configuration property.
Oracle WebLogic	Uses the <code>weblogic.security.services.Authentication</code> class to integrate with JAAS.

Transport-level Security

HTTP transport-level security can be used instead of message-level security. Any message-level security constraints previously enabled must be disabled in order to use transport-level security. To disable generated Web services from automatically being configured with WS-Security constraints, you should set the `AddWSSecurityAuthConstraint` WebServiceMaker-specific configuration property to **false**.

The following security constraints should be applied to the `web.xml` deployment descriptor (`sas.biws.war` module with the `sas.wip.services9.2.ear` application) as follows:

```

<security-constraint>
    <web-resource-collection>
        <web-resource-name>All-resources</web-resource-name>
        <url-pattern>/services/XMLA/*</url-pattern>
        <url-pattern>/services/generatedWebServiceName/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>

    <auth-constraint>
        <role-name>SASWebUser</role-name>
    </auth-constraint>
</security-constraint>

<login-config>
    <auth-method>BASIC</auth-method>
</login-config>

<security-role>

```

```
<role-name>SASWebUser</role-name>
</security-role>
```

The URL patterns in this code should include all of the Web services that should be secured by the HTTP transport. Notice that the WebServiceMaker service is not one of them because it must be configured for SAS authentication. Also, notice that the secured Web services must be invoked by a client that is in the SASWebUser role.

Update the JBoss Application Server Configuration

The name of the security domain is configured in the jboss-web.xml deployment descriptor. You must use this same name when creating a JAAS login configuration for this security domain. If for example the security domain name is **SASApplicationLogin**, your jboss-web.xml deployment descriptor contains the following:

```
<jboss-web>
  <security-domain>java:/jaas/SASApplicationLogin</security-domain>
</jboss-web>
```

The JAAS configuration file located in your server configuration (JBoss_Home/Server/SASServer1/conf/login-config.xml) looks like the following code:

```
<application-policy name="SASApplicationLogin">
  <authentication>
    <!-- place site-specific login modules here -->
    <login-module
      code="com.sas.services.security.login.TrustedLoginModule"
      flag="optional">
      <module-option name="host">SAS-metadata-serve-host</module-option>
      <module-option name="port">8561</module-option>
      <module-option name="domain">web</module-option>
      <module-option name="aliasdomain">DefaultAuth</module-option>
      <module-option name="trusteduser">sastrust@saspw</module-option>
      <module-option name="trustedpw">encoded-password</module-option>
    </login-module>
  </authentication>
</application>
```

You should add any site-specific login modules to the configuration. For example, if you want to use the UsersRolesLoginModule JBoss login module to authenticate users against a text file, follow these steps:

- 1 Add user names and passwords to the JBOSS_HOME/server/SASServer1/conf/props/sas-users.properties file. Format of this file is as follows:
username=password.
- 2 Add user names and role to the JBOSS_HOME/server/SASServer1/conf/props/sas-roles.properties file. Format of this file is as follows: *username=SASWebUser.*
- 3 Add the following login module to the SASApplicationLogin JAAS login configuration:

```
<login-module
  code="org.jboss.security.auth.spi.UsersRolesLoginModule"
  flag="required">
  <module-option name="usersProperties">
    props/sas-users.properties
  </module-option>
  <module-option name="rolesProperties">
```



```

        props/sas-roles.properties
    </module-option>
    <module-option name="unauthenticatedIdentity">
        Anonymous
    </module-option>
</login-module>

```

Update the WebSphere Application Server Configuration

The type of security that is configured determines the JAAS login configuration that is enabled. For example, if message-level security (WS-Security) is configured, then the WSLogin JAAS login configuration (or the JAAS login configuration overridden with the JAASLoginConfigName property) is enabled. Otherwise, if transport-level security is configured, then the WEB_INBOUND JAAS login configuration is enabled.

After you have determined what JAAS login configuration is active for your particular environment, use the administration console to add the SAS trusted login module to that configuration. Follow these steps:

- 1 Navigate to **Security ► Security administration, applications, and infrastructure ► Java Authentication and Authorization Service**. Select **Application logins** or **System logins** depending on the configuration that you need to modify. Select the JAAS configuration that you need to modify, then select **JAAS login modules**.
- 2 Click **New** and enter the following information:

```

Module class name: com.sas.services.security.login.websphere.WSTrustedLoginModule
Authentication strategy: OPTIONAL

```

Click **OK**.

- 3 Click the **Custom Properties** link and enter the following name/value pairs:

```

host: SAS-metadata-server-host
port: 8561
domain: web
aliasdomain: DefaultAuth
trusteduser: sastrust@saspw
trustedpw: encoded-password-for-sastrust

```

where *encoded-password-for-sastrust* is an encoded password that can be obtained from the SAS administrator. For more information about encoding passwords, see *SAS Intelligence Platform: Security Administration Guide*.

In order for the SAS trusted login module to be loaded successfully by the WebSphere run-time application class loader during authentication, the following SAS JAR files need to be copied from the SAS Versioned Jar Repository to WAS_HOME/lib/ext:

```

sas.core.jar
sas.oma.omi.jar
sas.svc.connection.jar
sas.svc.sec.login.jar
sas.svc.sec.login.websphere.jar
sas.security.sspi.jar

```

Lastly, configure the user account repository from which to authenticate against. Using the administration console, follow these steps:

- 1 Navigate to **Security ► Secure administration, applications, and infrastructure**.
- 2 In the User account repository, select from the available realm definitions (such as Local operation system) and configure it and set it as the current realm definition.

Additionally, if transport-level security is configured, then map the SASWebUser role to all authenticated users. Use the administration console to enable this mapping as follows:

- 1 Navigate to **Enterprise Applications ► SAS Web Infrastructure Platform Services ► Security role to user/group mappings**.
- 2 Select **All authenticated** for the SASWebUser role.

Update the WebLogic Application Server Configuration

Install and configure the Authentication Provider for your security realm. Follow these steps:

- 1 Copy the following SAS JAR files from the SAS Versioned Jar Repository to BEA_HOME/weblogic92/server/lib/mbeantypes. These JAR files allow you to add and configure the SAS trusted login module to the list of authenticated providers for a given security realm.

```
sas.svc.sec.login.weblogic..mbean.jar
sas.svc.sec.login.weblogic.mbean.nls.jar
```

- 2 Copy the following SAS JAR files from the SAS Versioned Jar Repository to BEA_HOME/weblogic92/server/lib/mbeantypes. These JAR files need to be available to the SAS trusted login module at run time.

```
sas.svc.sec.login.weblogic.jar
sas.svc.sec.login.weblogic.nls.jar
sas.svc.sec.login.jar
sas.svc.sec.login.nls.jar
sas.svc.connection.jar
sas.oma.omi.jar
sas.core.jar
sas.security.sspi.jar
```

- 3 Start the administrating server and then connect to it as follows:

```
http://host:port/console
```

- a Navigate to **Security Realms**, and then to your Realm.
- b Select the **Providers** tab and then select **Authentication** within that grouping. Typically, you see the configured default authenticator providers: DefaultAuthenticator, DefaultIdentifyAsserter.
- c Click **New** to add the SAS trusted login module to the chain of providers. Enter the following information:

```
Name: SASTrustedAuthenticator
Type: WLTrustedAuthenticator
```

- d Click on the newly created provider and enter the following information:

```
Common information:
  Control Flag: OPTIONAL
Provider Specific information:
  Host: SAS-metadata-server-host
  Domain: web
  Port: 8561
  Encrypt: false
  Trusted User: sastrust@saspw
  Trusted Password: encoded-password-for-sastrust
  Debug: false
```

where *encoded-password-for-sastrust* is an encoded password that can be obtained from the SAS administrator. For more information about encoding passwords, see the *SAS Intelligence Platform: Security Administration Guide*.

- e Click **Save** and **Activate changes**.
- f Restart the administration server.

- 4 Configure user information for the authentication provider. For example, if you are using the DefaultAuthenticator provider, you need to configure the internal WebLogic LDAP server with user and password information.

Navigate to **Security Realms**, select your Realm, and then select **Users and Groups**. Then click **New** to enter users.

Additionally, if transport-level security is configured, then map the SASWebUser role to individual authenticated users. You can perform this action using the administration console or you can modify the weblogic.xml deploy descriptor as follows:

```
<security-role-assignment>
  <role-name>SASWebUser</role-name>
  <principal-name>username</principal-name>
</security-role-assignment>
...
```

Update Remote Services Files

Modify the JAAS login configuration at the following location: SAS-config-dir/lev1/web/common/login.config. The following code is an example of what this configuration might look like:

```
PFS {
    com.sas.services.security.login.OMILoginModule required
        "host"="SAS metadata server host"
        "port"="8561"
        "repository"="Foundation"
        "domain"="DefaultAuth"
        "aliasdomain"="web"
        "trusteduser"="sastrust@saspw"
        "trustedpw"="encoded password for sastrust"
        "debug"="false";
};
```

Modify the remote services script, the wrapper.conf file, or both to add the following JAR files to the SAS application classpath (-Dsas.app.class.path) so that application server-specific JAAS principle and credential classes are available:

JBoss

```
JBoss_HOME/Server/SASServer1/lib/jbosssx.jar
```

WebSphere

```
WAS_HOME/lib/bootstrap.jar
WAS_HOME/lib/j2ee.jar
WAS_HOME/plugins/com.ibm.ws.runtime_6.1.0.jar
WAS_HOME/plugins/com.ibm.ws.emf_2.1.0.jar
WAS_HOME/plugins/org.eclipse.emf.ecore_2.2.1.v200609210005.jar
WAS_HOME/plugins/org.eclipse.emf.common_2.2.1.v200609210005.jar
```

WebLogic

WLS_HOME/server/lib/wls-ap.jar

Additional Administrative Tasks for SAS BI Web Services for Java

Enabling Java2 Security

Enabling Java2 Security is not recommended due to significant degradation in performance. However, if you must enable Java2 Security, then you should familiarize yourself with how the Axis2 engine loads Web services.

For example, when the Axis2 engine deploys a Web service, it explodes the service archive in a temporary directory designated by the application server's JVM and loads service classes using its own URL class loader. This is important to WebSphere administrators because they might think that codeBase permissions can be added to the was.policy file for Axis2 Web service classes. This does not work because classes that are not loaded by the WebSphere WAR class loader are not part of that protection domain. In this case, codeBase permissions regarding these service classes must be placed in a protection domain that covers these classes. A WebSphere administrator can use the app.policy file because it covers all classes in the cell, or the administrator can use the JVM java.policy file because it covers all classes in the JVM.

Running Exploded in the WebLogic Application Server

In order to support hot deployment, Axis2 requires the archive (sas.wip.services9.2.ear) to be deployed in exploded format. Because the WebLogic deployer does not automatically explode its archives like other containers do, the SAS Deployment Wizard manually performs this operation during configuration processing.

Hot deployment enables SAS BI Web Services for Java to dynamically generate new Web services and make them immediately available for consumption without having to stop and restart the server. Axis2 hot deployment is enabled by default. To configure SAS BI Web Services for Java in this manner, see the Deployment method configuration property for more details.

Java Development Kit (JDK) Requirement

Because SAS BI Web Services for Java must compile generated code while running in the application server, it requires access to a JDK at run time. Therefore, a Java Runtime Environment (JRE) is not sufficient. Normally, the application server is executed with a JDK and as a result SAS BI Web Services for Java has access to that JDK. However, in rare circumstances when the application server is executed with a JRE instead, it is necessary to configure SAS BI Web Services for Java to use a different JDK. For more information, see the JavaHome advanced SAS BI Web Services for Java configuration property.

Setting Up Clustering

Clustering is not automatically enabled for SAS BI Web Services for Java because hot deployment is not supported in this type of configuration. A more likely scenario is

to install SAS BI Web Services for Java in a non-clustered environment to develop new Web services. After these new Web services have been iteratively tested, modified, and retested, they can be promoted to a system that can be clustered. Because all SAS BI Web Services for Web services that are generated by Java are stateless, no additional Axis2-specific clustering needs to be configured other than making the Web services available to all nodes in the cluster. This can be accomplished in one of the following ways:

- Duplicate the Web services (and modules) in the Axis2 repository on each node.
- Use an Axis2 URL-based repository so that Web services (and modules) can be configured in a central location. To configure a central repository for Axis2, follow these steps:
 - 1 In the Axis2 configuration file (`sas.biws.war/WEB-INF/conf/axis2.xml`), comment out the following configuration parameters:

```
<parameter name="ServicesDirectory">...</parameter>
<parameter name="ModulesDirectory">...</parameter>
```

- 2 In the Web application deployment descriptor file (`sas.biws.war/WEB-INF/web.xml`), specify the `axis2.repository.url` AxisServlet initialization parameter as follows:

```
<servlet>
  <servlet-name>AxisServlet</servlet-name>
  ...
  <init-param>
    <param-name>axis2.repository.url</param-name>
    <param-value>http://host:port/axis2repos/</param-value>
  </init-param>
</servlet>
```

where *axis2repos* is a reachable path on a Web server that has been set up.

- 3 In this example, a new Web application is created with a context root of *axis2repos*. This application contains the Axis2 static services (and modules) configuration data that needs to be accessible from the previous step. The following illustrates the layout of the application:

```
axis2repos.war/
  modules/
    addressing.mar
    rahas.mar
    rampart.mar
    sas.biws.security.mar
    modules.list
  services/
    myService1.aar
    myService2.aar
    ...
    services.list
```

The `modules.list` file specifies the names of the modules in that directory. For example, it contains the following content:

```
addressing.mar
rahas.mar
rampart.mar
sas.biws.security.mar
```

The `services.list` file specifies the names of the services in that directory. For example, it contains the following content:

```
myService1.aar
myService2.aar
...
```

- 4 After the static files (services and modules) described in the previous step are accessible, clustered instances of SAS BI Web Services for Java (`sas.wip.services9.2.ear/sas.biws.war`) application can access them from this central location.

For more information about clustering applications in your particular environment, see the application server-specific documentation.

Enabling Application Scope

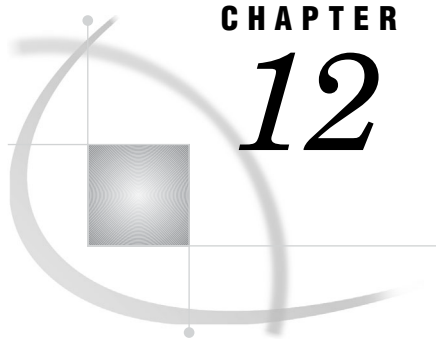
Web services are generated with the default request session scope. This enables clients to generate a Web service, remove that Web service, and regenerate that same Web service after making changes to it without having to worry about old classes that Axis2 might have cached. However, when a Web service is created with application session scope, the service's class is never unloaded even if the service is deleted or removed. In order to avoid this problem, new Web services are generated with request session scope so that a service's class is loaded on each request and classes are not cached. However, after a service is stabilized and is not changing, you might want to manually change the service's session scope to application to gain performance throughput. For example, you might change the `service.xml` deployment descriptor as follows:

```
<serviceGroup>
  <service name="myService" scope="application">
    ...
  </service>
</serviceGroup>
```

Using the Deploy As Web Service Wizard to Overwrite an Existing Web Service

When deploying a Web service with a name that already exists, the user is asked if the original Web service should be overwritten. In this case, the user should cancel this request and instead, delete the original Web service as a separate operation. Choosing to overwrite the existing Web service might not yield expected results if hot update is not enabled, because the Axis2 engine might not detect that the original Web service was actually deleted. The Axis2 engine assumes that hot update is being requested, not hot deployment. The Axis2 engine reads the repository contents every 10 seconds for any changes to its state.

Note: It is not recommended to enable hot update in a production environment because it could result in the system leading to an unknown state. △



CHAPTER

12

Administering SAS Web Application Themes

<i>Overview</i>	153
<i>Introduction to SAS Web Application Themes</i>	154
<i>Theme Components</i>	154
<i>The SAS Default Theme</i>	155
<i>How Custom Themes Are Created and Deployed</i>	155
<i>Steps for Defining and Deploying a New Theme</i>	156
<i>Overview</i>	156
<i>Step 1: Design the Theme</i>	156
<i>Options in Designing the Theme</i>	157
<i>Step 2: Create a Work Area for the Theme</i>	157
<i>Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates</i>	161
<i>Changing Colors</i>	161
<i>Changing Graphics</i>	162
<i>Changing Theme Templates</i>	163
<i>Additional Considerations</i>	163
<i>Step 4: Rebuild SAS Web Application Themes</i>	163
<i>Step 5: Deploy SAS Web Application Themes in Your Test Environment</i>	164
<i>Step 6: Test the New Theme</i>	164
<i>Step 7: Move the New Theme from Test to Production Environment</i>	164
<i>Step 8: Assign the Default Theme</i>	165
<i>Assign the Default Theme from SAS Management Console</i>	165
<i>Assign the Default Theme with the UpdateDefaultTheme.sas Program</i>	165
<i>Deploying SAS Web Application Themes on a Different Web Application Server</i>	165
<i>Modify Theme Metadata from the SAS Management Console</i>	166
<i>Modify Theme Metadata with the UpdateTheme.sas Program</i>	166
<i>Deleting a Custom Theme from the Metadata</i>	167
<i>Migrating Custom Themes</i>	167
<i>Overview</i>	167
<i>Migrating Cascading Style Sheets</i>	167
<i>Migrating Images</i>	168
<i>Migrating Theme Templates</i>	168
<i>Migrating Theme Descriptors</i>	168

Overview

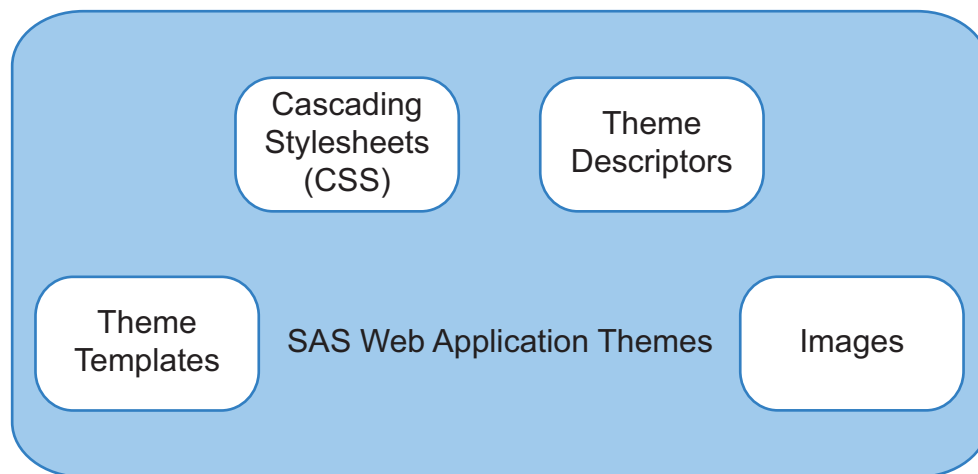
Introduction to SAS Web Application Themes

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.

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 ships 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 **.vtt1**) are provided that contain key/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 the **sas.themes.ear** archive file that you deploy to make 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 of the other themes in the archive). 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 4.2. Use themes to create branding in SAS Web Report Studio 4.2. A few properties for branding that existed in SAS WebReport Studio 3.1 are supported in SAS Web Report Studio 4.2. For information about these properties and usage, see “Customizing Report Styles for SAS Web Report Studio” on page 221. △

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 Design the theme. See “Step 1: Design the Theme” on page 156.
- 2 Create a work area for the theme. See “Step 2: Create a Work Area for the Theme” on page 157.
- 3 Make desired changes to the styles, graphics, and theme templates. See “Step 3: Make Desired Changes to the Styles, Graphics, and Theme Templates” on page 161.
- 4 Register the theme in metadata. See “Step 4: Rebuild SAS Web Application Themes” on page 163.
- 5 Rebuild SAS Web Application Themes and deploy in your test environment. See “Step 5: Deploy SAS Web Application Themes in Your Test Environment” on page 164.
- 6 Test the new theme in a test environment. See “Step 6: Test the New Theme” on page 164.
- 7 Move the new theme from test to production environment. See “Step 7: Move the New Theme from Test to Production Environment” on page 164.
- 8 Assign the default theme. See “Step 8: Assign the Default Theme” on page 165.

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

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 will be 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
```

- for z/OS:

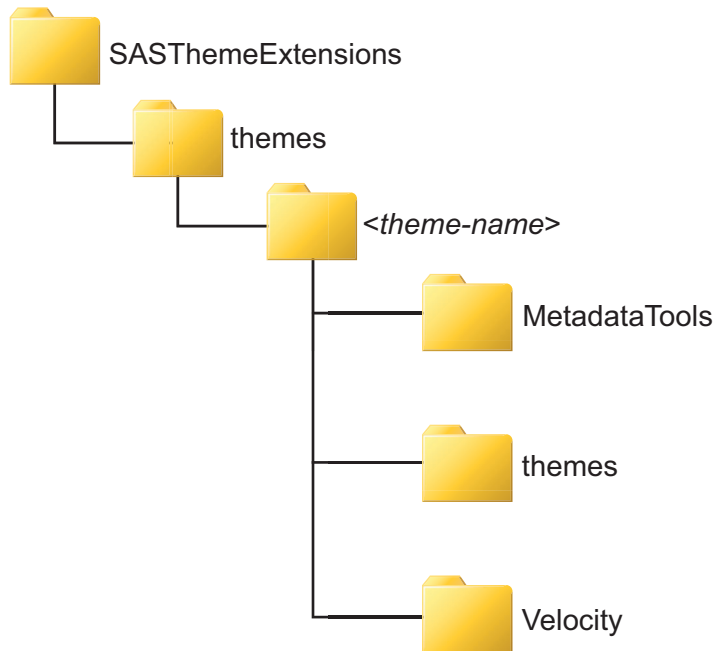
```
NewTheme.sh theme-name true
```

Beginning with the third maintenance release for SAS 9.2, the **NewTheme.sh** script is available for z/OS.

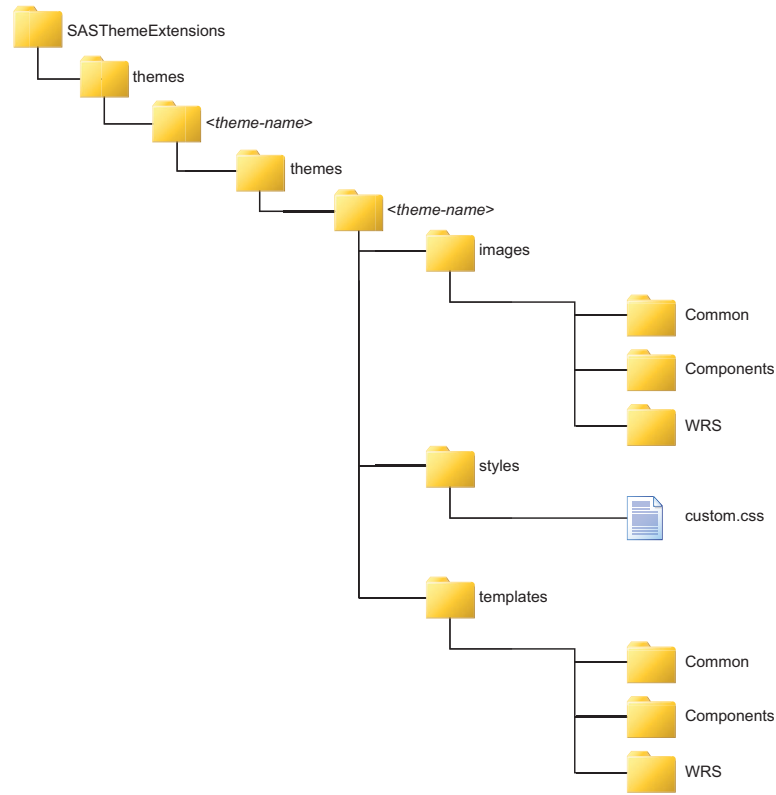
To use the Color Palette option, the **true** parameter is required in the command.

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`

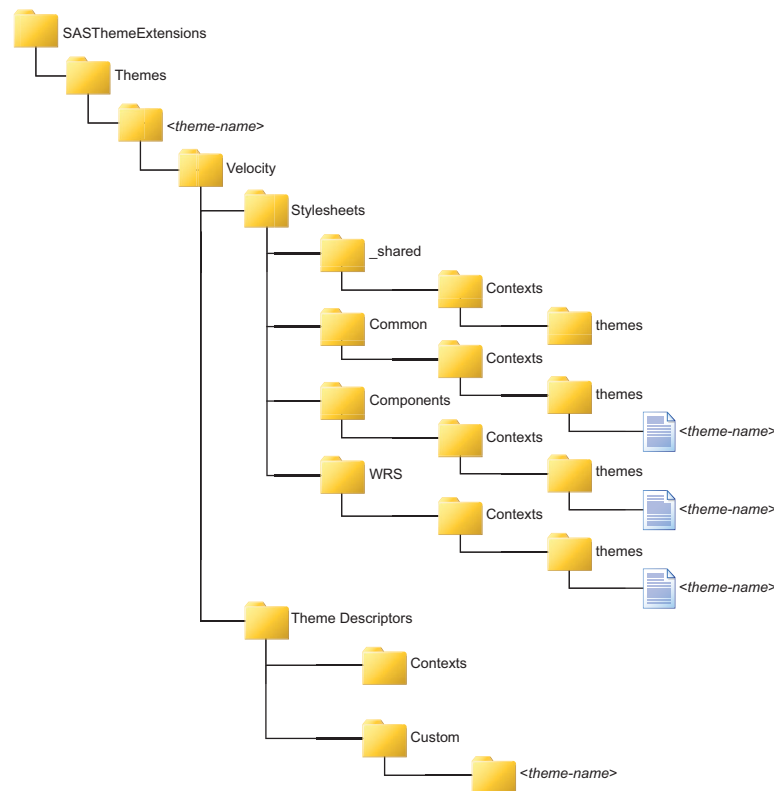
contains theme templates, which are HTML fragments that render specific portions of pages in SAS Web applications. The template files are divided into the following subdirectories by category:

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.vctx* 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.

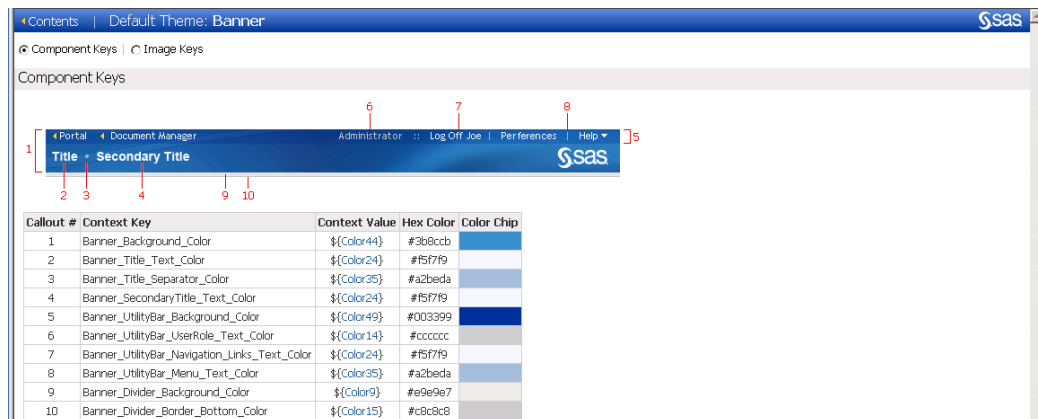
`\theme-name\Velocity\ThemeDescriptors\contexts\custom\theme-name`
contains theme descriptor template files for building the XML files that define the available collections of style sheets, theme templates, and images, including **ComponentsThemes.vtl**, **CustomThemes.vtl**, **SASThemes.vtl**, **SolutionsThemes.vtl**, and **WRSThemes.vtl**.

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

Changing Colors

To make style changes to specific page features, you must first identify the component key associated with that feature and then locate the CSS template file that sets the value for that key. For example, suppose your new theme design calls for changing the color for the title text in the banner at the top of SAS Web applications. The Banner specifications at the Themes Web site *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.



The screenshot shows the 'Default Theme: Banner' page on the SAS Themes Web site. It displays a visual representation of the banner with numbered callouts (1-10) pointing to various components. Below the visual is a table listing the context keys and their values.

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}	#a2bada	
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}	#a2bada	
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.

Changing Graphics

Image files are located in three subdirectories located in the *SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default* folder. These subfolders are: **Common**, **Components**, and **WRS**. The properties of each image are defined in the Theme Descriptors files.

The process for customizing images is similar to that for customizing styles. For example, suppose your new theme design calls for changing the background image for the banner at the top of SAS Web applications. A review of the Banner specifications at *SAS-configuration-*

directory\Lev1\Web\Utilities\SASThemeExtensions\specs\Default\index.html shows that the image key for the banner background is **banner_background**. A search for that string in the work area for the new theme shows the following IMAGE element in the **components.theme-name.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 **components.theme-name.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 **components.theme-name.vtl** context file for which the NAME= attribute has the value **logo**.

When customizing images, you should ensure that the replacement graphics have approximately the same dimensions as the original graphics. Otherwise, the images might disrupt the appearance of the applications in which they are used.

Changing Theme Templates

You should make changes to theme templates only in situations where you want to change the layout of a page element (for example, to tweak 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\context** subdirectory of the work area to provide a descriptive name for the new theme that will appear in the selection list of available themes in the Preferences page in SAS Web applications.

Step 4: Rebuild SAS Web Application Themes

To rebuild the EAR file for SAS Web Application Themes and register your themes in metadata, follow these steps:

- 1 Make sure that the SAS Metadata Server is running.
- 2 Access the SAS Deployment Manager.
 - On Windows, use the shortcut on the Start menu or run the **config.exe** file in the *SAS-installation-directory\SASDeploymentManager\9.2* directory.
 - On UNIX, run the **config.sh** file.
 - On z/OS, run the **config.rexx** file.
- 3 Select the **Rebuild Web Applications** option and click **Next**.
- 4 In the Select Configuration Directory\Level dialog box, specify the configuration directory and the level (for example, **Lev1**) in which you want to rebuild the applications.
- 5 In the Select Configuration Information dialog box, enter the user ID and password for an unrestricted administrative user (for example, **sasadm@saspw**).

- 6 In the Select Web Applications to Rebuild dialog box, select the **SAS Themes** check box.
- 7 In the Summary dialog box, click **Start** to start the rebuild.
- 8 When the rebuild is complete, click **Finish** to close the SAS Deployment Manager.

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 now contain a new Web archive (WAR) file for the new theme named **sas.theme.theme-name.war**.

Step 5: Deploy SAS Web Application Themes in Your Test Environment

To deploy the rebuilt SAS Web Application Themes to your Web application server in a test environment, see “Redeploying the SAS Web Applications” on page 97.

If you chose to configure your Web application server manually or deployed the SAS Web applications manually when you installed SAS 9.2, see your **Instructions.html** generated by the SAS Deployment Wizard.

If you chose to have the SAS Deployment Wizard configure your Web application server and deploy the SAS Web applications automatically, see the manual instructions for your Web application server on the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.

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 161. To view the new theme, log out of the portal. Then log into 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 156 until you are satisfied with the display of the new theme.

If you test the new theme several times, log out of 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 163.
- Rebuild SAS Web Application Themes and deploy to your Web Application Server. See “Step 5: Deploy SAS Web Application Themes in Your Test Environment” on page 164.

- Assign the new theme as the default theme. See “Step 8: Assign the Default Theme” on page 165.

Step 8: Assign the Default Theme

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 165.
- Use the **UpdateDefaultTheme.sas** program. See “Assign the Default Theme with the UpdateDefaultTheme.sas Program” on page 165.

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 the new EAR file by using the appropriate procedures for your Web application server.
- 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 SAS Remote Services and the Web Infrastructure Platform 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-nameMetadataTools* 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.

Deploying SAS Web Application Themes on a Different Web Application Server

Typically, SAS Web Application Themes are deployed along with other SAS Web applications on the same Web application server. If you want to deploy themes to a different Web application server, you should modify the theme metadata.

There are two ways to modify the theme metadata:

- Use SAS Management Console. See “Modify Theme Metadata from the SAS Management Console” on page 166.

- Use the **UpdateTheme.sas** program. See “Modify Theme Metadata with the UpdateTheme.sas Program” on page 166.

Modify Theme Metadata from the SAS Management Console

To deploy SAS Web Application themes to a different Web application server and modify the theme metadata, follow these steps:

- 1 Deploy the new EAR file by using the appropriate procedures for your Web application server.
- 2 In SAS Management Console, navigate to **Application Management ► Configuration Manager**, right-click on *Theme Name*, and select **Properties**.
- 3 On the **Connection** tab, complete the following:
 - Select the communication protocol (either http or https).
 - Enter the host name of the Web application server on which the theme is deployed.
 - Enter the port number of the Web application server.
 - Enter the name of the new theme in the **Service** field.
- 4 Click **OK** to save your changes.
- 5 To enable the new theme to go into effect, restart your Web application server.

Modify Theme Metadata with the UpdateTheme.sas Program

To deploy SAS Web Application themes to a different Web application server and modify the theme metadata, follow these steps:

- 1 Deploy the new EAR file by using the appropriate procedures for your Web application server.
- 2 Locate the **UpdateTheme.sas** program in the *SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\themes\theme-name\MetadataTools* directory.
- 3 Modify the following fields in the **UpdateTheme.sas**:

<pre>%let themeName=<i>Theme Name</i>;</pre>	Specify the name of the theme to update.
<pre>%let hostName=<i>Host Name</i>;</pre>	Specify the host name of the Web application server on which the theme is deployed.
<pre>%let port=<i>Port</i>;</pre>	Specify the port number of the Web application server.
<pre>%let URLPath=<i>base URL</i>;</pre>	Specify the application context root of the new theme as deployed on the Web application server.
<pre>%let protocol=<i>http</i>;</pre>	If you are using Secure Sockets Layer (SSL), specify <i>https</i> instead of <i>http</i> as the protocol for the URL.
- 4 Run the **UpdateTheme.sas** program.
- 5 To enable the new theme to go into effect, restart your Web application server.

Deleting a Custom Theme from the Metadata

To delete a custom-developed theme from the deployment for the SAS Information Delivery Portal, use the **DeleteTheme.sas** program located in the *SAS-configuration-directory\Lev1\Web\Utilities\SASThemeExtensions\themes\theme-name\MetadataTools* directory.

Migrating Custom Themes

Overview

SAS 9.2 includes significant updates to the theme infrastructure. In addition to introducing the templating mechanism, changes were made to enable more SAS Web applications to support themes. To apply a custom theme that you developed for an earlier release to SAS 9.2 Web applications, 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 157.
- 2 Migrate the cascading style sheets used in your theme.
- 3 Migrate the images used in your theme.
- 4 Migrate the theme templates.
- 5 Migrate the descriptors used in your theme.

Migrating Cascading Style Sheets

Before attempting to move any CSS files from an existing theme to the *\themes\theme-name\styles* subdirectory of the work area for the new theme, you should first review the specifications for the SAS 9.2 Default theme at *SAS-configuration-directory\Lev1Web\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 (**.vt1**) 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 (**.vt1**) 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 **a11** for its **PRODUCT=** attribute.

Migrating Images

Before attempting to move any image files from an existing theme to the `\themes\theme-name\images` subdirectory of the work area for the new theme, see the image specifications for the SAS 9.2 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 (`.vt1`) file in the `\Velocity\ThemeDescriptors\contexts\custom\theme-name` subdirectory of the work area for the new theme.

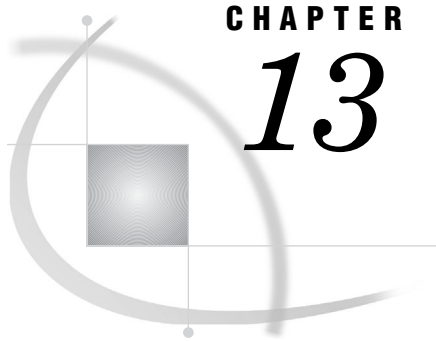
Migrating Theme Templates

Before attempting to move any theme template files from an existing theme to the `\themes\theme-name\templates` subdirectory of the work area for the new theme, you should consider carefully whether they are compatible with SAS 9.2 Web applications. SAS Web applications rely on the theme template structure being consistent with the versions that are shipped with the software. Theme templates must have the expected set of dynamic substitution variables in order for the applications to function properly.

Migrating Theme Descriptors

The theme descriptor template (`.vt1`) 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.



CHAPTER

13

Administering Multicast Options

<i>Overview of Multicasting</i>	169
<i>Configuring Multicast Options</i>	169
<i>Specify Multicast Options for SAS Remote Services</i>	170
<i>Specify Multicast Options for Web Application Servers</i>	170
<i>Key Multicast Properties</i>	170
<i>Configuring Multicast Security</i>	172
<i>Authentication Token for Multicast Security</i>	172
<i>Multicasting with JGroups</i>	173

Overview of Multicasting

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). When installation is performed with the SAS Deployment Wizard, the wizard supplies you with a default multicast address 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.

Multicast options are system properties that are specified as Java command line options on the Web application server to affect the behavior of the system. Multicast options provide the ability to tune and change the behavior of the multicast communication occurring within the SAS system.

The IP address and multicast UDP port number for the multicast host must match the values in the Web application server's startup script (for example, **SASServer1.bat**) and the **environment.properties** file located in the **SAS-configuration-directory\Lev1\Web\Applications\RemoteServices** directory.

Configuring Multicast Options

Multicast options should be changed in a synchronous manner among the following:

- Java virtual machine on which the SAS Remote Services reside. See "Specify Multicast Options for SAS Remote Services" on page 170.
- any Web application server on which a SAS application is deployed. See "Specify Multicast Options for Web Application Servers" on page 170.

- configuration for the Report Output Generation tool (if applicable). For information about this tool, see “Processing Reports Outside of SAS Web Report Studio” on page 248.

Specify Multicast Options for SAS Remote Services

You can make changes to the multicast configuration for the Java virtual machine where SAS Remote Services resides. Edit the appropriate file as needed.

On Windows:

- If you are running the SAS Remote Services from a command prompt, edit the **RemoteServices.bat** file.
- If you use the Window service, modify the **wrapper.conf** file located in the *SAS-configuration-directory\Lev1\Web\Applications* directory.

On UNIX and z/OS, edit the **RemoteServices.sh** file.

Note: Some multicast options for SAS Remote Services are specified in the **environment.properties** file located in the *SAS-configuration-directory\Lev1\Web\Applications\RemoteServices* directory. For more information about the **environment.properties** file, see “About the SAS Environment File” on page 403. △

Specify Multicast Options for Web Application Servers

You can make changes to multicast for any Web application server on which a SAS application is deployed. Edit the appropriate files as needed:

JBoss on Windows:

- Edit the *JBoss\bin\SASServer1.bat* file.
- If you use the Windows service, modify the *JBoss\server\SASServer1\wrapper.conf* file.

On UNIX and z/OS, edit the **SASServer1.sh** file.

For multicast configuration changes to WebLogic or WebSphere application servers, see your product documentation.

Key Multicast Properties

The following table shows some key multicast properties and provides an explanation of these properties.

Table 13.1 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, where X, Y, and Z are the last three octets of the metadata server's IP address. In an IP version 6 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: <ul style="list-style-type: none"> <input type="checkbox"/> encrypt: encrypt but do not require authentication <input type="checkbox"/> 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.

Configuring Multicast Security

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 your Web application server.

Authentication Token for Multicast Security

The multicast protocol is protected with encryption by default because it conveys credentials. By default, group communication is protected only with a fixed encryption key that is built into the software. If your middle tier is not running in an environment that is well isolated from end-user access, then you might want better protection against eavesdroppers and unauthorized group participants. For such situations, choose a multicast authentication token known only to your SAS middle-tier administrative staff.

The authentication token is a password-like string needed to connect to the group and create a site-specific encryption key. The SAS Deployment Wizard simplifies configuration by using the authentication token that is built into the software. This option is best used in development and other low-security environments. It might also be appropriate in higher-security environments where the multicast group communication is isolated from the user community either via firewall or TTL option, and where all data center administrative and operations staff have sufficient security approval.

If your multicast group communication is not contained within an isolated data center environment, or if the security procedures at your site require protections among administrative and operational staff in various roles, use an authentication token that is known only to the administrators of the SAS environment.

By default, there is a code level authentication token shared between all SAS middle-tier applications to prevent access to the multicast group from unauthorized listeners. If you choose to use a customized authentication token, use an authentication token value that meets your organization's security guidelines. The authentication token can be any password-like string. In a multi-tier configuration, the SAS Deployment Wizard displays this prompt for each tier that has an application participating in the SAS multicast groups. The same authentication token string must be specified for each tier in the same SAS deployment (each tier associated with the same metadata server).

Specify the authentication token as a command-line option for your Web application server:

```
-DMULTICAST_AUTHENTICATION_TOKEN=token
```

For information about how to set command line options, see your **Instructions.html** file. The **-Dmulticast** options are specified in the **RemoteServices.bat** file or the **RemoteServices.sh** file.

By default, clients who want to join a multicast group to receive messages are required to provide an authentication token for the join request. If you determine this process is causing an impact on performance, or that it is unnecessary, you can manually turn off the use of authentication tokens. If you specify **NONE** as an option, encryption and authentication are disabled. If you specify **ENCRYPT**, encryption is enabled with no authentication of the join request.

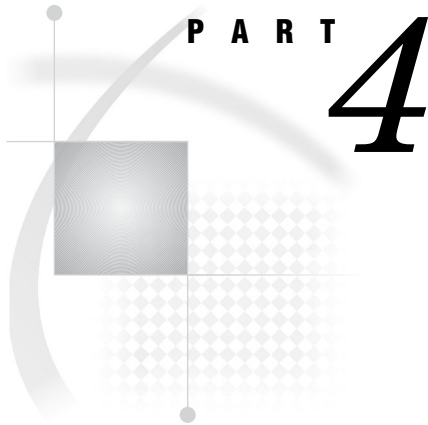
Multicasting with JGroups

SAS middle-tier applications use JGroups in the Web application server for inter-machine communication between applications and caching of application properties. The JGroups software uses IP multicasting as the protocol for group communications, and binds to a valid IP multicast address and UDP port. Typically, multihomed machines have multiple NIC cards, each with its own address. By default, the JGroups software selects the first non-loopback NIC as its bind address. In some cases, the Web application server selects the value of **InetAddress.getLocalHost().getHostName()** as the bind address for use by JGroups.

The communication protocol used by JGroups will not function correctly if the value chosen by JGroups for the SAS Remote Services application and the value chosen by the Web application server for the SAS middle-tier applications on a single machine do not match.

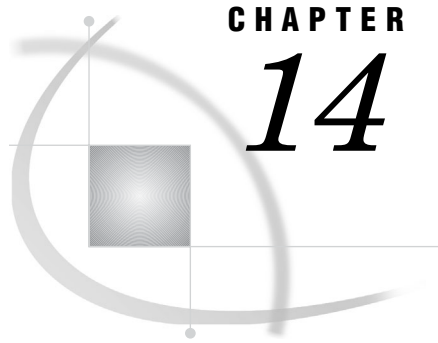
Beginning with the third maintenance release for SAS 9.2, the following JVM command line option is set by default to ensure that the Web application server and the JGroups software use the same bind address:

```
-Djgroups.bind_addr=ipaddress
```

SAS Web Report Studio Administration

<i>Chapter 14</i>	Introduction to SAS Web Report Studio Administration	177
<i>Chapter 15</i>	Configuring SAS Web Report Studio	183
<i>Chapter 16</i>	Managing SAS Web Report Studio Content and Users	197
<i>Chapter 17</i>	Customizing SAS Web Report Studio Report Styles	221
<i>Chapter 18</i>	Pre-generated Reports From SAS Web Report Studio	235



CHAPTER

14

Introduction to SAS Web Report Studio Administration

<i>Introduction to SAS Web Report Studio</i>	177
<i>About SAS Web Report Viewer</i>	178
<i>SAS Web Report Studio and the SAS Intelligence Platform</i>	178
<i>New Features in the Administration of SAS Web Report Studio 4.2</i>	178
<i>Prerequisites for Administering SAS Web Report Studio</i>	180
<i>Main Tasks for Administering SAS Web Report Studio</i>	180
<i>Overview of Main Tasks for Administering SAS Web Report Studio</i>	180
<i>Prepare Report Resources</i>	180
<i>Configure SAS Web Report Studio</i>	181
<i>Implement Security for SAS Web Report Studio</i>	181
<i>Perform Additional SAS Web Report Studio Administration</i>	182
<i>Additional Documentation for SAS Web Report Studio</i>	182

Introduction to SAS Web Report Studio

SAS Web Report Studio is a Web application that enables users to view, interact, create, and distribute both public and private reports. Users can interactively get the information that they need without having to understand a programming language. In addition, SAS predictive analytical results can be used by business professionals across the enterprise via their Web browsers.

The Report Wizard enables novice users to quickly create and distribute basic queries and reports based on either relational or multidimensional data sources in five easy steps. More advanced users can use either additional layout and query capabilities that are available, including the ability to define custom calculations and complex filter combinations, multiple queries, and SAS analytical results into a single document. A gallery of predefined layout choices expedites the report creation process, and an extensive range of advance report components enable users to create and interact with reports from their Web browsers. Reports and the application can be branded to match a corporate style. Reports can be shared with others or kept in private folders based on security settings. Certain pages of reports can be distributed to authorized users via e-mail or a subscription channel.

SAS Web Report Studio runs within a Web application server such as JBoss, Oracle WebLogic, or IBM WebSphere, and requires the SAS BI Reports Service Workspace. It can be invoked from within SAS Information Delivery Portal.

SAS Web Application Themes contain definitions for themes that are used by several SAS Web applications including the SAS Web Report Studio. Themes enable you to create and apply consistent, visual customization and company branding that will be applied to all theme-enabled SAS Web applications. In SAS Web Report Studio, themes apply to the user interface, including the dialog boxes that are used to view, create, edit, and share reports.

SAS Web Report Studio requires the following:

- SAS BI Report Services, which includes the report output generation tool. The report output generation tool (**rptbatch.bat** for Windows and **rptbatch.sh** for UNIX and z/OS) enables you to create pre-generated, static versions of reports from the command line. New parameters have been added to the report output generation tool.
- The SAS BI Report Services Configuration, which creates libraries that are used by SAS Web Report Studio and the report output generation tool. These libraries include the query cache library and the distribution library.

About SAS Web Report Viewer

If SAS Web Report Studio is not deployed at your site, the SAS Web Report Viewer application is deployed and used for viewing reports from SAS Information Delivery Portal.

SAS Web Report Viewer users can view existing reports and make temporary modifications, but they cannot save their changes or create new reports. In these circumstances, users see SAS Web Report Viewer in the application's banner.

You can further control SAS Web Report Viewer capabilities by using the set of predefined roles that are specific to SAS Web Report Viewer. For more information, see “Predefined Roles for SAS Web Report Viewer” on page 205. SAS Web Report Viewer can also be deployed as part of a custom application that is developed by using SAS AppDev Studio.

SAS Web Report Studio and the SAS Intelligence Platform

As an integral part of the SAS Intelligence Platform, SAS Web Report Studio leverages the analytical power of SAS by using the common SAS Open Metadata Architecture, which reduces administration tasks, and SAS Management Console, which enables administrators to perform administration tasks. SAS Web Report Studio uses SAS Information Maps, which are a business view of data created by SAS Information Map Studio, so that you do not have to understand complex data structures and databases. At the same time, SAS Web Report Studio ensures that enterprise data is used consistently. SAS Web Report Studio can leverage the work of analytical SAS tools, such as SAS Enterprise Guide, which makes it easy for a wide range of users to access SAS analytical intelligence. SAS solutions, which are domain-specific applications that are built on the SAS Intelligence Platform, leverage SAS Web Report Studio for reporting.

New Features in the Administration of SAS Web Report Studio 4.2

SAS Web Report Studio 4.2 offers several new features and enhancements that make administration easier, flexible, and convenient. These enhancements are as follows:

- Users' folders for SAS Web Report Studio can be located anywhere below the **SAS Folders** directory in the **Folders** tab of SAS Management Console. Previously, in SAS Web Report Studio 3.1, you were required to use predefined storage folders. This is no longer the case.

- Three predefined roles are available for SAS Web Report Studio with certain capabilities that are assigned to these roles initially. These predefined roles include Report Viewing, Report Creation, and Advanced. You are not required to use predefined roles. You can create roles and capabilities that meet the needs of your organization. You can determine the number of roles to create, which features are available for each role, and control other aspects of role-based behavior.
- The Report Distribution Wizard enables you to create and edit recipient lists by specifying recipient names and e-mail addresses, and channel information within the wizard's dialog boxes.
- When users need to view a report, and are transferred by an external application such as the SAS Information Delivery Portal, those users can be transferred to SAS Web Report Studio (if the application is installed). The functionality offered by SAS Web Report Studio is determined by the role capability assigned to these users.
- You can set security measures to limit SAS Web Report Studio to interact only with information maps that are in designated locations. For example, you might limit the availability of all relational information maps because some of those information maps include row-level permissions.
- As with all SAS applications, logging is accomplished by using the Logging Service Configuration dialog box within the Configuration Manager in SAS Management Console.
- In previous versions of SAS Web Report Studio, the **LocalProperties.xml** file offered the only practical method to override properties and their values. Although this file is available and supported in SAS Web Report Studio 4.2, it is recommended that you use the Configuration Manager in SAS Management Console to configure and set properties for SAS Web Report Studio. The Configuration Manager plug-in offers a consistent interface to set properties for all SAS applications.
- You can add disclaimer text to tables and graphics, and the text will appear below all tables and graphics in reports.
- In previous versions of SAS Web Report Studio, the **WebReportStudioProperties.xml** file was used. Now, the **Advanced** tab in the Web Report Studio Properties 4.2 dialog box is used to specify property names and property values.
- SAS Web Report Studio 4.2 maintains a working area that is hidden from users. This working area, which is located at **/System/Applications/SAS Web Report Studio/Web Report Studio 4.2**, is accessed by using the SAS Management Console. This location might store shared content such as images.
- Banner images are stored in the **/Web Report Studio 4.2/BannerImages** folder. Sample conditional highlighting image files are deployed in the **/Web Report Studio 4.2/ConditionalHighlightingImages** folder.
- In previous releases of SAS software, if client-side pooling was configured, SAS Web Report Studio 3.1 required that the pooling administrator's user name and password be stored in the metadata. With SAS Web Report Studio 4.2, this requirement has been waived. Previously, SAS Web Report Studio 3.1 stored credentials for the Web administrator (saswbadm) and pooling administrator in the **WebReportStudioProperties.xml** file. This is no longer true with SAS Web Report Studio 4.2.
- Beginning with the third maintenance release for SAS 9.2, the log file, **SASBIReportServices4.2.log**, is created when you first run the report output generation tool with default permissions.
- In the third maintenance release for SAS 9.2, you can change the location of the temporary workspace for SAS Web Report Studio and SAS Web Report Viewer.

Prerequisites for Administering SAS Web Report Studio

This documentation assumes that you have successfully installed and configured SAS Web Report Studio. Upon completion of installation, you should follow all of the post-installation steps that are provided in the **Instructions.html** file that is generated by the SAS Deployment Wizard.

For a comprehensive overview of installation, see the *SAS Intelligence Platform: Installation and Configuration Guide*. For instructions about using the SAS Web Report Studio interface, see the product's online Help, which includes the *SAS Web Report Studio User's Guide*.

Main Tasks for Administering SAS Web Report Studio

Overview of Main Tasks for Administering SAS Web Report Studio

After you have installed SAS Web Report Studio, you can perform a number of administrative tasks. For example, you should verify that SAS Information Maps are stored in a location that is accessible to SAS Web Report Studio in order for users to create reports from those information maps. SAS Web Report Studio 3.1 required that information maps be placed in a specific location; this requirement has been waived for SAS Web Report Studio 4.2.

The following sections summarize the administrative tasks that are specific to SAS Web Report Studio.

Prepare Report Resources

Before users can start creating reports, the necessary resources must be prepared and made available:

- Make sure that your data sources have been created.

In SAS Web Report Studio, the term *data source* refers to a SAS Information Map. If you have not already done so, create metadata for your databases and SAS data sets, and then create the information maps that will be used for reports. For details about creating metadata for your raw data, see the *SAS Intelligence Platform: Data Administration Guide*.

For details about creating information maps, see the SAS Information Map Studio online Help.

- Set up storage of reports and report-related objects.

Ensure that resources are stored in appropriate locations, and add folders to the storage structure in a way that facilitates access control of those folders. There are no preset requirements for creating folders. You have the flexibility to organize users' folders in a manner that suits your organizational requirements. For details, see "Overview of SAS Web Report Studio Folders" on page 197.

- Add content for use by report creators.

Make data sources (information maps), stored processes, banner images, fonts, and imported reports available to users of SAS Web Report Studio. See "Adding Content for Use by Report Creators" on page 207.

- Enable geographical maps.

If you want to display your multidimensional data in interactive geographical maps, then you must enable the geographical maps feature. See Appendix 1, “Configuring the ESRI Map Component,” on page 397.

Configure SAS Web Report Studio

Here are some configuration tasks that you might perform:

- Modify, update, or troubleshoot the configuration properties and settings in Web Report Studio 4.2 plug-in to SAS Management Console. See “Use the Configuration Manager to Configure SAS Web Report Studio Properties” on page 183.
- Assign users to SAS Web Report Studio’s predefined roles and capabilities. See “About Predefined Roles” on page 201.

The following configuration tasks are optional:

- Enable unregistered PUBLIC users, who do not have a user definition in the SAS metadata, to access SAS Web Report Studio. In SAS Web Report Studio 3.1, PUBLIC users were allowed access to the metadata server by default, but this has changed in SAS Web Report Studio 4.2. See “Enable Password Management to Provide Access for PUBLIC Users of SAS Web Report Studio” on page 185. Also, see “Manage Access” in the *SAS Intelligence Platform: Security Administration Guide*.
- Enable users to provide DBMS credentials interactively when logging into SAS Web Report Studio. When users enter their user IDs and passwords in SAS Web Report Studio, those credentials are valid for the current session only, because they are stored in memory and not in metadata. See “Provide DBMS Credentials Interactively in SAS Web Report Studio” on page 185.
- Customize SAS Web Report Studio’s pages by controlling the content displayed in the banner. Banner images make it easier for report consumers to identify the report and to distinguish the report from other reports. However, you have the option to customize SAS Web Report Studio’s pages and prevent the product title, report name, and company logo from appearing in the banner for reports. See “Hide Banner Properties for SAS Web Report Studio” on page 187.
- Configure logging for SAS Web Report Studio. The SAS 9.2 Intelligence Platform uses a standard logging facility to perform logging for SAS servers and to track and audit user actions for performance and security reasons. For details, see “Overview of Logging for SAS Web Report Studio” on page 189. Also, see “Administering Logging for SAS Servers” in *SAS Intelligence Platform: System Administration Guide*.
- Improve the performance of SAS Web Report Studio by managing memory and taking advantage of server-side pooling or client-side pooling capabilities. For details, see *SAS 9.2 Intelligence Platform: Application Server Administration Guide*.

Implement Security for SAS Web Report Studio

For general security tasks, see “Middle-Tier Security” on page 39. The following security tasks apply to SAS Web Report Studio:

- Set up users for SAS Web Report Studio.

Enable users to log on and manipulate reports by creating metadata identities for the users. You can also manage users by assigning users to predefined SAS Web Report Studio roles. For details, see “About Predefined Roles” on page 201.

- Manage access to reports.

Restrict access to reports in accordance with your security goals. See “Managing Access to Reports” on page 214.

- Configure a pooling workspace server to enforce row-level security. If your information maps have filters that prevent users from seeing particular rows in tables, then you should set up a separate pooled workspace server for SAS Web Report Studio. Doing so will prevent users from accessing the restricted tables through other methods. See *SAS 9.2 Intelligence Platform: Application Server Administration Guide*.

Note: For additional links to workspace server pooling topics, see “Improving the Performance of SAS Web Report Studio” on page 192. △

- Set up Web authentication.

You can configure SAS Web Report Studio to use Web authentication. For a detailed discussion of different types of authentication and configuration guidelines, see “Authentication Mechanisms” in the *SAS Intelligence Platform: Security Administration Guide*.

- Limit the availability of relational information maps that implement row-level security.

By default, SAS Web Report Studio can interact with information maps regardless of their location within the folder structure. However, you might choose to establish a more controlled environment. See “Limit the Availability of Relational Information Maps That Implement Row-Level Security” on page 212.

Perform Additional SAS Web Report Studio Administration

You might also want to do the following:

- Customize reports.

Add disclaimer text to reports or add custom report styles. For details, see Chapter 17, “Customizing SAS Web Report Studio Report Styles,” on page 221.

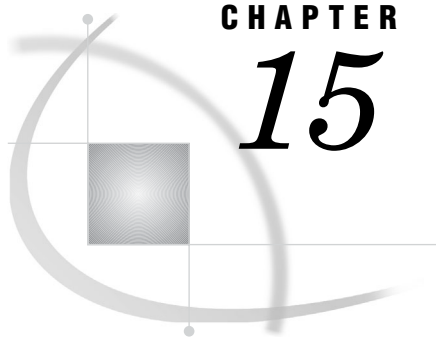
- Set up the scheduling and distribution of reports.

Schedule the creation of pre-generated reports so they will render quickly, and distribute reports to users. For details, see Chapter 18, “Pre-generated Reports From SAS Web Report Studio,” on page 235.

Additional Documentation for SAS Web Report Studio

The following additional documentation is available:

- SAS Web Report Studio online Help provides task instructions and information about the user interface.
- Chapter 3, “Best Practices for Configuring Your Middle Tier,” on page 23 contains information that is associated with middle-tier administration.



CHAPTER

15

Configuring SAS Web Report Studio

<i>Configuring SAS Web Report Studio</i>	183
<i>Use the Configuration Manager to Configure SAS Web Report Studio Properties</i>	183
<i>Configure the Analysis of SAS Web Report Studio Properties</i>	184
<i>Set Maximum Values for Report Filters</i>	184
<i>Enable Password Management to Provide Access for PUBLIC Users of SAS Web Report Studio</i>	185
<i>Provide DBMS Credentials Interactively in SAS Web Report Studio</i>	185
<i>Hide Banner Properties for SAS Web Report Studio</i>	187
<i>Edit LocalProperties.xml File to Set Properties for SAS Web Report Studio</i>	187
<i>Customize the Product and Browser Window Titles for SAS Web Report Studio</i>	187
<i>Change the Temporary Workspace Location</i>	188
<i>Configuring Logging for SAS Web Report Studio</i>	188
<i>Overview of Logging for SAS Web Report Studio</i>	189
<i>Change the Logging Levels</i>	189
<i>Configure Debug Logging Dynamically</i>	189
<i>Manage the Key User Action Log File</i>	190
<i>Understand Key User Action Log Output</i>	191
<i>Report Events in the Key User Action Log</i>	192
<i>Improving the Performance of SAS Web Report Studio</i>	192
<i>Suggestions for Improving the Performance of SAS Web Report Studio</i>	192
<i>Using the Query Cache</i>	193
<i>Overview of the Query Cache</i>	193
<i>Manage Host Access to the Query Cache Directory</i>	194
<i>Change the Location of the Query Cache Library</i>	195
<i>Disable the Query Cache</i>	196
<i>Redeploy SAS Web Report Studio</i>	196

Configuring SAS Web Report Studio

Use the Configuration Manager to Configure SAS Web Report Studio Properties

The Configuration Manager plug-in available in SAS Management Console enables you to perform various administrative tasks such as configuring properties and values for SAS Web Report Studio. In previous versions of SAS software, properties and values were defined in XML files. In SAS 9.2, properties and their values for SAS Web Report Studio are specified on the **Advanced** tab within the Web Report Studio 4.2 Properties

dialog box. For information about how to use the Configuration Manager plug-in to configure properties and values for SAS Web applications, see “Using Configuration Manager” on page 64.

For more information about the SAS Management Console, see “State of Your System” in the *SAS Intelligence Platform: System Administration Guide*.

Configure the Analysis of SAS Web Report Studio Properties

To ensure that a complete analysis of SAS Web Report Studio properties is accomplished at startup, and to add the results of the analysis to the SAS Web Report Studio log as a warning, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2** and right-click to display the Web Report Studio 4.2 Properties dialog box.
- 2 Click on the **Advanced** Tab.
- 3 Click **Add** to display the Define New Property dialog box.
- 4 Enter the property name as shown and specify the property value:
Property Name: wrs.pfs.logPropertiesAsWarn
Property Value: true
- 5 Click **OK** to exit the Define New Property dialog box.
- 6 Click **Add** in the Web Report Studio 4.2 Properties dialog box.
- 7 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 8 To enable the property to go into effect, restart your Web application server.

Set Maximum Values for Report Filters

To configure the maximum number of filter values that can be displayed when report creators define a filter, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2** and right-click to display the Web Report Studio 4.2 Properties dialog box.
- 2 Click on the **Advanced** Tab.
- 3 Click **Add** to display the Define New Property dialog box.
- 4 Enter the property name as shown and specify the property value:
Property Name: webreportstudio.max.filter.choices
Property Value: 1000
- 5 Click **OK** to exit the Define New Property dialog box.
- 6 Click **Add** in the Web Report Studio 4.2 Properties dialog box.
- 7 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 8 To enable the property to go into effect, restart your Web application server.

Report creators can configure their reports to prompt for filter values that are generated dynamically when the report is rendered. To enable this feature, report creators choose the **Prompting users to select values from a list** and the **allow users to query for values** options in the Create New Filter dialog box. When the report is rendered, report viewers click the **Get Values** button to load the values that are available for the filter.

You can configure the maximum number of prompt values that can be loaded when report viewers click the **Get Values** button. The default value is 1,000.

To configure the maximum number of prompt values, and specify the number that you want, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2** and right-click to display the Web Report Studio 4.2 Properties dialog box.
- 2 Click on the **Advanced** tab.
- 3 Click **Add** to display the Define New Property dialog box.
- 4 Enter the property name as shown and specify the property value:
Property Name: webreportstudio.max.prompt.choices
Property Value: 1000
- 5 Click **OK** to exit the Define New Property dialog box.
- 6 Click **Add** in the Web Report Studio 4.2 Properties dialog box.
- 7 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 8 To enable the property to go into effect, restart your Web application server.

For more information about dynamic prompt values, or for instructions about creating a filter, see the online Help for SAS Web Report Studio.

Enable Password Management to Provide Access for PUBLIC Users of SAS Web Report Studio

Unregistered (PUBLIC-only) users can pass through authentication but do not have a user definition in SAS metadata, and they are denied access to the SAS 9.2 metadata server. Previously, in SAS Web Report Studio 3.1, PUBLIC users were allowed access to the metadata server by default.


In order to enable PUBLIC users to access SAS Web Report Studio 4.2, you must access the Web Report Studio 4.2 Properties dialog box in the Configuration Manager within SAS Management Console and modify the value for the **Allow Public Users** field. When PUBLIC users are given access to SAS Web Report Studio 4.2, their history and preferences are retained and saved automatically for future use by SAS Web Report Studio.

Provide DBMS Credentials Interactively in SAS Web Report Studio

Typically, SAS Web Report Studio relies on stored credentials in the metadata in order to provide access to third-party DBMS data. The advantage of this approach is that the access is seamless. Users do not need to know a DBMS account ID and password. The disadvantage of this approach is that it requires that you store individual or group DBMS credentials in the metadata. Any change to the users' passwords in the third-party database (for example, Teradata) must be copied to the stored credentials in the metadata.

As an alternative to storing DBMS credentials for use by SAS Web Report Studio, you can require that users provide DBMS credentials interactively. Each user must supply additional credentials (for example, an Oracle account ID and password) one time in each SAS Web Report Studio session in which the user accesses third-party DBMS data (for example, Oracle tables).

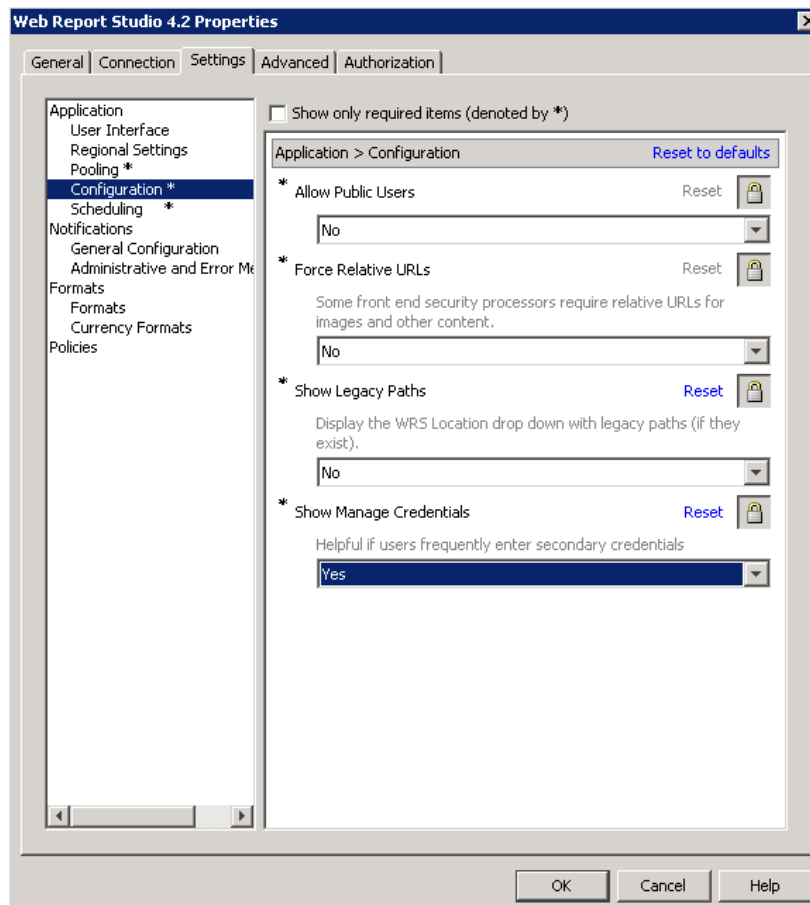
When users enter their user IDs and passwords in SAS Web Report Studio, those credentials are valid for the current session only, because they are stored in memory and not in metadata.

Note: SAS Web Report Studio users who schedule and distribute reports must have stored credentials in the metadata. 

SAS Web Report Studio can accept user-supplied credentials for secondary servers in two ways:

- If a user makes a request that requires DBMS credentials, and those credentials are not stored in the metadata, SAS Web Report Studio always presents the Missing report element dialog box with the **Manage Credentials** button.
- If the administrator configures the **Show Manage Credentials** option in the **Settings** tab within Web Report Studio Properties 4.2, SAS Web Report Studio presents users with the Manage Credentials dialog box. As a result, users can supply the DBMS credentials preemptively when accessing their third-party databases such as Teradata, Oracle, or DB2. For information about entering credentials in SAS Web Report Studio, see the *SAS Web Report Studio 4.2 User's Guide*.

The following figure shows that the administrator has set the value for the **Show Manage Credentials** field to **Yes**.



To allow users to enter their credentials by selecting **File ► Manage Credentials** in SAS Web Report Studio 4.2, follow these steps in the SAS Management Console:

- 1 On the **Plug-ins** tab under Configuration Manager, right-click Web Report Studio 4.2 and select **Properties**.
- 2 In the Web Report Studio 4.2 Properties dialog box, select the **Settings** tab.
- 3 In the Application Configuration dialog box, select **Yes** for the **Show Manage Credentials** field and click **OK**.
- 4 To allow this setting to go into effect, restart SAS Web Infrastructure Platform Services and SAS Web Report Studio applications.

Hide Banner Properties for SAS Web Report Studio

Pages in SAS Web Report Studio can be customized, and the product title, report name, and company logo can be hidden from being displayed in the banner.

To prevent the product title from being displayed in the banner, specify the following property name and property value on the **Advanced** tab within the Web Report Studio 4.2 Properties dialog box:

Property Name: `wrs.banner.product.title.hide`

Property Value: `true`

To prevent the report name from displaying in the banner, specify the following:

Property Name: `wrs.banner.report.name.hide`

Property Value: `true`

To prevent product title, report name, and company logo from displaying in the banner, specify the following property name and value:

Property Name: `wrs.banner.report.hide.all`

Property Value: `true`

To specify that a file with an XHTML fragment be added to the bottom of a report, specify the following property name and value:

Property Name: `wrs.footer.xhtml`

Property Value: `true`

To enable these properties to go into effect, restart your Web application server.

Edit LocalProperties.xml File to Set Properties for SAS Web Report Studio

In previous versions of SAS Web Report Studio, the **LocalProperties.xml** file offered the only practical method to override properties and their values. Although this file is available and supported in SAS Web Report Studio 4.2, it is recommended that you use the Configuration Manager in SAS Management Console to configure and set properties for SAS Web Report Studio. The Configuration Manager offers a consistent interface to set properties for all SAS applications.

If you prefer to create a **LocalProperties.xml** file, follow these steps:

- 1 Locate the sample file in the *SAS-configuration-directory\Lev1\Web\Applications\SASWebReportStudio4.2\customer* directory. The sample file is named **LocalProperties.xml.sample**.
- 2 Make a copy of this file in the *SAS-configuration-directory\Lev1\Web\Applications\SASWebReportStudio4.2\customer* directory, and name the copy **LocalProperties.xml**.

Notes: Changes that you make to **LocalProperties.xml** take effect after you restart your Web application server.

Customize the Product and Browser Window Titles for SAS Web Report Studio

You can replace the default titles for the banner and the browser window in SAS Web Report Studio by customizing the properties in the **LocalProperties.xml** file. You cannot use the Configuration Manager in SAS Management Console to complete this task.

To customize the titles for the banner and the browser window, follow these steps:

- 1 Locate the **LocalProperties.xml.sample** file in the *SAS-configuration-directory\Lev1\Web\Applications\SASWebReportStudio4.2\customer* directory.

- 2 Make a copy of the **LocalProperties.xml.sample** file in the same directory, and name the copied file as **LocalProperties.xml**.
- 3 Edit the **LocalProperties.xml** file by locating the following properties, uncommenting them, and specifying the custom values for the custom banner name and window browser title:

```
<webreportstudio.product.logo.text>
    your custom product title
</webreportstudio.product.logo.text>
<webreportstudio.page.title.text>
    your custom title for the browser window
</webreportstudio.page.title.text>
```

- 4 Save your changes to the **LocalProperties.xml** file.
- 5 To enable these changes to go into effect, restart your Web application server.

Change the Temporary Workspace Location

Beginning with the October 2010 release, the **wrs.io.tmpdir** property for SAS Web Report Studio 4.2 enables you to change the location of the temporary workspace for SAS Web Report Studio and SAS Web Report Viewer.

Here are some default locations for the Web application servers:

- JBoss:

```
JBOSS\jboss-4.2.0.GA\server\SASServer1\work\jboss.web\localhost
\SASWebReportStudio\sas.wrs
```

- WebLogic:

```
C:\SAS\EntBIServer\Lev1\Web\Temp\sas.wrs
```

- WebSphere:

```
C:\Users\userID\AppData\Local\Temp\2\sas.wrs
```

To change the location of the temporary workspace for SAS Web Report Studio 4.2, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2** and right-click to display the Web Report Studio 4.2 Properties dialog box.
- 2 Select the **Advanced** Tab.
- 3 Specify the following property names and property values on the **Advanced** tab within the Web Report Studio 4.2 Properties dialog box:

Property Name: **sas.webreportstudio.cleanup.temp.directory**
Property Value: *Locationoftemporaryworkspace*

Property Name: **wrs.io.tmpdir**
Property Value: *Locationoftemporaryworkspace*
- 4 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 5 To enable these properties to go into effect, restart your Web application server.

Configuring Logging for SAS Web Report Studio

Overview of Logging for SAS Web Report Studio

The SAS 9.2 Intelligence Platform uses a standard logging facility to perform logging for SAS servers. In SAS Management Console, the Logging Service dialog box enables you to manage performance, track security enforcement, and analyze specific situations.

For information about how to use logging for SAS Web applications, see “Administering Logging for SAS Web Applications” on page 105.

SAS Web Report Studio records events in two log files. By default, both log files are created in the *SAS-configuration-directory* \Lev1\Web\Logs directory. If clustering is enabled in your environment, the directory path for the **SASWebReportStudio4.2.log** file can be changed within SAS Management Console.

The following table summarizes the log files:

Table 15.1 Log Files

Log Context and Default Filename	Description
General Purpose Log (SASWebReportStudio4.2.log)	Logs events such as serious errors, application restarts, and users logging on.
Key User Action Log (SASWebReportStudio4.2_KeyActions.log)	Logs events such as application use, failed attempts to log on, report access, and batch report activities. For a list of all events, see “Understand Key User Action Log Output” on page 191.

Note: There are similar log files for SAS Web Report Viewer in the directory. △

Change the Logging Levels

To change the logging levels, see “Administering Logging for SAS Web Applications” on page 105.

Also, see “Administering Logging for SAS Servers” in the *SAS Intelligence Platform: System Administration Guide*.

Configure Debug Logging Dynamically

You can change the logging level for debugging in the Logging Service Properties dialog box as described in “Change the Logging Levels” on page 189. However, that procedure requires you to redeploy and restart SAS Web Report Studio. As an alternative, you can configure debug logging dynamically without restarting SAS Web Report Studio. You can do the following:

- activate a one-line notification for every action that occurs. This one-line message can be useful for providing debugging information about events.
- change the log level for events that are logged for **com.sas.apps.citation** or for one of its descendant contexts.

To implement this functionality, you manually edit the URL for SAS Web Report Studio in your browser. In order to perform this task, you must belong to the Advanced Role with the Manage Distribution List capability for SAS Web Report Studio. For information about SAS Web Report Studio roles and capabilities, see “Predefined Roles” on page 200.

There is no browser-based feedback for this debugging feature. All relevant information about events is placed in the General Purpose Log file.

To edit the URL, do either or both of the following:

- To activate the one-line notification, add the following string to the end of the URL:

```
debugLog.do?LogAllActions=true
```

Here is an example:

```
http://localhost:8080/SASWebReportStudio/debugLog.do?LogAllActions=true
```

Here is an example one-line message that might appear in the log file:

```
WRS 16:25:50,825 WARN report.OpenReportManagerAction
[da8ff705996908f9:14eaec9:107ed50f82f:-7fea]- DEBUG
logging of action requested: OpenReportManagerAction
```

When you have finished debugging, to suppress the one-line notification, change:

```
debugLog.do?LogAllActions=true
```

to:

```
debugLog.do?LogAllActions=false
```

- To change the logging level, add the following to the end of the URL:

```
debugLog.do?LogName=log.context&LogLevel=level
```

Provide the following values:

- Replace *log.context* with the log context that you want to debug. You can specify **com.sas.apps.citation**, or any context under **com.sas.apps.citation**. The **com.sas.apps.citation** context is the highest level, and represents the entire SAS Web Report Studio subsystem.
- Replace *level* with the log level that you want. You can specify any one of the following: DEBUG, INFO, WARN, ERROR, and FATAL.

Here is an example:

```
http://localhost:8080/SASWebReportStudio/
debugLog.do?LogName=com.sas.apps.citation&LogLevel=DEBUG
```

Manage the Key User Action Log File

For the Key User Action Log, SAS uses a rollover mechanism to manage the size and age of the log. SAS periodically archives the current log and creates a new one. To archive a log, SAS saves the log with a new name that includes the current date and time. SAS archives the current log based on configurable settings related to the size of the file and the duration since the last archive. SAS can also delete files after the number of archived files reaches a particular limit.

The following table shows the default values for rollover properties.

Table 15.2 Rollover Properties and Default Values

Rollover Properties	Default Values
<code>sas.wrs.keyUserActionLog.rollover.max</code>	-1
<code>sas.wrs.keyUserActionLog.rollover.sizeKBytes</code>	4000
<code>sas.wrs.keyUserActionLog.rollover.ageHours</code>	24

To manage the Key User Action Log and modify the values for rollover properties, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2** and right-click to display the Web Report Studio 4.2 Properties dialog box.
- 2 Select the **Advanced** Tab.
- 3 Click **Add** to display the Define New Property dialog box.
- 4 Enter the property name and property value and click **OK** in the Define New Property dialog box:

Property Name: `sas.wrs.keyUserActionLog.rollover.max`
Property Value: *Max Rollovers*

Property Name: `sas.wrs.keyUserActionLog.rollover.sizeKBytes`
Property Value: *Threshold Size for Rollovers*

Property Name: `sas.wrs.keyUserActionLog.rollover.ageHours`
Property Value: *Threshold Age for Rollovers*
- 5 Click **Add** in the Web Report Studio 4.2 Properties dialog box.
- 6 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 7 To enable these properties to go into effect, restart your Web application server.

If you are using SAS Web Report Viewer to render reports, then make similar changes for SAS Web Report Viewer. Periodically move or delete outdated archived log files.

Understand Key User Action Log Output

Events are logged to the `WebReportStudio_KeyActions.log` file in an XML format. Each event has a numeric code value that uniquely identifies the event.

The following table lists the events and their respective codes.

Table 15.3 Log Events and Their Codes

Event	Code
User logged on.	0
User attempted to log on but failed.	1
User logged off.	2
User saved a report.	3
User opened a report.	4
User deleted a report.	5
User moved a report.	6
User copied a report.	7
User renamed a report.	8
System start.	9
User scheduled a report. If the user scheduled a folder of reports, then the log file lists the folder.	10
User distributed a scheduled report.	11

Here are some entries from a sample log file:

```
<event><javaDate>1124136823696</javaDate><date>8/15/08</date><time>4:13PM</time><code>9</code><description>System Startup</description></event>
```

```

<event><javaDate>1124136826633</javaDate><date>8/15/08</date><time>4:13PM/
</time><user>saswbadm</user><code>0</code><description>Logon</description>
</event>

<event><javaDate>1124136878587</javaDate><date>8/15/08</date><time>4:14PM/
</time><user>dolson</user><code>0</code><description>Logon</description></event>

<event><javaDate>1124136923432</javaDate><date>8/15/08</date><time>4:15PM/
</time><user>dolson</user><code>4</code><description>Open</description>
<report>/ReportStudio/Shared/Reports/Deanna/Bursting/Orion 2 level bygroup -3
</report></event>

<event><javaDate>1124136977261</javaDate><date>8/15/08</date><time>4:16PM/
</time><user>dolson</user><code>3</code><description>Save</description>
<report>/ReportStudio/Shared/Reports/Deanna/Bursting/testReport</report></event>

<event><javaDate>1124136992808</javaDate><date>8/15/08</date><time>4:16PM/
</time><user>dolson</user><code>2</code><description>Logoff</description>
</event>

```

Report Events in the Key User Action Log

The information in the Key User Action Log can be imported into SAS data sets and presented in reports. To report the data, follow these steps:

- 1 Import the **WebReportStudio_KeyActions.log** data into a SAS data set. Here are the main steps:
 - a Assign a libref to an XML file that contains log data, and specify the XML engine.
 In the following example, **MyFile.xml** is a copy of a Key User Action Log file:


```
libname myxml xml 'C:\My Files\XML\MyFile.xml';
```
 - b Use the SAS data sets procedure to import the XML file into a SAS data set.
 Here is an example:


```
proc data sets library=myxml;
```

This code creates a data set named **EVENT** in the **MYXML** library.
 For more information, see SAS online Help and documentation.
- 2 In SAS Information Map Studio, create an information map based on the data set that you created in the preceding step. For the information map, you might want to provide the ability to filter based on the event code (<code> tag), the user name (<user> tag), the report name (<report> tag), or the date (<date> tag). For information about using SAS Information Map Studio, see the product Help.
- 3 In SAS Web Report Studio, define a report based on the information map that you created in the previous step. You can define the report to be refreshed manually, and then schedule the report to run at regular intervals.

Improving the Performance of SAS Web Report Studio

Suggestions for Improving the Performance of SAS Web Report Studio

To optimize the performance of SAS Web Report Studio, do the following:

- ❑ By default, SAS Web Report Studio is configured with server-side pooling. Server-side pooling is controlled by the server end spawner. Setting up a pool of workspace server processes eliminates the need to start a new process for each user request. Client-side pooling is also available as an option. Both types of pooling enable connections to relational workspace servers. For more information about server-side and client-side pooling, see the *SAS Intelligence Platform: Security Administration Guide*. Also, see *SAS Intelligence Platform: Application Server Administration Guide*.
- ❑ Modify the workspace server startup options to specify a work library, a buffer size for writing files to the work area, and a limit on SAS memory usage. For details, see the *SAS Intelligence Platform: Application Server Administration Guide*.
- ❑ Configure your middle tier as recommended in Chapter 3, “Best Practices for Configuring Your Middle Tier,” on page 23.
- ❑ Make appropriate use of pre-generated reports, such as manually refreshed reports. Use report scheduling to control when pre-generated reports are generated. Pre-generated reports offer faster performance than live reports. You can improve your site performance significantly by increasing the usage of pre-generated reports. You can schedule reports to be generated on a nightly, weekly, or monthly basis. For more information, see “Understanding Pre-generated Reports” on page 236.
- ❑ Use the query cache, which is enabled by default. You can change the location of the cache, and you can disable caching. For more information, see “Using the Query Cache” on page 193.

Using the Query Cache

Overview of the Query Cache

By default, SAS Web Report Studio (and SAS Web Report Viewer) use a large query cache to improve performance. For reports that contain more than one data-driven object, this cache maximizes efficiency. The query cache builds a temporary common data table that can fulfill the needs of all data-driven objects in the report. When the query cache is used, complex queries that include functions such as joins and filters are run only once (to build the common data table). Each data-driven object in the report can then run simple extraction queries against the common data table.

Note: The use of the cache is determined on a per-session basis, depending on the content of each report. In the current release, cache optimization is used only for reports that are based on relational data. △

During installation, the query cache is enabled and is associated with a SAS library. After installation, you can perform the following optional tasks:

- ❑ change the location of the query cache library
- ❑ disable the query cache

Using the query cache will likely increase performance if your reports have the following characteristics:

- ❑ joins from many tables
- ❑ many BY groups
- ❑ multiple report objects

- data sources other than SAS (for example, Teradata or DB2)
- formatted data values from data sources other than SAS

Conversely, using the query cache will *not* increase performance for a report that has all of the following characteristics:

- few joins from few tables
- few BY groups
- few report objects
- only SAS data tables as a source, or non-formatted data values from data sources other than SAS

There is no performance penalty for using the query cache unless the report uses a large native SAS table with report-ready data.

Manage Host Access to the Query Cache Directory

By default, the query cache directory is located at your equivalent of *SAS-configuration-directory\Lev1\SASApp\Data\wrstemp*. In order to protect any sensitive data in the cache, ensure full functionality, and optimize performance, it is important to carefully manage operating system access to this directory. The following table provides details.

Table 15.4 Who Needs Operating System Access to the Query Cache Directory

Server That Retrieves the Data	Account That Accesses the Directory
Server-side pooled workspace server	The launch credential of the server.
Client-side pooled workspace server	Each puddle login.
Standard workspace server	Each requesting user.

Here are some additional notes:

- In a new installation, appropriate initial protections are established for you. However, if you change your server configuration or relocate the directory, make sure that appropriate access is preserved.
 - On UNIX and z/OS, the directory's owner (the SAS Installer account) and the owner's primary group have read, write, and execute permissions for the directory. The SAS Spawnd Servers account (for example, sassrv) should be a member of the SAS installer's primary group. All other users have no access to the directory.
 - On Windows, the directory inherits access from its parent directory. The configuration grants read, write, and modify permissions to the SAS Spawnd Servers account (sassrv).
- As a precaution, a background process deletes any tables that are not cleaned up in the course of a transaction. Host access to the tables is achieved through the pooled server's launch credential or puddle login (if no pooling is involved, the SAS Spawnd Servers account, sassrv, is used).
- We recommend that you specify the same query cache for all SAS application server contexts within a deployment of SAS Web Report Studio.

Note: If you are setting up a restricted client-side pool in order to do secure row-level permissions, designate a separate query cache directory for the additional deployment of SAS Web Report Studio and give only the restricted puddle login account access to the query cache directory. △

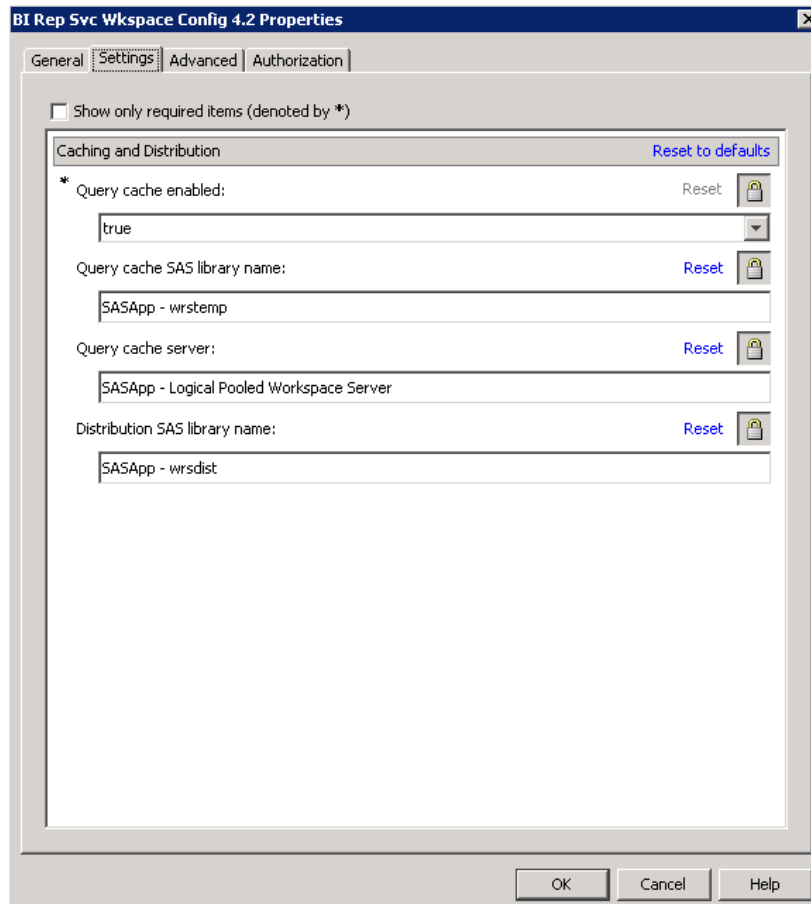
Change the Location of the Query Cache Library

The default location for the library that is used for the query cache is `SAS-configuration-directory\Lev1\SASApp\Data\wrstemp`. After installation, you can specify a different location for this library. For performance purposes, the library should be created on a dedicated fast drive that has plenty of disk space (approximately 100GB, but the needed size will vary based on your system's use and the number of users). Backups are unnecessary because the cache files are temporary. Temporary files are accessed by the relational workspace server. Therefore, the `wrstemp` library should ideally be located on the same server. In addition, RAID configurations decrease performance and are not recommended.

Note: Do not use the WORK or SAS WORK library for this feature. The query cache will not function correctly if you use the WORK or SAS WORK library. △

For clustered environments, the folder for this library needs to be exported to all nodes in the cluster (and you should specify the network address to this folder, not the local machine address). For non-clustered environments, or for a cluster that is restricted to a single physical machine, this folder does not need to be exported.

The following display shows the BI Rep Svc Wkspce Config 4.2 Properties dialog box where the settings for caching are configured:



To view the query cache library, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Application Management ► Configuration Manager**.
- 2 Right-click on **BI Web Services for Java 9.2** and select **Properties**.

- 3 In the BI Rep SVC Wkspce Config 4.2 Properties dialog box, select the **Settings** tab.
- 4 Verify that the appropriate values are specified for each of the following fields:
 - ☐ **Query cache enabled:** true
 - ☐ **Query cache SAS library name:** SASApp-wrstemp
 - ☐ **Query cache server:** SASApp --- Logical Pooled Workspace Server
 - ☐ **Distribution Library:** SASApp-wrsdist
- 5 Click **OK** to exit the dialog box.
- 6 If you modified the value for any property, you should restart the Web application server to enable the properties to go into effect.

Disable the Query Cache

To disable the query cache in the SAS Management Console, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Application Management > Configuration Manager**.
- 2 Right-click on **BI Rep Svc Wkspce Config 4.2** and select **Properties**.
- 3 In the BI Rep SVC Wkspce Config 4.2 Properties dialog box, select the **Settings** tab.
- 4 In the **Query cache enabled** field, select **false**.

If you are certain that you will not use the query cache in the future, modify the **Query cache SAS library name** field by leaving it blank. If applicable, edit the **autoexec_usermods.sas** file to remove the library assignment. If there is a possibility that you will re-enable the query cache, then you should leave the library in place.

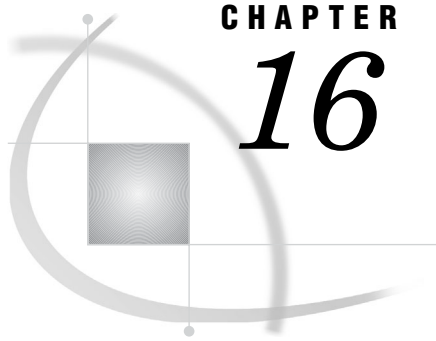
- 5 Click **OK** to exit the dialog box.
- 6 To enable this change to go into effect, restart your Web application server.

Redeploy SAS Web Report Studio

After initial installation, if you make configuration changes, then you should rebuild and redeploy SAS Web Report Studio. You would redeploy SAS Web Report Studio for the following reasons:

- ☐ SAS Web Report Studio is unconfigured and then reconfigured.
- ☐ Workspace servers are changed.
- ☐ Scheduling is configured.
- ☐ The name of the deployed Enterprise archive (EAR) file and context root used to access it is changed.
- ☐ Restrictive policy files are implemented or modified.
- ☐ Hot fix is used.

For information about rebuilding SAS Web applications with the SAS Deployment Manager, and manually redeploying SAS Web applications to a Web application server, see “Rebuilding the SAS Web Applications” on page 94.



CHAPTER

16

Managing SAS Web Report Studio Content and Users

<i>SAS Web Report Studio Folders</i>	197
<i>Overview of SAS Web Report Studio Folders</i>	197
<i>Storage Folders for SAS Web Report Studio Content</i>	198
<i>Display Users' Legacy Path Folders After Migration to SAS Web Report Studio 4.2</i>	199
<i>Predefined Roles</i>	200
<i>About Predefined Roles</i>	201
<i>Predefined Roles and Capabilities for SAS Web Report Studio</i>	201
<i>Predefined Roles for SAS Web Report Viewer</i>	205
<i>Adding Content for Use by Report Creators</i>	207
<i>Add Content for Use by Report Creators</i>	207
<i>Make Data Sources Available to SAS Web Report Studio</i>	207
<i>Make Stored Processes Available to SAS Web Report Studio</i>	207
<i>Make Images Available to SAS Web Report Studio</i>	209
<i>Make Fonts Available to SAS Web Report Studio</i>	210
<i>Make PDF Files Available to SAS Web Report Studio</i>	211
<i>Limit the Availability of Relational Information Maps That Implement Row-Level Security</i>	212
<i>Import Reports That Conform to the SAS Report Model</i>	212
<i>Import Legacy Reports</i>	213
<i>Managing Access to Reports</i>	214
<i>Overview of Managing Access to Reports</i>	214
<i>Change Access to Reports</i>	216
<i>Enable Permissions in Restrictive Policy File for Publishing Reports to Channels</i>	216
<i>Security Considerations for Pre-generated Reports</i>	216
<i>Considerations for Row-level Security</i>	217
<i>Protect Report Content in the WebDAV Server</i>	217
<i>Configure Content Mapping with a WebDAV Location</i>	217
<i>Verify SAS Trusted User's Permissions to Directories in the SAS Content Server</i>	218
<i>Verify SAS Trusted User's Access to the Directories in the SAS Content Server</i>	218
<i>Verify Users' Access to the SAS Directories in the SAS Content Server</i>	219
<i>Protect Data in Temporary Files Created by SAS Web Report Studio</i>	219

SAS Web Report Studio Folders

Overview of SAS Web Report Studio Folders

Proper storage of reports and report-related objects is important because storage of reports (and some report-related objects) must always be synchronized between your

metadata repository and your external content server. In addition, if you organize storage of reports appropriately for your environment, then controlling access to reports is easier.

You have the flexibility to choose the location of the SAS Web Report Studio user folders anywhere within the **SAS Folders** directory on the **Folders** tab of SAS Management Console. For example, you can create a folder called Shared and ensure that users' documents such as reports, information maps, and stored processes are stored in this folder. Another option is to create separate folders for reports, information maps, and other items.

For each folder that you create within the **SAS Folders** directory, a corresponding directory is also automatically created in the SAS Content Server. In this way, SAS Management Console preserves the necessary synchronization between the folders in the metadata repository and the content server.

For information and guidelines about how to set up your SAS folders, see the *SAS Intelligence Platform: System Administration Guide*.

To manage the availability of images, templates, and items for report distribution to users, SAS Web Report Studio administrators have access to the certain folders in the SAS Content Server and within SAS Management Console. These folders, which can be accessed by the SAS administrator, are as follows:

Web Report Studio 4.2/BannerImages

the folder where SAS Web Report Studio looks for banner images when building a report. Banner images are available for users to insert into headers and footers in their documents. You use SAS Management Console to manage the contents of the BannerImages folder.

Web Report Studio 4.2/ConditionalHighlightingImages

the top-level folder for conditional-highlighting images that can be included in reports.

Web Report Studio 4.2/ReportTemplates

the folder for the templates that are used when you create reports with custom layouts in SAS Web Report Studio.

Web Report Studio 4.2/Scheduling/DistributionDefs

the folder for items that handle report scheduling and distribution.

Storage Folders for SAS Web Report Studio Content

Storage folders for SAS Web Report Studio must exist in both the foundation SAS Metadata Repository *and* the SAS Content Server or the file system. The **SAS Folders/System/Applications/SAS Web Report Studio/Web Report Studio 4.2** folder in the metadata repository corresponds to the **sasfolders/System/Applications/SAS Web Report Studio/Web Report Studio 4.2** directory in the SAS Content Server. These top-level folders are created during installation, configuration, and deployment.

The parallel storage structures are necessary because reports and some report-related objects (such as images) have both a metadata component and a content component. For example, for each report that is saved in SAS Web Report Studio, two objects are stored:

- A metadata object that describes the report is stored in your metadata repository. The report metadata object contains information such as time stamps, authorship, access controls that provide security for the report, and other report-specific and application-specific properties.
- A report definition file is stored in your content server. The report definition file is an XML file that contains information about how the report is presented and what data is included in the report. The report definition is constructed according to the SAS Report Model, which is an XML specification for business reports. Reports

that comply with the SAS Report Model can be created, viewed, and modified by a variety of SAS applications.

In order to display a report, SAS Web Report Studio must retrieve both of these components. The parallel storage structures in the metadata repository and the content server facilitate this two-part retrieval.

CAUTION:

You must keep the report content files synchronized with their corresponding metadata objects. △

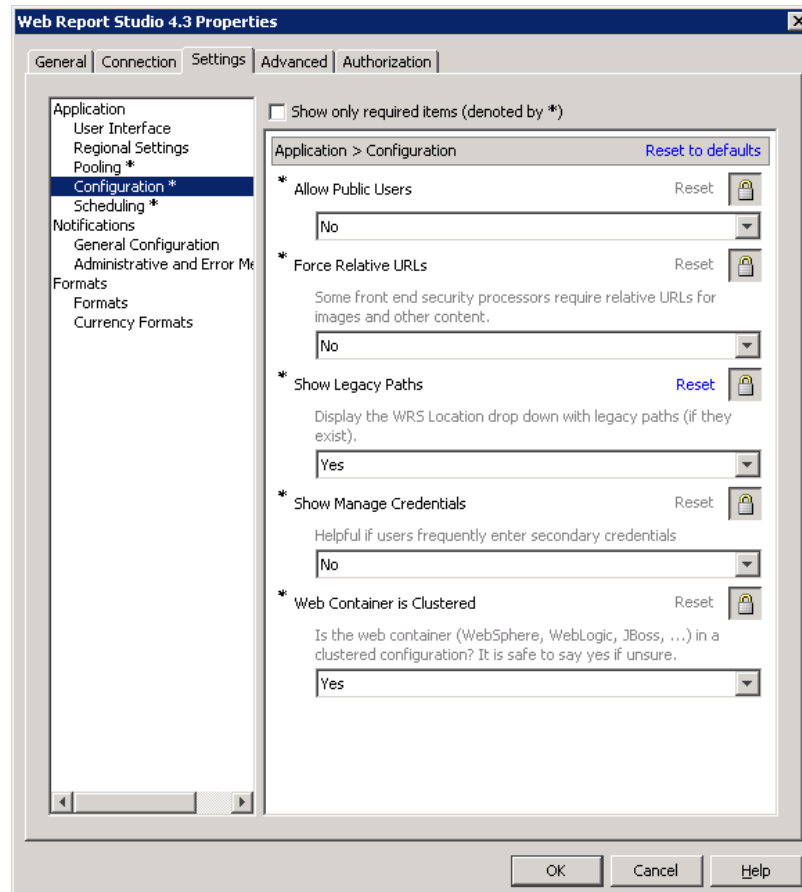
For details about how to keep report content files synchronized with their corresponding metadata objects, see “Best Practices for Managing SAS Folders” in the *SAS Intelligence Platform: System Administration Guide*.

Display Users' Legacy Path Folders After Migration to SAS Web Report Studio 4.2

If your site has migrated from SAS Web Report Studio 3.1 to SAS Web Report Studio 4.2, previous users of SAS Web Report Studio can navigate to their folders, which are located within the **SAS Folders** directory.

The following display shows the Web Report Studio 4.2 dialog box where the **Show Legacy Paths** field is set to **Yes** by default after migration to SAS Web Report Studio 4.2. If there was no migration, this field is set to **No** by default.

Display 16.1 Show Legacy Paths for SAS Web Report Studio



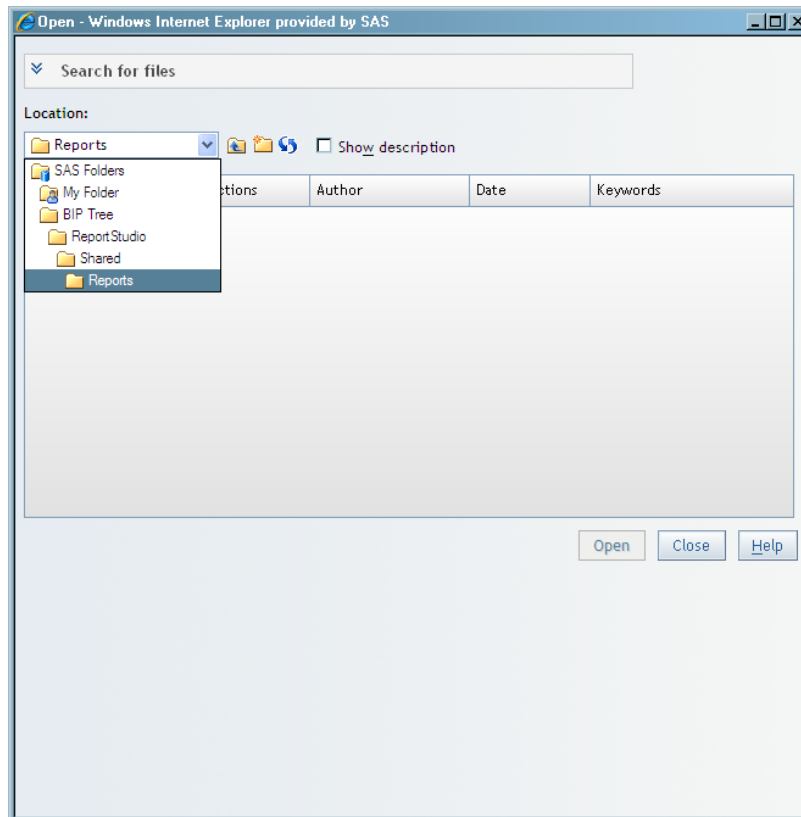
The legacy path folders are displayed in the drop-down **Location** menu in SAS Web Report Studio. The following table shows the legacy folder paths and types of files that are typically displayed when the drop-down menu is chosen under **Location**.

Table 16.1 Legacy Folder Paths in SAS Web Report Studio 4.2

Function in SAS Web Report Studio 4.2	Legacy Folder Paths
Open Report	/BIP Tree/ReportStudio/Shared/Reports
Open Info Map	/BIP Tree/ReportStudio/Maps
Locate Image to Include	/BIP Tree/ReportStudio/Shared/Images

In the following display, the user has chosen to open a report and is able to view the legacy folder path for **/BIP Tree/ReportStudio/Shared/Reports**.

Display 16.2 Reports in the Legacy Folder Path



Predefined Roles

About Predefined Roles

To enable the availability of specific capabilities provided by SAS Web Report Studio and SAS Web Report Viewer users, each user can be assigned to one or more predefined roles.

Note: You are not required to use predefined roles. However, it is recommended that you do not modify predefined roles. You can create roles to meet your own requirements, and assign capabilities to those roles. △

Predefined Roles and Capabilities for SAS Web Report Studio

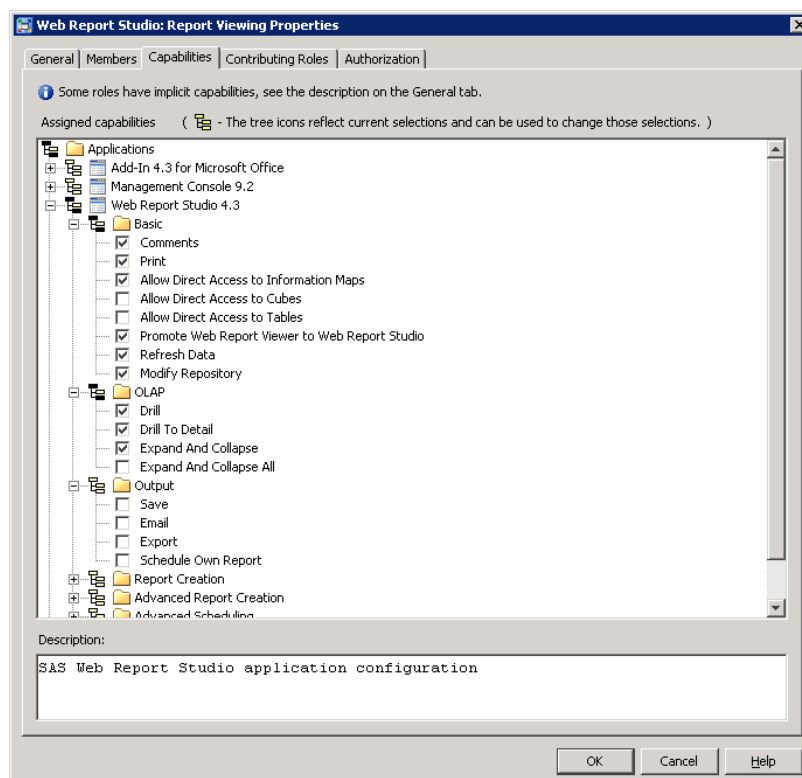
SAS Web Report Studio includes three predefined roles with certain capabilities that are assigned to these roles initially. These predefined roles are:

- ☐ Report Viewing
- ☐ Report Creation
- ☐ Advanced

You are not required to use predefined roles. You can create roles and capabilities that meet the needs of your organization.

The following display of the Web Report Studio: Report Viewing Properties window shows the capabilities that are assigned initially to the predefined role, Report Viewing. This window is accessed by navigating the SAS Management Console to the User Manager and selecting the **Capabilities** tab in the Web Report Studio: Report Viewing role window.

Display 16.3 Capabilities for the Report Viewing Role



The following table shows three predefined roles, and capabilities that are initially assigned to these roles in SAS Web Report Studio. Capabilities that are assigned initially to these roles are indicated by an X. You can modify the original settings, assign additional capabilities, or remove capabilities from these roles.

Certain capabilities are duplicated across the three predefined roles. For example, all of the assigned capabilities for Report Viewing are also available in the Report Creation and Advanced roles.

Note: If a user has been assigned multiple+ roles and multiple capabilities, removing a capability from one role does not negate that capability if the user was assigned that capability in a different role. For example, if John was assigned Print capability in all three roles (Report Viewing, Report Creation, and Advanced), removing the Print capability from the Advanced role alone does not prevent John from using that capability. In order to completely remove the Print capability for John, it must be deselected for all three roles to which John belongs. Alternatively, John can be assigned to a single role where the Print capability is not assigned. \triangle

Table 16.2 Roles and Capabilities for SAS Web Report Studio

Capability	Report Viewing	Report Creation	Advanced
Basic			
Comments	X	X	X
Print	X	X	X
Open Maps as Reports	X	X	X
Promote WRV to WRS	X	X	X
Refresh Data	X	X	X
Modify Repository	X	X	X
OLAP			
Drill	X	X	X
Drill to Detail	X	X	X
Expand and Collapse	X	X	X
Expand and Collapse All		X	X
Output			
Save		X	X
Email		X	X
Export		X	X
Schedule Own Report		X	X
Report Creation			
Create Report		X	X
Basic Edit		X	X
Aggregate or Detail			X
Select All Data		X	X
Select Data in View		X	X

Capability	Report Viewing	Report Creation	Advanced
Advanced Report Creation			X
Create Cascade Reports			X
Create Report Links			X
Advanced Edit			X
Repair Report			X
Advanced Scheduling			X
Distribute			X
Save Archive			X
Schedule Any Report			X
Schedule Folder			X
Administrative			
Manage Distribution List			

In order to be functional, some capabilities have prerequisites. Following is an explanation of each capability and prerequisites for certain capabilities:

Basic:Comments

provide access to Comment Manager.

Basic:Print

generate PDF output.

Basic:Open Maps as Reports

open Information Maps directly as reports. Without this capability, the information map is suppressed from the Open and Manage pages. With this capability, authors have an Edit dialog box in addition to the Open box.

Basic:Promote Web Report Viewer to Web Report Studio

displays the SAS Web Report Studio interface when a user requests to view a report from another product such as SAS Information Delivery Portal. Initially, this capability is assigned to all predefined roles. If this capability is unassigned, and a user's request to view a report from within another product is granted, then the SAS Web Report Viewer interface is used to present the report. When this capability is assigned to a role to which a user belongs, the SAS Web Report Studio interface is displayed when that user requests to view a report through the SAS Information Delivery Portal or other products.

Basic:Refresh Data

refresh data for reports.

Basic:Modify Repository

provide basic modifications to repository such as copy, move, rename, and delete. This capability does not grant the ability to save changes to reports.

OLAP:Drill

perform an OLAP drill operation.

OLAP:Drill to Detail

drill through to detail data for a single OLAP value.

OLAP:Expand and Collapse

expand or collapse OLAP data.

OLAP:Expand and Collapse All

expand or collapse an entire OLAP hierarchy. This capability requires the **Report Creation:Basic Edit** capability and the **OLAP:Expand and Collapse** capability.

Output:Save

save changes to reports. This capability requires that the user also has **Modify Repository** capability.

Output:E-mail

e-mail a report link.

Output:Export

export report content to Excel. This capability requires that the user also has the **Report Creation:Basic Edit** capability.

Output:Schedule Own Report

schedule reports authored by the users themselves. The **Schedule All Reports** capability is available in the SAS Web Report Studio's Advanced role.

Report Creation:Create Report

create new reports. The **Save** capability is a prerequisite for this capability.

Report Creation:Basic Edit

perform basic editing operations for tables and graphs. Operations include:

- ☐ Assign Data
- ☐ Total dialog box access
- ☐ Control of Total Type for OLAP reports only (pre-summarized versus visual)
- ☐ Percent of Total
- ☐ Filter and Rank
- ☐ Conditional Highlighting
- ☐ Rotate Table
- ☐ View Data Details
- ☐ Sort, Sort Priority, Remove All Sort
- ☐ Move
- ☐ Hide
- ☐ Replace / Swap
- ☐ All Table Properties
- ☐ All Graph Properties

Report Creation:Aggregate or Detail

specify whether data is detail data or aggregated data (relational reports only)

Report Creation:Select All Data

able to choose "Select All" data items from the Select Data dialog box.

Report Creation:Select Data in View

use the Select Data dialog box to select data while viewing a report.

Advanced Report Creation:Create Cascade Prompts

create cascading prompts.

Advanced Report Creation:Create Report Links

create linked reports.

Advanced Report Creation:Advanced Edit

enable users to perform the following functions:

- ☐ Isolate
- ☐ Member Properties

□ Suppress Empty

The Report Creation: Basic Edit capability is a prerequisite for this capability. In SAS Web Report Studio, the **Include Member with Only Missing Values** menu item is available only for multidimensional data sources. The Suppress Empty function in the Advanced Edit capability is required in order for users of multidimensional data sources to view and use the **Include Member with Only Missing Values** menu item under the **Data Menu** in SAS Web Report Studio.

Advanced Report Creation:Repair Report

update an invalid information map reference that is contained in a report definition.

Advanced Scheduling:Distribute

schedule a report distribution.

Advanced Scheduling:Save Archive

archive file versions. When creating a pre-generated version of a report (for example, a scheduled report), a version of the report is archived as a PDF file. Reports with archived versions are visually different in report selection dialog boxes, and access is allowed to these archived versions.

Advanced Scheduling:Schedule Any Report

schedule any report to which the users have WriteMetadata access, including the reports that they have authored.

Advanced Scheduling:Schedule Folder

schedule a folder.

Administrative:Manage Distribution List

create, edit, or delete a distribution list.

CAUTION:

Use of this capability creates a new physical table. Therefore, this capability must be restricted to few users. The distribution list contains e-mail addresses. Initially, this capability is not assigned to any SAS Web Report Studio role. An administrator can assign this capability to the Advanced role, or create a new role (for example, a role named as Manage Distribution) specifically for this capability and assign this role to a restricted number of users. △

Predefined Roles for SAS Web Report Viewer

SAS Web Report Viewer includes two predefined roles with certain capabilities assigned initially to these roles. These predefined roles are:

- Report Viewing
- Advanced

SAS Web Report Viewer does not allow users to create or modify reports.

The following table shows the two predefined roles for SAS Web Report Viewer. Capabilities that are initially assigned to these roles are indicated by an X. You can unassign existing capabilities or reassign capabilities to these roles. Unlike SAS Web Report Studio, SAS Web Report Viewer has a smaller subset of capabilities available for roles.

Table 16.3 Roles and Capabilities for SAS Web Report Viewer

Capability	Report Viewing	Advanced
Basic		
Comments	X	X
Print	X	X
Open Maps as Reports	X	X
Refresh Data	X	X
OLAP		
Drill	X	X
Drill to Detail	X	X
Expand and Collapse	X	X
Expand and Collapse All		X
Output		
E-mail		X
Export		X

Basic:Comments

provide access to Comment Manager.

Basic:Print

generate PDF output.

Basic:Open Maps as Reports

open Information Maps directly as reports. Without this capability, the information map is suppressed from the Open and Manage pages. With this capability, authors have an Edit choice dialog box in addition to the Open box.

Basic:Refresh Data

refresh data for reports.

OLAP:Drill

perform an OLAP drill operation.

OLAP:Drill to Detail

drill through to detail data for a single OLAP value.

OLAP:Expand and Collapse

expand or collapse operation for OLAP data.

OLAP:Expand and Collapse All

expand or collapse an entire OLAP hierarchy. This capability requires the Report Creation:Basic Edit capability and the OLAP:Expand and Collapse capability.

Output:E-mail

e-mail a report link.

Output:Export

export report content to Excel. This capability requires that the user has the Report Creation:Basic Edit capability.

Adding Content for Use by Report Creators

Add Content for Use by Report Creators

The resources that can be used as inputs to SAS Web Report Studio are information maps, stored processes, banner images, fonts, and existing reports from other locations. SAS Web Report Studio uses the metadata server to access these resources, so these resources must be registered in the metadata repository. The following sections describe how to add this metadata to the repository.

Typically, the addition and changes of content is accomplished from within SAS Web Report Studio. Alternatively, you can also add content from within SAS Management Console.

Note: If you plan to use geographical maps in reports, then see “About the ESRI Map Component” on page 397. △

Make Data Sources Available to SAS Web Report Studio

In SAS Web Report Studio, users do not interact directly with SAS data sets, third-party relational database tables, or SAS OLAP cubes. Instead, users interact with information maps that provide a business view of the underlying data. Information maps are created with SAS Information Map Studio or the INFOMAPS procedure.

In SAS Web Report Studio, the term *data source* refers to an information map. In a report that is created with SAS Web Report Studio, each section can use only one information map. However, you can have multiple sections per report.

Information maps can exist in any folder with one exception. If the special map accessibility check is selected in the Web Report Studio 4.2 Properties dialog box (typically used for sites with row-level permissions) and the information map is relational, then the map must reside below one of the folders specified by the administrator. In this case, the administrator can restrict SAS Web Report Studio to use only relational information maps from known locations.

To enable the special map accessibility check, see “Limit the Availability of Relational Information Maps That Implement Row-Level Security” on page 212.

In order for an information map to appear in the list of data sources in SAS Web Report Studio, the information map must meet all of these criteria:

- The information map must be stored in the main **Maps** folder or in one of the subfolders of that folder.
- The user of SAS Web Report Studio must have both Read and ReadMetadata permission to the information map.

Make Stored Processes Available to SAS Web Report Studio

Including a stored process in a report section is one of the ways in which to obtain data for the report. Each report section can contain multiple stored processes. When the report section is rendered, the output of each included stored process is displayed. Users can also run stored processes directly from within SAS Web Report Studio.

You cannot use SAS Web Report Studio to modify the query that is generated from the stored process, but you can use SAS Web Report Studio to add layout elements such as headers, footers, images, and text that are independent of the stored process output.

You can convert existing SAS programs into stored processes for use in SAS Web Report Studio. The programs can be parameterized, which enables users to input data in response to prompts. Prompted parameter values are transferred to the stored process as macro variables. To convert an existing program to a stored process, follow these steps:

- 1 Insert a `*PROCESSBODY` statement.
- 2 Insert a `%STPBEGIN` statement before a section of the code that produces output.
- 3 Insert a `%STPEND` statement after a section of the code that produces output.

For example, to alter this SAS program:

```
%let year=2002;
title "Sports & Outdoors Sales &year";
proc print data=sashelp.orsales;
    where year=&year;
run;
```

to become a stored process, change the code to look like this:

```
%global year;
*processbody;
%stpbegin;

title "Sports & Outdoors Sales &year";
proc print data=sashelp.orsales;
    where year=&year; /* &year is a parameter from a user prompt */
run;

%stpend;
```

Stored process output that will be included in a report must be generated through the Output Delivery System (ODS). Output that is generated in other ways, such as with PUT statements, is not accessible from SAS Web Report Studio. The `%STPBEGIN` and `%STPEND` macros in the stored process code ensure that ODS is used to generate the output.

The ODS output type for each stored process is determined by the manner in which the stored process is registered and executed. The ODS output type should not be controlled by making changes to the stored process code (neither by setting the value of the stored process input parameter `_RESULT`, nor by explicit ODS statements).

The following table indicates how the style is determined for stored process output.

Table 16.4 Style of Stored Process Output

Type of Output	Example	Style of Output
ODS text output	PROC PRINT listing	The style is determined by the user's preferences in SAS Web Report Studio. For example, "Seaside."
ODS graphical output	PROC GCHART graphs	The default is an ActiveX device.* ActiveX also uses the default style such as Seaside. GOPTIONS DEVICE=ACTIVEX;

* By default, the ACTIVEX device driver is used for graphs in stored process output. This format requires users to install a graph control on their local system in order to render the graph. However, to maintain a zero footprint on the client, SAS Web Report Studio does not require

this installation. Therefore, when the stored process is run in SAS Web Report Studio, the ACTXIMG device driver is substituted so that a static image is created. Similarly, if the JAVA device driver is specified, then the JAVAIMG device driver is substituted automatically.

To make a stored process available to users of SAS Web Report Studio, follow these steps:

- Register the stored process in the metadata by using either SAS Management Console or SAS Enterprise Guide.
- In SAS Management Console, navigate to the stored process under **SAS Folders**. On the **Authorization** tab for the stored process properties, verify that your SAS Web Report Studio users have ReadMetadata access to the stored process.

To learn more about stored processes, see "SAS Stored Processes" in *SAS 9.2 Stored Processes: Developer's Guide*.

Make Images Available to SAS Web Report Studio

Each report that is created in SAS Web Report Studio can include one or more images. The types of images that report creators can use are described in the following table.

Table 16.5 Images for SAS Web Report Studio

Type of Image	Details and Defaults
Banner images	<p>Any report can include a banner image in the header and footer of the report. Banner images make it easier for report consumers to identify the report and to distinguish the report from other reports. Banner images are stored in SAS Folders/System/Applications/SAS Web Report Studio/Web Report Studio 4.2/BannerImages.</p> <p>By default, the BannerImages folder is empty. Use the Folders tab in SAS Management Console to manage your banner images.</p>
Conditional highlighting images	<p>A report that includes tables can use images to draw attention to items that might be of particular interest to report consumers. A report creator can define conditions and, for each condition, select an image that is displayed in every table cell where the condition is met. Conditional highlighting images are stored in SAS Folders/System/Applications/SAS Web Report Studio/Web Report Studio 4.2/ConditionalHighlightingImages.</p> <p>Use the Folders tab in SAS Management Console to manage your conditional highlighting images.</p>
Other images	<p>Any report can include additional images for decorative or other purposes. These images can be stored under the SAS Folders/Shared/Images folder or in any other folder of the user's choice.</p> <p>By default, the Images folder is empty. Use SAS Web Report Studio to add images to this folder. For instructions, see the Help for SAS Web Report Studio. Alternatively, you can use SAS Management Console to add images to the folder. For instructions, see the Help for SAS Management Console.</p>

To make a banner image or conditional highlighting image available to users of SAS Web Report Studio, follow these steps in SAS Management Console:

- 1 Make sure that the SAS Content Server is running.

- 2 In SAS Management Console, navigate to the appropriate images folder below **SAS Folders**.
- 3 Right-click on the appropriate images folder to display a menu.
- 4 From the menu bar, select **Add Content From External File(s) or Directories**.
- 5 Select the file (or files) that you want to import and then click **Open**.

Note: If you select a folder, the folder and its contents are recursively imported. In SAS Web Report Studio, banner images or conditional highlighting images that are stored in subfolders of the **BannerImages** and **ConditionalHighlightingImages** folders are displayed in a single drop-down list. △

- 6 In the **Enter description** text box, enter the description that you want to be displayed for the graphic in SAS Web Report Studio. Image descriptions should be fewer than 20 characters.
- 7 Click **OK** to close the **Enter Description** text box. The imported images are available in SAS Web Report Studio within 10 minutes.
- 8 To make the images available immediately, restart the Web application server.

If an existing image is later modified, you can reimport the new image by using the preceding instructions. SAS Web Report Studio will detect and use the updated image.

To delete an image so it is no longer available to users of SAS Web Report Studio, follow these steps:

- 1 In SAS Management Console, navigate to the appropriate images folder below **SAS Folders** and select the image or images that you want to delete.
- 2 Select the image files to be deleted, and right-click to display a menu.
- 3 From the menu, select **Delete** and right-click.

Note: The minimum image resolution that is supported for clients (browsers) is 1024 x 768. △

Make Fonts Available to SAS Web Report Studio

You can customize the fonts that are available for tables and graphs in the report. SAS Web Report Studio (and SAS Web Report Viewer, if it is installed) uses the default fonts that are loaded from the following files:

- The **ServerFonts.xml** file lists fonts that are rendered on the server. These are the fonts that are available for graphs in a report. The fonts that are listed in this file should be installed on the middle-tier server where SAS Web Report Studio is deployed.
- The **ClientFonts.xml** file lists the fonts that are rendered on the client (user's) system. These fonts are available for tables, headers, and other text. These fonts should be installed on the client system where the browser is running.

The **ServerFonts.xml** and the **ClientFonts.xml** files are not modifiable. To supply custom font definitions, create and edit the following files:

- **LocalServerFonts.xml**
- **LocalClientFonts.xml**

You can create these files from the **LocalServerFonts.xml.sample** and **LocalClientFonts.xml.sample** files that reside in the *SAS-configuration-directory* \Lev1\Web\Applications\SASWebReportStudio4.2\customer folder. To create the files:

- 1 Open **LocalServerFonts.xml.sample** and save it using the name **LocalServerFonts.xml**.
- 2 Open **LocalClientFonts.xml.sample** and save it using the name **LocalClientFonts.xml**.

Each sample file contains information about adding fonts. Here is the general format for the font information in the file:

```
<?xml version="1.0" encoding="UTF-8"?>
<font>
  <font actualfont="Arial" displayfont="Arial" />
  <!-- more fonts -->
</font>
```

In the preceding code sample, the **actualfont** attribute is the font name that is stored in the report. The value for this attribute should match the name of the font on the system. If they differ, a font substitution can occur. The **displayfont** attribute is the font name that is displayed to users.

Note: XML tags, such as ****, are case-sensitive, and should be specified exactly as shown. △

In the UNIX and z/OS environments, fonts must be installed correctly and loaded by the JVM in order for SAS Web Report Studio to render them correctly. Both custom fonts, as well as true type fonts supplied by SAS, should be installed in the **jre.../lib/fonts** directory. See your UNIX or z/OS documentation for commands that apply to font installations.

If you plan to use SAS Web Report Viewer to render reports, create and edit the **LocalServerFonts.xml** and **LocalClientFonts.xml** file for SAS Web Report Viewer in a similar way. Make a copy of each respective sample file in the *SAS-configuration-directory\Lev1\Web\Applications\SASWebReportViewer4.2\customer* directory and add your fonts to the copy.

SAS Web Report Studio (and SAS Web Report Viewer, if applicable) must be reconfigured and redeployed after the custom font files are created or modified. For details, see “Rebuilding the SAS Web Applications” on page 94.

Make PDF Files Available to SAS Web Report Studio

SAS Web Report Studio enables you to open PDF files from within the application. To enable the availability of PDF files, upload the files to a folder that is visible to SAS Web Report Studio.

To upload PDF files for availability in SAS Web Report Studio, follow these steps:

- 1 In SAS Management Console, go to the **Folders** tab. Right-click on **SAS Folders** to display the menu bar.
- 2 Below **SAS Folders**, right-click on **Shared Data** and select **Add Content From External File(s) or Directories**.
- 3 In the Specify File(s) or Directories dialog box, select **All Files** from the drop-down list for **Files of type**.
- 4 Navigate to the folder that contains the PDF that you want to upload. Select the PDF file and click **Open**.
- 5 If desired, enter a description in the **File Description** field.

You can now view the PDF file or files within SAS Web Report Studio. For information about opening PDF files within SAS Web Report Studio, see *SAS Web Report Studio 4.2 User's Guide*.

Limit the Availability of Relational Information Maps That Implement Row-Level Security

By default, SAS Web Report Studio can interact with information maps regardless of their location within the folder structure. In some situations, you might choose to establish a more controlled environment. For example, you might limit the availability of all relational information maps because some of those information maps include row-level permissions.

To constrain SAS Web Report Studio so that it interacts only with information maps that are in designated locations, follow these steps in the SAS Management Console:

- 1 On the **Plug-ins** tab, under **Application Management ► Configuration Manager**, right-click **Web Report Studio 4.2** and select **Properties**.

In the Web Report Studio 4.2 Properties dialog box, select the **Advanced** tab.

- 2 Click **Add**.
- 3 In the Define New Property dialog box, enter each property name and property value as follows and click **OK**:

Property Name `wrs.map.accessibility.check.enabled`
Property Value `true`.

Property Name `wrs.map.accessibility.check.rootlocations`
Property Value *Path*
- 4 For the property name `wrs.map.accessibility.check.rootlocations`, specify one or more folders as the property value. If you list multiple locations, separate the entries with commas. An example of a path is **SBIP://METASERVER/DepartmentFolder/InfoMaps**.
- 5 Click **OK** to save your changes.
- 6 To enable these properties to go into effect, restart the Web application server.

You can specify more than one path for the **rootlocations** property. If you list multiple locations, separate the entries with commas. With these settings, a relational information map is eligible for use in SAS Web Report Studio only if it is located in the folder path (or its subdirectories).

Note: These settings are typically used in conjunction with metadata layer permissions. For example, if you specify a folder path that is the only valid location for relational information maps, you should also limit the WriteMetadata and WriteMemberMetadata permissions on that folder. △

Note: These settings do not constrain the availability of information maps that use OLAP data. △

Import Reports That Conform to the SAS Report Model

In addition to enabling users to create new reports, SAS Web Report Studio enables users to work with reports that were created elsewhere. Importing a report is the process of retrieving the XML file that defines a report and then adding that report to your report storage structure. The retrieved XML file is written to the appropriate directory within your content server, and a corresponding metadata object is created and stored in a parallel location in the metadata repository.

A report that you import into a new metadata repository will render properly only if all of the report's underlying components (such as an information map, a stored process, and the data sources) are available in the appropriate locations in the new repository's report storage structure. To import a report, follow these steps:

- 1 Log on to SAS Management Console with a metadata profile that connects to the metadata server into which you will import the report.
- 2 Navigate to the appropriate folder below **SAS Folders**, and select the folder into which you will import the report.
- 3 Right-click on the appropriate folder to display a menu bar.
- 4 From the menu bar, select **Add Content From External File(s) or Directories**.
- 5 In the Specify Source File(s) or Directories dialog box, select the report that you want to import and click **Open**.
- 6 In the Enter description dialog box, enter a description for the report and click **OK**.

Import Legacy Reports

You can use the Output Delivery System (ODS) to make legacy SAS reports available in a SAS Intelligence environment. For example, you might have a collection of legacy reports that were created using a SAS program editor, SAS Enterprise Guide, or SAS InTrNet. You can use ODS to write those reports directly to the **ReportStudio** storage structure. SAS Web Report Studio treats the ODS output as a report, allowing a user to display, move, rename, and delete the output as with any other report. However, this type of report cannot be edited from SAS Web Report Studio.

To write ODS output directly to the report storage structure, use the SAS Report XML tag set and the SASXPGRP access method in the FILENAME statement. When you use the SASXPGRP access method in the FILENAME statement, a SAS Business Intelligence Protocol (SBIP)* URL identifies the external file to which you want to write. If your process generates multiple files in the same location, the SBIP URL should refer to a directory rather than to a specific file.

A trailing slash in the SBIP URL is required when specifying a directory. If the specified file or directory already exists, it is overwritten.

The following options to the SASXPGRP access method are required unless otherwise indicated:

USERID="*user ID*"

specifies the user ID to access the server.

PASSWORD="*password*"

specifies the password to access the server.

DOMAIN="*domain*"

specifies the authentication domain name for the server.

OMRHOST="*host*"

specifies the network name of the machine hosting the metadata repository.

OMRPORT="*nnnn*"

specifies the port number of the metadata server for accessing the repository.

OMRUSER="*user ID*"

specifies the metadata user ID to access the repository. This can be the same as the server user ID, or it can be different.

OMRPASSWORD="*password*"

* SBIP is a proprietary protocol for specifying the location of resources in a SAS Metadata Repository. For example, this path **SBIP://METASERVER/Department1/2008/Reports/MyReport.srx** specifies the location of a report named **MyReport** within a repository named **Foundation**.

specifies the password to access the repository. This can be the same as the server password, or it can be different.

OMRREPOSNAME="*name*"

specifies the name of the repository.

For example, the following code outputs SAS Report XML to the specified report storage container:

```
filename dest sasxprp "SBIP://RepName/Department1/2008/Users/xyz/Reports"
    userid="xyz" password="bip2004" domain="thisDomain"
    OMRHost="bipsvrxyz.na.sas.com" OMRPort="9999" OMRUser="xyz"
    OMRPassword="bip2004" OMRRepoName="RepName"
;

option noovp;
ods sasreport file="myreport.xml" path=dest;
proc print data=sashelp.class;
run;
ods sasreport close;
```

Managing Access to Reports

Overview of Managing Access to Reports

The following table summarizes the basic security considerations for reports.

Table 16.6 Report Security Considerations

Item	Action to Secure the Item
Report definitions	Metadata objects that are associated with the reports
	Physical storage location of the report definitions
Underlying report data	Metadata objects that are associated with the report data
	Physical storage location of the report data
	Information maps that reference the report data
	Stored processes that reference the report data
	Report definition (if the report includes embedded data)
	Generated report (if the report is a batch report)

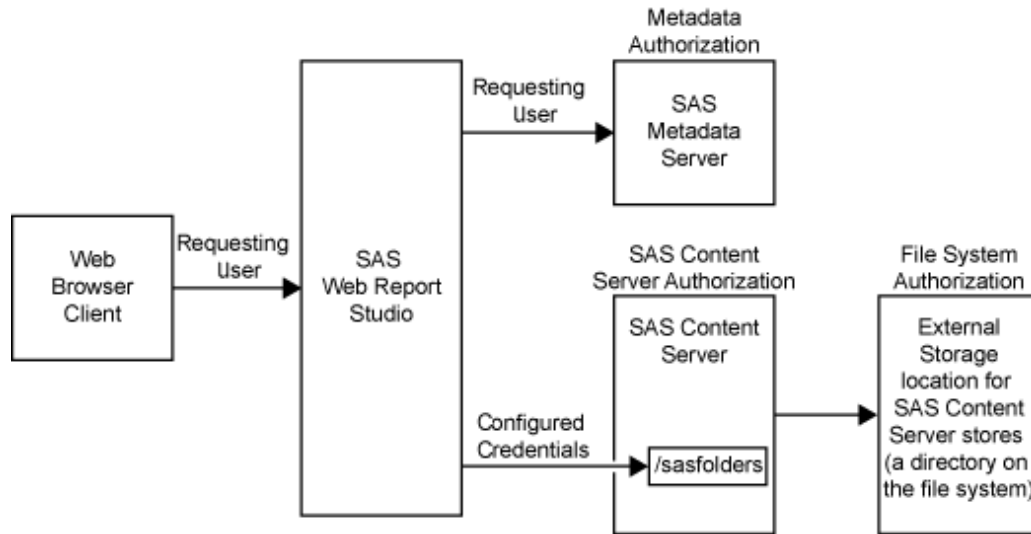
Different types of reports require different security measures. For example, if there is no embedded data of a sensitive nature in a report definition, then the report definition can be considered secure if the report's underlying data, information maps, stored processes, and output are secure. However, pre-generated reports (and some reports that are created through ODS) can include embedded data, so these reports must be protected with access controls that parallel the access controls on the underlying information maps and stored processes.

CAUTION:

Do not rely on restricting access to the underlying information maps or stored processes to ensure that batch reports are viewed only by the appropriate users. △

Access to a report can be affected by multiple layers of controls. For example, the following figure depicts the authorization layers that affect access to reports in a deployment that is using a SAS content server with the SAS document store located in a directory in the file system.

Display 16.4 Authorization Layers That Affect Access to Reports



In the figure, the requesting user's access to reports is subject to controls in the metadata, SAS Content Server, and file system authorization layers. However, the only layer in which the requesting user's permissions matter is the metadata layer, because this is the only layer in which the requesting user's identity is known. In the metadata layer, each user's access to reports is based on the user's individual identity and group memberships. When you work with metadata access controls for reports, consider these points:

- The effective permissions on a report folder are inherited by all of the reports within that folder.
- The ability to view or work with a report can be affected by access to each of the report's underlying components.
- If your organization uses publication channels to deliver reports, the reports can also be protected by controlling access to the publication channels. To enable all registered users to add channels or subscribers or to add items to a particular channel, see "Permissions by Item" in the *SAS Intelligence Platform: Security Administration Guide*.

As the preceding figure depicts, SAS Web Report Studio uses only one account to connect to the external storage location, so you cannot make access distinctions between individual users by setting operating system access controls on specific items within the external storage location. However, you should set operating system controls that allow only the identity under which the SAS Content Server process is running (local system in this example) to access this physical file location.

Similarly, SAS Web Report Studio uses only one account, the trusted user `sastrust`, to communicate with the WebDAV content server, so you cannot make access distinctions between individual users by setting access controls in the SAS Content Server. However, you should use this layer to protect your SAS report content. For information about how to set permissions by using the SAS Content Server admin console, see "Using the SAS Content Server Administration Console" on page 122.

For more information about access controls and access management, see “Access Management Tasks” in the *SAS Management Console: Guide to Users and Permissions*.

Change Access to Reports

Each new report subfolder that you create in metadata inherits the effective access controls of its parent folder. For information about how to limit access to each subfolder or the items within the subfolders, see “Access Management Tasks” in the *SAS Management Console: Guide to Users and Permissions*.

Enable Permissions in Restrictive Policy File for Publishing Reports to Channels

When a SAS Web Report Studio report is saved as a pre-generated report in PDF format, the output is published to a channel. The Web application server must have explicit policy permissions granted to allow read and write access to this directory.

An express or typical installation completed with the SAS Deployment Wizard creates a SAS environment that does not use restrictive policy files. By default, the **sas.all.permissions.policy** is used to allow access to SAS Web applications. As a result, SAS Web applications can access the necessary content.

If you implement restrictive policies at your site for JBoss or IBM WebSphere application servers, you must ensure that read and write access is granted to the directory where pre-generated reports are published to channels.

CAUTION:

SAS strongly discourages the use of restrictive policy files on SAS middle-tier applications because they provide no end-user security, they are difficult to maintain, and they can be very detrimental to application performance. △

For more information about restrictive policy files, see “Configuring and Deploying Restrictive Policy Files” on page 45.

Security Considerations for Pre-generated Reports

When a pre-generated report is created, the content of the report reflects the access that the generating user ID has to objects such as data sources and stored processes. For example, if a pre-generated report is configured to use an identity named SalesMgr1, then the report can include anything that SalesMgr1 is able to access. Regardless of who actually views the report, the report content is always based on the access controls that apply to SalesMgr1. This means that any user who has ReadMetadata permission for a pre-generated report can view that report, even if other metadata access controls deny the user access to the report’s underlying components (such as data sources and stored processes). For this reason, you must give careful consideration to the identity that each pre-generated report uses for generation, and you must secure the pre-generated reports that you create. Pre-generated reports are saved and open to anyone who has permissions to the Operating System (OS) location.

Note: When a user refreshes the report data while viewing the report in SAS Web Report Studio, that user sees only the content that he or she has permission to see. △

Considerations for Row-level Security

If you implement BI row-level access to data, it is recommended that you configure a pooled workspace server that is dedicated for use by SAS Web Report Studio. For BI row-level access to data, you need a pooled workspace server to prevent the workspace server processes from running under the accounts of requesting users. Pooled workspace servers run under one or more designated accounts that are called puddle accounts or puddle logins. You need a dedicated workspace server to isolate the row-level security puddle account from applications that do not fully enforce row-level security.

In a dedicated deployment of SAS Web Report Studio, a client-side pooled workspace server is required.

For information about pooled workspace servers, see the *SAS Intelligence Platform: Application Server Administration Guide*.

For more information about row-level security, see “BI Row-Level Permissions” in the *SAS Intelligence Platform: Security Administration Guide*.

Protect Report Content in the WebDAV Server

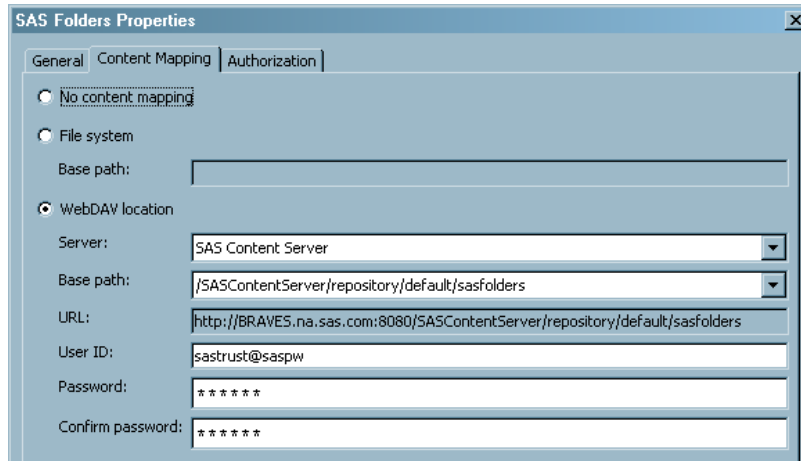
On a publicly accessible WebDAV server, the area where SAS report content is stored should be protected against access by components that do not enforce SAS metadata permissions. For example, applications from other vendors and the DAV navigator portlet should not be able to access content in this area.

The **sasfolders** folder in the SAS Content Server can be configured for WebDAV content mapping and 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. Applications that are not aware of SAS metadata do not have access to the sastrust user ID and password, so these applications cannot access the area of the content server.

Configure Content Mapping with a WebDAV Location

Content Mapping associates a location where report content is stored with a tree of folders and objects in a metadata repository. During installation, content mapping is created for the **/SASContentServer/repository/default/sasfolders** WebDAV directory. The **Content Mapping** tab is available only for folders that are directly below the root node of the SAS Folders navigation tree.

The following display shows the Content Mapping dialog box:

Display 16.5 Content Mapping Dialog Box


SAS Folders Properties

General | **Content Mapping** | Authorization

☐ No content mapping

☐ File system

Base path:

☒ WebDAV location

Server:

Base path:

URL:

User ID:

Password:

Confirm password:

If you want to view or modify content mapping with a WebDAV Location, follow these steps:

- 1 In SAS Management Console, go to the **Folders** tab, and navigate to **SAS Folders**. Right-click on **SAS Folders** to display the menu bar.
- 2 In the menu bar, select **Properties**.
- 3 In the SAS Folders Properties dialog box, select the **Content Mapping** tab.
- 4 In the Content Mapping dialog box, select **WebDAV location**.
- 5 From the menu for **Server**, select **SAS Content Server**. The server specifies a WebDAV-enabled HTTP server or HTTPS server where the report content will be stored.

The **Base path** and **URL:** fields are filled with the folder path in SAS Management Console and the folder path in the SAS Content Server respectively. The **Base path** specifies the path to the report content. You can use this **Base path** field only when you select **WebDAV location**.

- 6 When the WebDAV location is chosen for content mapping, also fill in the fields for **User ID**, **Password**, and **Confirm Password**.

Verify SAS Trusted User's Permissions to Directories in the SAS Content Server

To verify the SAS Trusted User's permissions to the SAS Content Server, follow these steps:

- 1 Access the SAS Content Server's administration console by opening a Web browser to `<machine-name>:8080/SASContentServer/dircontents.jsp` and logging on as the SAS Content Server administrator.
- 2 Select the permissions icon for the **sasfolders** directory, and verify that the SAS Trusted User has exclusive, full access to this directory (and to this directory's subdirectories and files).

Verify SAS Trusted User's Access to the Directories in the SAS Content Server

To verify that the SAS Trusted User (sastrust) can access the **sasfolders** directory, follow these steps:

- 1 Access the SAS Content Server's administration console by opening a Web browser to `<machine-name>:<port>/SASContentServer/repository/default` and logging on as the SAS Content Server administrator.
- 2 Authenticate as the `sastrust` user.
- 3 Verify that you can see the contents of the directories.

Verify Users' Access to the SAS Directories in the SAS Content Server

To verify that other users cannot directly access the **SAS Folders** area in the SAS Content Server, access the SAS Content Server's administration console by opening a Web browser to `<machine-name>:<port>/SASContentServer/repository/default` and logging on as the SAS Demo User. You should see a "Page Not Found" message.

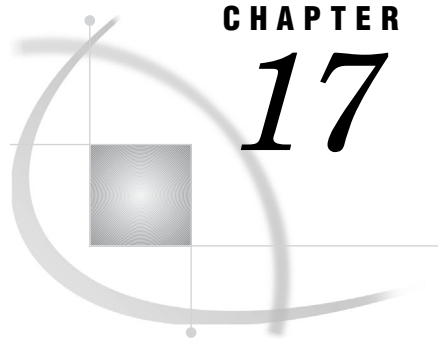
Protect Data in Temporary Files Created by SAS Web Report Studio

SAS Web Report Studio writes temporary files that might contain data that should be protected. These temporary files are stored in the following locations:

- in the `tmpnull` and `tmpuser` subfolders within the folder where SAS Web Report Studio is deployed. For example, if you are using the JBoss application server on Windows, this location might be `C:\JBoss\server\SASServer1\work\jboss.web\localhost\SASWebReportStudio\sas.wrs`.
- in the Java temporary folder on the server where SAS Web Report Studio is running. The location of this folder is defined by the Java property `java.io.tmpdir`.

To protect the data in these temporary files, do the following:

- Use operating system protections to limit access to the computer on which SAS Web Report Studio is deployed.
- Set additional operating system protections on the folders that contain the temporary files. Only system administrators who require access to all folders should be able to access these folders.



Customizing SAS Web Report Studio Report Styles

<i>SAS Web Application Themes and Custom Report Styles</i>	221
<i>Customizing Report Styles for SAS Web Report Studio</i>	221
<i>Overview of Providing Custom Report Styles</i>	222
<i>Specify Property Names and Values for Styles</i>	222
<i>CSS Formats for Custom Report Styles</i>	224
<i>About Cascading Style Sheet Formats</i>	224
<i>Supported Properties</i>	224
<i>Tables</i>	226
<i>Graphs</i>	228
<i>Text</i>	231
<i>Synchronized Objects Container</i>	231
<i>Display Filters</i>	231
<i>Add Disclaimer Text to Graphs and Tables</i>	232
<i>Specify Fonts for PDF Reports Generated by SAS Web Report Studio</i>	233

SAS Web Application Themes and Custom Report Styles

SAS 9.2 includes SAS Web Application Themes that enable you to create consistent visual customization and company branding that can be applied to all theme-enabled SAS Web applications including SAS Web Report Studio, SAS Information Delivery Portal, and SAS BI Dashboard. SAS Web Application Themes apply to the user interface of SAS Web Report Studio, including the dialog boxes that are used to view, create, edit, and share reports. For more information about SAS Web Application Themes and how to create custom themes, see Chapter 12, “Administering SAS Web Application Themes,” on page 153.

While themes apply to SAS Web applications, SAS Web Report Studio provides customizable styles that apply to the content within reports. Custom report styles affect the content of reports that are created with SAS Web Report Studio. Specifically, the report styles affect the colors, fonts, and other elements that are used in tables and graphs. Cascading style sheets enable the customized display of the contents of reports that are generated with SAS Web Report Studio.

The following sections describe customization of report styles.

Customizing Report Styles for SAS Web Report Studio

Overview of Providing Custom Report Styles

Report styles affect the colors, fonts, and other elements that are used in tables and graphs. Report viewers can select one of the following styles (which ship with SAS Web Report Studio) for a report: Plateau, Meadow, Seaside, or Festival. The default report style is Plateau. You can add your own custom styles to the list of available styles.

Note: The ability to apply custom styles is available for applications that run in all locales worldwide. △

SAS Web Report Studio relies on cascading style sheets (CSS) to render styles for reports. To supply a custom style, follow these steps:

- 1 Create a CSS file and define the formats that you want for the style. For information about the supported formats as well as a sample CSS file, see “CSS Formats for Custom Report Styles” on page 224.
- 2 In the Web Report Studio 4.2 Properties dialog box within SAS Management Console Configuration Manager, provide information that SAS Web Report Studio needs in order to locate and render the style. For instructions, see “Specify Property Names and Values for Styles” on page 222.

Specify Property Names and Values for Styles

In the Web Report Studio 4.2 Properties dialog box within the SAS Management Console, you must provide information that SAS Web Report Studio needs in order to locate and render the style that you want to use.

The following table shows the property names that you specify, and explains the property values that you enter in the **Advanced** tab of the Web Report Studio 4.2 Properties dialog box.

Table 17.1 Property Names of Report Styles Specified in the Web Report Studio 4.2 Properties Dialog Box

Property Name	Description
<css>	<p>Provides the fully qualified path to one or more external CSS files from which style schemes will be read. If you specify multiple files, separate them with a comma.</p> <p>If you remove a filename from this element, then any report that has been created with the corresponding style might not render correctly. The rendering behavior is undefined if the CSS file has been removed.</p>
<schemelist>	<p>Specifies the list of styles that are available to SAS Web Report Studio users.</p> <p>You must add your custom style name to the list in order for that style to be available for use. The name must match exactly the name of a CSS file in the <css> list (but without the file path or CSS filename extension). Any mismatches cause the name not to be available in SAS Web Report Studio. If you specify multiple styles, separate the style names with a comma.</p> <p>Default styles are Plateau, Meadow, Seaside, and Festival. If you remove any of these names from the list, then the corresponding styles will no longer be available to users. However, existing reports that reference the styles will continue to render properly because these styles are built in and inherently known by SAS Web Report Studio.</p>
<defaultscheme>	<p>Defines the default styles that will be applied to new reports. If no style is specified, then the default style is Plateau.</p>

To specify the styles that you want to use, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2**.
- 2 Right-click and select **Properties** to display the Web Report Studio 4.2 Properties dialog box.
- 3 Click on the **Advanced** tab.
- 4 Click **Add** to display the Define New Property dialog box.
- 5 Enter the property names as shown and insert your own property values for **sas.wrs.style.css**, **sas.wrs.style.schemelist**, and **sas.wrs.style.defaultscheme**.

Property Name: **sas.wrs.style.css**

Property Value: *Comma-delimited list of CSS files*

Property Name: **sas.wrs.style.schemelist**

Property Value: *your custom style names*

Property Name: **sas.wrs.style.defaultscheme**

Property Value: *default style*
- 6 Click **OK** to exit the Define New Property dialog box.
- 7 Click **Add** to add another property.
- 8 When you are finished adding properties, click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 9 To enable these properties to go into effect, restart your Web application server.

CSS Formats for Custom Report Styles

About Cascading Style Sheet Formats

In order to provide custom report styles, you create one or more Cascading Style Sheet (CSS) files. CSS files are text files that can be edited with a text editor. CSS file enables specified formats (CSS rule sets) to be available for users to modify in SAS Web Report Studio. These files are stored in the

C:\SAS\config\Lev1\Web\Applications\SASWebReportStudio4.2\customer folder. A default CSS file is also available at that location.

When CSS files are created and their filenames are specified on the **Advanced Tab** of the Web Report Studio 4.2 Properties, the following elements can be modified by users in SAS Web Report Studio:

- ☐ tables, both list and crosstabulation
- ☐ graphs
- ☐ text objects
- ☐ headers and footers
- ☐ containers for synchronized objects
- ☐ display filters

In the CSS file, lines that start with `<` or `-` are considered comments. These lines are ignored by SAS Web Report Studio.

SAS Web Report Studio does not support at-rules, such as `@import`. Such directives are ignored.

A sample CSS file is available to help you develop your own custom styles. The file **Seaside_CSS.css** was copied to the **customer** folder when you installed and then configured SAS Web Report Studio. This CSS is based on the built-in Seaside style.

For instructions about making the CSS formats available to SAS Web Report Studio, see “Specify Property Names and Values for Styles” on page 222. For information about CSS files in general, see the W3C organization’s Web site at <http://www.w3.org/TR/CSS21/>.

Supported Properties

This table lists the properties that are supported for the property types that are explained through examples later in this chapter.

Table 17.2 Supported Properties for CSS Formats

Property Type	Supported Properties
text	font-family font-weight color background-color text-align font-size text-decoration font-style
minimal text	font-family font-weight text-color font-size font-style
border	border border-color border-top-color border-bottom-color border-right-color border-left-color border-width border-top-width border-bottom-width border-right-width border-left-width border-style border-top-style border-bottom-style border-right-style border-left-style
cell	padding padding-top padding-bottom padding-left padding-right
graph data styles	color marker-symbol* marker-size line-thickness

* Possible values are: TRIANGLEFILLED, SQUAREFILLED, STARFILLED, HEXAGONFILLED, CIRCLEFILLED, CROSSFILLED, FLAGFILLED, CYLINDERFILLED,

PRISMFILLED, X, SPADEFILLED, DIAMONDFILLED, HEARTFILLED, CLUBFILLED, POINT, NONE.

Tables

The following figure shows a sample list table.

Figure 17.1 List Table

Figure 17.1 shows a sample list table with the following structure:

- 1** points to the **Table Title**.
- 2** points to the **Age** column header.
- 3** points to the **Name** column header.
- 4** points to a data row (e.g., 14, 64.3, Judy, F, 90).
- 5** points to the **Totals** row (253, 1184.4, Total, 1900.5).

Applied filters: None				
Age	Height	Name	Sex	Weight
14	69	Alfred	M	112.5
13	56.5	Alice	F	84
13	65.3	Barbara	F	98
14	62.8	Carol	F	102.5
14	63.5	Henry	M	102.5
12	57.3	James	M	83
12	59.8	Jane	F	84.5
15	62.5	Janet	F	112.5
13	62.5	Jeffrey	M	84
12	59	John	M	99.5
11	51.3	Joyce	F	50.5
14	64.3	Judy	F	90
12	56.3	Louise	F	77
15	66.5	Mary	F	112
16	72	Philip	M	150
12	64.8	Robert	M	128
15	67	Ronald	M	133
11	57.5	Thomas	M	85
15	66.5	William	M	112
253	1184.4	Total		1900.5

Here are the supported style formats for elements in the list table.

Table 17.3 CSS Formats for List Tables

Numbered Item	Selector	Supported Property Types
1 Title	Table Caption	Text
2 Headings	Table Column Label	Text, cell, border
3 Border	Table	Border
4 Cells	Table Column Cell	Text *, cell, border
5 Totals	Table Rows Summary	Text, cell, border

* The alignment (text-align property) for cells is overridden based on data type (numeric versus text).

Note: In the CSS file, you must define the Table format before you define any of its descendant formats, such as Table Caption or Table Column Label. △

For more information about the supported property types, see “Supported Properties” on page 224.

The following figure shows a sample crosstabulation table.

Figure 17.2 Crosstabulation Table

		SOUTHEAST		SOUTHWEST		Total	
		Number of Sales	Cost of Sales	Number of Sales	Cost of Sales	Number of Sales	Cost of Sales
2000	Catalog						
	Collectibles	5	\$117.00	11	\$251.00	16	\$368.00
	Gardening	37	\$1,114.00	21	\$612.00	58	\$1,726.00
	Pets	72	\$471.00	35	\$198.00	107	\$669.00
	Software	30	\$3,229.65	21	\$1,831.09	51	\$5,060.74
	Sports	88	\$5,995.00	84	\$4,255.00	172	\$10,250.00
	Toys	60	\$1,517.37	51	\$1,241.24	111	\$2,758.61
Subtotal: 2000		292	\$12,444.02	223	\$8,388.33	515	\$20,832.35
2001	Collectibles	4	\$104.00	6	\$133.00	10	\$237.00
	Gardening	33	\$934.50	40	\$1,421.00	73	\$2,355.50
	Pets	74	\$539.00	61	\$438.00	135	\$977.00
	Software	21	\$2,096.49	27	\$3,430.39	48	\$5,526.88
	Sports	100	\$6,207.00	88	\$5,251.00	188	\$11,458.00
	Toys	50	\$1,838.91	62	\$1,461.03	112	\$3,299.94
Subtotal: 2001		282	\$11,719.90	284	\$12,134.42	566	\$23,854.32
Total		574	\$24,163.92	507	\$20,522.75	1081	\$44,686.67

Here are the supported style formats for elements in the crosstabulation table.

Table 17.4 CSS Formats for Crosstabulation Tables

Numbered Item	Selectors	Supported Property Types
1 Title	Table Caption	Text
2 Headings	Table Rowgroup Label	Text
2 Headings	Table Rowgroup Row Label	Cell
2 Headings	Table Columngroup Label	Border
3 Border	Table Columngroup Column Label	Border
4 Cells	Table Rowgroup Row Cell	Text *
	Table Columngroup Column Cell	Cell
		Border
5 Totals	Table Rows Summary	Text
	Table Columns Summary	Cell
		Border

Numbered Item	Selectors	Supported Property Types
⑥ Subtotals	Table Rowgroup Rows Summary	Text
	Table Columngroup Columns	Cell
	Summary	Border
⑦ Subheads	Table Rowgroup Values	Text
	Table Columngroup Values	Cell
		Border

* The alignment (text-align property) for cells is overridden based on data type (numeric versus text).

Note: In the CSS file, you must define the Table format before you define any of its descendant formats, such as Table Caption or Table Column Label. △

For more information about the supported property types, see “Supported Properties” on page 224.

Graphs

Like tables, graphs support styles for different aspects of their rendering. However, when subgroups are used in a graph, you should specify a unique format for each subgroup value in order to distinguish between the values. Subgrouping is data-dependent (for example, one subgroup might have three values, whereas the same subgroup on different data might have nine values). Therefore, SAS Web Report Studio supports a flexible collection of rules called *graph data styles*. A report scheme can consist of up to 12 specified graph data styles. Each graph data style can in turn be used for a particular subgroup of data.

The following example shows three sample graph data styles:

```
Graph GraphDataStyle1
{
    color : red;
    marker-symbol : DIAMONDFILLED;
    marker-size : 10px;
    line-thickness : 2px;
}

Graph GraphDataStyle2
{
    color : green;
    marker-symbol : CIRCLEFILLED;
    marker-size : 10px;
    line-thickness : 2px;
}

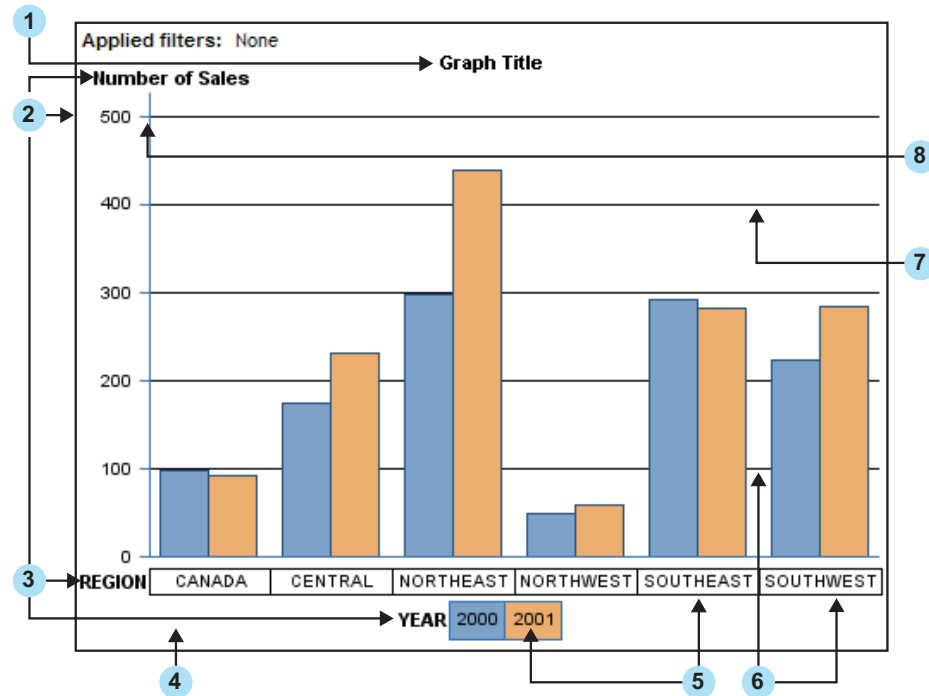
Graph GraphDataStyle3
{
    color : blue;
    marker-symbol : SQUAREFILLED;
    marker-size : 10px;
    line-thickness : 2px;
}
```

This method enables you to define graph schemes that supply common formats across different types of graphs. Not all the graph data styles are used for each graph.

Note: The progressive bar chart and the geographical map do not support the `GraphDataStylen` formats. The supported formats for these charts are described later in this section. △

The following figure shows a sample graph, followed by a list of the supported formats for elements in the graph.

Figure 17.3 Progressive Bar Chart



The following table shows the list of supported elements for graphs.

Table 17.5 CSS Formats for Graphs

Numbered Item	Selector	Supported Properties and Property Types
① Title	Graph TitleText	Text
② Border	Graph BorderLines	Line-color property
③ Axis and legend labels	Graph LabelText	Minimal text
④ Background	Graph BackFill	Fill-color property
⑤ Axis and legend values	Graph ValueText	Minimal text
⑤ Axis and legend values	Graph LegendFill	Fill-color property
⑥ Grid lines	Graph GridLines	Line-color property

Numbered Item	Selector	Supported Properties and Property Types
⑦ Data	Graph GraphDataStyle <i>n</i>	Graph data styles
⑧ Horizontal and vertical axis	Graph AxisLines	Line-color property Line-thickness property

For more information about the supported property types, see “Supported Properties” on page 224.

The geographical (ESRI) chart supports only the border style.

The progressive bar chart does not support the GraphDataStyle*n* formats. Instead, the chart uses three different formats for its data styles. These formats are unique to the progressive bar chart.

Figure 17.4 Sample Progressive Bar Chart

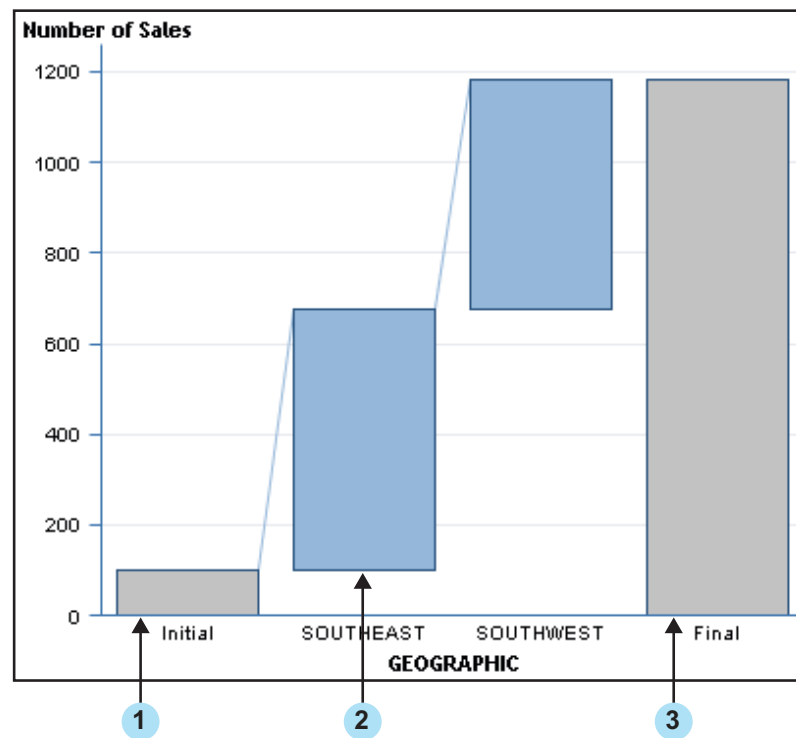


Table 17.6 CSS Formats for Progressive Bar Charts

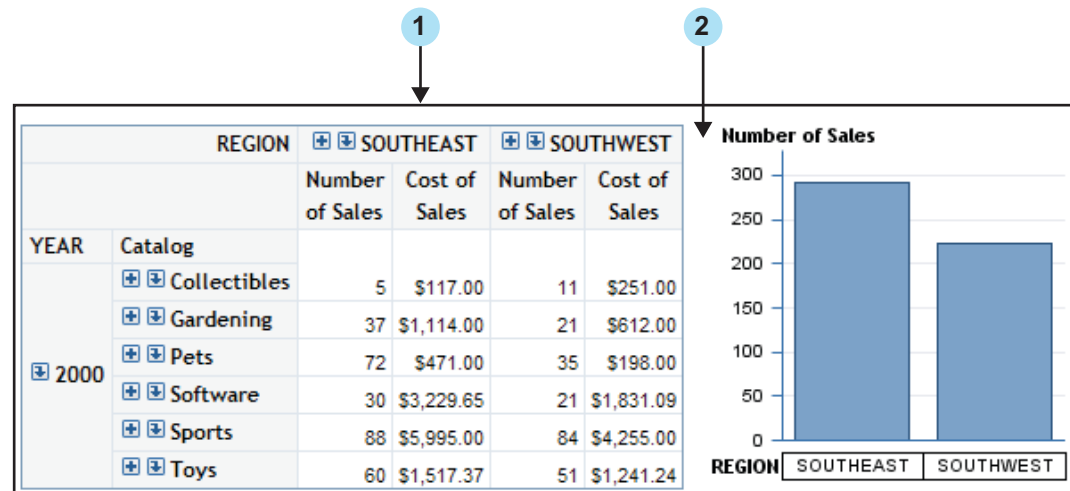
Numbered Item	Selectors	Supported Properties
❶ Initial bar	Graph InitialDataStyle	Fill-color
❷ Positive or negative bars	Graph ThreeColorRamp	Fill-gradient-start-color
	Graph ThreeColorAltRamp	Fill-gradient-end-color
❸ Final bar	Graph FinalDataStyle	Fill-color

Text

Text elements, including headers and footers, use the **text** property type, and support all text formats.

Synchronized Objects Container

The following figure shows a container for synchronized objects, followed by a list of the supported formats.

Figure 17.5 Sample Synchronized Objects Container**Table 17.7** CSS Formats for a Synchronized Objects Container

Numbered Item	Selector	Supported Properties or Property Types
❶ Container border	LinkedContainer	Border
❷ Container	LinkedContainer	Background-color property Padding property

For information about the supported property types, see “Supported Properties” on page 224.

Display Filters

Three types of Display Filters are available are:

- graphs
- tables
- containers for synchronized objects

Display Filters for graphs, tables, and the containers for synchronized objects must be specified individually. The following figure shows display filters for a table. Display filters are similar for graphs and synchronized object containers.

Figure 17.6 Sample Display Filters

Applied filters: Geographic equal to SOUTHEAST, SOUTHWEST

Table Title							
REGION		+ - SOUTHEAST		+ - SOUTHWEST		Total	
		Number of Sales	Cost of Sales	Number of Sales	Cost of Sales	Number of Sales	Cost of Sales
YEAR	Catalog						
+ - 2000	+ - Collectibles	5	\$117.00	11	\$251.00	16	\$368.00
	+ - Gardening	37	\$1,114.00	21	\$612.00	58	\$1,726.00
	+ - Pets	72	\$471.00	35	\$198.00	107	\$669.00
	+ - Software	30	\$3,229.65	21	\$1,831.09	51	\$5,060.74
	+ - Sports	88	\$5,995.00	84	\$4,255.00	172	\$10,250.00
	+ - Toys	60	\$1,517.37	51	\$1,241.24	111	\$2,758.61
Subtotal: 2000		292	\$12,444.02	223	\$8,388.33	515	\$20,832.35
+ - 2001	+ - Collectibles	4	\$104.00	6	\$133.00	10	\$237.00
	+ - Gardening	33	\$934.50	40	\$1,421.00	73	\$2,355.50
	+ - Pets	74	\$539.00	61	\$438.00	135	\$977.00
	+ - Software	21	\$2,096.49	27	\$3,430.39	48	\$5,526.88
	+ - Sports	100	\$6,207.00	88	\$5,251.00	188	\$11,458.00
	+ - Toys	50	\$1,838.91	62	\$1,461.03	112	\$3,299.94
Subtotal: 2001		282	\$11,719.90	284	\$12,134.42	566	\$23,854.32
Total		574	\$24,163.92	507	\$20,522.75	1081	\$44,686.67

When the tables and graphs are synchronized objects, then the LinkedContainer selector must be used, because the filters are displayed for the container that holds the tables and graphs.

In the preceding figure for Sample Display Filters, the selectors are Graph DisplayFilter, Table DisplayFilter, and LinkedContainer DisplayFilter. The supported property types and properties are text, border, and margin-bottom.

Note: For each of the formats, you must define the parent format before you define any of its descendants in the CSS file. For example, you must define a LinkedContainer format before you define a LinkedContainer DisplayFilter format. △

Add Disclaimer Text to Graphs and Tables

SAS Web Report Studio enables you to add disclaimer text to graphs and tables. You can use the disclaimer text to provide a copyright statement or some general disclaimer of usage.

To add disclaimer text to graphs and tables, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2**.
- 2 Right-click and select **Properties** to display the Web Report Studio 4.2 Properties dialog box.
- 3 Click on 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 with your own disclaimer text:
Property Name: `wrs.disclaimer.tableAndGraph`
Property Value: *My Disclaimer*
- 6 Click **OK** to exit the Define New Property dialog box.
- 7 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 8 To enable this property to go into effect, restart your Web application server.

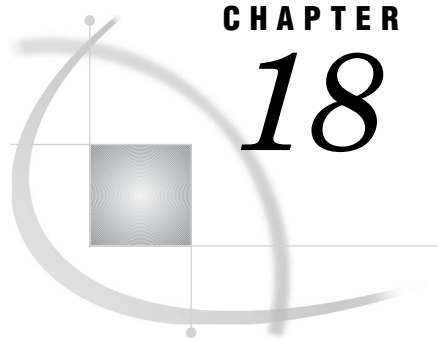
The disclaimer text does not affect existing reports. The text will appear beneath the tables and graphs of all new reports.

Specify Fonts for PDF Reports Generated by SAS Web Report Studio

When printing SAS Web Report Studio reports in non-Western languages (for example, Chinese), the generated PDF report might fail to display the content if the fonts are not configured in the SAS Management Console Configuration Manager. For example, the default font for SAS Web Report Studio reports is Times Roman. However, the Times Roman font does not contain glyphs that are used in certain languages. To ensure that special fonts in different languages display correctly in the PDF reports, follow these steps:

- 1 In SAS Management Console on the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► Web Report Studio 4.2**.
- 2 Right-click and select **Properties** to display the Web Report Studio 4.2 Properties dialog box.
- 3 Click on the **Advanced tab**.
- 4 Click **Add** to display the Define New Property dialog box.
- 5 Enter the property name as shown (note that there is a period at the beginning of the property name), along with the property value as shown:
Property Name:
`.vmwide.com.sas.report.render.view.pdf.itext.font.fontDirectories`
Property Value: *Font_path_string*

To specify multiple font directory paths in the UNIX or z/OS environment, follow this format: *Font_path_string1:Font_path_string2:Font_path_string3*. In the Windows environment, follow this format:
Font_path_string1;Font_path_string2;Font_path_string3
- 6 Click **OK** to exit the Define New Property dialog box.
- 7 Click **OK** to exit the Web Report Studio 4.2 Properties dialog box.
- 8 To enable this property to go into effect, restart your Web application server.



CHAPTER

18

Pre-generated Reports From SAS Web Report Studio

<i>Overview of Pre-generated Reports from SAS Web Report Studio</i>	235
<i>Understanding Pre-generated Reports</i>	236
<i>Scheduled Reports and Distributed Reports</i>	236
<i>How Report Scheduling Differs from Report Distribution</i>	237
<i>Required Role Capabilities for Scheduling and Distributing Reports</i>	238
<i>How Users Interact with Manually Refreshed Reports</i>	239
<i>Configuring a Scheduling Server</i>	239
<i>Verifying Permissions for the Distribution Library</i>	241
<i>Setting Up a Recipient List for Report Distribution</i>	241
<i>Overview: Setting Up a Recipient List</i>	241
<i>(Optional) View the Library for Recipient Lists</i>	241
<i>Creating a Recipient List for Report Distribution</i>	242
<i>Understanding How Recipient Lists Enable Report Distribution</i>	242
<i>Create a Recipient List</i>	243
<i>Alternative Example: Create a Recipient List Using PROC SQL</i>	247
<i>Considerations for Creating Recipient Lists</i>	248
<i>Processing Reports Outside of SAS Web Report Studio</i>	248
<i>Overview of Processing Reports Outside of SAS Web Report Studio</i>	249
<i>Requirements for Report Batch Processing</i>	249
<i>The Report Output Generation Tool</i>	249
<i>Parameters for the Report Output Generation Tool</i>	250
<i>Logging Directory Permissions for Report Output Generation Tool</i>	252
<i>Viewing Console Output</i>	252
<i>Using the Report Output Generation Tool to Distribute Reports</i>	253
<i>Obtaining the Values for Running the Report Output Generation Tool on Windows and UNIX</i>	254
<i>Obtaining the Values for Running the Report Output Generation Tool on z/OS</i>	254
<i>Report Output Generation Tool Examples</i>	255
<i>Example: Running a Report</i>	255
<i>Example: Running a Scheduled Job</i>	256
<i>Example: Using a Prompts File as Input for a Report</i>	256
<i>Example: Distributing a Report</i>	257
<i>Migrating from the Batch Generation Tool to the Report Output Generation Tool</i>	257
<i>Comparison Overview</i>	257
<i>Parameter Comparison</i>	257

Overview of Pre-generated Reports from SAS Web Report Studio

Understanding Pre-generated Reports

SAS Web Report Studio enables users to view reports with results from pre-generated queries. One advantage of providing pre-generated results is improved performance. Pre-generated reports are stored in the repository and are available for users to view. It takes less time to view these reports because the queries have already been processed and the results have already been generated. However, because these reports are non-interactive, they cannot accept input from a requesting user. When a pre-generated version of a report includes prompts (questions that require user input), the prompt values that were provided when the report was generated are used. The non-interactive nature also has important implications for security.

SAS Web Report Studio users can create pre-generated reports in several ways:

- Users can save any report as a static report in PDF format. The report is saved in the repository and is made available to other authorized users who can distribute the report through a publication channel or to a list of e-mail addresses. Users cannot interact with a report that has been saved in PDF format.
- When users save a report, they can select **Data can be manually refreshed** from the Save As dialog box in SAS Web Report Studio. This type of report typically requires a manual refresh each time it is opened.
- Users can schedule saved reports to be run at a specified time. This feature enables users to refresh report data at specified intervals or times.
- Users can create a snapshot of report data, and then distribute the static results to recipients.

The following sections describe the administration tasks that are associated with scheduling manually refreshed reports and distributing static snapshot reports.

Scheduled Reports and Distributed Reports

Scheduled reports and distributed reports vary slightly in their functions. The following table summarizes these two types of reports.

Table 18.1 Summary of Scheduled and Distributed Reports

Type of Report	Description
Scheduled report	<p>The report is scheduled to be updated at pre-determined times. When users open the report in SAS Web Report Studio, they can update the report data manually.</p> <p>These reports can be archived in order to maintain older versions of the report.</p>
Distributed report	<p>The report is scheduled to be updated at pre-determined times. When the report is updated, SAS Web Report Studio creates a snapshot of report data, and then distributes the static results by e-mail to recipients and to publication channels that you specify.</p>

For distributed reports, the snapshot that you distribute might include all or part of the original report. For example, suppose that your organization has sales teams in different countries, and you want to provide high-level sales information to the team managers in each country. When you set up a distribution, you can group the report based on the country, and then specify which managers should receive the sales

information for each country. SAS Web Report Studio e-mails the appropriate version of the report to the specified recipients in either PDF or HTML format. Recipients cannot refresh or interact with the report that they receive.

Authorized users can use the report scheduling and distribution features to update pre-generated reports at specified times. In addition, these users have the option to send the reports directly to a specified list of users via e-mail. In order to enable users to schedule reports, a scheduling server must be configured. For information about configuring a scheduling server, see *Scheduling With SAS 9.2*. For information about scheduling reports in SAS Web Report Studio, see the *SAS Web Report Studio 4.2 User's Guide*.

How Report Scheduling Differs from Report Distribution

The processes that you use to schedule and distribute reports are similar in several ways. Both tasks use wizards, and both rely on the **rptbatch.bat** executable to create output. In UNIX and z/OS, this executable is referred to as **rptbatch.sh**. The **rptbatch.bat** file resides in the *SAS-configuration-directory\Lev1\ReportBatch* directory. The report output generation tool, **rptbatch.bat** replaces the batch generation tool (**batchgen.exe**) that was included with previous releases. The **rptbatch.bat** file functions as a wrapper around the **outputgen.exe** file and is used to create output.

However, there are several differences between scheduling and distribution. The following table summarizes these differences:

Table 18.2 Differences Between Scheduling and Distributing Reports

Report Scheduling	Report Distribution
Reports are generated and stored in a repository.	Reports are generated and e-mailed to recipients that you specify in a recipient list.
Reports can be pushed to a publication channel. Publication channels are specified in the Schedule Report wizard. If publication channels were not defined in the deployment, publication channel options will not appear in the Schedule Report Wizard.	Reports can be pushed to a publication channel. Publication channels are specified in the recipient list.
The full report is generated.	The full report can be distributed. Alternatively, the report can be divided by group breaks so that each recipient gets a subset of the report.
Users can schedule a folder for report generation. All reports in the folder are generated.	Users can set up distribution for only one report at a time.
Users can enable archiving for a report. If archiving is enabled, the latest report is archived when a new one is generated. The archive can retain multiple versions of the report.	Reports cannot be archived.

Report Scheduling	Report Distribution
Scheduling does not include the ability to preview a schedule. No e-mail is sent, so a review is not required.	Users can run a test to preview the distribution of a report. To allow review of e-mail and prevent spamming to e-mail recipients, the test returns a recipient list either in the user interface or via an e-mail.
Users who receive the report can refresh and interact with the report. This action requires the report to be regenerated.	Users cannot refresh or interact with the report.

Note: A publication channel is an information repository that has been established by using the SAS Publishing Framework in SAS Management Console and which can be used to publish information to users and applications. If you publish your report to a publication channel, then authorized users and applications can access your report by subscribing to the channel. For example, the SAS Information Delivery Portal can list the content of a publication channel. △

Required Role Capabilities for Scheduling and Distributing Reports

In order to schedule or distribute reports, users must be assigned to an Advanced role to which certain scheduling and report distribution capabilities have been assigned. The Advanced role is a predefined role for users and administrators of SAS Web Report Studio. For information about predefined roles for SAS Web Report Studio, see “Predefined Roles” on page 200.

Identify which scheduling or distribution capabilities you want to assign to the Advanced role, and assign users to that role. Here is a list of scheduling and distribution capabilities available in the Advanced role:

Advanced:Distribute

Enable a distribution

Advanced:Save Archive

Archive file versions. When creating a pre-generated version of a report (for example, a scheduled report), a version of the report is archived as a PDF file. Reports with archived versions are visually different in report selection dialog boxes, and access is allowed to these archived versions.

Advanced:Schedule Any Report

Schedule any report to which the users have WriteMetadata access, including the reports that they have authored.

Advanced:Schedule Folder

Schedule a folder. Every report in the chosen folder will be processed into a pre-generated report.

Advanced:Manage Distribution List

Create, edit, or delete a distribution list.

CAUTION:

Use of this capability creates a new physical table. Therefore, this capability must be restricted to few users. Initially, this capability is not assigned to any SAS Web Report Studio role. △

Users who have been assigned this role capability must also have ReadMetadata and WriteMetadata permissions to the directory where the table will be created.

These permissions are required to enable the users to create, edit, or delete a distribution list. The table will be created in the directory referenced by the distribution library (for example, **SASApp-wrsdist**). For easier management of users assigned with this role capability, you can create a new role with this single capability added to that role, and then assign users to this role capability. Users assigned with this role capability can maintain these tables.

Note: In addition to the preceding role requirements, you must have configured a SAS Trusted User (sastrust) in order to schedule or distribute reports. You configured the SAS Trusted User during installation. The SAS Trusted User is used to establish a trust relationship with the metadata server. △

How Users Interact with Manually Refreshed Reports

Scheduled reports can be manually refreshed when users open the report.

The following list explains how users interact with manually refreshed reports:

- A user cannot interact with a manually refreshed report until the user refreshes the report.
- If a user saves changes to the live version of a report, then the original, static version of the report is deleted.
- If a user saves changes to the live version of a report and specifies that the report can be manually refreshed, then a new static version of the report is generated and saved (along with the revised live report). If the user selects the “Retain previous instances of output not to exceed” option, the report’s archive is created and or updated.
- If a user saves changes to the live version of a report using a different name for the report, then the original version of the report is preserved. In this case, a version of the revised report is not generated.

Configuring a Scheduling Server

A scheduling server must be configured to enable the scheduling of reports. The following table provides a brief overview of the high-level tasks associated with configuring a scheduling server.

Table 18.3 Tasks Associated with Configuring a Scheduling Server

Scheduling	Description
Platform Suite for SAS	Platform Suite for SAS, an integrated enterprise job scheduler, is designed to manage job flows that are generated by SAS software and includes several components such as Process Manager Server, Platform Flow Manager, Platform Calendar Editor, Platform Load Sharing Facility (LSF), and Platform Grid Management Services. The installation of Platform Suite for SAS is complex, but it provides management and execution that is independent of the SAS Web Report Studio installation and supports clustered SAS Web Report Studio environments. For more information, see “Setting Up Scheduling Using Platform Suite for SAS” in <i>Scheduling in SAS</i> .
SAS In-process Scheduling for SAS Web Report Studio	With in-process scheduling, you can schedule jobs from SAS Web Report Studio. An in-process scheduling server runs as a process inside of SAS Web Report Studio, which eliminates the need to send the jobs to an external scheduling server. In-process scheduling is easier to install and configure, and it does not require a separate license. In addition, using an in-process scheduling server simplifies authentication, because scheduled jobs do not have to be authenticated to an external scheduling server. In-process scheduling is not supported in a clustered environment. In-process scheduling servers are supported only in SAS Web Report Studio. For more information, see “Setting Up Scheduling Using SAS In-Process Scheduling” in <i>Scheduling in SAS</i> .
Schedule Manager	The Schedule Manager is a SAS Management Console plug-in that works with scheduling servers to schedule jobs that you create with SAS Web Report Studio and other applications. For more information about using the Schedule Manager to create a flow, specify dependencies, add jobs to a deployed flow, set deployed flow properties, and schedule flows, see “Scheduling Jobs Using Scheduling Manager” in <i>Scheduling in SAS</i> .
Scheduling Reports	Reports can be scheduled in two ways. You can enable report scheduling with In-Process Scheduling or with Platform Suite for SAS. For more information about enabling report scheduling, see “Enabling the Scheduling of Reports” in <i>Scheduling in SAS</i> .

After you have defined the in-process scheduling in SAS Management Console, define recipients to enable the distribution of reports to those recipients. SAS Web Report Studio users can start to schedule reports. If users are authorized to schedule reports inside SAS Web Report Studio, they can set up the report schedule and submit the report for scheduling. For information about scheduling reports, see *SAS Web Report Studio: User's Guide*.

Note: Operating system scheduling is not supported for SAS Web Report Studio. △

Verifying Permissions for the Distribution Library

SAS Web Report Studio stores information about report distribution (such as channels and recipients' e-mail addresses) in SAS data sets. By default, the storage is set up as follows:

- Data sets are in your equivalent of the `SAS-configuration-directory\Lev1\SASApp\Data\wrsdist` directory on the metadata server.
- A corresponding SAS library is registered in metadata (for example, **SASApp - wrsdist**) within the `\Products\SAS BI Report Services` folder path.
- The SAS library, **SASApp-wrsdist**, is designated as the distribution library for SAS Web Report Studio and can be viewed or managed in the BI Rep Svc Wkspace Config 4.2 Properties dialog box within the SAS Management Console Configuration Manager.
- The launch credential of the server-side pooled workspace server (for example, `sassrv`) has full access to this directory. In the standard configuration, only this account reads from or writes to this directory.

If you adjust your workspace server configuration, it might be necessary to adjust host access to this directory as follows:

- If you alter the standard configuration so that SAS Web Report Studio uses standard workspace servers, give users who create distribution lists full access, and make sure that users who distribute reports have read access.
- If you alter the standard configuration so that SAS Web Report Studio uses client-side pooled workspace servers, give every puddle login full access.

CAUTION:

The primary security concern for the wrsdist directory is limiting the write access so that users cannot add themselves to inappropriate distribution lists. △

If pooling is enabled, pooled users in the Windows environment need modify access to the directory. Pooled users in the UNIX and z/OS environments require read, write, and execute access to the directory.

Setting Up a Recipient List for Report Distribution

Overview: Setting Up a Recipient List

In order to distribute reports to recipients, you must define those recipients in the metadata repository. A library with tables for recipients lists was created during installation and configuration. The sections that follow explain how to verify the library for recipient lists, and create the recipient list tables for your reports.

(Optional) View the Library for Recipient Lists

To view the distribution library for recipient lists and assign SAS servers, follow these steps in the SAS Management Console:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Environment Management ► Data Library Manager ► Libraries**.
- 2 In the Libraries plug-in, select the distribution library (for example, **SASApp - wrsdist**). Right-click on it and select **Properties**. The SASApp-wrsdist Properties window is displayed. Note that in the **Name** field, the SAS library (referred to as Libref) is displayed. Supply a description of the distribution library (optional).
- 3 (Optional) Click the **Assign** Tab. Select one or more SAS servers. The library is assigned to the server or servers that you select from this list.
- 4 Click **Finish** to save the settings or exit the window after viewing the distribution library.

Creating a Recipient List for Report Distribution

Understanding How Recipient Lists Enable Report Distribution

Suppose that you want to distribute a report to employees in different locations around the world. The following report summarizes geographic data by country.

Continent Name	City Name	Postal Code
Australia/Pacific	Abbotsford	3067
Australia/Pacific	Acacia Ridge	4108
Australia/Pacific	Adelaide	5065
Australia/Pacific	Aitkenville	4180
Australia/Pacific	Albert Park	5014
Australia/Pacific	Albert Street	4000
Australia/Pacific	Alexandria	1405
Australia/Pacific	Alfords Point	2234
Australia/Pacific	Alice Springs	871
Australia/Pacific	Allansford	3277
Australia/Pacific	Altona	3018
Australia/Pacific	Alyangula	885
Australia/Pacific	Ambeley	4308
Australia/Pacific	Angaston	5353
Australia/Pacific	Applecross	6153
Australia/Pacific	Applecross	6193
Australia/Pacific	Applecross	6953
Australia/Pacific	Appelthorpe	4378
Australia/Pacific	Argenton	2284

This sample report was created with a group break on a variable named Country Abbreviation. The result is a separate report page for each value of Country Abbreviation.

In order to distribute the relevant report to each employee, you must create a recipient list that maps each Country Abbreviation value to one or more recipients. A recipient list, which can be created in SAS Web Report Studio, is a SAS table that contains one or more group break values along with e-mail addresses and publication channels. After you create the recipient list, you can schedule the report to be generated and distributed to the specified recipients.

Note: You can create recipient lists for reports that have more than one group break. △

Create a Recipient List

There are two ways to create a recipient list:

- Use the Distribute Report Wizard in SAS Web Report Studio.
- Use Base SAS (for example, DATA step or PROC SQL). You might want to do this if you already have a list of e-mail addresses in a mail directory, and you want to import those addresses into the table. For a sample program, see “Alternative Example: Create a Recipient List Using PROC SQL” on page 247.

Names in a recipient list must conform to SAS data set naming conventions. In order to create a recipient list by using the Distribute Report Wizard in SAS Web Report Studio, you must have the Advanced role capability of Manage Distribution List assigned to you. Initially, this capability is not assigned to the Advanced role. You can create a new role, add this capability to that role, and assign users to this role and its capability. Alternatively, you can assign the Manage Distribution List capability to the Advanced role. For information about managing roles, see “Predefined Roles” on page 200. To create a recipient list, you first use SAS Web Report Studio to create an initial list that can include your group breaks. When creating the recipient list, you add e-mail addresses and publication channels to that list.

For detailed instructions and an explanation of each dialog box in the Distribute Report wizard, see the online Help for SAS Web Report Studio.

To create a recipient list, follow these steps:

- 1 Log on to SAS Web Report Studio.
- 2 In SAS Web Report Studio, select the report that you want to distribute.
- 3 From the **File** menu, select **Distribute** to launch the Distribute Report wizard.

Distribute Report - Windows Internet Explorer provided by SAS

Step 1 of 3: Define execution time, date, and recurrence

Report: Orion Geographical Report

Run report:

☐ Now

☒ One time only

☐ Daily

☐ Weekly

☐ Monthly

☐ Yearly

Run report at: Hour: 16 Minute: 10

Starting on: August 05, 2008

Ending on: ☒ No end date ☐ August 05, 2008

< Back Next > Finish Cancel Help

- 4 In the Define execution time, date, and recurrence dialog box, choose the appropriate options to specify when you want the report to run and click **Next**. The Specify output and recipient rules dialog box is displayed.

Distribute Report - Windows Internet Explorer provided by SAS

Step 2 of 3: Specify output and recipient rules

Output

Include report as: ☒ .pdf file attachment ☐ Embedded .html

Subject line text: Geographic Data

Sender: Jim.Saunders@mgb.com

Sender display name: Jim Saunders

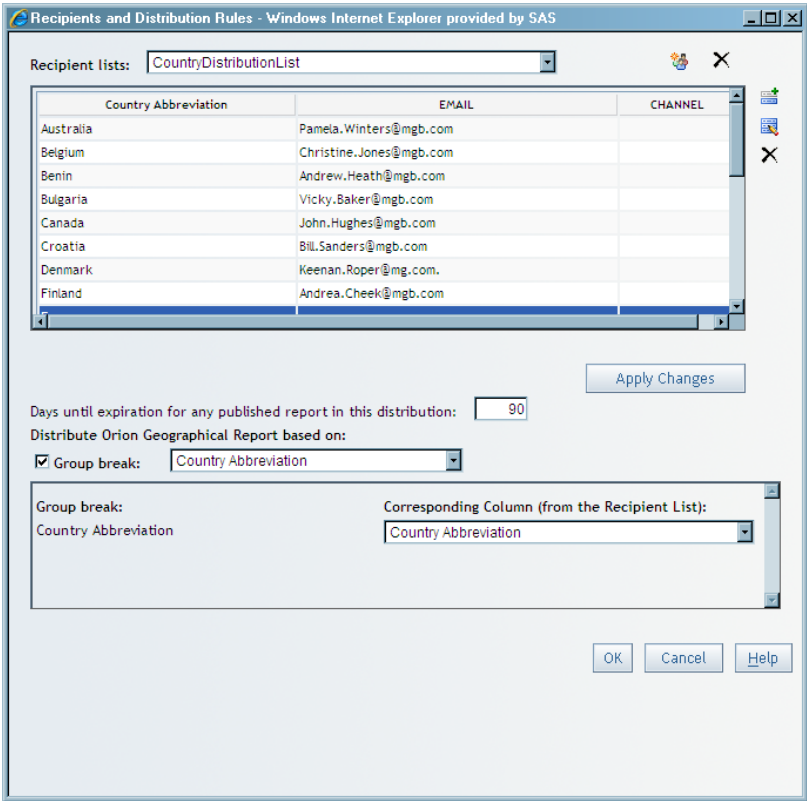
Message: Report on International Postal Codes

Recipients and Distribution Rules ...

< Back Next > Finish Cancel Help

- 5 In the Specify output and recipient rules dialog box, select the appropriate radio button to specify whether the report should be sent as a PDF file attachment or as embedded HTML.




6 Click **Recipients and Distribution Rules**. The Recipients and Distribution Rules dialog box is displayed.





Note: The icons on the top right corner of the window are displayed if you belong to the role capability of Manage Distribution List. If you are not assigned to this role capability, these icons are not displayed. △

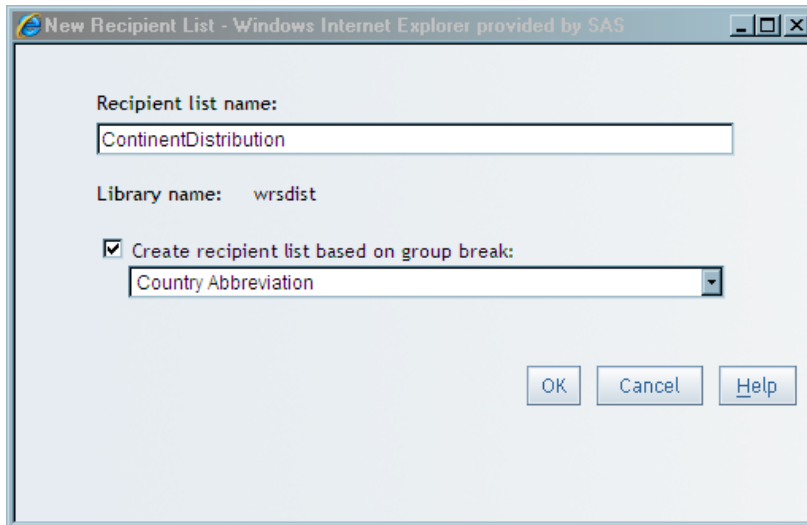
The following table explains the purpose of each icon.

Table 18.4 Icons for Managing Recipient Lists

Icon	Purpose
	Create a recipient list based on this report. For a report with group breaks, one row per group break value will be added to the resulting table. These rows allow you to specify different recipients for different group break values.
	Remove selected recipient list. If you remove a list, any distribution that references the list becomes nonfunctional.
	Add a new row to this recipient list table, and include group by values, an e-mail address, a publication channel, or both. If you do not add an e-mail address or a publication channel, that recipient is ignored during the distribution of the report. The addition of a row is needed only if the group by data changes to include a new value. The group by values can be changed.

Icon	Purpose
	Edit a selected recipient by specifying or modifying the recipient's e-mail address and publication channel.
	Delete a selected recipient.

- 7 To create a new recipient list, click on the  icon. The New Recipient List dialog box is displayed.



- 8 The following table explains the fields in this dialog box:

Table 18.5 Fields in the New Recipient List Dialog Box

Field	Description
Recipient list name	The name that you specify for the list of recipients. This name must be a valid SAS data set name.
Library name	The name of the library created in metadata for recipient lists. This field cannot be edited.
Create recipient list based on group break	The check box indicates whether you want to subset the report. When the check box is selected, you can select the group break that you want to use to subset the report. This field is available only if the section contains a group break.

- 9 Enter the name of the recipient list, and specify whether you want to subset the report. If the report contains nested group breaks, you can specify the group break level upon which the report can be subdivided. For example, you can divide on country, or on both country and city. Click **OK** to return to the main dialog box.
- 10 You can either cancel out of the wizard or continue defining a distribution. Either way, the recipient list has been created as a SAS data set within the specified Distribution Library.

Next, specify the actual recipient e-mail addresses, publication channels, or both. If you do not specify an e-mail address or a publication channel, that

recipient is ignored during the report distribution. This table has one row per Group Break value for the group break level specified previously.

- 11 Click **Apply Changes** to save the recipient list.
- 12 Click on the icon in the Recipients and Distribution Rules to launch the Add Recipient dialog box.
- 13 In the Add Recipient dialog box, fill in the information in the fields. Enter the recipient's e-mail address (not the user names as they are stored in the metadata repository), the publication channels, or both. To send a report to a group of people, enter each of their e-mail addresses or the distribution list e-mail ID (if your mail system handles distribution lists).
- 14 Click **OK** to close the Add Recipients and Distribution Rules dialog box. Click **OK**.
- 15 The Summary dialog box is displayed:

The screenshot shows a web browser window titled "Distribute Report - Windows Internet Explorer provided by SAS". The main content area is titled "Step 3 of 3: Summary". It contains the following fields and options:

- Report:** Orion Geographical Report
- Execution date/recurrence:** Runs now
- Distribute report based on:** Country Abbreviation
- Recipient list:** CountryDistributionList
- ☒ Send a copy of these results to the sender (Jim.Saunders@mgb.com)
- Distribution test options:**
 - ☒ Send a copy of these test results to the sender (Jim.Saunders@mgb.com)
 - ☒ View distribution test results now
- Run Test** button
- Navigation buttons at the bottom: **< Back**, **Next >**, **Finish**, **Cancel**, and **Help**.

In the Summary dialog box, complete the following tasks:

- ☐ Select the check box to distribute the report and send a copy of the results to yourself.
- ☐ Select one or both check boxes for Distribute test options and click **Run Test**. Before actual distribution of a report, you perform this task to verify whether the distribution is satisfactory.

- 16 Click **Finish** to exit the Distribute Report wizard.

Alternative Example: Create a Recipient List Using PROC SQL

As an alternative to creating the recipient list in SAS Web Report Studio, you can create a table manually in Base SAS. This example illustrates one way to create a table manually using PROC SQL. After you create the table, you must import the table into your metadata repository by using the Data Library Manager in SAS Management Console or other SAS code.

The following example uses a SAS library that is named `wrsdist`, a SAS table named `Burst`, and a group break variable named `Year`.

```
libname wrsdist ''\server\c$\DataSources\SAS\wrsdist'';
proc sql;
create table wrsdist.Burst (Year num, EMAIL char(256), CHANNEL char(256));
insert into wrsdist.Burst
values (2000, 'email1@abc.com', 'channelname')
values (2001, 'email2@abc.com', 'channelname')
values (2002, 'email3@abc.com', 'channelname')
;
quit;
```

Considerations for Creating Recipient Lists

Here are some things to consider when you create recipient lists:

- Although each recipient list is based on a report, a single recipient list can be used to distribute more than one report. When you use a recipient list for multiple reports, you reduce the overall number of recipient lists that must be created and maintained.

For example, the sample recipient list shown earlier can be created with multiple group break levels.

In this example, the recipient list (`CountryDistributionList`) can be used for the report that is grouped by `Country`. In addition, the same list can be used for any report that is grouped by `Country` and then by `City`. In this example, you specify the e-mail address, the publication channel, or both for a recipient in each `City`.

- There is no limit to the number of group levels that you can use in a distribution. Note, however, that each new level exponentially increases the number of recipient e-mails to define.

You can also have a recipient list with no group break column. This means that you will distribute the entire report to all recipients that are specified in the `EMAIL` or `CHANNEL` columns.

- You can leave recipient `EMAIL` or `CHANNEL` cells empty for some or all of the group breaks. When a row has empty `EMAIL` and `CHANNEL` cells, the corresponding group break report is not generated.
- Users can select the proper recipient list when they create a distribution. For that reason, you should provide descriptive names for your recipient lists. SAS Web Report Studio does not validate the relationship between group break columns in a recipient list and group breaks in the corresponding report.
- The initial rows of the distribution list, which is limited to approximately 200 rows, are displayed so the user can verify that they have selected the intended recipient list. Users assigned to the role capability of `Manage Distribution List` see all rows.
- SAS Web Report Studio stores all the lists in a single library that you defined during installation. Use a naming convention that makes sense for your organization and that prevents collisions in the event that multiple administrators create lists.

Overview of Processing Reports Outside of SAS Web Report Studio

SAS Web Report Studio enables you to run queries against reports and provide the pre-generated, static results to users. From within SAS Web Report Studio, pre-generated reports can be created manually, created on a scheduled basis, or distributed. As an alternative to using SAS Web Report Studio, you can create pre-generated, static versions of reports from the command line by using the report output generation tool. You can then use scheduling software to run the command and update the reports at specified times or intervals. For example, you might use the tool when you want to update reports on a system that does not have SAS Web Report Studio installed.

The report output generation tool can also be used to distribute reports to users. You must first create the distribution list. Then you can use Platform Suite for SAS or In-process Scheduler to run the report output generation tool and redistribute the report at specified times.

Requirements for Report Batch Processing

These are the requirements for running the report output generation tool:

- SAS BI Report Services (previously SAS Query and Reporting Services) must be installed on the system.
- To schedule report jobs using Platform Suite for SAS or In-process Scheduler, you must first enable the scheduling of reports. For more information, see *Scheduling in SAS*.

You can use alternate scheduling software with the report output generation tool. However, if you intend to distribute reports, then you will need Platform Process Manager.

- To distribute reports, you must first set up a recipient list. See “Setting Up a Recipient List for Report Distribution” on page 241.

The Report Output Generation Tool

There are three modes for running the report output generation tool:

- The *extract* mode extracts prompt information (report parameters) from reports and then writes that information to a specified file. The extract mode is used to generate a file that is then used as input for the batch or burst mode. This two-part process enables you to edit prompt values before you generate a report. Prompts enable users to input report parameters at run time. For example, a prompt for the year can be used to narrow the focus of a sales report.

If you run a report that has prompts and do not provide prompt information, then the report output generation tool uses default prompt values if they have been defined. If no default values have been defined, then the command fails.

- The *batch* mode generates a static report file.
- The *burst* mode distributes a report snapshot to specified recipients. For an overview of distribution using the report output generation tool, see “Using the Report Output Generation Tool to Distribute Reports” on page 253.

Here are the commands for the report output generation tool:

Table 18.6 Commands for the Report Output Generation Tool

Platform	Command
Windows	rptbatch.bat --extract --batch --burst <parameters>
UNIX and z/OS	rptbatch.sh --extract --batch --burst <parameters>

You use the report output generation tool (**rptbatch.bat** on Windows and **rptbatch.sh** on UNIX and z/OS) to call upon **outputgen.exe** file. The **rptbatch.bat** file is located in the *SAS-configuration-directory* \Lev1\ReportBatch directory. The **outputgen.exe** file is located in the *SAS-installation-directory* \SASBIReportSevices\4.2 directory.

Parameters for the Report Output Generation Tool

The following table describes the parameters for the report output generation tool.

Note: If a parameter value contains one or more spaces, then enclose the value in quotation marks. \triangle

Table 18.7 Report Output Generation Tool Parameters

Parameter	Description	Mode
--batch, -a	Specifies that the tool will run in batch mode.	Batch
--burst, -b	Specifies that the tool will run in burst mode.	Burst
--extract, -e	Specifies that the tool will run in extract mode.	Extract
-ps	Specifies an .xml file that contains values that are passed to the report when it is run.	All
--username <userName>, -u <userName>	Is the login ID of a user registered in the metadata repository that is specified with the --repository parameter. The user should have all the permissions necessary for generating the report, including permission to access the underlying information maps or stored processes. If you want to run a report that has been saved in a metadata schedule job definition, then provide a trusted account for the --username option (for example, sastrust), add the --trusted option to indicate that this is a trusted account, and provide the fully qualified ID of the job.	All
--password <password>, -pw <password>	Is the password that belongs to the user specified for --username .	All
--trusted, -tr	(Optional). Specifies a Boolean value that indicates whether the value supplied for --username is a trusted user. If this option is specified, then the source identifier should be a fully qualified ID for a job.	All
--repository <repository>, -r <repository>	Specifies the name of the SAS metadata repository in which the report resides.	All

Parameter	Description	Mode
--spring-xml-file <filePath>, -sxf <filePath>	<p>Used at run time to connect to the configuration service. The configuration service is used to obtain metadata connection information such as host, port, user name, mail server host, and port. The spring.xml file is in XML format, and is automatically deployed and configured in the <i>SAS-configuration-directory\Lev1\Web\Applications\SASBIReportServices4.2</i> directory. Do not edit this file.</p>	All
--source <identifier>, -s <identifier>	<p>Specifies a value such as a path URL, metadata key, or a fully qualified ID for the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> report (extract and batch mode) <input type="checkbox"/> job (extract, batch, and burst mode) <input type="checkbox"/> burst definition (extract, batch, and burst mode) <p>The path URL specifies an individual report or a directory in the report repository using a path URL that starts with SBIP:// and has a root tree name and an entity name. If the report name contains any spaces, the path URL should be enclosed within quotation marks (for example, "SBIP://METASERVER/Enterprise Reporting/SalesReports/MyReport.srx").</p> <p>A metadata key specifies an individual report or a directory in the report repository by referencing a unique identifier for a particular metadata object. The advantage of using a metadata key instead of a report URL is that the metadata key remains the same when the report is moved or renamed.</p> <p>A fully qualified ID uses <i>xxxxxxxx.xxxxxxx</i> notation to identify objects in a metadata repository (for example, A0000001.FR569JWX).</p> <p>The distribution definition is the command that is displayed for the distribution job in the Schedule Manager's properties dialog box. A distribution definition is associated with a single report.</p>	All
--log-file <logFilePath>, -l <logFilePath>	(Optional) Specifies the path and name of the log file. By default, the command logs to the console.	All
--log-level <level>, -v <level>	(Optional) Specifies the level of detailed information provided in the log. The value can be one of the following: debug , info , warn , error , or fatal . The default value is warn .	All
--help, -?	Prints the syntax for the command.	All
--prompts-file <promptsFile>, -pf <promptsFile>	<p>(Optional)</p> <p>In extract mode, this option is used to create the prompts file for the report that is specified for --source.</p> <p>In batch or burst mode, this is the name and path of the file that contains prompt information for the report to be generated.</p>	All
--no-pdf, -n	(Optional) Specifies a Boolean value that indicates whether a PDF file is generated. If this option is not specified, then a PDF is generated.	Batch

Parameter	Description	Mode
--channels <channelIdentifiers>, -ch <channelIdentifiers>	(Optional) Specifies a comma-separated list of channels to which the report will be published. The channel identifier can be a channel name, metadata key, path URL, or fully qualified ID. Channels are conduits for publishing particular categories of information. You can set up a channel for a particular topic, organizational group, user audience, or any other category.	Batch Burst
--channel-age <channel-age> --ca <channel-age>	(Optional) Specifies the number of days that the report will remain in the channel before it is deleted. If this parameter is not specified, the report does not expire.	Batch Burst
--mail-server <mailServer>, -ms <mailServer>	(Optional) Specifies the host name or IP address of the e-mail server that is used to distribute reports.	Burst
--mail-port <mailPort>, -mp <mailPort>	(Optional) Specifies the port that is used by the e-mail server. Typically, the port value is 25 .	Burst
--test-run , -t	(Optional) Specifies a Boolean value that indicates that a test distribution will be performed. This enables you to verify that your distribution configuration is correct.	Burst

Logging Directory Permissions for Report Output Generation Tool

The log directory created during installation and configuration is a placeholder for **rptbatch.bat** file. On UNIX and z/OS, the file is **rptbatch.sh** file. The log file, **SASBIReportServices4.2.log**, is typically located in the *SAS-configuration-directory\Lev1\Applications\SASBIReportServices4.2\Logs* directory.

Beginning with the third maintenance release for SAS 9.2, the **SASBIReportServices4.2.log** file is created when you first run the report output generation tool with default permission.

In order to generate a log file, the user who is running the **rptbatch.bat** must be granted read, write, and execute permissions for the log directory. If multiple users will be running the **rptbatch.bat**, each of those users must be granted read, write, and execute permissions to both the log file and the directory. This access must be granted both to users who run the tool directly from the command line, and any scheduler user such as the LSF user.

For information about logging, see “Administering Logging for SAS Servers” in the *SAS Intelligence Platform: System Administration Guide*. For information about logging that applies to SAS Web Report Studio, see “Configuring Logging for SAS Web Report Studio” on page 188.

Viewing Console Output

To view console output, modify the **rptbatch_usermods.bat** for Windows or the **rptbatch_usermods.sh** file (for UNIX and z/OS) to call upon **outputgen_console.exe**. These files correspond to the **rptbatch.bat** and the **rptbatch.sh** file. Any editing should be performed only in the **rptbatch_usermods.bat** or the **rptbatch_usermods.sh** file. You can also add command parameters to these files.

Here is an example of the **rptbatch_usermods.sh** file:

```
#!/bin/sh
#
```

```
# javabatchsrv_usermods.sh
#
# Script for managing the sasapp - Logical SAS Java Batch Server
#
# Uncomment the set -x to run in debug mode
# set -x

# Source usermods file
. /local/install/SAS/92/configdirBIDashBIRepsrv/lev1/ReportBatch
/rptbatch_usermods.sh

Quoteme() {
    if [ $# -gt 1 ]; then
        quoteme="\$*\\"
    else
        quoteme=$1
    fi
}

cmd="/local/install/SAS/92/SASBIReportServices/4.2/outputgen"

for arg in "$@" ; do
    Quoteme $arg
    tmp="$quoteme"
    cmd="$cmd $tmp"
done

eval exec $cmd $USERMODS_OPTIONS
```

The following Java batch server command invokes **outputgen.exe** with the usermods in the **rptbatch_usermods.bat** file:


```
C:\SAS\Config92\Lev1\ReportBatch\rptbatch.bat -spring-xml-file
C:\SAS\Config92\Lev1\Web\Applications\SASBIReportServices4.2\
spring.xml -repository Foundation
```

Using the Report Output Generation Tool to Distribute Reports

If you have created a report for distribution, then you can use the report output generation tool to perform that distribution.

To distribute reports, follow these steps:

- 1 Create a SAS report in SAS Web Report Studio. The report must exist in the metadata repository.
- 2 Set up a distribution library and recipient list data set.

Note: You can use PROC SQL to create the data set, and then use SAS Management Console to register the data set. For an example, see “Alternative Example: Create a Recipient List Using PROC SQL” on page 247. 

- 3 Use the report output generation tool to distribute the report.

Obtaining the Values for Running the Report Output Generation Tool on Windows and UNIX

To run the report output generation tool for a distribution job at the command line, obtain the command syntax and values from Schedule Manager in SAS Management Console.

To obtain the command syntax and values for a specific distribution job on Windows and UNIX, follow these steps:

- 1 In SAS Management Console, navigate to **Environment Management ► Schedule Manager**.
- 2 Expand Schedule Manager node and locate the distribution job that you want to run. Here is an example of the name of a distribution job:
jsaunders_FinanceReport_1261077179801
- 3 Click on the distribution job to select it.
- 4 Right-click on the distribution job, and select **Properties**.
- 5 In the dialog box that appears, select the **Scheduling Details** tab.
- 6 Click **Advanced**.
- 7 In the Advanced Properties dialog box that appears, retrieve the entire text string by copying it so that you can use this syntax to run the report output generation tool at the command line. Here is an example of a complete text string that you can run at the command line.

```
"C:\SAS\configdir\Lev1\ReportBatch\rptbatch.bat"
--spring-xml-file file:C:\SAS\configdir\Lev1
\Applications\SASBIReportServices4.2\spring.xml
--repository Foundation
--source "A6C9MMI.B0000CFU"
--burst --channel-age 50
--metadata-key "BurstDefinition+omi:\\Foundation\reposname=Foundation
\Transformation;id=A6C9MMI.B0001TUJ"
```

Note that the value displayed for the **id=** portion of the **--metadata-key** parameter should be specified for the **source** parameter when you run the report output generation tool with the **--prompts-file** parameter.

- 8 Click **Cancel** to exit from the dialog box for the distribution job.

Obtaining the Values for Running the Report Output Generation Tool on z/OS

To run the report output generation tool for a distribution job at the command line, follow these steps:

- 1 From the Schedule Manager in SAS Management Console, retrieve the server's fully qualified path name to the generated .in or .jcl file.
- 2 Navigate to the *SAS-configuration-directory/Lev1/ReportBatch/JCL/* directory.
- 3 If an .in file is present, copy the contents of the .in file into the command line text used to run the distribution job.
- 4 If an .in file is absent and a .jcl file is present, copy the relevant portion of the command syntax and values from the .jcl file into the command line text used to run the distribution job.

To obtain the command syntax and values for a specific distribution job on z/OS, and run the report output generation tool for the distribution job, follow these steps:

- 1 In SAS Management Console, navigate to **Environment Management ► Schedule Manager**.
- 2 Expand Schedule Manager node and locate the distribution job that you want to run. Here is an example of the name of a distribution job:
jsaunders_FinanceReport_1261077179801
- 3 Click on the distribution job to select it.
- 4 Right-click on the distribution job, and select **Properties**.
- 5 In the dialog box that appears, select the **Scheduling Details** tab.
- 6 Click **Advanced**.
- 7 In the Advanced Properties dialog box that appears, retrieve the server's fully qualified path name to the generated .in or .jcl file. Make a copy of the complete text string.

You will need this information to locate the corresponding .jcl or .in file for it, and run the report output generation tool from the command line. Here is an example of a complete text string:

```
SAS-configuration-directory/Lev1/ReportBatch/JCL/  
jsaunders_FinanceReport_1261077179801.jcl
```

- 8 Make a note of the complete text string.
- 9 Click **Cancel** to exit from the dialog box for the distribution job.
- 10 Navigate to the *SAS-configuration-directory/Lev1/ReportBatch/JCL/* folder on your WebSphere application server.

In the command line text for this distribution job, if all of the text strings are 80 characters or less in length, a .jcl file will be present in the folder. If one or more text strings in the command line text for this distribution job exceed 80 characters, an .in file will be present along with the .jcl file in the folder.

- 11 In the *SAS-configuration-directory /Lev1/ReportBatch/JCL/* folder, verify whether an .in file is present for the distribution job.
- 12 If an .in file is present in the folder, open the .in file and copy the contents of the .in file into the command line text used to run the distribution job. If an .in file is not present in the folder, open the .jcl file. Then, copy the appropriate text from the .jcl file into the command line text used to run the distribution job.

Here is an example of text in a .jcl file that is retrieved and copied into the command line text used to run the distribution job:

```
SH rptbatch.bat --batch  
--spring-xml-file file:C:\SAS\configdir\Lev1\Applications  
  \SASBIReportServices4.2\spring.xml  
--username sastrust@saspw  
--password {sas001}VHJ1c3QxMjM=  
--repository Foundation  
--source A0000001.FR569JWX
```

Report Output Generation Tool Examples

Example: Running a Report

The following command (executed as one line without any breaks) generates a version of the report **MyReport.srx**:

```
rptbatch.bat --batch
--spring-xml-file file:C:\SAS\configdir\Levl\Applications
\SASBIReportServices4.2\spring.xml
--username sasdemo
--password {sas001}VHJ1c3QxMjM=
--repository Foundation
--source SBIP://METASERVER/MyDepartment/Shared/Reports/MyReport.srx
```

In the command, the value provided for the source is the fully qualified name of the report.

In the Windows environment, the user name parameter should include the domain name or machine name that is followed by the user name:

```
--username <domain>\sasdemo
```

If you have spaces in the report name, insert the path within quotation marks:

```
--source "SBIP://METASERVER/MyDepartment/Shared/Reports/MyReport.srx"
```

As an alternative, you can specify the report's metadata key instead of the URL. For example, the following command generates a version of the report that is identified by the metadata key:

```
rptbatch.bat --batch
--spring-xml-file file:C:\SAS\configdir\Levl\Applications
\SASBIReportServices4.2\spring.xml
--username sasdemo
--password {sas001}VHJ1c3QxMjM=
--repository Foundation
--source 'Report+omi:\\Sales\reposname=Sales\Transformation;
id=A528654F.AY002MVJ'
```

Example: Running a Scheduled Job

The following command (executed as one line without any breaks) generates the report that is associated with a scheduled job:

```
rptbatch.bat --batch
--spring-xml-file file:C:\SAS\configdir\Levl\Applications
\SASBIReportServices4.2\spring.xml
--username sastrust@saspw
--password {sas001}VHJ1c3QxMjM=
--repository Foundation
--source A0000001.FR569JWX
```

The value provided for the source is the fully qualified ID for the job.

Example: Using a Prompts File as Input for a Report

Use the extract mode to generate an XML file that contains report prompts. You can then supply the name of the file when you run the report output generation tool in burst or batch mode.

The following command creates an XML file with the prompts extracted from a report named myReport. The command creates the file **myPromptFile.xml** in **C:\temp** directory.

```
rptbatch.bat --extract
--spring-xml-file file:C:\SAS\configdir\Levl\Applications
```

```

\SASBIReportServices4.2\spring.xml
--username sasdemo
--password {sas001}VHJ1c3QxMjM=
--repository Foundation
--prompts-file c:\temp\myPromptFile.xml
--source SBIP://METASERVER/MyDepartment/Shared/Reports/MyReport.srx

```

The following command uses the prompts file as input to generate the report.

```

rptbatch.bat --batch
--spring-xml-file file:C:\SAS\configdir\Lev1\Applications
\SASBIReportServices4.2\spring.xml
--username sasdemo
--password {sas001}VHJ1c3QxMjM=
--repository Foundation
--prompts-file c:\temp\myPromptFile.xml
--source SBIP://METASERVER/MyDepartment/Shared/Reports/MyReport.srx

```

Example: Distributing a Report

The following command (executed as one line without any breaks) generates the report associated with the ID value that is specified for the source:

```

rptbatch.bat --burst
--test-run
--spring-xml-file file:C:\SAS\configdir\Lev1\Applications
\SASBIReportServices4.2\spring.xml
--source 'A5578MBC.AL002DND'
--metadata-key "Transformation+omi://Sales/reposname=Sales/Transformation;
id=A5578MBC.AY004FR7"

```

Migrating from the Batch Generation Tool to the Report Output Generation Tool

Comparison Overview

Previous releases of SAS software included a batch generation tool (**batchgen.exe**). If you created jobs to run **batchgen.exe**, then you should be able to migrate most of those jobs to the **outputgen.exe** file, which is invoked by the report output generation tool **rptbatch.bat**.

In addition to these differences, when you set up the scheduling of jobs, the command that is supplied for the Java Batch server is different. For more information about the command, see *SAS 9.2 Scheduling*.

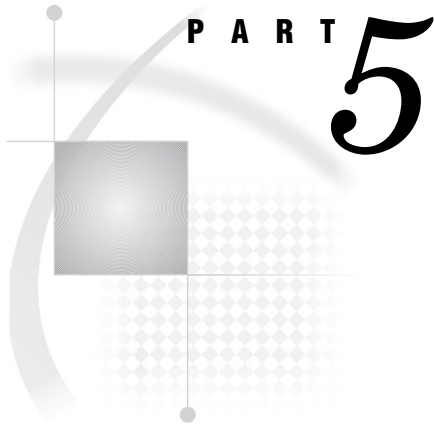
Parameter Comparison

The following table shows the parameters for **batchgen.exe** and their **rptbatch.bat** counterparts:

Table 18.8 batchgen.exe and rptbatch.bat Tool Parameters

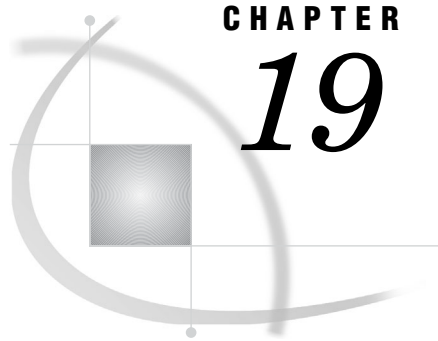
batchgen.exe		rptbatch.bat (outputgen.exe)	
Parameter	Mode (Run, Extract)	Parameter	Mode (Batch, Extract)*
run	Batch	--batch, -a	Batch
extract	Extract	--extract, -e	Extract
-username	Both	--username, -u	Both
-password	Both	--password, -pw	Both
-repository	Both	--repository, -r	Both
-workspaceserver	Both	Not applicable	Not applicable
-url -file -metadatakey	Both	--source, -s Supports the url and metadatakey options, but not the file option.	Both
-nopdf	Run	--no-pdf, -n	Batch
-channels	Run	--channels, -ch	Batch
-logfile	Both	--log-file, -l	Both
-outputFile	Extract	No exact equivalent, though --prompts-file is used to create a prompts file.	Extract
-excludePrompts	Extract	Not applicable	Not applicable
-recursive	Extract	Not applicable	Not applicable

* The burst mode is not included in this table because that mode was not supported by the Batch Generation tool.



SAS Information Delivery Portal Administration

<i>Chapter 19</i>	Overview of the SAS Information Delivery Portal	261
<i>Chapter 20</i>	Introduction to SAS Information Delivery Portal Administration	267
<i>Chapter 21</i>	Administering Portal Authorization	277
<i>Chapter 22</i>	Adding Content to the Portal	291



CHAPTER

19

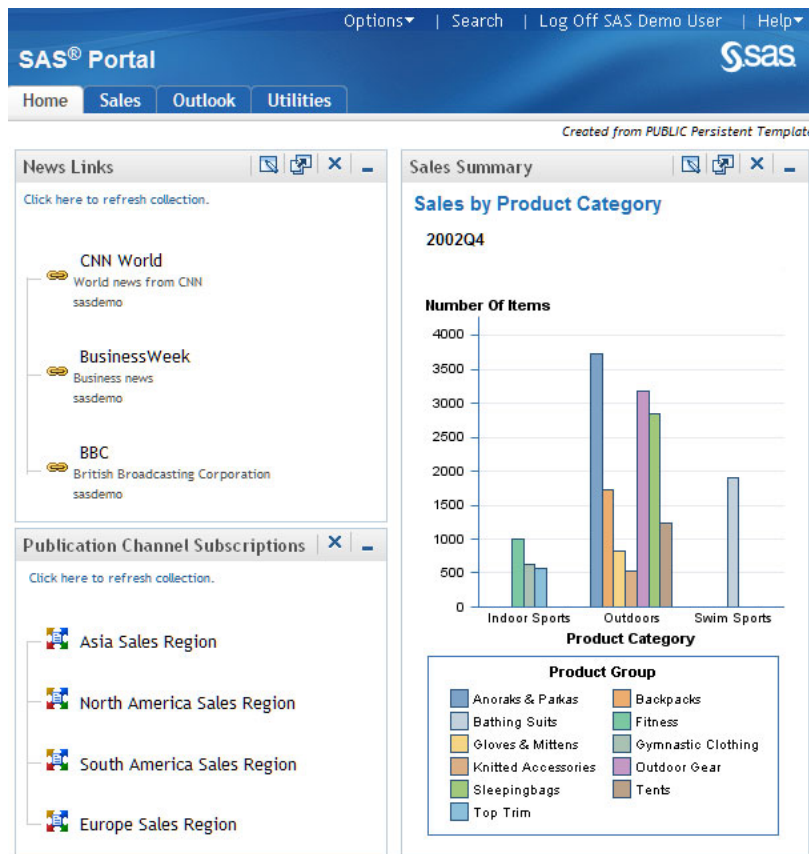
Overview of the SAS Information Delivery Portal

<i>Introduction to the SAS Information Delivery Portal</i>	261
<i>Understanding the SAS Information Delivery Portal</i>	262
<i>The SAS Information Delivery Portal</i>	262
<i>Features of the SAS Information Delivery Portal</i>	263
<i>Understanding the Portal Components</i>	263

Introduction to the SAS Information Delivery Portal

The SAS Information Delivery Portal provides a Web-based user interface that enables users to navigate and access a wide variety of information. This information includes reports, charts, Web applications, documents, and links to internal or external Web pages. You can configure security in order to ensure that users access only the information that they are authorized to see.

The portal uses portlets to organize information about Web pages. Here is a sample portal that contains links to Web sites that provide business or world news:



The portal includes portlet templates and several predefined portlets that all conform to industry-standard design patterns. In addition, developers in your organization can use the SAS application programming interfaces (APIs) to create custom portlets. These SAS APIs also provide tools to facilitate secure integration and information sharing with remote applications.

Note: In this documentation, *portal* is a generic term that refers to the SAS Information Delivery Portal. △

Understanding the SAS Information Delivery Portal

The SAS Information Delivery Portal

The SAS Information Delivery Portal provides the following capabilities:

- ☐ personalization features that enable users to create and customize their own pages and portlets
- ☐ the ability to subscribe to publication channels, and to publish content to channels or to a Web-Based Distributed Authoring and Versioning (WebDAV) repository
- ☐ support for running SAS Stored Processes in the background and receiving alert messages when processes are finished
- ☐ support for syndicated, continuously updated Web content from information providers

- access to SAS Information Maps and SAS reports via the portal, if SAS Web Report Viewer and SAS Information Map Studio are installed

Features of the SAS Information Delivery Portal

In addition to the preceding features, the SAS Information Delivery Portal provides the following features:

- Users can access the personalization options in order to update their personal views of the portal. By using these options, users can do these tasks:
 - create new portal pages, and edit or remove existing pages. Users who are authorized as content administrators can also share pages with groups of users.
 - choose the portlets that are to appear on each page, and arrange portlets in a grid layout in which the portal page is divided into one, two, or three columns and a specified number of rows.
 - create, edit, and remove collection portlets and URL display portlets. A collection portlet contains a list of content items; a URL display portlet accesses a specific URL, and then displays the returned information inside the portlet's borders.
 - create, edit, and remove WebDAV navigator portlets. These portlets enable users to access files of any type that are stored on a WebDAV server.
 - create links to intranet locations, external Web sites, or any other content that is accessible through a URL.
 - set user preferences, including country and language (locale), and theme.
 - move the portal's navigation bar to the top or side of the browser window, and change the order in which tabs appear on the navigation bar.

For details about using these options, click the Help link in the banner of the user interface.

- Users can choose to run SAS Stored Processes in the background and receive alert messages when processes are finished.
- Users can view content that has been published to SAS publication channels, manage their own subscriptions to publication channels, and publish content from the portal to a publication channel. Users can also publish portal content to a WebDAV repository and view packages that have been published to WebDAV.
- Syndicated, continuously updated Web content from information providers can be provided to users through the portal. The portal provides support for the Rich Site Summary (RSS) standard, a lightweight XML format that is designed for sharing news headlines and other syndicated Web content.
- If your organization has installed SAS Web Report Studio and SAS Information Map Studio, then users can access SAS Information Maps and SAS reports by using the portal. Depending on the software that is installed and the role capability assigned to users, the portal uses either SAS Web Report Studio or SAS Web Report Viewer to display reports.

For information about using the report and information map features, click the Help link in the banner of the user interface.

Understanding the Portal Components

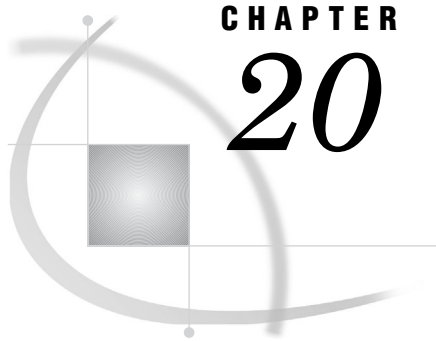
The SAS Information Delivery Portal runs in a Web application server, and requires a Java 2 Software Development Kit (SDK). The portal uses the SAS Web Infrastructure

Platform for authentication, security, and other common Web infrastructure services. The portal also uses the SAS Foundation Services for both local and remote service functionality. Finally, the portal connects with the SAS Metadata Server in order to store and obtain user, resource, and security information. For more information about this Web environment, see “Understanding the Middle-Tier Environment” on page 7.

Here are the main components of the SAS Information Delivery Portal:

- The portal, which includes the following:
 - Portal Java Classes: The foundation of the portal consists of Java classes contained in the Portal API. For complete documentation of the Java classes included in these SDKs, see the Portal API class documentation. If you want, you can use these classes to develop your own custom portlets for deployment in the portal. For details, see “Using the Portlet API” in *Developing Portlets for the SAS Information Delivery Portal*.
 - Portal Java Servlets, JSPs, and JavaBeans: The portal Web application servlets, JSPs, and JavaBeans are the active components of the portal. Using the portal Web application Java classes, these servlets, JSPs, and JavaBeans interact with the metadata server, and the SAS Workspace Server to deliver portal functionality and content to users.
 - Package Viewer: The Package Viewer enables users to display packages in the portal.
 - Visual Data Explorer: The Visual Data Explorer enables users to display SAS Information Maps in the portal.
- Custom Themes: Themes control the appearance of SAS Web applications including the portal. A theme consists of cascading style sheets (CSSs) and graphical elements, including the portal’s banner, background image, and logo. You can create your own custom themes. For information about themes, see Chapter 12, “Administering SAS Web Application Themes,” on page 153.
- SAS Stored Process Server Web application: The Stored Process Server Web application enables users to run stored processes. The Stored Process Web application can run as a stand-alone process or through the portal. The Stored Process Web application uses the Stored Process Viewer to provide input to and display output from stored processes.
- SAS Documentation Web application: The SAS Documentation Web application is a Web application that manages SAS documentation for the portal and other Web applications.
- SAS Preferences Web application: The SAS Preferences Web application manages user preferences for the portal and other SAS applications.
- SAS Web Report Studio (optional): SAS Web Report Studio is a Web application that enables users to create and view reports stored in the SAS Report Model format.
- SAS Web OLAP Viewer for Java (optional): The SAS Web OLAP Viewer for Java enables users to display OLAP data in the portal.
- SAS Content Server: The Web server also manages content that is accessible to HTTP clients. This content might be accessible through Uniform Resource Locators (URLs), or it might be accessible only through Web applications. WebDAV provides services to help manage and locate content stored on the Web server. WebDAV enhancements to the HTTP protocol enable the Web to serve as a document database. Through this database, users in remote locations can collaborate in creating and editing documents (such as SAS Reports, word processing files, images, and SAS packages) that are stored in folders (called collections) within a hierarchical file system. The portal requires WebDAV to enable users to do the following:

- run stored processes in the background and save stored process results to a WebDAV server
- use the portal alert features
- use the WebDAV Navigator portlets
- access files
- access WebDAV-based publication channels
- use WebDAV-based subscription management
- publish content to WebDAV
- Custom Portlets (optional): You can develop your own custom portlets that take advantage of the portal's content, metadata, and security services. For details about developing portlets, see *Developing Portlets for the SAS Information Delivery Portal*.
- SAS BI Portlets: Portal users can use SAS BI Portlets to access, view, or work with content items that reside in either the SAS metadata server or the SAS Content Server. SAS BI Portlets, which are available with the October 2009 Release and later, are also compatible with the WebSphere Portal. The suite of SAS BI portlets that are compliant with JSR 168 include the SAS Collection Portlet, SAS Navigator Portlet, SAS Report Portlet, and SAS Stored Process Portlet.
- Custom Applications (optional): You can develop your own custom Web applications using the SAS Foundation Services (and other Business Intelligence Services). When a foundation service-enabled Web application is invoked from the portal, the portal passes the application the session and application context, which can then be used to obtain the authenticated user (and allow single signon).
- Other SAS Solutions Web Applications: The middle tier might also manage other Web applications, such as solutions or custom Web applications.



CHAPTER 20

Introduction to SAS Information Delivery Portal Administration

<i>Prerequisites for Administering the Portal</i>	267
<i>What You Should Know</i>	267
<i>What You Should Do</i>	268
<i>Who Can Administer the Portal</i>	269
<i>Main Tasks for Administering the Portal</i>	270
<i>Provide Portal Content</i>	270
<i>Implement Security for the Portal</i>	271
<i>Set Up Portal Views</i>	271
<i>Customize the Portal's Appearance</i>	272
<i>Perform Routine Maintenance</i>	272
<i>Promote Portal Content</i>	272
<i>Remove the Configuration or Redistribute the Portal</i>	273
<i>Suggestions for Verifying Portal Operation</i>	273
<i>Important Portal Administrative Files</i>	274
<i>Logging for SAS Information Delivery Portal</i>	274
<i>Overview of Logging for the SAS Information Delivery Portal</i>	274
<i>Additional Documentation for the Portal</i>	275

Prerequisites for Administering the Portal

What You Should Know

WRITER'S NOTES - Updated on 4/23/10 to reflect 4.3 features.

In order to administer the SAS Information Delivery portal, you should familiarize yourself with the following:

- the concepts that are listed in "Prerequisites for Administering the Web Applications" on page 3.
- the SAS Web Infrastructure Platform. For full details, see Chapter 2, "Working in the Middle-Tier Environment," on page 7.
- which SAS users have permissions to administer the portal, and what additional users you should define for administration purposes. See "Who Can Administer the Portal" on page 269.
- how to use the portal, including information about the following:
 - how to start the servers and log on. See "Starting the Web Applications" on page 17.

- what the portal enables users to accomplish, and the main tasks that users can perform in the portal. For a general introduction and tour of the portal, see *SAS Information Delivery Portal: Introduction*. Also, see the online Help for the portal.
- how to create a page in the portal, add portlets to the page, add links and other items to portlets, search for items, view and navigate information maps and reports, and other tasks. For instructions, see the online Help that is provided with the portal. Also, see “Introduction to Adding Content” on page 293.
- the software components that are required for portal operation. For a description of these components, see “Understanding the Portal Components” on page 263.
- location of important portal files. See “Important Portal Administrative Files” on page 274.
- the main tasks for administering the portal. See “Main Tasks for Administering the Portal” on page 270.

What You Should Do

Before you can administer the SAS Information Delivery Portal, the portal must be functional. Make sure that you have completed the following tasks:

- 1 installed and configured the Java Software Development Kit (SDK) for JBoss or WebLogic application server. The WebSphere application server ships with its own version of the JDK which is installed when you install WebSphere.
- 2 installed and configured a Web application server such as JBoss, WebLogic, or WebSphere application server. During installation, the SAS Deployment Wizard can configure the Web application server.
- 3 using the SAS Deployment Wizard, you installed and configured the middle-tier software that is required for portal operation. In summary, the portal:
 - relies on the Web Infrastructure Platform, SAS Web Application Themes, and SAS Foundation Services
 - uses SAS Web Report Studio and SAS Stored Process Web Application
 - uses SAS application and data servers (SAS Workspace Server, SAS Stored Process Server, and SAS OLAP Server) to render SAS data and run SAS programs
 - uses the SAS Content Server for read-write HTTP functionality
- 4 defined the required SAS users on the host and in the SAS Metadata Server.
 - For a summary of these users, see “Understanding the State of Your System” in the *SAS Intelligence Platform: System Administration Guide*.
 - For information about how to create users and assign them to groups and roles, see the *SAS Management Console: Guide to Users and Permissions*.
 - To understand how role assignments affect users’ ability to perform the system administration tasks that are documented in this guide, see Chapter 3, “Who Can Do What: Credential Requirements for SAS Management Console Tasks” in the *SAS Intelligence Platform: System Administration Guide*.

In addition, you might have created additional credentials to access SAS application and data servers (SAS Workspace Server, SAS Stored Process Server, and SAS OLAP Server).

- 5 verified that your portal operates correctly. See “Suggestions for Verifying Portal Operation” on page 273.

Who Can Administer the Portal

The following table shows the recommended users who should have administrator rights and the type of permissions for each. Depending on the administrative tasks that you want to perform, you can log on to the SAS Information Delivery Portal as one of these users.

Note: Except as noted in the table, the permissions are configured during installation. You can verify the permissions for each user in SAS Management Console by looking at the authorization properties in the user’s permission tree. For more information about permission trees, see “Managing Portal Permission Trees in Metadata” on page 288. △

For a detailed explanation of user accounts and SAS administrator accounts, see “Understanding the State of Your System” in the *SAS Intelligence Platform: System Administration Guide*.

Table 20.1 Users Who Administer the Portal

User Name or Function	Default Metadata User ID	What the User Administers	Required Metadata Permissions	What the Permissions Allow
SAS Administrator	sasadm	Creates and manages metadata on the SAS Metadata Server. This account should <i>never be used</i> to log on to the portal. This account should be used only for administering metadata in SAS Management Console.	All permissions	Unrestricted user
SAS Trusted User	sastrust a user who has WriteMetadata permission granted in the portal ACT. This user has full access to portal content. By default, the sastrust user is a Portal Administrator	This user has WriteMetadata permission granted in the portal ACT and has full access to portal content. By default, this user is a portal administrator. It must be used only to log on to the portal to modify or remove content that cannot be managed through a normal account. It should never be used as a content administrator.	(All portal content) - ReadMetadata - WriteMetadata	Unrestricted access to portal content; this account must be used cautiously.

User Name or Function	Default Metadata User ID	What the User Administers	Required Metadata Permissions	What the Permissions Allow
Group content administrator	(varies)	Administers content with the group for which this user is administrator. The SAS administrator must manually configure permissions for a group content administrator. A group content administrator can be configured for the PUBLIC group. See “Configure a Group Content Administrator” on page 282.	(Group and personal content) - ReadMetadata - WriteMetadata	Create portal content and share it with the respective group. View, edit, share, unshare, and delete all content that has been granted access to or shared with the group, including content that others create.
Defined portal users	(varies)	Administer personal portal content, and perform tasks that are available from the portal’s Options menu. The SAS administrator must manually define portal users. For more information, see <i>SAS Management Console: Guide to Users and Permissions</i> .	(Personal content only) - ReadMetadata - WriteMetadata	View any portal content that has been granted public access or access to a group to which this user belongs. Create, view, edit, and delete personal portal content.

By default, when you first install the portal, all users who can access the metadata server are members of the PUBLIC group. The preceding table does not list PUBLIC group members, however, because it is expected that you will restrict the PUBLIC permissions when you set up security for your portal deployment. Actual administration should be reserved only for those users who are listed in the table.

Main Tasks for Administering the Portal

Provide Portal Content

Determine the types of content that you want to provide, and then add content items to the SAS Information Delivery Portal environment. In general, content falls into the following two main categories:

- SAS content, such as information maps, reports, stored processes, and packages that are created by the SAS Publishing Framework
- Other Web content, including Web applications, documents, links to internal or external Web pages, and syndication channels that provide syndicated, continually updated Web content

For details and instructions, see Chapter 22, “Adding Content to the Portal,” on page 291.

Developers in your organization will create many of these content items. In addition, your organization can develop custom portlets and themes for the portal. For more information, see *Developing Portlets for the SAS Information Delivery Portal*.

Implement Security for the Portal

For general security tasks, see “Middle-Tier Security” on page 39. The following security tasks apply specifically to the portal:

- Set up users for the portal.

Enable users to log on to the portal by creating metadata identities for the users. For instructions about adding users and groups, see “User Administration” in the *SAS Management Console: Guide to Users and Permissions*. See also “Planning for Portal Users and Groups” on page 278.

If you want particular users to help administer portal content for their respective groups, then you can configure these users as group content administrators. Group content administrators can create portal content and share it with members of the group. Group content administrators can also edit or remove content that has been shared with the group. For instructions, see “Configure a Group Content Administrator” on page 282.

- Manage access to content.

You implement authorization in order to control which users have which permissions for which resources. You can implement authorization for the portal in the following ways:

- Configure permissions for the users and groups that are defined in SAS metadata. You can add portal users to groups that you define in SAS metadata, grant the necessary permissions to those groups, and then limit the permissions for the PUBLIC group.
- Set up authorization for the portal content that you deploy. When you set up authorization for content, only users who have the proper authorization can access the content. The method that you use to control access varies with the type of content. For details, see “Understanding Portal Authorization” on page 280.

- Set up Web authentication.

(Optional) You can configure the portal to use Web authentication. For a detailed discussion of different types of authentication and configuration guidelines, see “Authentication Mechanisms” in the *SAS Intelligence Platform: Security Administration Guide*. For instructions on configuring Web authentication for JBoss, IBM WebSphere, or Oracle WebLogic, see the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.

Set Up Portal Views

The SAS Information Delivery Portal gives each user a personalized virtual workplace within a Web browser. This workplace is referred to as a portal view. When you deploy the portal, you can create initial portal views for different groups of users by sharing pages and content with the groups.

For example, suppose that you want to provide different types of information to engineers, to sales people, and to managers. You might first create an "engineers," "sales," and a "managers" group identity in SAS metadata. Then, you might create

pages, add information to the pages, and share the pages and information with the appropriate group. When users log on to the portal, those users who belong to one of these groups will see the pages that were shared with the group. This enables you to ensure that users have access only to the information that is appropriate for them. (To make information available to everyone in your organization, you can share information with the PUBLIC group.)

To facilitate the process of deploying views, you can designate a group content administrator for a group that is defined in SAS metadata. This person can then assume responsibility for sharing information with the respective group. For instructions, see “Configure a Group Content Administrator” on page 282.

Customize the Portal's Appearance

You can make some changes to the appearance of the SAS Information Delivery Portal that affect all portal views:

- You can set up a default theme. When users log on to the portal, they see the theme that you specify as default. In addition, you can make new themes available to portal users. For information about themes, see Chapter 12, “Administering SAS Web Application Themes,” on page 153.
- You can change the application name that appears in the banner.
- You can change the default preferences that were set during installation. For example, you can change the locale, date format, time format, and other preferences on the Configuration Manager's **Settings** tab. See “Using Configuration Manager” on page 64.

The changes will be seen by all users who log on to the portal.

All users can personalize their portal views. For example, users can change the order in which pages appear, the number of columns on a page, and other aspects of their portal views.

Perform Routine Maintenance

Here are some maintenance tasks that you might need to perform:

- Change passwords for users as needed. See “User Administration Tasks” in the *SAS Management Console: Guide to Users and Permissions*.
- Add new users, groups, and roles. See “User Administration Tasks” in the *SAS Management Console: Guide to Users and Permissions*.
- Add new custom-developed portlets, Web applications, themes, and other content.
- Update existing portal pages.
- Delete portal content items from the portal environment. For more information, see the online Help that is provided with the portal.
- Remove old publication channel files from the file system or from the WebDAV repository. (The expiration date for a package does not delete files; it only removes them from the channel.)
- If necessary, remove and then reinstall portal metadata to its initial state (the state that it was in after you installed the portal).

Promote Portal Content

Beginning with SAS Information Delivery Portal 4.3, a content promotion tool is available. This tool consists of stand-alone batch scripts, shell scripts, and metadata

extraction templates. These scripts and templates use the metadata server's import and export capabilities to promote portal metadata from a SAS 9.2 system to another SAS 9.2 system. You can promote the following types of content ins SAS 9.2 deployments:

- Portal Application Tree
- User Permissions Tree
- Portal page template
- Portal content object
- Portlet instance
- Portal page

Remove the Configuration or Redistribute the Portal

The SAS Deployment Manager enables you to remove the configuration for the SAS applications, including the SAS Information Delivery Portal. You can then use the SAS Deployment Wizard to reconfigure the portal. For more information, see “Using the SAS Deployment Manager” on page 94.

Files that are specific to the configuration of the portal are located in the *SAS-configuration-directory* /**Lev1/Logs/Configure** directory. The filenames for the portal configuration begin with `javaportal_configure` and have a time and date stamp appended to the filename. When you remove the portal's configuration, a `javaportal_unconfigure` file is also created.

You might need to move portal components to different hosts. For example, if you initially installed the portal on the same machine as other SAS components in order to develop and test custom portal content, then you can later move some or all of the portal components to different machines. The Web applications are designed to operate in a tiered environment using various servers, each of which can run on a separate machine.

Suggestions for Verifying Portal Operation

Instructions for installing the SAS Information Delivery Portal are provided with the *SAS Intelligence Platform: Installation and Configuration Guide* and SAS Deployment Wizard.

Here are some suggestions for verifying portal operation after you have completed the installation:

- 1 Start the portal. For instructions, see “Starting the Web Applications” on page 17.
- 2 Log on to the portal as the SAS Demo User (`sasdemo`) or a group content administrator. For help logging on, see the online Help that is included with the portal. If you have configured an alternate authentication provider (Web, LDAP, or Active Directory), then be sure to use the appropriate format for your logon credentials.
- 3 In the portal, view the online Help by clicking the **Help** link in the banner pane of the window. This step verifies that the SAS Documentation Web application has been successfully installed.
- 4 In the portal, set some preferences (by using the portal's **Options ► Preferences** menu). This step verifies that the SAS Preferences Web application has been successfully installed.
- 5 When you are finished testing the portal, log off the portal.

Important Portal Administrative Files

Here are the locations of important portal files:

Table 20.2 Portal Files

Files	Location
installation files	<i>SAS-installation-directory\SASInformationDeliveryPortal\4.3</i>
configuration files	<i>SAS-configuration-directory\Lev1\Web\Applications\SASPortal4.3</i> For example, on Windows, these files might be found in C:\SAS\Config\Lev1\Web .
initial configuration instructions	<i>SAS-configuration-directory\Lev1\Documents\instructions.html</i> file (for a planned installation). These configuration instructions are provided during installation. You might refer to these files to verify parts of your configuration.
log file	<i>SAS-configuration-directory\Lev1\Web\Logs\SASPortal4.3.log</i>

Logging for SAS Information Delivery Portal

Overview of Logging for the SAS Information Delivery Portal

The SAS Intelligence Platform uses a standard logging facility to perform logging for SAS servers. Logging is managed through the Logging Service window within the SAS Management Console. For an overview and guidelines about logging, see “Administering Logging for SAS Servers” in the *SAS Intelligence Platform: System Administration Guide*.

For information about how to modify and customize options in the Logging Service Configuration, see “Modifying Service Configurations” in the *SAS Foundation Services: Administrator’s Guide*. For a brief overview of logging as it applies to SAS Web applications, see “Administering Logging for SAS Web Applications” on page 105.

The logging output for the portal is stored in the **SASPortal4.3.log** file located at the *SAS-configuration-directory\Lev1\Web\Logs* directory.

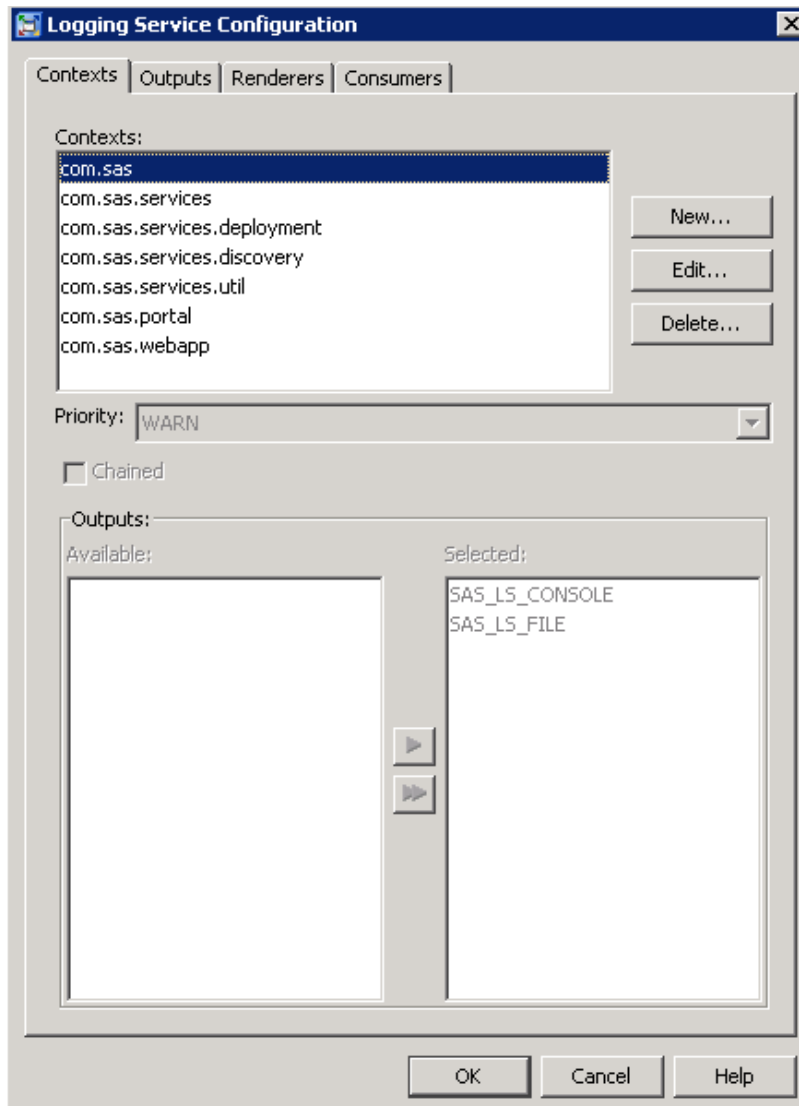
To access the Logging Service Configuration window, in SAS Management Console, navigate to **Plug-ins ► Environment ► Foundation Services Manager ► SASPortal 4.3 Local Services ► Core ► Logging Services**. Right-click and select **Properties** to display the Logging Service Properties window.

Message logging is accomplished with a logging context. A logging context is usually the fully qualified class name of the class where the logging message originated. Logging contexts are created or edited in the Logging Service Configuration window. Every logging context can have outputs associated with it. As a result, different log messages might go to different locations. By default, all portal log messages go to both

the application server console and the portal log file. In the Logging Service Configuration window, the following logging contexts are specific to the SAS Information Delivery Portal:

- com.sas.portal
- com.sas.webapp

The following figure shows the Logging Service Configuration window:

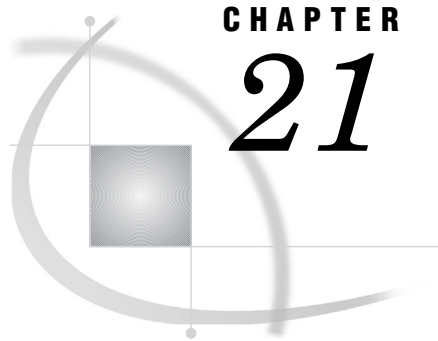


Additional Documentation for the Portal

Here is additional documentation that is available for the portal:

- For a general introduction and tour of the portal, see *SAS Information Delivery Portal: Introduction*.
- Online Help in the portal's interface provides concepts and procedures that explain how to create pages in the portal, add portlets to a page, add links and other items to portlets, search for items, view and navigate information maps and reports, and other tasks. To access the online Help, click the *Help* link in the portal.

- The **Instructions.html** file, located in the *SAS-configuration-directory* Lev1/Documents directory, contains instructions for redeploying the portal.
- For more information about middle-tier administration, see Chapter 3, “Best Practices for Configuring Your Middle Tier,” on page 23.
- For information about authentication, single sign-on, Secure Sockets Layer, and other security-related administration, see Chapter 4, “Middle-Tier Security,” on page 39.
- For information about how to develop your own custom portlets, see *Developing Portlets for the SAS Information Delivery Portal*.



CHAPTER

21

Administering Portal Authorization

<i>Overview of Portal Authorization Tasks</i>	277
<i>Planning for Portal Users and Groups</i>	278
<i>Overview of Planning for Portal Users and Groups</i>	278
<i>Step 1: Analyze and Upload Content</i>	279
<i>Files</i>	279
<i>Packages Published on a SAS Content Server</i>	279
<i>SAS Publication Channels on a SAS Content Server</i>	279
<i>SAS Application Servers</i>	279
<i>Step 2: Analyze and Group Users</i>	280
<i>Step 3: Assign Group Content Administrators</i>	280
<i>Understanding Portal Authorization</i>	280
<i>Overview of Authorization</i>	280
<i>Methods Used to Implement Authorization</i>	281
<i>Portal Items That Require Authorization, and Their Respective Authorization Methods</i>	282
<i>Configure a Group Content Administrator</i>	282
<i>Sharing Content in the Portal</i>	283
<i>Overview: Sharing Portal Content</i>	283
<i>Who Can Share Portal Content</i>	285
<i>Types of Changes That Can Be Made to Shared Content</i>	285
<i>About Shared Pages</i>	286
<i>Sharing Items That Contain Other Items</i>	286
<i>When Can You Share Content?</i>	287
<i>Suggestions for Sharing Content with Multiple Groups of Users</i>	287
<i>Setting Up Authorization for SAS Publication Channels</i>	287
<i>Managing Portal Permission Trees in Metadata</i>	288
<i>Overview of Permission Tree Folders</i>	288
<i>How Permission Tree Folders Are Created</i>	288
<i>How Permission Tree Folders Are Removed</i>	289
<i>Verify Permission Tree Folders and Permissions</i>	289

Overview of Portal Authorization Tasks

Portal authorization is accomplished by controlling access to the content that is added to the SAS Information Delivery Portal. Portal authorization concepts include the following:

- **Portal administrator.** The SAS Trusted User (sastrust) is the portal administrator, and has full access to the portal content. This user account should be used only to log on to the portal to modify or remove content that cannot be managed through a normal account. It should never be used as a content administrator. This user account should be used with caution.

- The SAS Administrator. The sasadm account creates and manages metadata on the SAS Metadata Server. This account should never be used to log on to the portal. This account should be used only for administering metadata in SAS Management Console.
- Portal users and groups. Carefully consider the types of content that you plan for the portal, and which groups you should create for that content. See “Planning for Portal Users and Groups” on page 278.
- Group content administrators. A group content administrator is delegated to managing content for a particular group of users. Group content administrators assume responsibility for creating and sharing portal content with their respective groups. For details, see “Configure a Group Content Administrator” on page 282 and “Sharing Content in the Portal” on page 283.
- Portal content. In addition to sharing content, there are other ways to control access to portal content. After you have organized your users into groups, you can configure authorization for portal content in order to allow or restrict access for members of these groups. See “Understanding Portal Authorization” on page 280.
- SAS Content Server. To authorize access to content on a SAS Content Server, users and groups that are defined in a SAS Metadata Repository can be defined. See Chapter 10, “Administering the SAS Content Server,” on page 121.
- Permission trees. The portal stores all permissions in SAS metadata, and displays the permissions in Authorization Manager in the SAS Management Console. The resources for which a portal user or group has permissions are all grouped under a folder that is designated for the user or group. These folders are called *permission tree folders*. Permission trees are created for groups when the portal administrator logs on to the portal, or the Web application server is restarted. You can also create permission tree folders manually by running the `initPortalData.bat` file on Windows or the `initPortalData.sh` file on UNIX and z/OS. This option is recommended when you have a large number of new groups that require permission tree folders.

Planning for Portal Users and Groups

Overview of Planning for Portal Users and Groups

When you define users to access the SAS Information Delivery Portal, it is recommended that you organize the users into groups. You can then grant these groups access to content based on the sensitivity of the data and the group's need for information. The use of groups is particularly important if the users have different information needs and different rights to view content.

The use of groups simplifies the process of administering and maintaining portal security, and reduces the chance for errors. Here are some guidelines to follow when creating users and groups, and administering portal content.

- Portal content includes users' personal content and group-based content. As new content is added to the portal, you can make it available to the appropriate groups based on the type of information and its level of sensitivity. This process is much simpler than giving access to a long list of individual users.
- As new users are added, you can assign them to the appropriate groups and they will automatically have access to the appropriate content.
- Users who are authorized as group content administrators can share their pages with members of the groups for which they are a group content administrator.

- The Portal ACT is used to set permissions on the Permissions trees.
- The SAS Trusted User, who is also the portal administrator, is responsible for administering the portal, and is a member of the Portal ACT. Additional portal administrators can be created by adding users to the Portal ACT and giving them ReadMetadata and WriteMetadata permissions.
- The SAS Trusted User can share any user's content with any group. However, it is highly recommended that content administration for each group is performed by the group content administrator.
- Group content administrators are responsible for administering content for specific groups. Although the SAS Trusted User, who is the portal administrator, can share any user's content with any group, this is strongly discouraged.

The following steps outline basic tasks for planning your user groups.

Step 1: Analyze and Upload Content

The SAS Information Delivery Portal contains both group content that is displayed to group members, and personal content that belongs to users. Personal content is accessed individually by each user who is granted access to the portal. For each category of content, determine the authorization restrictions, if any, that apply to the content. If restrictions are needed to view the content, then identify the types of users and groups that should and should not be authorized to access the content.

Groups can be defined for a variety of portal content including pages and portlets, Web applications, links, portlets, and syndication channels. Some of the groups that were already created for SAS Reports, SAS Information Maps, or SAS Stored Processes can be used for the portal content. For general access to the portal, users belong to the PUBLIC group, the SASUSERS group, or both.

Files

If you are storing file content on the SAS Content Server's WebDAV repository, then you must set up groups for access to the appropriate group folders.

Packages Published on a SAS Content Server

If you are publishing packages to the SAS Content Server, then it is recommended that you set up a group that contains all of the users who need the ability to publish to the server.

In addition, you should plan for the personal and group folders in which you will publish and access the packages.

SAS Publication Channels on a SAS Content Server

If you are publishing packages to a SAS publication channel on the SAS Content Server, then it is recommended that you set up a group that contains all the users who need the ability to publish to the server.

In addition, you should plan for the WebDAV personal and group folders in which you will publish and access the packages.

SAS Application Servers

Groups can be defined based on users' need to access data on particular Integrated Object Model (IOM) servers. IOM servers include SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers. In addition, if an IOM server's authentication

domain is different from the authentication domain associated with the SAS Metadata server or the Web application server, then you should set up a group definition for users to access the IOM server.

Step 2: Analyze and Group Users

After analyzing the content, you can identify groups of users. These user groups might be based on your organization's structure. However, it is more important to group users that have similar data access needs.

You might start by identifying large groups of users. You can then subdivide those large groups into smaller groups if necessary. For example, you might create an Accounting user group that needs access to financial files through the SAS Information Delivery Portal. Within that group, you can identify a subgroup of users who need access to salary information files that should not be accessed by the rest of the group. Make a list of the groups that you need to create, and identify the users to belong to the various groups.

The goal is to organize the user base in a way that reduces the number of cases in which specific users must be granted access to specific data. By keeping exception situations to a minimum, you will simplify maintenance tasks and reduce the chance for errors.

Step 3: Assign Group Content Administrators

It is strongly recommended that each group be assigned with a group content administrator who is responsible for managing the content for that group. Identify a user in each group that can function as the group content administrator. The SAS Trusted User, who functions as the portal administrator, can also perform content administration. Group content administrators can create personal pages and share their personal pages with all members of their respective group. For instructions about configuring a group content administrator, see "Configure a Group Content Administrator" on page 282.

Understanding Portal Authorization

Overview of Authorization

The SAS Information Delivery Portal uses the authorization (access control) metadata on the SAS Metadata Server to determine who can view content in the portal.

All users who log on to the portal must have ReadMetadata and WriteMetadata permissions on the Default ACT of the Foundation repository. Each portal user has access to their own personal portal content, and to the group content of any group to which they belong as a member. As part of your security implementation, you will set up authorization for particular portal content in order to allow or restrict user access to that content. For example, if the portal displays SAS reports that contain employee salary information, you should ensure that only managers can see those reports.

The methods for implementing authorization for content vary depending on the type of content. Before using any of these methods, it is generally helpful to first organize the potential users of the portal into groups. Each group should contain users who have similar job functions or similar information needs. A user can be assigned to more than one group.

Methods Used to Implement Authorization

You can implement authorization for the SAS Information Delivery Portal in the following basic ways:

- Specify ownership (personal or shared) for content in the SAS Information Delivery Portal.

By default, content that any user creates in the portal is personal. *Personal content* is content that can be edited, viewed, and deleted only by the user who created it, or by a portal administrator. When you create content in the portal, the content is added to the appropriate permission tree in SAS metadata. For example, if you log on to the portal as the SAS Demo User and create a personal page, that page is added to the SAS Demo User's permission tree.

The portal enables you to share content with a group that is defined in SAS metadata. The group can be all portal users (PUBLIC or SASUSERS) or a group that you define, such as "Sales Managers." When you share portal content with a group, the content is moved to the group's permission tree in metadata. To share portal content with a group, you must be a group content administrator for the respective group. Although a portal administrator can share portal content with a group, this is not a recommended practice.

- Specify authorization in SAS metadata.

When you create content apart from the portal, you can specify access control that explicitly allows or disallows specific types of access to individual users or groups of users. For example, if you create an information map, stored process, or publication package, then you define the authorization for the item that you created. Depending on the content type, there are several ways that you can set up this authorization:

- Use SAS Management Console to specify authorization for SAS content such as publication channels. This option provides flexibility in controlling access to portal content. For more information, see "Setting Up Authorization for SAS Publication Channels" on page 287.
- Specify authorization for custom-developed portlets in the portlet's descriptor file. Portlets that allow users to create new instances (for example, `userCanCreateMore=true`), can also be shared by using the portal's share feature.

For information about using a portlet deployment descriptor file to specify which users or groups are authorized to access the portlet, see *Developing Portlets for the SAS Information Delivery Portal*.

- Specify authorization for page templates, Web applications, and syndication channels when you run a **.sas** program that loads the respective metadata. (You can also share page templates, Web applications, and syndication channels from the portal.) For details, see the applicable topic for adding page templates, Web applications, or syndication channels in Chapter 22, "Adding Content to the Portal," on page 291.
- Specify authorization for folders in the SAS Content Server by using the SAS Content Server Administration Console. For more information, see "Using the SAS Content Server Administration Console" on page 122.

Portal Items That Require Authorization, and Their Respective Authorization Methods

For a summary of the different types of content that should have authorization configured, and how authorization is configured for each type, see “Summary of Content That Can Be Added to the Portal” on page 294.

Configure a Group Content Administrator

A group content administrator is a user who has WriteMetadata permission for the respective group, and the group’s Portal permission tree. A group content administrator can share personal content with the group, and can edit or remove content that has been shared with the group. (The SAS administrator and the SAS Trusted User has WriteMetadata permission for all group permission trees that are defined in metadata.)

Prerequisites: Before you can assign a content administrator for a group, all of the following must be true:

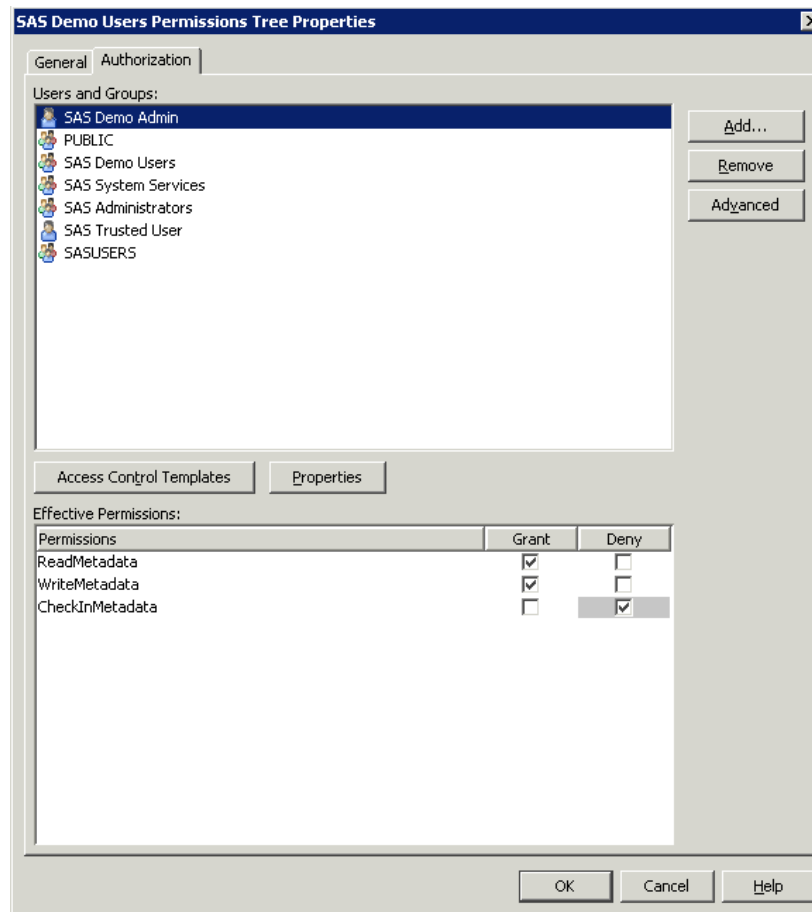
- The person who will be a content administrator must have a user identity that is defined in SAS metadata.
- This user identity must be a member of the group that the person will administer.
- A group permission tree folder must exist in metadata for the group. To verify that a permission tree folder exists, or to create one, see “Overview of Permission Tree Folders” on page 288.

To configure a group content administrator for the Portal Application Permissions tree, follow these steps:

- 1 Log on to SAS Management Console as the SAS Administrator (sasadm).
- 2 On the **Plug-ins** tab in SAS Management Console, navigate to **Environment Management ► Authorization Manager ► Resource Management ► By Type ► Tree**.
- 3 Right-click on the permissions tree for the group and select **Properties**.
- 4 In the permissions tree properties dialog box, select the **Authorization** tab.
- 5 Select the **Add** button to display the Add Users and Groups dialog box.
- 6 In the Add Users and Groups dialog box, select and move the group content administrator under Available Identities to Selected Identities. Note that the group content administrator must be a person, and not a group.
- 7 Click **OK** to exit the dialog box.
- 8 When you return to the **Authorization** tab, make sure the appropriate user is selected in the **Users and Groups** list box.
- 9 To modify the permissions for the selected user, in the permissions list row for the WriteMetadata permission, select **Grant**.

Important Note: Ensure that the permission is explicit. The check box for a permission that comes from a directly assigned access control entry (ACE) has no added background color. If the check box for a permission has a background color, to remove the background color and designate the permission as a directly assigned permission, click the check box again.

The following example display of the **Authorization** tab shows the permissions for the SAS Demo Users group tree. The WriteMetadata permission is directly assigned to the content administrator (in this example, the SAS Demo Admin user).



10 In the properties dialog box, click **OK** to save your changes.

The user that was configured as a group content administrator can now log on to the portal and share personal content with that group.

Sharing Content in the Portal

Overview: Sharing Portal Content

After defining groups in SAS metadata and initializing their respective group permission trees, the portal administrator can log on to the SAS Information Delivery Portal and create group content administrators who can manage and share portal content with their groups.

In SAS 9.1.3, the portal created permission trees for identity groups that defined the roles. In SAS 9.2, the portal does not create the permission trees associated with the roles.

Group permission trees are created when the SAS administrator logs into the portal, the Web application server is restarted, or by running the `initPortalData.bat` file. For details about permission trees, see “Overview of Permission Tree Folders” on page 288. The portal’s share feature provides an easy and efficient way to control access to particular types of portal content.

The following content items can be shared from the portal:

- ☐ pages
- ☐ portlets
- ☐ applications
- ☐ links
- ☐ syndication channels

When a content item is created, the group content administrator can share the item with a user group that is defined in SAS metadata. The group can be all portal users (PUBLIC) or a group that you define, such as "Sales Managers." When you share an item with a group, the item is owned by the group rather than by an individual. Portal users who belong to the group can access the shared item, but only a group content administrator should edit the content. Although, a portal administrator can also edit content, this practice is not recommended.

Note: The portal uses the authorization metadata of the SAS Metadata Server to determine who can view the content on a page and in a portlet. If a user is not authorized to view particular content on a page or portlet that has been shared with the user's group, then the content will not appear in that user's portal view. △

A content item can be shared with only one group. If you want to share content with users who belong to multiple groups, there are ways to work around this limitation. See "Suggestions for Sharing Content with Multiple Groups of Users" on page 287.

The *location* of a content item indicates whether it has been shared. If a content item is not shared, then the content definition is located in the user's permission tree in SAS metadata. If a content item is shared, then the content definition is located in the group's permission tree.

You can specify the location when you create the content item. For example, the following display illustrates the creation of a new page in the portal. When you select a group in the **Location (group)** drop-down list, you share the page with that group:

Add Pages to Profile

Create Search

*Name: Main Page

Description:

Keywords:

Page rank: 100 Pages are ordered by rank from lowest to highest.

Location (group): PUBLIC

Share type: Persistent

Add

Done

Note: The **Location** and **Share type** fields are displayed only if the user is a group content administrator. △

Who Can Share Portal Content

You must log on to the portal with the appropriate permissions in order to share content. Here are the types of users that can share content:

Table 21.1 Who Can Share Portal Content

User	Share Permissions
Portal Administrator	Can create and share portal content with any group that is defined in SAS metadata.
Group content administrator	Can create portal content and share it with the respective group. The SAS administrator must manually configure permissions for a group content administrator. A group content administrator can be configured for the PUBLIC group. See “Configure a Group Content Administrator” on page 282.

For more information about the permissions that are granted to these users in SAS metadata, see “Who Can Administer the Portal” on page 269.

Types of Changes That Can Be Made to Shared Content

After content has been shared with a group, group content administrators can do the following for their group:

- Edit the shared content. When you edit shared content, the changes that you make appear in all of the users’ portal views where that content is displayed.
- Unshare the content, or change the group with which the content is shared. When content is unshared with a group (for example, a page that was shared with a group is now unshared), and it is shared with another group, only that particular content item is moved. The moved page is not removed from the page list of each user. The moved portlet is also not removed from the page with which it is associated. Because the moved content is not displayed to the users, it appears as if the content has been removed. If the same page is shared again with the original group, users will see that page again. An exception applies when a user unshares a page. When a user unshares a page, the portal prompts the user to respond and confirm if portlets, associated applications, links, and syndication channels should be moved. If the user selects to move all of this content, then the entire content is moved. However, the user’s page is not removed from the page lists.
- Remove the shared content from your portal view. When a shared item is displayed in your portal, you can remove it from your view without affecting the portal views of other users.

Note: All portal users can remove a shared page from their portal views under some conditions. See “Shared Pages” on page 300. △

- Permanently delete the shared content from all portal views. When you delete shared content, the content is removed from all of the portal views where that content is displayed. The content is also permanently deleted from the portal environment.
- Change the scope (pages only). You can change the scope of a shared page (PERSISTENT, DEFAULT, AVAILABLE). For more information about scope, see “Page Attributes: AVAILABLE, DEFAULT, and PERSISTENT” on page 299.

You can make these changes for *all* content that has been shared to the group for which you are an administrator, including content that others have created. In order to modify content that another user created, you might first need to search for the content.

About Shared Pages

After you share a page with a group, when users who belong to the group log on to the portal, the shared page is available to them. The share type (DEFAULT, AVAILABLE, or PERSISTENT) that you apply to the page determines how portal users access the page.

If you share a page that contains portlets, then you can specify whether you also want to share the portlets and their contents. For details, see “Sharing Items That Contain Other Items” on page 286.

When you log on to the portal as the portal administrator, a DEFAULT or PERSISTENT page is not added automatically to your page list. You will need to add the page manually. The reason is that a portal administrator has access to all user and group content. When users log on, the pages for every group that they have access to are initialized. This can have a large performance impact when a portal administrator logs on.

For more information, see “Shared Pages” on page 300.

Sharing Items That Contain Other Items

When you share portal content, a list of contained content items is displayed. This list contains any created content that is owned by the same identity as the content being shared (page or collection portlet). In the displayed list, you can select the content that you want to share. For example, displaying only the content that is owned by the current identity helps prevent a shared PUBLIC item from being moved accidentally.

If you share a page that contains portlets, then you can specify whether you also want to share those portlets. The portal displays a list of all the portlets that are on the page and that you are authorized to share, and you choose whether to share them. Collection portlets, which display content created in the SAS Information Delivery Portal, are shared. Collection portlets can contain links, applications, or syndication channels. When you share a collection portlet, you can specify whether you also want to share the applications, links, and syndication channels that are contained in the portlet.

Note: When you share a page that contains a Bookmarks portlet, or a Publication Channel Subscriptions portlet, these portlets will not be shared. If you want to provide these portlets to users, consider creating a page template instead. △

The following is a list of portlets that cannot be shared:

- ☐ Bookmarks
- ☐ Stored Process Alerts
- ☐ Personal Repository Navigator
- ☐ Results Navigator
- ☐ Stored Process Navigator
- ☐ Tree Navigator
- ☐ Information Map Navigator
- ☐ Report Navigator
- ☐ Publication Channel Subscriptions

Within the shared pages and portlets, individual users will see only the content that they are authorized to view. Content that was created outside the portal environment,

such as SAS Stored Processes, SAS Publication Channels, SAS Packages, SAS Information Maps, SAS Reports, and files that are on a SAS Content Server, all retain the permissions that have been assigned to them in SAS metadata. Only authorized users can view the content. For example, suppose a page that you share contains two portlets, one with salary information and one with company news items. If a user who is not authorized to view salary information accesses the page, only the news items will be visible to that user.

When Can You Share Content?

Group permission trees must exist in SAS metadata before you can share content with the groups. To verify that a permission tree folder exists, or to create one, see “Managing Portal Permission Trees in Metadata” on page 288.

In the SAS Information Delivery Portal, you can share content with a group in the following situations:

- when you create a new page, portlet, application, link, or syndication channel
- when you edit the properties of a page or a portlet
- when you edit an application, link, or syndication channel

For complete instructions, see the online Help that is provided with the portal.

Suggestions for Sharing Content with Multiple Groups of Users

The SAS Information Delivery Portal enables you to share a content item with only one group at a time (though you can later switch to a different group). If you want to share content with multiple types of users simultaneously, then there are ways to work around this limitation and accomplish your goal.

Recall that the target group can be either all portal users (PUBLIC) or a group that you define in metadata, such as "Sales Managers." The group can be of any size, and it can contain other groups. If you want to share content with multiple groups, you might combine the groups into a new group that you define (for example, "All Sales"). You can then create a group content administrator for that new group to share content with the group.

Recall also that, within the shared portlets on a shared page, users are shown only the content that they are authorized to see. It is recommended that instead of providing individual access controls to portal content, you share portlets with different groups and not with specific users.

Setting Up Authorization for SAS Publication Channels

The Publishing Framework plug-in for SAS Management Console enables you to define SAS publication channels, which are conduits for publishing particular categories of information. You can set up a channel for a particular topic, organizational group, user audience, or any other category. After you have defined publication channels, authorized users can subscribe to them and automatically retrieve information whenever it is published to those channels.

Users of SAS Information Delivery Portal can manage their subscriptions from within the portal. The portal enables users to select channels to subscribe to, specify the desired delivery transport (such as an e-mail address or message queue), and specify filters that indicate which information is to be included or excluded.

In order to publish to a channel, users must have Write permission to the channel. A WriteMetadata permission is required only if a channel has an archive persistent store. To add channels or subscribers, the WriteMemberMetadata permission is required on the relevant parent folder.

For instructions about assigning permissions to channels, see the *SAS 9.2 Security Administration Guide*. For more information about the Publishing Framework and channels, see the online Help in the SAS Management Console. Also, see the *SAS 9.2 Publishing Framework: Developer's Guide*.

Managing Portal Permission Trees in Metadata

Overview of Permission Tree Folders

All portal users must have appropriate permissions in order to view, create, or edit portal content. Permissions are granted to users for particular content and resources. For example, you must give a group content administrator permission to edit the content that is associated with the respective group. All portal users are automatically granted permissions to view and edit content that they create in their personal portal views.

The SAS Information Delivery Portal stores all permissions in SAS metadata and displays the permissions in Authorization Manager in the SAS Management Console. The resources for which a portal user or group has permissions are grouped under a folder that is designated for the user or group. These folders are called *permission trees*. Permission trees can be created only in the Foundation repository.

For example, suppose that you have created a Finance group in metadata. In the Authorization Manager, a folder named **Finance Permission Tree** appears in the **Tree** list. To view the **Tree** list on the SAS Management Console **Plug-ins** tab, navigate to **Environment Management ► Authorization Manager ► Resource Management ► By Type ► Tree**. If you inspect the properties for the Finance permissions tree folder, you will find the permissions that are defined for the contents of the folder. (If a folder does not appear in the list, then you can create the folder by using one of the options described in the section “How Permission Tree Folders Are Created” on page 288.)

When you add new users or groups to the metadata server, the portal must add permission trees to the metadata before you can administer those users or groups. For example, if you create a new group in metadata, then the portal must create a permission tree folder for that group before you can share content with the group or configure a content administrator for the group. User permission trees are never modified by the portal administrator.

How Permission Tree Folders Are Created

Every user and group that is defined in metadata has its own permission tree folder. The methods that the SAS Information Delivery Portal uses to create permission tree folders depend on whether the metadata identity is a user or a group:

- User permission trees: The portal creates a permission tree for a user entity that is defined in metadata when you log on to the portal as that user.
- Group permission trees: The portal creates a permission tree for one or more groups that are defined in metadata when you do any *one* of the following:

- Restart the Web application server. The portal software creates permission tree folders for new groups each time the Web application server is started.
- Log on to the portal as a portal administrator (for example, sastrust).
- Create permission tree folders manually by running the **initPortalData.bat** utility on Windows or the **initPortalData.sh** utility on UNIX and z/OS. The **initPortalData.bat** and the **initPortalData.sh** utilities are located in the *SAS-configuration-directory\Levn\Web\Applications\SASPortal4.2\InitializePortal* directory. This option is recommended when you have a large number of new groups that require permission tree folders.

How Permission Tree Folders Are Removed

After you delete a user or group identity from SAS metadata, the SAS Information Delivery Portal removes the corresponding permission tree when you do any *one* of the following:

- Restart the Web application server.
- Log on to the portal as the SAS administrator (sasadm).
- Run the **initPortalData.bat** utility on Windows or the **initPortalData.sh** utility on UNIX and z/OS.

Once you remove a permission tree from the metadata, that tree is permanently gone. The tree will not be restored if you later create a user or group with the same name.

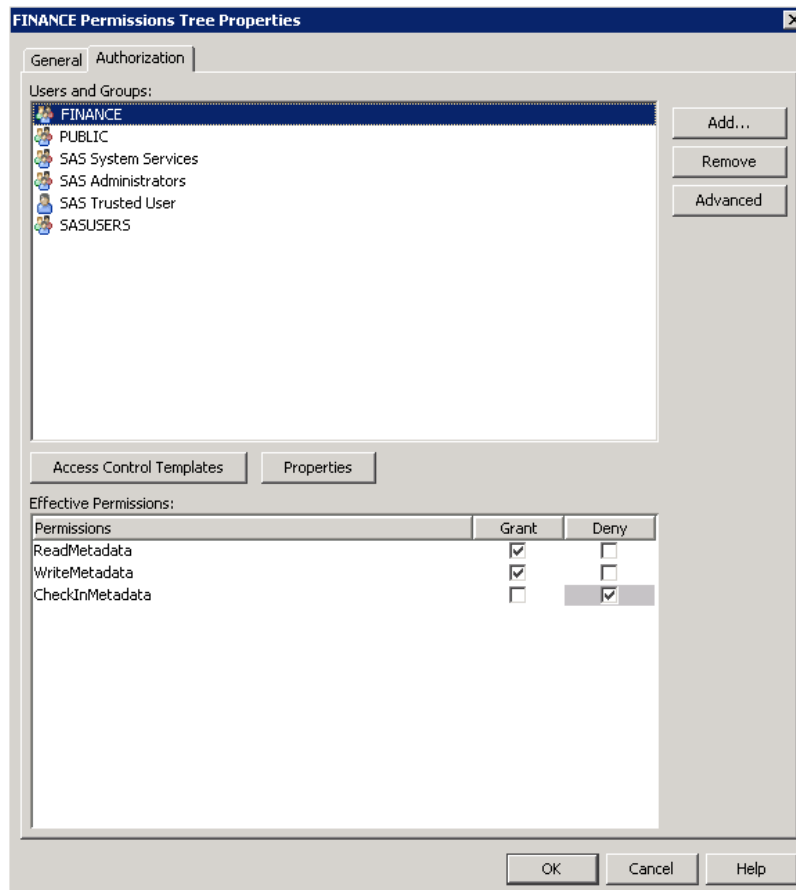
Verify Permission Tree Folders and Permissions

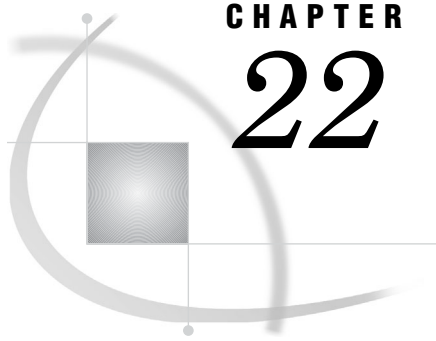
You can verify that a permission tree folder has been created for a particular user or group. You can also verify the permissions that have been granted for the resources that are associated with the user or group.

To verify that a permission tree folder has been created for a group, follow these steps:

- 1 Log on to SAS Management Console as the SAS Trusted User or as the SAS Administrator.
- 2 Navigate to **Authorization Manager ► Resource Management ► By Type ► Tree**, select the group (in our example, this is FINANCE Permissions Tree), and right-click and select **Properties**.
- 3 In the Permissions Tree Properties dialog box, select the **Authorization** tab. The permissions for the group appear in the **Permissions** list box. These permissions apply to all the resource items that are listed under the permission tree folder. (You can manually override the permissions for any of these items.)
- 4 Click **OK** to exit the dialog box for the Permission Tree Properties.

The following display of the **Authorization** tab shows the properties for the **Finance** permission tree.





CHAPTER 22

Adding Content to the Portal

<i>Overview of Adding Content</i>	293
<i>Introduction to Adding Content</i>	293
<i>SAS Application Server Requirements</i>	293
<i>Metadata Requirements</i>	294
<i>Summary of Content That Can Be Added to the Portal</i>	294
<i>Understanding Pages and Page Templates</i>	296
<i>About Pages</i>	296
<i>Who Can Administer Pages</i>	298
<i>Understanding Customized Page Deployment</i>	298
<i>Page Attributes: AVAILABLE, DEFAULT, and PERSISTENT</i>	299
<i>Personal Pages</i>	300
<i>Shared Pages</i>	300
<i>Types of Edits That Can Be Made to a Page</i>	301
<i>Page Templates</i>	301
<i>Overview of Page Templates</i>	301
<i>Main Features of Page Templates</i>	301
<i>The Home Page Template</i>	302
<i>Adding, Editing, and Removing Pages</i>	302
<i>Add and Share a Page</i>	303
<i>Edit a Page</i>	303
<i>Remove a Page from the Portal</i>	304
<i>Adding, Editing, and Removing Page Templates</i>	304
<i>Add a Page Template</i>	304
<i>Edit or Remove a Page That Was Created from a Page Template</i>	309
<i>Edit a Page Template</i>	310
<i>Delete a Page Template from the Portal</i>	310
<i>Understanding Portlets</i>	312
<i>Overview of Portlets</i>	312
<i>Custom-Developed Portlets</i>	313
<i>Portlet Templates (Editable Portlets)</i>	313
<i>Predefined Portlets That Are Provided with the SAS Information Delivery Portal</i>	315
<i>SAS BI Portlets</i>	315
<i>Main Steps to Add a Portlet</i>	316
<i>Adding WebDAV Graph Portlets</i>	317
<i>Overview of Adding WebDAV Graph Portlets</i>	317
<i>Step 1: Prepare the Data Set That You Want to Graph</i>	318
<i>Step 2: Create an XML File and Add It to WebDAV</i>	318
<i>Step 3: Create and Share a WebDAV Graph Portlet</i>	320
<i>Understanding Portlet Deployment</i>	321
<i>Overview of Portlet Deployment</i>	321
<i>Deploying Portlets</i>	321

<i>How Local and Remote Portlets Execute</i>	322
<i>Hiding Portlets from Users</i>	322
<i>Overview of Hiding Portlets from Users</i>	322
<i>Associating the Portlet with a Group</i>	323
<i>Hide a Portlet</i>	323
<i>Adding Custom-Developed Portlets</i>	324
<i>Overview of Adding Custom-Developed Portlets</i>	324
<i>Step 1: Design and Code the Portlet</i>	325
<i>Step 2: Deploy the Portlet in the SAS Information Delivery Portal</i>	325
<i>Step 3: Ensure That the Appropriate Resource Metadata Is Added to the SAS Metadata Repository</i>	325
<i>Step 4: For Remote Portlets Only, Add the Permission Statements for the Portlet to the Required Policy Files</i>	326
<i>Step 5: Implement Authorization for the Portlet</i>	326
<i>Step 6: Add the Portlet to the SAS Information Delivery Portal</i>	326
<i>Removing Portlet Configurations</i>	326
<i>Adding Links</i>	327
<i>Adding Files</i>	328
<i>Overview of Adding Files</i>	328
<i>Step 1: Add the File to the SAS Content Server</i>	328
<i>Step 2: Implement Authorization (Access Control) for the File</i>	329
<i>Step 3: Make the File Available to Portal Users</i>	329
<i>Adding Custom Web Applications</i>	329
<i>Overview of Adding Web Applications Enabled for SAS</i>	329
<i>Step 1: Design and Code the Web Application</i>	329
<i>Step 2: Ensure That the Appropriate User or Group Permissions Tree Is Created in the SAS Metadata Repository</i>	330
<i>Step 3: Add the Web Application's Metadata to the SAS Metadata Repository</i>	330
<i>Step 4: Add the Permission Statements for the Web Application to the Required Policy Files</i>	332
<i>Step 5: Implement Authorization for the Web Application</i>	332
<i>Step 6: Make the Web Application Available in the Portal</i>	332
<i>Step 7: Update or Remove the Web Application</i>	333
<i>Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java</i>	333
<i>Overview: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java</i>	333
<i>Step 1: Design and Code the Web Application</i>	334
<i>Step 2: Deploy the Web Application to the Web Application Server</i>	334
<i>Step 3: Ensure that the Appropriate Group Metadata Exists in the SAS Metadata Repository</i>	334
<i>Step 4: Add the Application's Metadata to the SAS Metadata Repository</i>	334
<i>Step 5: Implement Authorization (Access Control) for the Web Application</i>	336
<i>Step 6: Make the Web Application Available in the Portal</i>	336
<i>Step 7: Update or Remove the Web Application</i>	336
<i>Adding Syndication Channels</i>	337
<i>Overview of Adding Syndication Channels</i>	337
<i>Step 1: Add the Syndication Channel's Permission Statement to the Appropriate Policy File</i>	337
<i>Step 2: Ensure That the Appropriate User or Group Permissions Tree Is Created in the SAS Metadata Repository</i>	338
<i>Step 3: Add the Syndication Channel's Metadata to the SAS Metadata Repository</i>	338
<i>Step 4: Implement Authorization for the Syndication Channel</i>	340
<i>Step 5: Make the Syndication Channel Available in the Portal</i>	340
<i>Step 6: Update or Remove the Syndication Channel</i>	340
<i>Adding SAS Packages</i>	340
<i>Overview of Adding SAS Packages</i>	340
<i>Step 1: Publish a Package</i>	341
<i>Step 2: Define Authorization for the Package</i>	341

Step 3: Make the Package Available in the SAS Information Delivery Portal	341
Adding SAS Publication Channels	342
Overview of Adding SAS Publication Channels and Subscribers	342
About SAS Publication Channels	342
Subscribers	343
E-Mail Transport Restriction	343
WebDAV Publication Channel Considerations	343
Step 1 (Optional): Add Archive Permission Statement for the SAS Publication Channel	343
Step 2: Assign Permissions for Folders and Files in the SAS Content Server	344
Step 3: Create a SAS Publication Channel	345
Step 4: Add Subscribers to the SAS Publication Channel	345
Step 5: Implement Authorization (Access Control) for the SAS Publication Channel	345
Step 6: Make the SAS Publication Channel Available to Content Subscribers	345
Executing SAS Stored Processes from the SAS Information Delivery Portal	345
What Is a Stored Process?	346
How Stored Processes Are Executed from the SAS Information Delivery Portal	346
Characteristics of Non-Streaming Stored Processes	346
Main Tasks for Administering Stored Processes	347
About Stored Process Alerts	348
Adding SAS Information Maps	349
Overview of Adding SAS Information Maps	349
Step 1: Control Access the SAS Information Map	349
Step 2: Make the SAS Information Map Available in the SAS Information Delivery Portal	349
Adding SAS Reports	350
Overview of Adding SAS Reports	350
Step 1: Control Access to the SAS Report	350
Step 2: Make the SAS Report Available to Portal Users	350

Overview of Adding Content

Introduction to Adding Content

The SAS Information Delivery Portal 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. Depending on the software that your organization has installed, you can make some or all of these types of content available to portal users. For a list of the content that is available based on your software configuration, see “Understanding the Portal Components” on page 263.

The SAS Information Delivery Portal also provides a secure environment for sharing information with users. This chapter provides instructions for adding content to the portal and for controlling access to that content.

SAS Application Server Requirements

To add particular SAS content items, you must ensure that the appropriate servers for that content are already defined and deployed. SAS application servers are required for the following SAS content:

- packages
- publication channels
- stored processes
- information maps
- reports

Metadata Requirements

All content requires the addition of metadata to the SAS Metadata Repository. The SAS Information Delivery Portal stores its metadata in the Foundation repository. This metadata consists of the following:

- content metadata, or metadata that describes the particular content
- authorization metadata, or metadata that specifies which SAS users and groups are authorized to access the content

The metadata for SAS Information Delivery Portal content can be categorized as follows:

- For some types of content, such as SAS Publication Channels, you administer both content and authorization metadata.
- For content that is created in the portal, such as links and pages, the portal administers both content and authorization metadata.
- For WebDAV content, when you add the content to the SAS Content Server repository, the content metadata becomes available to the portal. You administer authorization metadata separately.
- For SAS content, such as information maps and reports, content and authorization metadata are administered in the SAS Information Map Studio or SAS Web Report Studio application.

For a summary of the methods that are used to add metadata for portal content, see “Summary of Content That Can Be Added to the Portal” on page 294.

Summary of Content That Can Be Added to the Portal

The following table provides a quick reference for the types of content that can be added to the portal. For each content item, the table shows the administration tools that are used to add the item’s metadata (both content and authorization metadata) to the SAS Metadata Repository or to the SAS Content Server repository. For more information about an item, or for instructions about adding the item, see the applicable topic in this chapter.

Table 22.1 Summary of Portal Content and Metadata

Content Type	Description	Where to Specify Metadata	
		Content Metadata	Authorization Metadata
Web application	A Web-based computer program.	Portal (create feature), or a SAS program	Portal (share feature), or a SAS program
File	A file of any type. Files enable the SAS Information Delivery users (who have access) to view a variety of document types in the portal.	SAS Content Server	SAS Content Server
Link	Content that is addressable using a universal resource locator (URL). Users can create links to sites on the Web or on a local intranet.	Portal (create feature)	Portal (share feature)
Page template	A Web page in the portal that is a template page that contains portlets.	Portal (create feature), or a SAS program	Portal (share feature), or a SAS program
Page	A Web page in the portal that contains portlets.	Portal (create feature)	Portal (share feature)
Custom-developed portlet	A rectangular display component of the portal in which content and links to content are displayed. Administrators can add custom-developed local or remote portlets to the SAS Information Delivery Portal.	Portlet Deploy Mechanism	Portal (share feature)*
Portlet	A rectangular display component of the portal in which content and links to content are displayed. Users can create portlets from template portlets or add predefined portlets to a page in the portal.	Portal (create feature)	Portal (share feature)
Syndication channel	A channel that provides syndicated, continuously updated Web content. For example, Really Simple Syndication (RSS) feed is available in XML format with headlines and content	Portal (create feature), or a SAS program	Portal (share feature), or a SAS program

		Where to Specify Metadata	
Content Type	Description	Content Metadata	Authorization Metadata
SAS Publication Channel	A channel created by the SAS Publishing Framework. Publication channels can be used to provide access to archived content published through the SAS Publishing Framework.	SAS Management Console	SAS Management Console
Package	A collection of structured and unstructured content that has been published to a publication channel or to SAS Content Server.	The content metadata is part of the metadata for the SAS publication channel or SAS Content Server repository	The authorization metadata is part of the metadata for the SAS publication channel or SAS Content Server repository
SAS Stored Process	A SAS program that is stored in a central location and is available to be executed on a request basis.	SAS Management Console, or SAS application (such as SAS Enterprise Guide)	SAS Management Console, or SAS application (such as SAS Enterprise Guide)
SAS Information Map	Business-oriented view of multidimensional and relational data, which can be used to develop reports. SAS Information Maps are available in the portal if your organization has installed SAS Information Map Studio.	SAS Information Map Studio	SAS Information Map Studio
SAS Report	A visual representation of data models and the results of analysis and summarization of the data from SAS procedural output. A SAS report is stored in the SAS Report Model format.	SAS Web Report Studio, or other SAS application that can create SAS Reports	SAS Web Report Studio, or other SAS application that can create SAS Reports

* Custom-developed portlets can be shared in the portal if the portlet's descriptor file contains settings that enable sharing.

Understanding Pages and Page Templates

About Pages

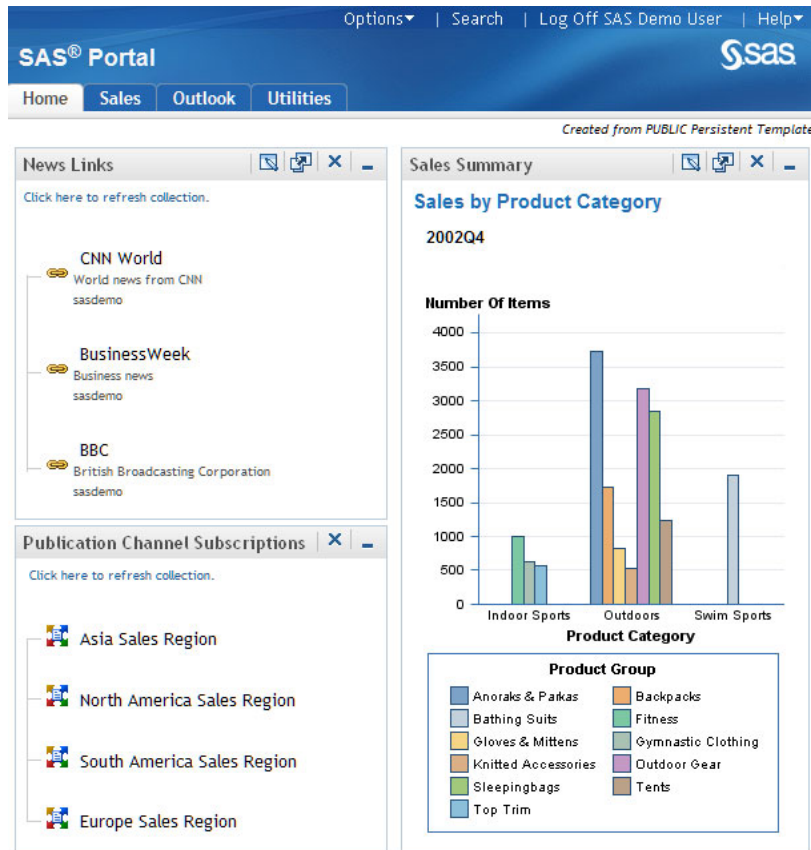
When SAS Information Delivery Portal samples are loaded during portal configuration, the following templates and links are created by default:

- home page template with a collection portlet and bookmarks portlet added to users' home pages
- link to the SAS home page
- link to SAS Integration Technologies

In addition to the home page template and the default portal links, the Portal Application tree is created when either of the following is true:

- Portal samples are loaded.
- Portal is started for the first time in the Web application server.

The SAS Information Delivery Portal uses pages to present and organize information. Here is an example of a portal that contains several pages, which are represented by links in the navigation bar:



In this example, the page named Home is the active Web page. Notice that the name of the page is highlighted in the navigation bar. To display one of the other pages, you click the page's name in the navigation bar.

Each page contains one or more portlets. *Portlets* are the rectangular display components that contain the links, graphs, reports, and other information that is available in the portal. A page can contain any number of portlets.

A banner spans the top of the portal, and the top right corner of the banner contains several links. The **Options** link in the banner displays a menu from which you can perform common management tasks. For example, you can create, edit, remove, and share pages by using this menu.

Every page inherits the theme that is applied to the portal. A theme defines the portal's colors, fonts, banner, and graphical elements. (You deploy themes separately

from pages. For more information, see Chapter 12, “Administering SAS Web Application Themes,” on page 153.)

Each page in the portal has an associated *rank* number that determines the order in which pages are listed in the navigation bar. The pages are ordered by rank from lowest to highest. Pages with equal rank are listed in the order in which they were created. The default value is 100. In the portal, you can choose to override page ranks by explicitly defining the order of pages.

You determine how rank values are used for your organization. You might use rank to group different categories of pages. For example, you might reserve a range of 1–24 for the most important types of pages in your organization. The next range of 25–49 could be used for pages that are slightly less important. When you rank pages by using a range of values, you provide the flexibility to order pages for a wide range of portal users.

Who Can Administer Pages

In addition to creating, editing, and deleting your own personal pages, you can perform the following tasks depending on your permissions:

Table 22.2 Page Administrators

User Type	What the User Can Administer
SAS Trusted User	Can share, unshare, edit, and delete any page in the SAS Information Delivery Portal, including a page that someone else has created. Although the SAS Trusted User (known as the portal administrator) can administer the pages, it is highly recommended that the group content administrator perform these tasks. By default, the SAS Trusted User is a portal administrator. A portal administrator is granted WriteMetadata permission in the Portal ACT and has full access to the portal content.
Group content administrator	Can share any page that has been created by the group content administrator. Can unshare, edit, and delete any page that has been shared with the respective group, including a page that someone else has created. The SAS administrator must manually configure permissions for a group content administrator. A group content administrator can be configured for the PUBLIC group.
All portal users	Can create, edit, and delete only personal pages.

For more information about the permissions that are granted to these users in SAS metadata, see “Who Can Administer the Portal” on page 269.

Understanding Customized Page Deployment

The SAS Information Delivery portal gives each user a personalized virtual workplace within a Web browser. This workplace is referred to as a *portal view*. When you deploy the portal, you can customize views of the portal for different groups of people by deploying different pages to those groups. This enables you to ensure that users have access only to the information that is appropriate.

Users who are deploying a page can complete the following tasks:

- 1 create the page
- 2 add portlets to the page
- 3 add content to the portlets

In addition to the tasks performed by users, content administrators can perform the following tasks:

- set up access control for the content. This task can be accomplished with the SAS Management Console Configuration Manager plug-in.
- share the page with a group of users who have permission to access the content.

For example, suppose that you want to provide different types of information to engineers, to sales people, and to managers. First, make sure that a group identity has been defined in SAS metadata for each type of user, and that the group contains the applicable user definitions. Next, you create pages and share them with the appropriate group. The portal users who belong to the group see only the pages that are shared with their group.

After users log on to the portal, they can edit pages, add new pages, and personalize their views. For example, users can subscribe to content channels that are of interest to them, create links to Internet sites that are visited frequently, and change the navigation scheme.

To facilitate the process of deploying views, you can designate a group content administrator for each group that is defined in SAS metadata. This person can assume responsibility for creating and sharing pages with the respective group.

Page Attributes: AVAILABLE, DEFAULT, and PERSISTENT

Pages have attributes that define how a page is associated with a portal view. These attributes determine the following:

- whether the page is added automatically to portal views, or whether users add the page manually to their views
- whether users can remove the page from their portal views after the page has been added

The following table shows the three attributes that associate pages with portal views:

Table 22.3 Page Attributes

Attribute	Description
AVAILABLE	The page is added manually to portal views. Users typically search for the page and then add it to their portal views. After adding the page, users can later remove the page from their portal views if they no longer need the page.
DEFAULT	The page is added automatically to portal views. Users see the page when they log on to the SAS Information Delivery Portal. Users can later remove the page if they do not want it in their views.
PERSISTENT	The page is added automatically to portal views. Users <i>cannot</i> remove the page from their portal views.

DEFAULT is the default value for every page that you create in the portal. After you create a page, if you share that page, then you can apply any one of the three attributes to the shared page. For more information, see the section “Shared Pages” on page 300.

Personal Pages

When you create a new page in the portal, that page by default is a personal page. A *personal page* is owned by the person who created it. All users who can log on to the portal can create personal pages in their portal views.

Here are the characteristics of a personal page:

- The user who created the page owns that page. This user can edit and remove the page.
- No other users can access, edit, or remove the personal page, except a portal administrator.
- If the owner of a page is also the group content administrator, that owner can share that page with other groups. Shared pages are described in the next section.

Shared Pages

If you have administrative permissions, you can share a page with a user group that is defined in SAS metadata. You can share pages with either of the following groups:

- PUBLIC group, which contains all portal users. It is convenient to share pages with the PUBLIC group because all users in that group, by default, have access to the pages.
- A specific group that you define, such as “Sales Managers.” In this case, the pages are shared only with that specific group you defined.

When you share a page with a group, you do not create multiple instances of the page. There is only one instance of the page, but that page is owned by a group rather than by an individual.

If you share a page that contains portlets, you can specify whether you also want to share the portlets and their contents.

Note: When you share a page that contains a Stored Process Alerts portlet, a Bookmarks portlet, a Navigator portlet, or a Publication Channel Subscriptions portlet, these portlets cannot be shared. If you want to provide these portlets to users, consider creating a page template instead. See “Page Templates” on page 301. △

After you share a page with a group, when users who belong to the group log on to the portal, the shared page is available to them. The share type attribute (DEFAULT, AVAILABLE, or PERSISTENT) that you apply to the page determines how portal users access the page:

- Pages that have a share type of DEFAULT or PERSISTENT are added automatically to portal views for the respective group members. (The page is not added automatically to the portal view for portal administrators.)
- For pages that have a share type of AVAILABLE, group members must search for the page before they can see it. Group members can add the AVAILABLE page to their portal views.

Only users who are authorized as an administrator for the group can edit a shared page. You can edit both the content and the properties of a shared page. For example, you can add or remove portlets, and you can unshare the page. Any changes that you make to a shared page are seen by all users who can access the page.

You can also permanently delete a shared page from the portal. When you permanently delete a page, that page is removed from all portal views.

Portal users who can access the shared page can remove the page from their individual portal views if the shared page has an attribute of DEFAULT or AVAILABLE. If the shared page has an attribute of PERSISTENT, the portal

administrator or group content administrator can remove the page from a portal view. However, it is strongly advised that the group content administrator perform this task.

For general information about sharing portal content, see “Sharing Content in the Portal” on page 283.

Types of Edits That Can Be Made to a Page

If you have the appropriate permissions, you can edit a page in several ways:

- ☐ change the label, description, or keywords for the page
- ☐ change the number of columns on the page, and re-position the portlets within the columns
- ☐ add predefined or custom portlets to the page, or create new portlets and add them to the page
- ☐ remove portlets from the page
- ☐ share a page, unshare a page, or change the group to which the page is shared
- ☐ change the attribute that is applied to a shared page (for example, you can change the shared page from AVAILABLE to DEFAULT)
- ☐ change the page rank that determines the order in which pages appear in the portal

You edit pages by using the portal **Options** menu. For more information, see the online Help that is provided with the portal.

Page Templates

Overview of Page Templates

A page template is a page definition that is stored in SAS metadata and that is associated with a group (either PUBLIC or a group that is defined explicitly). The page template must have an attribute of either DEFAULT or PERSISTENT. You can create a page template from an existing page in the SAS Information Delivery Portal, or you can create a page template by running a SAS program.

When you define a page template and add it to SAS metadata, the following occurs:

- 1 The portal’s metadata associates the page template with the group that you specified when you created the page template.
- 2 When a user logs on to the portal, the portal checks to see whether the user belongs to the specified group. If the user belongs to the group and does not yet have a page associated with this template, the portal creates a page from the template and adds that page to the user’s portal view.
- 3 The portal checks for new templates every time the user logs on, and adds new pages as appropriate.

Note: Page templates are not related to portlet templates or themes. These are different entities that use the name “template.” For information about portlet templates, see “Understanding Portlets” on page 312. For information about SAS Web Application Themes, see Chapter 12, “Administering SAS Web Application Themes,” on page 153. △

Main Features of Page Templates

Page templates offer an alternative to shared pages, and there are several reasons why you might want to implement page templates. Page templates provide the following features:

- Page templates enable you to deploy pages without logging on to the SAS Information Delivery Portal. Unlike shared pages, the portal does not need to be running in order for you to create a page template and add it to metadata. A SAS program, `LoadPageTemplateExample.sas`, is provided with the portal.
- A separate page is created for each user. Each user can edit the page that is added to his or her portal view without changing the pages that have been added to other user's views.

Note: All portal users can edit the portlets on the page by adding or removing content. △

- Page templates can contain portlets that are not shareable. For example, a page template can contain a Stored Process portlet, an Alerts portlet, a Bookmarks portlet, or a Publication Channel Subscriptions portlet.
- As with shared pages, when you add content to portlets on page templates, users are shown only the content that they are authorized to see.
- Page templates are defined as either `DEFAULT` or `PERSISTENT`. This means that the pages are always added automatically to portal views.
- Portal users can remove the page from their portal views as follows:
 - If the page is defined as `DEFAULT`, users can remove the page from their portal views.
 - If the page is defined as `PERSISTENT`, users cannot remove the page unless the portal administrator first removes the page template from SAS metadata.

After you have created a page template, you cannot edit it. However, you can delete the page template and create it again. For more information, see “Adding, Editing, and Removing Page Templates” on page 304.

The Home Page Template

After you install and configure the SAS Information Delivery Portal, when you first log on to the portal, you see the Home page. The purpose of the Home page is to help users as they begin to add content to their portal views. Here are the characteristics of the Home page template:

- The template has an attribute of `PERSISTENT`, and is associated with the `PUBLIC` group. This means that all users see the Home page when they log on to the portal, and they can search for and add the page to their portal view.

Users cannot remove the Home page from their portal views unless the portal administrator first removes the Home page template from SAS metadata.

- The page contains a collection portlet and a Bookmarks portlet.
- All users can edit the portlets by adding or removing content. Users can also edit the page. See “Types of Edits That Can Be Made to a Page” on page 301.

Although the portal administrator can see every user's Home page, it is not a good practice to use the portal administrator's account to perform regular tasks such as viewing the Home page or adding a Home page because it is difficult to distinguish whether the Home page belongs to a particular user or to the portal administrator.

Adding, Editing, and Removing Pages

Add and Share a Page

Content administrators are responsible for deploying custom views for particular groups of portal users. To accomplish this, you first create the pages, add content to those pages, apply security constraints to the content, and finally share the pages with a user group.

For basic concepts related to pages, see “Understanding Pages and Page Templates” on page 296.

Here is a summary of the steps required to add and share a page. For complete instructions, see the online Help that is provided with the portal:

- 1 Using the SAS Management Console, verify that a permissions tree folder exists for the group with which you want to share the page. If necessary, create a permission tree folder. See “Managing Portal Permission Trees in Metadata” on page 288.
- 2 Log on to the portal as a group content administrator for the respective group.
- 3 Use the portal **Options** menu to create a new page or to add an existing page to your portal view.
- 4 Add the portlets and the content that are appropriate for the group with which you intend to share the page. For instructions, see the online Help that is provided with the portal.
- 5 Implement authorization for the contents on the page. Take any necessary steps to control access to files, reports, or other items that have been added to the portlets on this page. For general information about access control, see “Understanding Portal Authorization” on page 280.
- 6 Edit the page in order to share the page publicly or with a group that is defined in SAS metadata. When you share a page, you specify an attribute of DEFAULT, AVAILABLE, or PERSISTENT. For a description of these attributes, see “Page Attributes: AVAILABLE, DEFAULT, and PERSISTENT” on page 299.

If the page contains portlets that you have permission to share, you can specify whether you also want to share the portlets. When you share a portlet, you can specify whether you also want to share any applications, links, and syndication channels that are contained in the portlet. For details about sharing portal content, see “Sharing Content in the Portal” on page 283.

Note: All users can add personal (unshared) pages to their portal views by using the portal **Options** menu. For more information about personal pages, see “Personal Pages” on page 300. △

Edit a Page

You edit a page using the **Options** menu in the SAS Information Delivery Portal. For instructions, see the online Help that is provided with the portal.

Here are some main points related to editing pages:

- Log on to the portal using a login that is appropriate for the type of page that you want to edit. For example, a group content administrator can edit a page that has been shared with the respective group. To see who can administer particular types of pages, see “Who Can Administer Pages” on page 298.
- After logging on, to edit a page that someone else created, you might first have to search for the page. You can add the page to your portal view.
- Any changes that you make to a shared page are seen by all users who can access the page.
- For a summary of the types of edits that you can make to a page, see “Types of Edits That Can Be Made to a Page” on page 301.

Note: The information presented here does not apply to page templates. For information about page templates, see “Adding, Editing, and Removing Page Templates” on page 304. △

Remove a Page from the Portal

You remove a page using the **Options** menu in the SAS Information Delivery Portal. For instructions, see the online Help that is provided with the portal.

Here are some of the main points related to removing pages:

- Log on to the portal using a login that is appropriate for the type of page that you want to remove. To see who can administer particular types of pages, see “Who Can Administer Pages” on page 298.
- You have the following options for removing a page:

Table 22.4 Options for Removing a Page

Option	Description
Remove the page from your personal portal view.	The page no longer appears in your personal portal view, but remains in the SAS metadata so that you can add it later. If the page is shared, this option has no effect on other portal views.
Remove and delete the page from the system permanently.	The page is deleted from the SAS metadata, and therefore is not available to add later. If the page is shared, this option removes the page from all portal views. If the page contains one or more portlets that you are authorized to delete, then you can delete those portlets. If authorized, you can also delete the contents of the portlets.

- All users can remove pages according to the following rules:
 - All portal users can remove or permanently delete personal pages that they created.
 - If a page was shared with a **DEFAULT** or **AVAILABLE** attribute, all portal users who can access the page can remove the page from their portal views.
 - If a page was shared with a **PERSISTENT** attribute, only the portal administrator or the associated group content administrator can remove the page.

Note: The information presented here does not apply to page templates. For information about page templates, see “Adding, Editing, and Removing Page Templates” on page 304. △

Adding, Editing, and Removing Page Templates

Add a Page Template

A page template is a specific implementation of a page definition. Page templates enable an administrator to define which pages new users see the first time they log on

to the portal. Page templates are always associated with a group that is defined in SAS metadata.

For more information about page templates, see “Page Templates” on page 301.

Before you can create a page template and associate it with a group, you must create a permissions tree in SAS metadata for the group. To verify that a permissions tree exists, or to create one, see “Managing Portal Permission Trees in Metadata” on page 288.

There are two ways to create a page template and add it to the portal:

- In the portal, create a page template from an existing page.

When you create a page template in the portal, the portal adds the page template’s metadata to the metadata repository.

Here is a summary of the steps that are required to create and share a page template. For complete instructions, see the online Help that is provided with the portal:

- 1 Log on to the portal as the group content administrator for the group.
- 2 Create a page in the portal, and add the contents that are appropriate for the page.
- 3 Use the **Options** menu to convert the page to a template.
- 4 When you convert the page to a template, you can share the template with a group that is defined in SAS metadata. For details about sharing portal content in general, see “Sharing Content in the Portal” on page 283.

- Create a page template by running a SAS program.

To edit and run a SAS program that creates a page template and adds the template metadata to the SAS metadata repository, follow these steps:

- 1 Modify the SAS program **LoadPageTemplateExample.sas**, which is located in *SAS-configuration-directory* \Lev1\Web\Applications\SASPortal4.2\sasJobs directory. In the **LoadPageTemplateExample.sas** file, specify the appropriate variables for your page template.
- 2 After you have modified **LoadPageTemplateExample.sas**, save your changes and run the program. After you run the program, a page is created for each user who is a member of the specified group.

Here are descriptions of the variables that are in **LoadPageTemplateExample.sas**:

options metaserver=“host”

Specify the host name of the SAS Metadata Server. Use the value of the `iomsrv.metadatasrv.host` property in the **configuration** file located in the *SAS-configuration-directory* \Lev1\Utilities directory. For example:

```
localhost
machine
machine.mycompany.com
```

metaport=port

Specify the port number of the SAS Metadata Server. This value is a number between 0 and 65536. Use the value of the `iomsrv.metadatasrv.port` property in the **configuration** file located in the *SAS-configuration-directory* \Lev1\Utilities folder.

metauser=“user ID”

Specify the user ID to use to connect to the SAS Metadata Server; this user ID is typically the `sastrust` user (default, `sastrust@saspw`).

metapass=“password”

Specify the password for the *metauser*. Make sure that this file is secure, or delete the file when you are finished. This file contains the password for the SAS Trusted User.

```
metarepository="repository";
```

Specify the name of the SAS Metadata Repository in which your portal metadata is stored, followed by a semicolon (;). Use the value of the `oma.repository.foundation.name` property in the **configuration** file located in `SAS-configuration-directory\Lev1\Utilities` directory.

```
%let repositoryName="Foundation";
```

```
%let groupName=SAS Group;
```

Specify the group that you want to add the data to, followed by a semicolon (;). This group must be the same as the group that you verified or created in Step 1 in this topic. Before you can run *LoadPageTemplateExample.sas*, the group permissions trees must be created in the SAS Metadata Repository.

```
%let pageName=Page Template Name;
```

Specify the name of the page template that you want to create, followed by a semicolon (;).

```
%let pageDescription=page template description;
```

Specify the description of the page template that you want to create, followed by a semicolon (;).

```
%let shareType=Default | Sticky;
```

Specify whether the page is Persistent or Default, followed by a semicolon (;).

Default Default user or group pages are automatically added to the portal of the user, or of all users in a group. The users can later remove the page.

Sticky Persistent group pages are automatically added to the user's portal, or all users in the group. Users cannot remove the page.

```
%let profile=DESKTOP_PROFILE;
```

Do *not* change this value. (This metadata exists in order to allow for future expansion. Currently, only desktop profiles are supported.)

```
%let role=DefaultRole;
```

Do *not* change this value. (This metadata exists in order to allow for future multiple roles. Currently, only the default role is supported.)

```
%let pageRank=pageRank;
```

Specify the page rank that you want for this page template, followed by a semicolon (;). All pages that are created from this page template have this page rank.

Pages are ordered in the portal by rank from lowest to highest. Pages with equal rank are listed in the order in which they were created. Portal users can choose to override page ranks by explicitly defining the order of pages.

```
data pageTemplate;
```

Specifies the SAS data set that defines the data values for the page template contents.

Do *not* modify the section between the "data pageTemplate" line and the "cards4" line, which describes the data in the data set. The data between the "cards4" line and the ";;;" line describes each portlet (1 per line) that you want to place on the page template. The line is formatted as a comma-separated value (CSV) line, and each column denotes a different value for the portlet. For example, the following lines create four portlets, with two portlets in each column on the page:


```

1,1,Bookmarks,Bookmarks portlet Description,Bookmarks template
1,2,Alerts,Alerts portlet Description,Alerts template
2,1,My Links,Description of portlet,Collection template
2,2>Welcome>Welcome portlet Description>Welcome template
;;;

```

Specify the information for each portlet as follows:

columnNum

Specify the column number in which to place the portlet on the page. Each page in the portal can have up to three columns.

portletPos

Specify the position of the portlet within the column. The portlets are positioned in ascending order from top (lowest number) to bottom (highest number).

portletName

Specify the name of the portlet to add. This is the name that identifies the portlet on the page. If a portlet already exists with the same *portletName*, the existing portlet is used, and a new portlet is not created. This field cannot contain a comma.

portletDescription

Specify the description of the portlet to add. This field cannot contain a comma.

prototypeName

Specify the name of the prototype that was created when the portlet was deployed. The prototype is the name of the portlet's template. Here are examples of pre-defined prototype names:

```

Alerts template
Bookmarks template
Collection template
DAVContent template
InformationMapNavigator template
PersonalRepositoryNavigator template
PubChannelSubscriptions template
ReportNavigator template
ResultsNavigator template
StoredProcessNavigator template
TreeNavigator template
WebDAVNavigator template
WebDAVGraph portlet
InfoMapView portlet
DisplayURL portlet
Welcome template

```

The template name for custom portlets is "<portletName> template". This field cannot contain a comma.

data properties;

Specify the SAS data set that defines any additional data properties that are needed by the portlets. This variable enables you to add default starting data to template portlets.

If there are no additional data properties for any portlets, delete the section between the "cards4;" and ";;;" lines.

Do not modify the section between the "data properties" line and the "cards4" line, which describes the data in the data set as follows:

```

data properties;
length colPos $80 propName $80 propValue $80;
infile cards4 delimiter=',';
input colPos propName propValue;
cards4;

```

The data between the “cards4” line and “;;;” lines describes each property (one per line) that you want to define. The line is formatted as CSV, and each column denotes a portion of the property definition. For example, the following lines create four properties, one for the first portlet in the first column and three for the first portlet in the second column.

```

1_1,Name0,DefaultValue 0
2_1,MyCollectionPortletPropertyName1,DefaultValue 1
2_1,MyCollectionPortletPropertyName2,DefaultValue 2
2_1,MyCollectionPortletPropertyName3,DefaultValue 3
;;;

```

Specify the information for each property as follows:

colPos

Specify the column and position of the portlet to which you are adding this property. The format is <column>_<position>.

propName

Specify the name of the property to add. This field cannot contain a comma.

propValue

Specify the value of the property to add. This field cannot contain a comma.

data collectionData;

Specify the SAS data set that defines any collection or bookmark links to add to collection or Bookmarks portlets.

If no links are required for any of the portlets, delete the section between the “cards4” and “;;;” lines.

Do *not* modify the section between the “data collectionData” line and the “cards4” line, which describes the data in the data set as follows:

```

data collectionData;
length colPos $80 dataType $80 searchStr $80;
infile cards4 delimiter=',';
input colPos dataType searchStr;
cards4;

```

The data between the “cards4” line and the “;;;” line describes each link (one per line) that you want to add to Collection or Bookmarks portlets. The line is formatted as CSV, and each column denotes a portion of the link definition. For example, the following lines create three links, one for the first portlet in the first column and two for the first portlet in the second column.

```

1_1,Document,@Name=' SAS '
2_1,Document,@Name=' CNN '
2_1,Document,@Name=' CNNSI '
;;;

```

Specify the information for each link as follows:

colPos

Specify the column and position of the portlet to which you are adding this link. The format is <column>_<position>.

dataType

Specify the metadata type of the data object to be added to the collection or Bookmarks portlet. Acceptable values include the metadata types for any object that can be added to a collection portlet. Here are the supported object types (available in a portal search) along with the metadata *dataType* for each:

Object Type Metadata	dataType
Application	Document
Link	Document
Package **	ArchiveFile
Portlet	PSPortlet
Publication Channel	ITChannel
Information Map	Transformation
SAS Report	Transformation
SAS Stored Process	ClassifierMap
Syndication Channel	Document

** A publication package must be stored in SAS metadata in order to be referenced in the page template. The package cannot be stored in WebDAV.

searchStr

Specify an *XMLSelect* string that uniquely locates the data to add to the collection or Bookmarks portlet. The XMLSelect string is in the form *@Name='name'*, where *'name'* corresponds to the *Name* metadata attribute for the object.

If the data is not found, this program continues executing, ignores this entry, and prints a WARNING to the log.

Edit or Remove a Page That Was Created from a Page Template

You can edit or remove pages that have been created from a page template in the same way that you edit or remove any page. Here are the general rules:

Table 22.5 Rules for Editing or Removing Pages Created From the Page Template

Action	Instructions and Rules
Edit a page	<p>Edit a page by using the Options menu in the SAS Information Delivery Portal. For more information, see the online Help that is provided with the portal.</p> <p>The changes that you make to the page in your portal view have no effect on the pages that have been added to other portal views.</p> <p>All portal users can edit a page that has been added to their portal views.</p> <p>For a summary of the types of edits that you can make to a page, see “Types of Edits That Can Be Made to a Page” on page 301.</p>
Remove a page	<p>If the template page has an attribute of DEFAULT, then you can remove the page (created from the template) from your portal view by using the portal Options menu. Note that the page created from a DEFAULT template is not shared. When you remove such a page from your portal view, you do not affect the pages that have been added to other portal views. All portal users can remove a DEFAULT page that has been added to their portal views.</p> <p>If the template page has an attribute of PERSISTENT, then you cannot remove the page created by this template from your portal view unless you first delete the page template from SAS metadata. For more information, see “Delete a Page Template from the Portal” on page 310.</p>

Edit a Page Template

After you have created a page template, you cannot edit it. However, you can delete the page template, and create a new page template that contains the revised content. Note that when you delete a template page, users lose any changes they made to the page created by the page template.

Depending on the method that you use to delete a page template, you can choose also to delete any pages that were created from the template. This feature is especially useful during the implementation phase for a new portal. After you create a page template, if you decide that it needs changes, you can delete the page template along with all pages that might have been created in other users’ portal views. You can now create a new page template that contains the revised content.

For more information about deleting a page template, see “Delete a Page Template from the Portal” on page 310.

Delete a Page Template from the Portal

You might need to delete a page template if you decide that an existing page template is obsolete or that it needs to be modified. If a page template needs to be modified, you can delete the existing page template, and create a new page template that contains different content. When you delete a page template, you permanently remove the page template from SAS metadata and from the portal environment.

There are two ways to delete a page template:

- Delete the page template in the portal.

- Delete a page template by running a SAS program.

When you delete a page template in the portal, the portal removes the page template's metadata from the metadata repository.

Here is a summary of the steps that are required to delete a page template. For complete instructions, see the online Help that is provided with the portal:

- 1 Log on to the portal as the group content administrator for the group with which the page template is shared.
- 2 In the portal, search for and delete the page template.
- 3 When you delete the page template, you can specify whether you also want to delete all of the pages that were created from the template.

When you delete a page template by running a SAS program, you do not remove any pages that were created from the template. After the page template has been removed, any pages that were created from the template remain in portal views, but are no longer associated with a template. If you create a new template, the new template does not affect any pages that were created based on the previous template. As a result, users will see a new page along with the original page in their portal views. Users can review the changes that appear in the new page, remove the original page from their portal views, and re-personalize the new page.

To delete a page template by running a SAS program, follow these steps:

- 1 Modify the SAS program **RemovePageTemplate.sas**, which is located in the *SAS-configuration-directory\Lev1\Web\Applications\SASPortal4.2\sasJobs* directory. In the **RemovePageTemplate.sas** file, specify the appropriate variables for the page template that you want to remove.
- 2 After you have modified **RemovePageTemplate.sas**, save your changes and run the program.

Here are descriptions of the variables that are in **RemovePageTemplate.sas**:

options metaserver="host"

Specify the host name of the SAS Metadata Server. Use the value of the *iomsrv.metadatasrv.host* property in the **configuration** file located at *SAS-configuration-directory \Lev1\Utilities* folder. For example:

```
localhost
machine
machine.mycompany.com
```

metaport=port

Specify the port number of the SAS Metadata Server. This value is a number between 0 and 65536. Use the value of the *iomsrv.metadatasrv.port* property in the **configuration** file located at *SAS-configuration-directory \Lev1\Utilities* folder.

metauser="user ID"

Specify the user ID to use to connect to the SAS Metadata Server; this user ID is typically the SAS Trusted User (sastrust) (default, *sastrust@saspw*).

metapass="password"

Specify the password for the *metauser*. Make sure that this file is secure, or delete the file when you are finished. This file contains the password for the SAS Trusted User.

metarepository="repository";

Specify the name of the SAS Metadata Repository in which your portal metadata is stored, followed by a semicolon (;). Use the value of the *oma.repository.foundation.name* property in the **configuration** file (located in the *SAS-configuration-directory\Lev1\Utilities* directory).

```
%let groupName=SAS Group;
```

Specify the group that you want to remove the data from, followed by a semicolon (;).

```
%let pageName=Page Template Name;
```

Specify the name of the page template that you want to remove, followed by a semicolon (;).

```
%let shareType=Sticky | Default;
```

Specify the page type as either is persistent (sticky) or default, followed by a semicolon (;).

```
%let profile=DESKTOP_PROFILE;
```

Do *not* change this value. (This metadata exists in order to allow for future expansion. Currently, only desktop profiles are supported.)

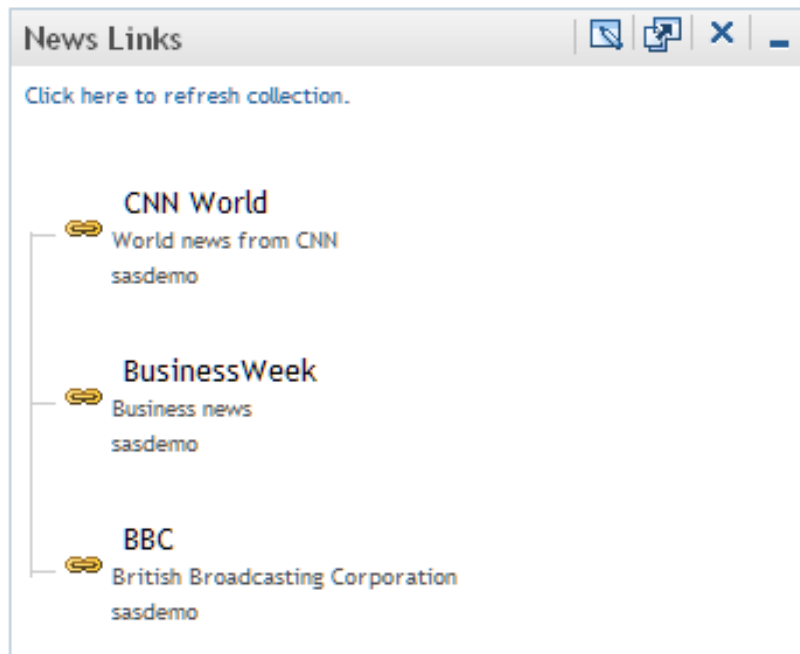
```
%let role=DefaultRole;
```

Do *not* change this value. (This metadata exists in order to allow for future multiple roles. Currently, only the default role is supported.)

Understanding Portlets

Overview of Portlets

Portlets are the rectangular display components of the portal, and are used to organize a portal's contents on a page. Each portlet is surrounded by a border and has a title bar that contains a label and icons. Here is a sample portlet that contains links to Web sites that provide business or world news.



Note: All users can add portlets to the portal. △

For instructions about adding any of the portlets that are described here, see the online Help that is provided with the portal (see the topic “About Portlets” in the Help). The portal supports the following basic types of portlets:

- ☐ custom-developed portlets
- ☐ portlet templates (editable portlets)
- ☐ predefined portlets that are provided with the portal
- ☐ SAS BI Portlets

The following sections describe these different types of portlets.

Custom-Developed Portlets

You can create custom portlets by using the portlet development kit. For more information, see *Developing Portlets for the SAS Information Delivery Portal*.

Portlet Templates (Editable Portlets)

A portlet template enables users to create their own portlet instances. When a user creates a portlet that is based on a portlet template, the user selects the template from a drop-down list. Here are the portlet templates that are provided with the SAS Information Delivery Portal:

Table 22.6 Portlet Templates

Template Name	Description
Alerts Portlet	Available with SAS Shared Services, which is included in the SAS Enterprise Intelligence software bundle.
Information Map Viewer Portlet	Displays information map views that users create by using bookmarks in the Visual Data Explorer component. The Visual Data Explorer has its own bookmark mechanism that is similar to the portal’s bookmarks. This mechanism, called a <i>data exploration</i> , enables users to bookmark one or more views of an information map. When users reopen the information map, they can display the view that was bookmarked. For more information about data explorations, see the online Help that is provided with the SAS Information Delivery Portal (see the topics “About Data Explorations and Bookmarks” and “About Information Map Viewer Portlets” in the Help).
Business Intelligence Dashboard Portlet	Displays one or more graphical indicators. If your installation includes SAS Business Intelligence Dashboard, which is included in the SAS Enterprise Intelligence software bundle, then you can add dashboard portlets to portal pages. An indicator is a composite of one or more related objects. Each indicator has a data source, one or more gauges, hyperlinks to additional information, and range settings for the gauges. You must make the data sources available before users can display dashboard content. See Chapter 24, “Administering SAS BI Dashboard,” on page 363.

Template Name	Description
URL Display Portlet	<p>Displays the Web content at a specific URL. For example, users can configure a URL Display Portlet to display their company's Web site.</p> <p>When users create a URL display portlet, they will typically specify a URL that points to a complete HTML page. The content can be located at any URL that the user is able to access from the browser. For this type of content, users should choose the Show full HTML content in an inline frame option, which is enabled by default. This option displays the URL content in an inline frame (IFRAME) within the portlet.</p> <p>Portlet content that is not displayed in an IFRAME is subject to the portal's security policies. This means that you must add the appropriate URL access permissions in the portal's policy file. Therefore, only administrators should add a URL display portlet that does not display in an IFRAME.</p>
WebDAV Content Portlet	<p>Displays the contents of an HTML fragment that is stored in the portal's WebDAV repository. You must add an HTML fragment to WebDAV before it becomes available for use. An HTML fragment is an HTML file that does not include opening and closing HTML tags, HEAD tags, or BODY tags, and which can be displayed successfully in the cell of an HTML table.</p> <p>When preparing the HTML fragment, you should be aware of the following considerations:</p> <ul style="list-style-type: none"> □ If the HTML fragment contains style definitions with class names that also occur in the portal theme, then the appearance of the portal might be affected when the portlet is displayed. □ If the HTML fragment contains JavaScript, use namespaces for the JavaScript functions to prevent conflicts with portal processing. □ If the HTML fragment contains text in a language that uses multibyte characters (such as Japanese, Korean, Simplified Chinese, or Traditional Chinese), then you must convert the text to UTF-8 in order for the portlet to work correctly. <p>For more information, see the "WebDAV Content Portlets" topic in the online Help.</p>
WebDAV Graph Portlet	<p>Displays a graph that uses data from the portal's WebDAV repository. You must create the data files that are used for the graphs and add those data files to WebDAV. For details, see "Adding WebDAV Graph Portlets" on page 317.</p>
WebDAV Navigator	<p>Enables a user to explore the contents of a WebDAV repository.</p>

When a user adds one of these portlets, the user can view only the WebDAV content that the user is authorized to access.

Predefined Portlets That Are Provided with the SAS Information Delivery Portal

A predefined portlet is automatically deployed when you install the SAS Information Delivery Portal. These portlets often cannot be edited. Predefined portlets also include portlets that you or someone else created previously and that are available for general use. Authorized users can edit those portlets.

Users can search for and add a predefined portlet to their pages in the portal.

The following table shows the predefined portlets that are deployed when you install the portal.

Table 22.7 Predefined Portlets

Portlet Name	Description
Stored Process Alerts Portlet	Displays an electronic notification when a stored process has finished running in the background.
Bookmarks Portlet	Enables users to view and work with content that they have found by using the Search tool and have bookmarked for later use.
Information Maps Navigator Portlet	Enables users to explore information maps in the metadata repository.
Personal Repository Navigator Portlet	Enables users to explore the contents of their personal WebDAV repositories.
Publication Subscription Portlet	Lists all of the publication channels that the user subscribes to and enables users to view content that has been published to them.
Reports Navigator Portlet	Enables users to explore reports in the metadata repository.
Results Navigator Portlet	Enables users to explore stored process results in the WebDAV repository.
Stored Process Navigator Portlet	Enables users to explore stored processes in the metadata repository.
Tree Navigator Portlet	Enables users to explore all content items (that they are authorized to access) in the metadata repository.
Welcome Portlet	Displays localized text by using the user's locale (language and country) preference. The Welcome Portlet is not interactive.

SAS BI Portlets

SAS BI Portlets are available in the October 2009 Release. For a detailed description of the SAS BI Portlets, see “Introduction to SAS BI Portlets” on page 353. The following table shows the predefined SAS BI portlets that are deployed when you install the SAS BI Portlets.

Table 22.8 SAS BI Portlets

SAS BI Portlet Name	Description
SAS Collection Portlet	Enables users to create a list of heterogeneous SAS content items that can be accessed by a launching a content viewer.
SAS Navigator Portlet	Enables users to navigate repository folders in the metadata server and locate SAS content items such as reports and stored processes.
SAS Report Portlet	Enables users to display SAS reports in static HTML format..
SAS Stored Process Portlet	Enables users to display stored process output.

Main Steps to Add a Portlet

One of the administrative tasks of a group content administrator is to deploy custom information for particular groups of portal users. To help accomplish this goal, you can create and share portlets with groups that are defined in SAS metadata.

To learn about the different types of portlets, see “Understanding Portlets” on page 312.

Here is a summary of the steps that are required to add and share a portlet. For complete instructions, see the online Help that is provided with the SAS Information Delivery Portal:

- 1 In SAS Management Console, verify that a permissions tree folder exists for the group with which you want to share the portlet. If necessary, create a permissions tree folder. See “Managing Portal Permission Trees in Metadata” on page 288.
- 2 Log on to the portal as a group content administrator (in order to share the portlet with the respective group).
- 3 You can create a new portlet and add it to a page, create a new portlet independently of a page, or add an existing portlet to a page.
- 4 Edit the portlet in order to add links, applications, or other content to the portlet.
- 5 If you are creating a URL display portlet that is *not* displayed in an inline frame (IFRAME), you must enable the portal to access the URL. To enable this access, add permissions statements for the portlet to the Java policy file for the portal’s Web application server.

If the URL display portlet accesses a Web site that uses relative URL paths for its graphics, those graphics will not display. To ensure that those graphics display correctly, enable the “Show URL Content Inside an I-Frame” option in inline frame (IFRAME).

The URL specifies the protocol and address of the HTML file to display. The Java permissions that are needed to access the HTML file depend on whether the URL protocol is for a file system or an HTTP server. To add a permissions statement to the policy file, depending on the URL type, do one of the following:

- For the file protocol, add a `java.io.FilePermission` statement that grants access to all of the files that make up the HTML fragment; these files include the HTML file and any resources that it uses (such as images, CSS, and JavaScript). The following permission grants access to the entire C drive and all subdirectories:

```
permission java.io.FilePermission "C:\\-","read";
```

- For the HTTP protocol, add a `java.net.SocketPermission` statement that grants access to the host and port of the machine serving up the HTML fragment. The following permission grants access to the Web server running on host.domain:


```
permission java.net.SocketPermission "host.domain:80",
    "connect, resolve";
```

For more information about policy files, see “Configuring and Deploying Restrictive Policy Files” on page 45. For more information about URL display portlets, see “Understanding Portlets” on page 312. Refer also to the online Help that is provided with the portal.

- 6 Implement authorization for the contents on the portlet. Take any necessary steps to control access to files, reports, or other items that have been added to the portlet. For general information about access control, see “Understanding Portal Authorization” on page 280.
- 7 Make the portlet available in the portal.

Edit the portlet in order to share the portlet with a group that is defined in SAS metadata. If the portlet contains applications, links, or syndication channels that you are authorized to share, you can specify whether you also want to share those contents.


When you share the portlet with a group, all members of the group can search for and add the portlet to their pages.

Note: In the portal, users can arrange portlets in columns by using width percentages for the columns. These percentages suggest how the portlets will fit on a page, but are not absolute column widths. Some portlets require a minimum width in order to be displayed, regardless of the percentage that is associated with the portlet’s column. In addition, a portlet’s size can vary based on the content that it contains. If a particular portlet cannot fit within a column, the percentage that you specified for the column will be overridden by the width that is actually required in order to display the portlet. 

Alternatively, you can add the portlet to a page that has been shared or that you intend to share with the group. Depending on the share type, group members will either see the page the next time that they log on, or group members can search for and add the page.

After you have created a portlet, you can edit the portlet, remove the portlet from a page, or delete the portlet permanently from the portal environment. Any changes that you make to a shared portlet are seen by all users who can access the portlet. If you permanently delete a shared portlet, the portlet is removed from all portal views.

For complete instructions about creating, sharing, editing, or deleting a portlet, see the online Help that is provided with the portal. For information about sharing portal content in general, see “Sharing Content in the Portal” on page 283.

Note: All users can create portlets and add portlets to their pages by using the portal **Options** menu. Only users who are authorized as an administrator for a group can share a portlet with the group, or can edit a shared portlet. 

Adding WebDAV Graph Portlets

Overview of Adding WebDAV Graph Portlets

A WebDAV graph portlet creates and displays a line graph, a bar chart, or a pie chart of data that is stored in a WebDAV repository. After you add properly formatted XML

data files to the WebDAV repository, you can create and share WebDAV graph portlets that provide particular views of those data files. All portal users can create their own WebDAV graph portlets.

The following steps explain how to create XML data files and add them to the WebDAV repository, and how to create and share WebDAV graph portlets.

Step 1: Prepare the Data Set That You Want to Graph

Developers in your organization must provide a valid SAS data set to use for the WebDAV graph. The developer should understand your organization's data models and should have SAS programming experience.

Often, you can use a SAS data set without any modifications. Other times, you might want to subset a SAS data set, or perform summary statistics. In general, the simpler the data (more processing that is completed by using the SAS programming language), the easier it is to display a graph in the SAS Information Delivery Portal. For storage and performance reasons, it is preferable to presummarize a large data set (one that has 5000 or more entries). In addition, a summary report is very appropriate for achieving a dashboard effect that enables users to see important information at a glance.

Step 2: Create an XML File and Add It to WebDAV

The SAS Information Delivery Portal provides a macro that creates a properly formatted XML file from your SAS data set, and adds the XML file to the WebDAV repository. The XML file uses the standard SAS Report Model format that is used by other SAS applications, such as SAS Web Report Studio. The macro provides additional formats and labels that the WebDAV graph portlet requires in order to generate a graph.

To create and add the XML file, follow these steps:

- 1 Create a SAS program that invokes the **publishToWebDAV** macro. The macro file is located in the *SAS-configuration-directory\Lev1\Web\Applications\SASPortal4.2\sasJobs* directory.
- 2 In your SAS program, when you invoke **publishToWebDAV**, you must pass a set of arguments that specify the name of the SAS data set, the WebDAV location, and the credentials that are required to write to that location. The **publishToWebDAV** syntax is described after these steps.
- 3 Save and run the SAS program.

If there are any DBCS characters in the parameters (for example, davloc) the **publishToWebDAV** macro must run in a language-specific SAS environment. Another option is to save the SAS program in UTF-8 format, and use the Batch Submit with SAS 9.2 (UTF8) program to run the job.

The **publishToWebDAV** macro creates an XML file named **data.xml** in the WebDAV location that you specified.

Note: If you later update the SAS data set, you must run the macro again to re-create the **data.xml** file in order to see those updates in WebDAV graph portlets. \triangle

Here is the syntax for the **publishToWebDAV** macro:

```
%publishToWebDAV (sasdsn, davloc, userid, passwd)
```

All parameters are required in order to create and publish the XML file. Here are descriptions of the parameters:

<i>sasdsn</i>	Specify the full name of the SAS data set, in the format <i>libref.SAS-data-set</i> , and enclosed in single quotation marks. For example:
---------------	--

	<code>'sashelp.class'</code>
<i>davloc</i>	<p>Specify the URL for the WebDAV location, enclosed in single quotation marks. For example:</p> <pre>'http://<WebDAVHost>:8080/SASContentServer/repository/default/sasdav/Users/sasdemo/graph1'</pre> <p>In this example, the macro creates a graph1 directory under sasdav/Users/sasdemo on the WebDAV server, and creates an XML file named data.xml in that directory.</p> <p>Be sure to specify a unique location for each data set to be published. If the location that you specify already exists on the WebDAV server, you will overwrite the existing XML file.</p> <p>The macro does not create nested directories within the parent directory. If you want nested directories, create the directory structure before running the macro.</p> <p>For example:</p> <pre>'http://<WebDAVHost>:8080/SASContentServer/repository/default/sasdav/graph1/ClassData'</pre> <p>The top level WebDAV directory (sasdav) should already exist. The intermediate directory (graph1) must be manually created before running this macro. The parent directory (ClassData) is automatically created by this macro if the directory does not exist. If it already exists, the parent directory (and its contents) are removed and the directory is re-created.</p>
<i>user ID</i>	<p>Specify the user ID that is used to connect to the WebDAV server, enclosed in single quotation marks. If the WebDAV server runs on a Windows system, the user ID for an external user (for example, a user defined on the host system) should be qualified with either the domain name or the machine name. For example:</p> <pre><machine or Windows domain>\'sasdemo'</pre> <p>The user ID that you specify must be authorized to write to the WebDAV location that is specified for the <i>davloc</i> parameter.</p>
<i>passwd</i>	<p>Specify the password that is used to authenticate the user that you specified in the previous argument, enclosed in single quotation marks. It is recommended that you encrypt the password, but you are not required to do so.</p> <p>Use SAS proprietary 32-bit encryption to encrypt passwords. For example, to encrypt a password of SASDemo1, submit this code in the SAS Program Editor:</p> <pre>proc pwencode in='SASDemo1' method=sasenc; run;</pre> <p>The encrypted password is written to your SAS log. When you use <i>method=sasenc</i>, the first part of the password is {sasenc}.</p> <p>Here is a sample SAS program that includes and invokes the publishToWebDAV macro:</p> <pre>%include 'publishToWebDAV.sas'; %publishToWebDAV('sashelp.class',</pre>

```
'http://localhost:8080/SASContentServer/repository/default/sasdav/graph1/ClassData',
'sasadm@saspw', '{sasenc}E19F24391F7B576C45200E543F0B37B4');
```

Step 3: Create and Share a WebDAV Graph Portlet

After you have successfully completed Step 2, you can create and share a WebDAV graph portlet.

Here is a summary of the steps that are required to create and share a WebDAV graph portlet. For complete instructions, see the online Help that is provided with the SAS Information Delivery Portal:

- 1 Verify that a permissions tree folder exists for the group with which you want to share the portlet. If necessary, create a permissions tree folder. For more information, see “Managing Portal Permission Trees in Metadata” on page 288.
- 2 Log on to the portal as a group content administrator (in order to share the portlet with the respective group).
- 3 Create the WebDAV graph portlet. You can create a new portlet and add it to a page, create a new portlet independently of a page, or add an existing portlet to a page.
- 4 Edit the contents of the portlet in order to specify the location of the XML data file and other properties. You can specify the following properties:
 - ☐ the type of graph or chart that is to be created (line graph, bar chart, or pie chart).
 - ☐ the size of the graph or chart (small, medium, or large).
 - ☐ a title and a description.
 - ☐ the data variables that will be used for the horizontal and the vertical axes. You can also specify a subcategory variable to provide more detail for the horizontal axis.
 - ☐ a statistic that is to be calculated based on the response variable (the variable that is used for the vertical axis). If you do not choose a statistic from the list that is provided, *Sum* is used as the default statistic.
 - ☐ a link to detailed information. If you want the portlet to display a link to other information, such as a SAS report or a stored process, enter the path for the link. The target file can be stored in SAS metadata or in the portal's WebDAV repository.

Create the path for a link for a metadata-based object (all objects except files and WebDAV packages):

```
METASERVER/+Products/SAS Intelligence Platform/Samples+/+Sample:Hello World
```

The following example illustrates a location in metadata for the file

SAS_Email_Message.html:

WebDAV/Templates/notification/en

Adding the two items results in **WEBDAV/Templates/notification/en/SAS_Email_Message.html**.

Note: To determine the path, you can search for the file in the portal. (To find a WebDAV file, search on the Files category.) The path is displayed in the search results.

If you share the WebDAV graph portlet with a group of users, make sure that the group has permissions to access the target file. △

For complete descriptions of the preceding properties, see the online Help that is provided with the portal.

- 5 Edit the properties of the portlet in order to share the portlet with a group of portal users. When you share the portlet with a group, only members of that group can access the portlet.

Note: In order to view the graph, group members must have read permissions for the XML file that you created on the WebDAV server. 

If you have added XML data files to the WebDAV server, all portal users can create WebDAV graph portlets and add those portlets to their pages by using the portal **Options** menu. The group content administrators can use the portal **Options** menu to create portlets. In either case, only users who are authorized as an administrator for a group can share a portlet with the group or edit a shared portlet.

Understanding Portlet Deployment

Overview of Portlet Deployment

The SAS Information Delivery Portal provides several functions with regard to portlets:


- hot deployment of portlets. After you copy the PAR file into the appropriate portlet deployment directory, the portal automatically deploys the portlets via a hot deployment mechanism that runs when the portal's Web application server starts.
- state management of portlets. The portal manages portlet state and keeps track of the portlet context.
- routing of user requests. The portal routes user requests to the appropriate portlet. These portlets might be local portlets or remote portlets. For details about how local and remote portlets run in the portal, see "How Local and Remote Portlets Execute" on page 322.

For instructions about adding custom-developed portlets to the portal, see "Adding Custom-Developed Portlets" on page 324.

Deploying Portlets

To deploy portlets in the SAS Information Delivery Portal, copy the PAR file to the portlet deployment directory. For instructions, see "Step 2: Deploy the Portlet in the SAS Information Delivery Portal" on page 325. When the Web application server starts, the portal deploys the portlets through the portal's hot deployment mechanism. The portal handles portlets as follows:

- deploying additional portlets: If you add a portlet and its resources to the Web application server while the portal is running, the portal automatically deploys the new portlet into the portal.

Note: The portal makes one attempt to deploy the PAR file. If the hot deployment is not successful, the portal does not attempt to deploy the PAR file again. 

- updating or removing portlets: If you update or remove a portlet and its resources in the Web application server, the portal does not automatically update or remove the portlet from the portal. To update or remove a portlet, you must rebuild the **sas.portal9.2 .ear** file, and stop and restart the Web application server. The

portal checks the portlet deployment directory and updates or removes the appropriate portlets from the portal.

How Local and Remote Portlets Execute

From an administration and performance perspective, it is important to understand how portlets are executed. You can develop and deploy two types of portlets: local portlets and remote portlets. For details, see *Developing Portlets for the SAS Information Delivery Portal*.

A *local portlet* is deployed inside the SAS Information Delivery Portal and executes inside the portal's Web application server. Because a local portlet executes in the portal's Web application server, it consumes the computing resources (for example, CPU, memory, and disk storage) of the server machine on which the portal's Web application server runs. When local portlets are deployed, they might also include resources such as Web pages, style sheets, images, resource bundles, and Java classes that are deployed inside the portal.

A *remote portlet* might not execute within the same Web application server and Web application as the portal. Remote portlets enable data from external applications to be incorporated into a Web application. Therefore, a remote portlet might consume computing resources (for example, CPU, memory, and disk storage) on a different machine than the server machine on which the portal's Web application server runs.

For details about the steps to develop a remote portlet, and a detailed sample, see "Creating a Remote Portlet" and "Sample: Remote Portlet (HelloUserRemotePortlet)" in *Developing Portlets for the SAS Information Delivery Portal*.

From a user's perspective, the local portlet and remote portlet look the same. When a user interacts with a remote portlet, the remote portlet looks like a local portlet.

Hiding Portlets from Users

Overview of Hiding Portlets from Users

After you have created and deployed a portlet, you can make that portlet unavailable to users temporarily while you continue development or make changes to the portlet. You can make a portlet unavailable in order to ensure that users do not access the portlet while you are working on it.

The simplest way to make a portlet unavailable to users is to apply metadata authorization controls in order to hide the portlet. To hide a portlet, use SAS Management Console to deny users ReadMetadata permission on the portlet's template. This method effectively hides all instances of the portlet that might have been added to users' portal views. This method is useful for hiding two types of portlets:

- your organization's custom portlets
- portlets that are created from the portal's built-in portlet templates

For information about portlet templates, see "Understanding Portlets" on page 312.

Associating the Portlet with a Group

It is recommended that you associate the portlet with a group before denying access to the portlet. It is much easier to set permissions for a group than it is to set permissions for a large number of individual users.

Here are some key points related to working with groups:

- In the portlet's XML file, associate the portlet with a group. The group can be PUBLIC or any user-defined group. For example, you might associate the portlet with a group called Sales. For more information about the XML file, see "Creating a Deployment Descriptor" in *Developing Portlets for the SAS Information Delivery Portal*.

After you make changes to the portlet's XML file, you must redeploy the portlet before those changes can take effect. See "Understanding Portlet Deployment" on page 321.

- Make sure that the group exists in SAS metadata. Create the group if necessary, and add users to the group.
- For routine maintenance, such as adding content to the portlet, create a group content administrator for the group. For instructions, see "Configure a Group Content Administrator" on page 282.

Hide a Portlet

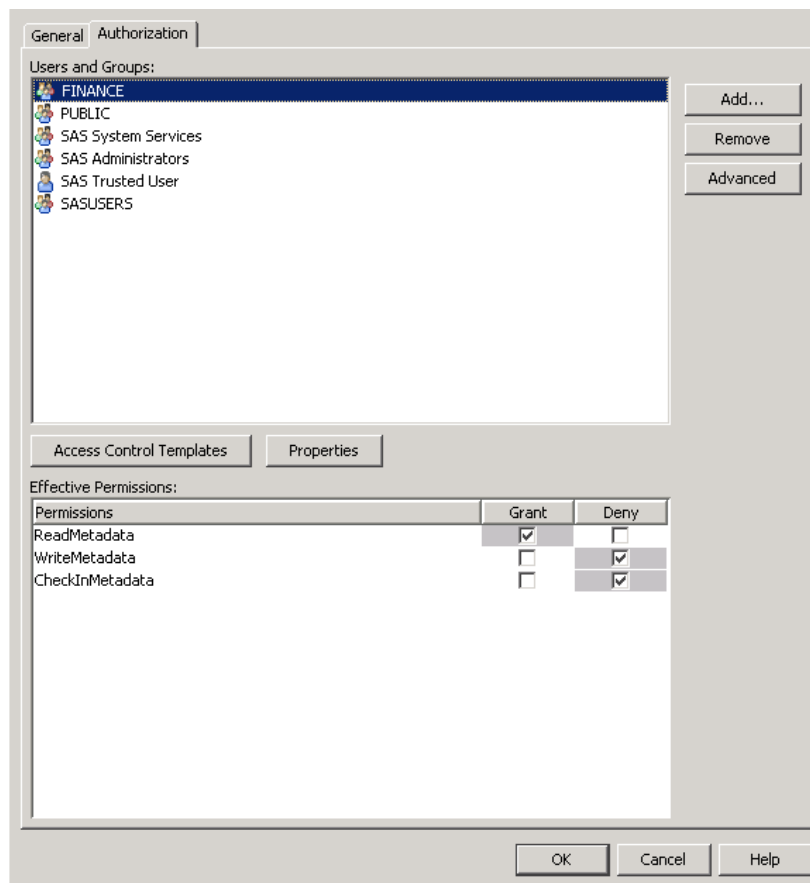
To deny ReadMetadata permissions for a portlet template, follow these steps:

- 1 Log on to SAS Management Console as the SAS Administrator.
- 2 Navigate to the portlet template in the following location in metadata:
Environment Management ► Authorization Manager ► Resource Management ► By Type ► Prototype
- 3 Select the template for the portlet that you want to hide.
- 4 From the main menu, select **File ► Properties**.
- 5 In the Properties dialog box, select the **Authorization** tab.
- 6 In the **Names** list box, select the group or users who will be denied access. If a particular user or group is not listed, click **Add** and add the user or group. When you return to the **Authorization** tab, make sure that the appropriate user or group is selected in the **Users and Groups** list box.
- 7 To modify the permissions for the selected user or group, in the permissions list row for the ReadMetadata permission, select the **Deny** check box.
- 8 When you are finished, click **OK**.

When members of the group log on to the portal, they will not see any portlet that was generated from the portlet template. All instances of the portlet will be hidden from any user or group that is denied ReadMetadata permission on the portlet's template.

To enable the portlet again when you are finished with development, repeat these steps, but grant ReadMetadata permission to the group or users.

The following display of the **Authorization** tab shows the users and groups who have permissions for a portlet template.



Adding Custom-Developed Portlets

Overview of Adding Custom-Developed Portlets

A portlet is a rectangular display component of the SAS Information Delivery portal in which content and links to content are displayed. You can develop a custom portlet and add it to the portal. Your custom-developed portlet can be either a local portlet (contained in a PAR file) or a remote portlet that is contained in a PAR file and a Web application archive (WAR) file.

To design and develop a custom local or remote portlet for deployment in the portal, you should have a working knowledge of JavaServer Pages (JSPs), Java servlets, and the Java programming language. Once you have developed a portlet, you can add it to the portal.

Developing Portlets for the SAS Information Delivery Portal provides guidelines for developing custom portlets.

The following sections describe the steps for developing a custom portlet and adding it to the portal.

Step 1: Design and Code the Portlet

For details about designing and coding the portlet, see *Developing Portlets for the SAS Information Delivery Portal*.

Step 2: Deploy the Portlet in the SAS Information Delivery Portal

To deploy a portlet to the SAS Information Delivery Portal, follow these steps:

- 1 Move or copy the portlet's PAR files to the portlet deployment directory located on the machine where the portal application resides.

If a custom portlet was developed by following the instructions provided in *Developing Portlets for the SAS Information Delivery Portal*, and the portal resides on the same machine where the portlet is located, there is no need to manually copy the PAR files.

The location of the PAR files is defined by the `webapp.sasportlets.deployed.dir` property in the **configuration** file located at `SAS-configuration-directory\Lev1\Utilities` folder. The portal automatically adds the portlet to the SAS Metadata Repository. For example, on a Windows system you might copy **portlet.par** to the `SAS-configuration-directory\Lev1\Web\Applications\SASPortlets4.2\Deployed` directory.

- 2 With the SAS Deployment Manager, rebuild the **sas.portal4.2.ear** file. For instructions, see "Rebuild One or More Web Applications" on page 95.
- 3 Redeploy the **sas.portal4.2.ear** file from the `C:\SAS\Config\Lev1\Web\Staging` directory to the `Web-application-server\server\SASServer1\deploy_sas` directory.
- 4 Stop and restart the Web application server.

If the portlet is a remote portlet, you must also do the following:

- 1 Deploy the associated WAR file in the Web application server.
- 2 Add permissions to the policy files of the SAS Services Application. For details, see "Access Permissions for Custom Portlets and Web Applications" on page 50.

For details about PAR and WAR files, see *Developing Portlets for the SAS Information Delivery Portal*.

Step 3: Ensure That the Appropriate Resource Metadata Is Added to the SAS Metadata Repository

If your data resources have already been defined in the metadata repository, you can skip this step.

To enable the portlet to leverage the SAS Information Delivery Portal's content and security features, you must ensure that the appropriate metadata for each resource has been added to the portal's SAS Metadata Repository. Resources might include SAS Stored Processes, SAS Information Maps, SAS packages, SAS publication channels, and SAS Reports. The SAS servers, spawners, and logins associated with the resources must also be defined.

For an overview of metadata requirements for content, see Chapter 22, "Adding Content to the Portal," on page 291).

In addition, when you add SAS publication channels, syndication channels, and servers, you must enable your portlet to access the content by specifying the appropriate permissions in your Web application server's policy file.

Note: Although the metadata for portlet data resources must be added to the SAS Metadata Repository, it is not necessary for these data resources to be surfaced in portlets. △

Step 4: For Remote Portlets Only, Add the Permission Statements for the Portlet to the Required Policy Files

Add the remote portlet's codebase and permissions, and any additional permissions for the SAS Information Delivery Portal and SAS Services Application codebases to the required policy file. For details, see "Access Permissions for Custom Portlets and Web Applications" on page 50.

Step 5: Implement Authorization for the Portlet

When a portlet is developed, authorization metadata that specifies whether all users or specific groups can access the portlet can be included in the descriptor file for the portlet. In order to enable the sharing of the portlet, the descriptor file can also contain an attribute that enables authorized portal users to create new instances of the portlet.

For information about using the portlet deployment descriptor file to specify whether all users or specific groups are authorized to access the portlet, see "Creating a Deployment Descriptor" in *Developing Portlets for the SAS Information Delivery Portal*.

Note: When a portlet is deployed within the scope of users, all users can use the portlet. If the same portlet is deployed within the scope of a group, only the specified group can use it. △

Step 6: Add the Portlet to the SAS Information Delivery Portal

To add your custom-developed portlet to the portal, see "Main Steps to Add a Portlet" on page 316.

Removing Portlet Configurations

If you have portlets that you no longer use, you can remove the configuration of those portlets. To register portlets, the SAS Information Delivery Portal copies the portlet deployment descriptor file, **portlet.xml**, from each portlet archive file (PAR) in the *SAS-Configuration-Directory/Lev1/Web/Applications/SASPortlets4.2/Deployed* directory to the **sas.portal4.2.ear** file.

To remove a proprietary SAS portlet configuration, follow these steps.

- 1 Undeploy the portal from the Web application server where it is running.
- 2 Remove your portlet's PAR file from the *SAS-configuration-directory/Lev1/Web/Applications/SASPortlets4.2/Deployed* directory.

Note: If you cannot remove the PAR file because it is locked, stop the Web application server that is running the portal. During the portlet registration process, the portal loads classes from the PAR files. As a result, the portal has a lock on all PAR files that it deploys. Some Web application servers require the server to be stopped in order to break that lock. △

- 3 Follow the standard unconfiguration practices for your portlet and the Web application server. This task might include undeploying any Web application or other resource that supports your portlet.


- 4 To remove the `portlet.xml` file from the `sas.portal4.2.ear` file, use the **Rebuild Web Applications** option in the SAS Deployment Manager to rebuild the EAR file for the portal. See “Rebuilding the SAS Web Applications” on page 94.
- 5 In step 2, if you stopped the Web application server where the portal was deployed, restart the Web application server now.
- 6 Use the SAS Deployment Manager to redeploy the `sas.portal4.2.ear` file. See “Redeploying the SAS Web Applications” on page 97.

Adding Links


A link is portal content that is addressable using a universal resource locator (URL). You can create links to sites on the Web or on a local intranet, and share the links with groups of users. Users who can access the links can include the links in collection portlets and display them on pages in the portal. You can also add links to portlets that you share with users.

To create and share a link, follow these steps:

- 1 Verify that a permissions tree folder exists for the group with which you want to share the link. If necessary, create a permissions tree folder. See “Managing Portal Permission Trees in Metadata” on page 288.
- 2 Log on to the SAS Information Delivery Portal as a group content administrator (in order to share the link).
- 3 You can create a new link and add it to a portlet, create a new link independently of a portlet, or add an existing link to a portlet. When you create the link, you can share the link with a group that is defined in SAS metadata. If you are adding an existing link, you can edit the link in order to share it.

Note: For instructions about creating and sharing a link, see the online Help that is provided with the portal (see the “Links” section). 

- 4 Implement authorization (access control) for the target content. Take any necessary steps to control access to files, reports, or other items that the link targets. For example, if the link opens to a page that contains a report, you might want to implement authorization on the report. For general information about authorization, see “Understanding Portal Authorization” on page 280.

Note: A link’s URL might target an HTTP document that is independent of the portal or outside of the portal environment. The portal does not secure the physical document for this type of link. However, you can secure the document through Web server security. See the documentation for your Web application server. 

- 5 Make the link available in the portal by sharing the link. For general information about sharing content, see “Sharing Content in the Portal” on page 283.

When you share the link with a group, all members of the group can search for and add the link to their collection portlets. You have other options for making the link appear in the portal, including these options:

- ☐ You can edit a collection portlet in order to add the link to the portlet. You can share the portlet with a group, including the PUBLIC group. Group members can search for and add the portlet to their pages.
- ☐ After adding the link to a collection portlet, you can add the portlet to a page that has been shared or that you intend to share with a group. Depending on the page’s share type, group members will either see the page the next time they log on, or group members can search for and add the page to their portal views.

Note: If you logged on as a portal administrator, you can edit any portlet or page in the portal. If you logged on as a group content administrator, you can edit only portlets and pages that you have created, or portlets and pages that have already been shared with the group for which you are administrator. △

- You can also search for the link and publish it to a SAS publication channel, and add either the SAS publication channel or SAS package to a collection portlet.

After you have created a link, you can edit the link, remove the link from a portlet, or delete the link permanently from the portal environment. Any changes that you make to a shared link are seen by all users who can access the link. If you permanently delete a shared link, the link is removed from all portal views.

For complete instructions about creating, sharing, editing, or deleting a link, see the online Help that is provided with the portal. For general information about sharing portal content, see “Sharing Content in the Portal” on page 283.

Note: All portal users can create and add links to their collection portlets. Only users who are authorized as a content administrator for a group can share a link with the group, or can edit a shared link. △

Adding Files

Overview of Adding Files

A file can be a file of any type. You must add files to the SAS Content Server repository in order to make them available in the SAS Information Delivery Portal. Files enable portal users (who have access) to view a variety of document types in the portal. When a user selects a file for viewing, the browser displays it using the appropriate software based on the MIME type that is assigned to the file.

The following sections describe the steps to add files to the portal.

Step 1: Add the File to the SAS Content Server

Each user can view only the content that is located under the SAS base path root (`/sasdav`) and that has the appropriate access control configured (Read permission for the user).

You can add the file to a personal folder or to a group folder, as follows:

- To add the file to the SAS Content Server repository for a user’s personal access, follow these steps:
 - 1 Log on to SAS Information Delivery Portal as the user in order to create the user’s personal repository directory on the SAS Content Server.
 - 2 Log on to the SAS Web Administration Console tool to access the SAS Content Server, and locate the appropriate user folder for the content.
 - 3 Use a WebDAV client tool, such as Microsoft Web Folders, to add content to the appropriate folder. You can also use the SAS DAVTree utility. See “Using the DAVTree Utility to Manage WebDAV Content” on page 79.
- To add the file to the WebDAV repository for group access, follow these steps:
 - 1 Determine which SAS groups will access the content.

- 2 Use the SAS Web Administration Console to access the SAS Content Server, and to create the appropriate group folders for the content, and to assign the correct access permissions.
- 3 Use a WebDAV client tool, such as Microsoft Web Folders, to add content to the appropriate folders. You can also use the SAS DAVTree utility. See “Using the DAVTree Utility to Manage WebDAV Content” on page 79.

Step 2: Implement Authorization (Access Control) for the File

Take any necessary steps to control access to the file by setting the appropriate permissions in the SAS Content Server.

Step 3: Make the File Available to Portal Users

Depending on who has access permission to the file, either you or a group content administrator can edit a portlet to add a file to a collection portlet on a page. You can share the portlet or the page with a group of users.

All users can also add a file to a collection portlet on a page in their personal portal view. Users might need to search for the file first. DAV files can be viewed with navigator portlets.

Users can also publish the file to a SAS Publication Channel and either add the SAS Publication Channel or SAS Package to a collection portlet, or add the Publication Channel Subscriptions Portlet to their portal.

For more information, see the portal’s online Help.

Adding Custom Web Applications

Overview of Adding Web Applications Enabled for SAS

Custom SAS-enabled Web applications can be accessed both from within the portal and from outside the portal. For examples that show how to add SAS Web applications to the portal, see “Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java” on page 333.

The following sections describe the steps for adding Web applications to the portal.

Step 1: Design and Code the Web Application

Developers in your organization can design and code Web applications for some or all users. This step is not required for SAS Web applications, such as SAS Web Report Studio and SAS Web OLAP Viewer for Java, which have been developed to support single sign-on with all supported authentication providers. For more information about single sign-on, see *SAS Intelligence Platform: Security Administration Guide*.

Note: If your custom application sends data to the portal, then the application must set its encoding to UTF-8 when it sends the data. △

Step 2: Ensure That the Appropriate User or Group Permissions Tree Is Created in the SAS Metadata Repository

If you are a group content administrator for specific groups, you can share a Web application with that user group (which is defined in SAS metadata). You can share pages with either of the following groups:

- a PUBLIC group, which contains all portal users. It is convenient to share Web applications with the PUBLIC group because all users in that group, by default, have access to the Web applications.
- A specific group that you define such as “Portal Web Report Studio Users” or “Portal Web OLAP Viewer Users.” In this case, the Web applications are shared only with that specific group that you defined.

For details about defining groups, see “User Administration” in the *SAS Intelligence Platform: Security Administration Guide*.

After you add the Web application’s location metadata to the metadata repository (as described in Step 4), the group will be granted ReadMetadata permission to enable the group members to view the content. Group members can also add the Web application to one of their collection portlets. Only the group that you specify will be able to access the Web application.

If you add the Web application’s location metadata by running a SAS program, you can associate the Web application with a user instead of a group. The user must have a permissions tree in metadata. To create a permissions tree for a user, log on to the portal as that user. When you run the SAS program, the user will be granted ReadMetadata and WriteMetadata permissions to view and edit the content. You can later log on to the portal as the group content administrator in order to share the Web application with a group. (You might need to search for the Web application first.)

Step 3: Add the Web Application’s Metadata to the SAS Metadata Repository

There are two ways to define a Web application’s location in metadata:

- Create the Web application in the SAS Information Delivery Portal. When you create a Web application in the portal, the portal adds the Web application’s location metadata to the metadata repository. Here are the high-level steps:
 - 1 Log on to the portal as the group content administrator for the group with which you want to share the Web application.
 - 2 Either create the application and add it to a collection portlet, or create an application that is independent of any portlet.
 - 3 When you create an application, you can share it with a group that is defined in SAS metadata. For general information about sharing portal content, see “Sharing Content in the Portal” on page 283.

Note: For complete instructions about creating and sharing Web applications, see the online Help that is provided with the portal. △

- Create the Web application by running a SAS program. To edit and run a SAS program that creates the application’s location metadata to the SAS metadata repository, follow these steps:
 - 1 Modify the SAS program **LoadWebApplicationExample.sas**, which is located in the *SAS-configuration-directory\Lev1\Web\Applications\SASPortal4.2\sasJobs*

directory. In the **LoadWebApplicationExample.sas** file, specify the appropriate variables for your Web application.

- 2 After you have modified **LoadWebApplicationExample.sas**, save your changes and run the program.

Here are descriptions of the variables that are in **LoadWebApplicationExample.sas**:

options metaserver="host"

Specify the host name of the SAS Metadata Server. Use the value of `iomsrv.metadatasrv.host` property in the **configuration** file located in the *SAS-configuration-directory\Lev1\Utilities* directory. For example:

```
localhost
machine
machine.mycompany.com
```

metaport=port

Specify the port number of the SAS Metadata Server. This value is a number between 0 and 65536. Use the value of the `iomsrv.metadatasrv.port` property in the **configuration** file located in the *SAS-configuration-directory\Lev1\Utilities*.

metauser="user ID"

Specify the user ID to connect to the SAS Metadata Server; this user ID is typically the SAS trusted user (`sastrust@saspw`).

metapass="password"

Specify the password for the metauser.

%repositoryname=Foundation

Specify the name of the SAS Metadata Repository where your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the `oma.repository.foundation.name` property in the **configuration** file located in the *SAS-configuration-directory\Lev1\Utilities* folder.

%let GroupOrUserName=SAS User or Group

Specify the SAS group or user that you want to add the data to, followed by a semicolon (;). Before you can run this program, the SAS group and user permissions trees must be created in the SAS metadata repository.

%let identityType=IdentityGroup | Person

Specify the type of identity, followed by a semicolon (;). Use `IdentityGroup` for a group and `Person` for an individual.

%let webappName=Web application name;

Specify the name of the Web application that you want to create, followed by a semicolon (;).

%let webappDescription=Web application Description

Specify the description of the Web application that you want to create, followed by a semicolon (;).

%let webappURI=Web application URI;

Specify a valid URL for the Web application, followed by a semicolon (;). For example:

```
%let webappURI=http://<hostName>:<Port>/SASWebReportStudio/logonFromPortalWRS.do;
```

If your Web application has request parameters, you must encode the parameters as HTML markup, add the parameters to the `webappURI` value, and enclose the entire string in single quotation marks ('). For example, a URL that takes the form:

```
http://<hostName>:<Port>/webapp?param1=value1&param2=value2
```

is entered in **LoadWebApplicationExample.sas** as:

```
%let webappURI='http://<hostName>:<Port>/webapp?param1=value1&param2=value2';
```

Note that & is replaced with *&*, and the entire URL string is enclosed in single quotation marks.

Here is an example:

```
options metaserver="localhost"
    metaport=8561
    metauser="sastrust@saspw"
    metapass="Password"
    metarepository="Foundation";

%let groupOrUserName=PUBLIC;
%let identityType=IdentityGroup;
%let webappName=SAS Web Report Studio;
%let webappDescription=The SAS Web Report Studio Web application;
%let webappURI=http://localhost:8080/SASWebReportStudio/
    loginFromPortalWRS.do;
```

Step 4: Add the Permission Statements for the Web Application to the Required Policy Files

Add the Web application's codebase and permissions, and any additional permissions for the SAS Information Delivery Portal and SAS Services codebases to the required policy file. If the Web application uses the SAS Foundation Services API, you must add permissions to the SAS Services Application's policy file.

For more information, see "Access Permissions for Custom Portlets and Web Applications" on page 50.

Step 5: Implement Authorization for the Web Application

If you create the Web application by running a SAS program, access is limited to the user or group that you specify in **LoadWebApplicationExample.sas**.

If you create the Web application in the portal, a group's content administrator can share the application with a group of users. Only the users in that group can access the application.

In addition, if you configured Web authentication, access is controlled via a Web user role in the Web application's configuration XML file.

Note: When you implement authorization, access to content is controlled only from within the portal Web application. Users outside of the portal Web application will be able to use the Web application's URL to access the Web application. △

Step 6: Make the Web Application Available in the Portal

When you share a Web application with a group, the Web application becomes available to members of that group. Members can search for and add the Web application to their collection portlets.

Here are some other options for making your application appear in the SAS Information Delivery Portal:

- You can edit a collection portlet in order to add the Web application to the portlet. You can share the portlet with a group, including the PUBLIC group. If the SAS Information Delivery Portal is installed, group members can search for and add the portlet to their pages.
- After adding the Web application to a portlet, you can add the portlet to a page that has been shared or that you intend to share with a group. Depending on the page's share type, group members will either see the page the next time they log on, or group members can search for and add the page.

If you logged on as a portal administrator, you can edit any portlet or page in the portal. If you logged on as a group content administrator, you can edit only portlets and pages that you have created, or portlets and pages that have already been shared with the group for which you are administrator.

Note: All portal users can create and add Web applications to their collection portlets. Only users who are authorized as a content administrator for a group can share a Web application with the group or edit a shared Web application. △

Step 7: Update or Remove the Web Application

After you have created a Web application, you can edit it, remove it from a portlet, and delete it permanently from metadata. From the SAS Information Delivery portal, you can edit or delete any Web application that exists in metadata.

Any changes that you make to a shared Web application are seen by all users who can access the Web application. If you permanently delete a shared Web application, that Web application is removed from all portal views.

For instructions about editing, removing, or permanently deleting a Web application, see the online Help that is provided with the portal. Deleting a Web application only removes the URL stored by the portal for the Web application. The Web application and any metadata that was created to deploy the Web application are not impacted.

Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java

Overview: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java

This example uses the instructions from “Adding Custom Web Applications” on page 329 in order to illustrate how you might add SAS Web Report Studio, SAS Web OLAP Viewer for Java, or both to the SAS Information Delivery Portal. When you follow the instructions that are provided here, you will implement either or both of these as stand-alone applications that have the following characteristics:

Authorized users can access the SAS Web applications that you add to the portal without an additional prompt for their logon credentials. For more information about single sign-on, see “Using Single Sign-On Among Web Applications” on page 41. Applications that support single sign-on have the following characteristics:

- invoked from the portal, but executed remotely

- support single sign-on
- use Metadata Server authentication
- use the SAS Foundation Services API

Although this section uses SAS Web Report Studio and SAS Web OLAP Viewer for Java for examples, you can choose to add other applications to the portal.

The following sections describe these steps to add SAS Web Report Studio and SAS Web OLAP Viewer for Java to the SAS Information Delivery Portal. If you want to add one or the other application, but not both, follow the instructions only for the application that you want to add.

Step 1: Design and Code the Web Application

You do not need to perform this step when you add SAS Web Report Studio and SAS Web OLAP Viewer for Java. By default, both applications are designed to support single sign-on invocation, and do not require additional code.

Step 2: Deploy the Web Application to the Web Application Server

If you have already installed and configured SAS Web Report Studio and SAS Web OLAP Viewer for Java, their enterprise archive (EAR) files should be deployed into the Web application server.

If you have not installed the applications, you can do so by using the SAS Deployment Wizard. You can also manually configure and deploy the applications. For more information, see the SAS third-party Web site at <http://support.sas.com/resources/thirdpartysupport/v92>.

The EAR files that should be deployed are located in the `SAS-configuration-directory\Lev1\Web\Staging` directory. These are the names of the EAR files:

- `SASWebReportStudio4.2.ear`
- `SASWebOLAPViewer4.2.ear`

Step 3: Ensure that the Appropriate Group Metadata Exists in the SAS Metadata Repository

If you have administrative permissions, you can share a Web application with a user group that is defined in SAS metadata. You can share pages with any of the following groups:

- a PUBLIC group, which contains all portal users. It is convenient to share pages with the PUBLIC group because all users in that group, by default, have access to the Web applications
- Portal Web Report Studio Users
- Portal Web OLAP Viewer Users

For details about defining groups, see “User Administration” in the *SAS Intelligence Platform: Security Administration Guide*.

Step 4: Add the Application’s Metadata to the SAS Metadata Repository

There are two ways to define a Web application in metadata:

- Create the Web application in the SAS Information Delivery portal
- Create the Web application by running a SAS program

Run the program separately for SAS Web Report Studio and SAS Web OLAP Viewer for Java. To run the SAS program, follow these steps:

- 1 Make a backup copy of the SAS program **LoadWebApplicationExample.sas**, which is located in the *SAS-configuration-directory\Lev1\Web\Applications\SASPortal4.2\sasJobs* directory.
- 2 In **LoadWebApplicationExample.sas**, specify the following application-specific variables for either SAS Web Report Studio or SAS Web OLAP Viewer for Java:
 - SAS Web Report Studio example:

```
%let groupOrUserName=PUBLIC;
%let identityType=IdentityGroup
%let webappName=SAS Web Report Studio;
%let webappDescription=The SAS Web Report Studio Web application;
%let webappURI=/SASWebReportStudio/logonFromPortal.do;
```

- SAS Web OLAP Viewer for Java example:

```
%let groupOrUserName=PUBLIC;
%let identityType=IdentityGroup
%let webappName=SAS Web OLAP Viewer;
%let webappDescription=The SAS Web OLAP Viewer for
Java Web application;
%let webappURI=/SASWebOLAPViewer/visualdataexplorerer.do;
```

Notes:

- In this example, **webappURI** is specified as a relative location within your Web application server. For example, if SAS Web Report Studio is installed on JBoss, the line **%let webappURI=/SASWebReportStudio/logonFromPortal.do;** would be resolved to:

```
http://<PortalJBOSHost>:8080/SASWebReportStudio/
logonFromPortal.do
```

If SAS Web Report Studio were installed on a remote system instead, you would specify the location as a fully qualified URL that includes the remote host and port, such as:

```
%let webappURI=http://<remotePortalJBOSHost>:8090/
SASWebReportStudio/logonFromPortal.do
```

- For complete descriptions of the variables in **LoadWebApplicationExample.sas**, see “Adding Custom Web Applications” on page 329.
- 3 Save your changes and run the **LoadWebApplicationExample.sas** program.
 - 4 If applicable, repeat these steps a second time. For example, if you modified and ran **LoadWebApplicationExample.sas** for SAS Web Report Studio, you might modify and run it for SAS Web OLAP Viewer for Java.

Step 5: Implement Authorization (Access Control) for the Web Application

If you have correctly installed and configured the application that you want to add (SAS Web Report Studio or SAS Web OLAP Viewer for Java), no additional steps are required for access control.

Note: When you implement authorization, access to content is controlled only from within the SAS Information Delivery Portal. Users outside of the portal will be able to use the Web application's URL to access the Web application. △

For general information about access control, see “Understanding Portal Authorization” on page 280.

Step 6: Make the Web Application Available in the Portal

When you share a Web application with a group, the Web application becomes available to members of that group. Members can search for and add the Web application to their collection portlets.

You have other options for making your application appear in the SAS Information Delivery Portal:

- You can edit a collection portlet in order to add the Web application to the portlet. You can share the portlet with a group, including the PUBLIC group. Group members can search for and add the portlet to their pages.
- After adding the Web application to a portlet, you can add the portlet to a page that has been shared or that you intend to share with a group.

Although the portal administrator can edit any portlet or page in the portal, this is not a good practice. The group content administrator should edit the portal content. If you logged on as a group content administrator, you can edit only portlets and pages that you have created, or portlets and pages that have already been shared with the group for which you are administrator.

Note: All portal users can create and add Web applications to their collection portlets. Only users who are authorized as a content administrator for a group can share a Web application with the group, or can edit a shared Web application. △

For instructions about adding or sharing portlets and pages, see the online Help that is provided with the portal.

Step 7: Update or Remove the Web Application

After you have created a Web application, you can edit it, remove it from a portlet, and delete it permanently from metadata. You can edit or delete any Web application that exists in metadata (including Web applications that were created by running **LoadWebApplicationExample.sas**).

Any changes that you make to a shared Web application are seen by all users who can access the Web application. If you permanently delete a shared Web application, the Web application is removed from all portal views.

For instructions about editing, removing, or permanently deleting a Web application, see the online Help that is provided with the portal.

Adding Syndication Channels

Overview of Adding Syndication Channels

A syndication channel is a channel that provides syndicated, continuously updated Web content. The SAS Information Delivery portal provides support for the emerging Rich Site Summary (RSS) standard, a lightweight XML format designed for sharing news headlines and other syndicated Web content. By incorporating RSS content into the Web application, you can give users access to high-quality, continually updated news that is relevant to their roles in the organization. The BBC, CNN, and Forbes channels are just a few examples of RSS channels that are available publicly.

RSS documents contain metadata, or summary information, about content that is available on the provider's Web site. Each content item consists of a title, a link, and a brief description. By clicking on a link, the user can display the full text for a content item.

The following sections describe the steps for adding a syndication channel.

Step 1: Add the Syndication Channel's Permission Statement to the Appropriate Policy File

To connect to the site that is syndicating content for the syndication channel, you must add a permission statement to the policy file that grants permission to the SAS Information Delivery Portal to connect to the site.

To add a permission statement to the policy file, add a statement with the following format:

```
permission java.net.SocketPermission "machine.domain:80",
    "connect, resolve";
```

where *machine.domain* is the domain-qualified host on which the XML file for the syndicated content is located.

When the SAS Information Delivery Portal's machine is running IPv6, the *machine.domain:80* host address format might not be valid for the permission statement. In these cases, you must either enable all socket permissions or determine the appropriate host address format to use in the policy file.

For example, if you want to add a syndication channel from *rssnews.acme.com*, make a copy of the example file located in the *SAS-configuration-directory\Lev1\Web\Common\SASServer1\SASPortal4.2\PolicyFileInputs\ears\sas.portal* directory, and add the following statement about *rssnews* and its port number:

```
grant codeBase "file:${sas.deploy.dir}/sas.portal4.2ear/-" {
...
permission java.net.SocketPermission
    "rssnews.acme.com:80", "accept,connect,listen,resolve";
...
};
```

For more information about policy files, see “Configuring and Deploying Restrictive Policy Files” on page 45.

Step 2: Ensure That the Appropriate User or Group Permissions Tree Is Created in the SAS Metadata Repository

If you have administrative permissions, you can share a syndication channel with a user group that is defined in SAS metadata. Before you can define a syndication channel and share it with a group, you must create a permissions tree in SAS metadata for the group. To verify that a permissions tree exists, or to create one, see “Managing Portal Permission Trees in Metadata” on page 288. You can share pages with either of the following groups:

- a PUBLIC group, which contains all portal users. It is convenient to share pages with the PUBLIC group because all users in that group, by default, have access to the syndication channels.
- a specific group that you define such as “Sales Managers.” In this case, the pages are shared with the specific group you defined.

After you add the syndication channel metadata to the metadata repository (as described in Step 3), the group will be granted ReadMetadata permission to enable the group members to view the content. Group members can also add the syndication channel to one of their collection portlets. Only the group that you specify will be able to access the syndication channel.

If you add the syndication channel metadata by running a SAS program, you can associate the syndication channel with a user instead of a group. The user must have a permissions tree in metadata. (To create a permissions tree for a user, log on to the portal as that user.) When you run the SAS program, the user will be granted ReadMetadata and WriteMetadata permissions to view and edit the content. You can later log on to the portal as a group content administrator to share the syndication channel with a group (you might need to search for the syndication channel first).

Step 3: Add the Syndication Channel’s Metadata to the SAS Metadata Repository

There are two ways to define a syndication channel in metadata:

- Create the syndication channel in the SAS Information Delivery portal. When you create a syndication channel in the portal, the portal adds the syndication channel’s metadata to the metadata repository.

Here is a summary of the steps that are required to create a syndication channel in the portal. For complete instructions, see the online Help that is provided with the portal:

- 1 Log on to the portal as the group content administrator for the group with which you want to share the syndication channel.
- 2 Either create a syndication channel and add it to a collection portlet, or create a syndication channel that is independent of any portlet.
- 3 When you create a syndication channel, you can share it with a group that is defined in SAS metadata.

For general information about sharing portal content, see “Sharing Content in the Portal” on page 283.

- Create the syndication channel by running a SAS program. To edit and run a SAS program that creates a syndication channel and adds the channel’s metadata to the SAS metadata repository, follow these steps:

- 1 Modify the SAS program **LoadSyndicationChannelExample.sas**, which is located in the *SAS-configuration-directory\Lev1\Web\Applications\SASPortal4.2\sasJobs* directory. In the **LoadSyndicationChannelExample.sas** file, specify the appropriate variables for your syndication channel.
- 2 After you have modified **LoadSyndicationChannelExample.sas**, save your changes and run the program.

Here are descriptions of the variables that are in **LoadSyndicationChannelExample.sas**:

options metaserver="host"

Specify the host name of the SAS Metadata Server. Use the value of the `iomsrv.metadatasrv.host` property in the **configuration** file located in the *SAS-configuration-directory\Lev1\Utilities* folder. For example:

```
localhost
machine
machine.mycompany.com
```

metaport=port

Specify the port number of the SAS Metadata Server. This value is a number between 0 and 65536. Use the value of the `iomsrv.metadatasrv.port` property in the **configuration** file located in the *SAS-configuration-directory\Lev1\Utilities* folder.

metauser="user ID"

Specify the user ID to use to connect to the SAS Metadata Server. This user ID is typically the `sastrust` user.

metapass="password"

Specify the password for the metauser. Make sure that this file is secure, or delete the file when you are finished. This file contains the password for the SAS Trusted User.

metarepository="repository";

Specify the name of the SAS Metadata Repository in which your portal metadata is stored, followed by a semicolon (;). Use the value of the `oma.repository.foundation.name` property in the **configuration** file located in the *SAS-configuration-directory\Lev1\Utilities* folder.

%let groupOrUserName=SAS User | Group;

Specify the SAS group or user that you want to add the data to, followed by a semicolon (;).

%let channelName=syndication channel name;

Specify the name of the syndication channel that you want to create, followed by a semicolon (;).

%let channelDescription=syndication channel description;

Specify the description of the syndication channel that you want to create, followed by a semicolon (;).

%let channelURI=syndication channel URI;

Specify a valid URL for the syndication channel followed by a semicolon (;). For example:

```
%let channelURI=http://www.sas.com/news/preleases/SASRecentPress.xml;
```

Step 4: Implement Authorization for the Syndication Channel

You implement authorization for a syndication channel as follows:

- If you create the syndication channel by running a SAS program, access is limited to the user or group that you specify in `LoadSyndicationChannelExample.sas`.
- If you create the syndication channel in the portal as a content administrator for a group, you can share the syndication channel with a group of users. Only the users in that group can access the syndication channel.

Step 5: Make the Syndication Channel Available in the Portal

When you share a syndication channel with a group, the syndication channel becomes available to members of that group. Members can search for and add the syndication channel to their collection portlets.

You have other options for making the syndication channel appear in the portal syndication channel:

- You can edit a collection portlet in order to add the syndication channel to the portlet. You can share the portlet with a group, including the PUBLIC group. Group members can search for and add the portlet to their pages.
- After you add the syndication channel to a portlet, you can add the portlet to a page that has been shared or that you intend to share with a group. Depending on the page's share type, group members will either see the page the next time they log on, or group members can search for and add the page.

Note: If you logged on as a portal administrator, you can edit any portlet or page in the portal. If you logged on as a group content administrator, you can edit only portlets and pages that you have created, or portlets and pages that have already been shared with the group for which you are administrator. △

Step 6: Update or Remove the Syndication Channel

After you have created a syndication channel, you can edit it, remove it from a portlet, and delete it permanently from metadata. You can edit or delete any syndication channel that exists in metadata.

Any changes that you make to a shared syndication channel are seen by all users who can access the syndication channel. If you permanently delete a shared syndication channel, the syndication channel is removed from all portal views.

For instructions about editing, removing, or permanently deleting a syndication channel, see the online Help that is provided with the SAS Information Delivery Portal.

Adding SAS Packages

Overview of Adding SAS Packages

A package is a collection of structured and unstructured content that has been published using the Publishing Framework by running a SAS Stored Process on a SAS Workspace Server. Users can view SAS packages from the portal.

Packages are used to deliver the following:

- the content of publication channels, which publish information by using the Publishing Framework. If you publish a package from the SAS Information Delivery Portal, the package might include any of the following archived content types:
 - files
 - links
 - SAS Information Maps
 - SAS Stored Processes
- SAS Stored Process output, which can be published to a WebDAV server or to a SAS publication channel.

Users can view packages from the portal if the packages have been published to a SAS publication channel or to a SAS Content Server repository. In order to view packages within the portal, the SAS Package Viewer is required.

The following sections describe how to add a package to the portal.

Step 1: Publish a Package

Create a package by publishing content to one of the following locations:

- SAS publication channel
- WebDAV repository on a SAS Content Server

A package can be created in several ways:

- You can develop a SAS Stored Process that runs on a SAS Workspace Server and produces packages. These packages can be stored on a SAS Content Server or published to a SAS publication channel. For details, see the *SAS Stored Processes: Developer's Guide*.
- You can use the Publishing Framework plug-in within SAS Management Console to create a package. For details, see the online Help for Publishing Framework.
- Users can use the **Publish** icon in the portal's toolbar for some items, or in the bookmark portlet to publish a package and add the package (content) metadata to the SAS Metadata Repository. For details about using the portal to publish a package, see the portal's online Help. When users publish the package to a SAS publication channel or to a SAS Content Server repository, the Web application adds the metadata for the package to the SAS Metadata Repository or WebDAV repository.
- You can publish a package in SAS Enterprise Guide.

Step 2: Define Authorization for the Package

The authorization for a package is part of the metadata for the SAS publication channel or SAS Content Server repository to which the package is published. Take any necessary steps to control access to files, SAS Information Maps, or other items that have been added to the package.

Step 3: Make the Package Available in the SAS Information Delivery Portal

You view channels or packages from the SAS Information Delivery Portal by using the search capability, the Publication Channel Subscriptions portlet, or the Collection

portlet. The authorization given to users for the channels or to the SAS Content Server repository determines which channels and packages they can view in the portal.

To view packages available to you in the portal:

- Search for packages and publication channels.
- Add a package or publication content type to a collection portlet on a portal page.
- Add a Publication Channel Subscriptions portlet to a portal page. This portlet lists all the channels to which you have subscriptions.
- Add a WebDAV navigator portlet and view the package from a WebDAV navigator portlet.

Note: When users access the SAS Information Delivery Portal, the packages that are published to the SAS publication channel at the top-level folder are displayed. In order for users to display and view the packages that are published to subfolders, they can use either the search option within the portal, or the WebDAV navigator portlet. △

Adding SAS Publication Channels

Overview of Adding SAS Publication Channels and Subscribers

About SAS Publication Channels

The SAS Information Delivery Portal provides access to the publish and subscribe features of SAS. These features enable users and applications to publish information to other authorized users. Channels carry information from the publishers who created it to the subscribers who want it. Information can be published using various delivery methods, including SAS publication channels, message queues, e-mail, and files.

From the portal, users can publish a package that might include any of the following archived content types:

- files
- links
- SAS Information Maps (which are published as a link that will display the SAS Information Map in the Visual Data Explorer of the portal)
- stored processes

A SAS publication channel is a channel created with the SAS Publishing Framework plug-in available in SAS Management Console. Publication channels can be used to provide access to archived content published through SAS Publishing Framework. This feature relies on the SAS Publishing Framework software, which is part of SAS Integration Technologies.

For detailed documentation, see the following:

- For descriptions and instructions related to subscribers, channels, and delivery transports, see the online Help for Publishing Framework plug-in within SAS Management Console.
- For information about implementing the Publishing Framework capabilities in your applications, see “Publishing Framework” in the *SAS Publishing Framework: Developer’s Guide*.

Subscribers

The portal provides an interface through which subscribers can subscribe to SAS publication channels.

A subscriber is a person who has a need for information that is published by the Publishing Framework. Before a user can receive information from a channel, you must create a subscriber profile for that user.

You can create two different types of subscribers by using the Publishing Framework plug-in SAS Management Console:

- A content subscriber is a subscriber who is configured to receive packages. A package is a bundle of one or more information entities such as SAS data sets, SAS catalogs, or almost any other type of digital content
- An event subscriber is a subscriber who is configured to receive events. An event is a well-formed XML document that can be published to an HTTP server, a message queue, or a channel that has event subscribers defined for it. However, the SAS Information Delivery Portal does not use event subscribers.

Users' subscriber profiles contain information about how the information that is published to the channels is to be delivered. A user can choose either e-mail, WebDAV, or the portal as the delivery transport. In order for the channel to display in the Publication Channel Subscription portlet, the user must create a subscriber profile that specifies the portal as the delivery transport.

After subscribing to a channel, users can use the portal's Publication Channel Subscription portlet to view archived content that is published through the channel.

For instructions about how to create a subscriber profile, see the online Help for the portal.

E-Mail Transport Restriction

When a user publishes a package to a channel from within the SAS Information Delivery Portal, the package will not be delivered to channel subscribers who selected the e-mail transport. To deliver to those subscribers, you must publish the package by using SAS Enterprise Guide or CALL routines within a SAS program or SAS stored process. For more information, see the *SAS Publishing Framework: Developer's Guide*.

WebDAV Publication Channel Considerations

If you are setting up a WebDAV publication channel, you must enable users to publish to the SAS Content server. For details, see "Planning for Portal Users and Groups" on page 278.

The following sections describe how to set up a publication channel in the portal. After the publication channel is created, users can publish information to the channel.

Step 1 (Optional): Add Archive Permission Statement for the SAS Publication Channel

To enable the SAS Information Delivery Portal to connect to the file system in order to publish to an archive path, you must add a permission statement to the policy file that grants read and write access to the path.

To add a permission statement to the Web application server's policy file, add a statement with one of the following formats:

- To grant read and write access to a specific directory:

```
permission java.io.FilePermission
    "path", "read,write";
```

- To grant read and write access to all the files in a path:

```
permission java.io.FilePermission
    "path/*", "read,write";
```

- To grant read and write access to all the files and subdirectories (recursively) in a path:

```
permission java.io.FilePermission "path/-", "read,write";
```

For example, if you are running a JBoss application server, to grant read and write access to all the files and subdirectories in the path **/sas/PubSub/**, add the following statement:

```
grant codeBase "file:${java.home}/webapps/Portal/-" {
    ...
    permission java.io.FilePermission
        "/sas/PubSub/-", "read,write";
    ...
};
```

For more information about policy files, see “Configuring and Deploying Restrictive Policy Files” on page 45.

Step 2: Assign Permissions for Folders and Files in the SAS Content Server

When you create channels and assign subscribers in the Publishing Framework plug-in within SAS Management Console, you should select WebDAV for persistent store, choose a base path (**/SASContentServer/repository/default/sasdav**), and specify a relative folder for the base path in the metadata repository. You also need to create a new folder below the **/SASContentServer/repository/default/sasdav** folder on the SAS Content Server or use an existing **sasdav** folder, and assign permissions in SAS Content Server. The name of the relative path specified in the Publishing Framework plug-in must match the folder name created in the SAS Content Server.

Note: Selecting a base path for persistent store in the Publishing Framework plug-in, and assigning a relative path does not automatically create the relative path folder on the SAS Content Server. △

WebDAV content is published only to the **/SASContentServer/repository/default/sasdav** folder, or any folder below the **sasdav** folder in the SAS Content Server. Any search within the SAS Information Delivery Portal does not display the contents of the **/SASContentServer/repository/default/sasfolders** path.

Assume that you are going to specify a relative path called **Publish** in the SAS Publishing Framework plug-in’s properties dialog box. Log on to the SAS Content Server Administration Console, and create the **Publish** folder below the **/SASContentServer/repository/default/sasdav** folder. Select the principal for the folder (for example, **jcr:authenticated**), and enable the following permissions to the **Publish** folder’s subfolders and files: Read, Write, Delete, Inherit Write, and Inherit Delete.

For instructions about how to assign permissions to folders in the SAS Content Server, see “Modify Permissions for WebDAV Folders and Files” on page 124.

Step 3: Create a SAS Publication Channel

To add a channel to the SAS Metadata Repository, log on to SAS Management Console as the SAS Administrator and use the Publishing Framework plug-in to define the channel in the metadata repository. If you are publishing to a channel with WebDAV as the persistent store, the relative path should match the folder created on the SAS Content Server. For detailed instructions about defining channels, see the online Help for the Publishing Framework plug-in.

Step 4: Add Subscribers to the SAS Publication Channel

Use the Publishing Framework plug-in in SAS Management Console or the portal **options** menu to add content subscribers. For instructions, see the online Help for Publishing Framework or the portal.

Step 5: Implement Authorization (Access Control) for the SAS Publication Channel

To enable the publication of content to an archive persisted channel, grant Write and WriteMetadata permissions to SASUSERS on that channel's **Authorization** tab (WriteMetadata permission is required only if a channel has an archive persistent store). To enable all subscribers to add channels or subscribers, grant WriteMemberMetadata permission on the relevant parent folder. For instructions about performing these tasks, see the *SAS 9.2 Security Administration Guide*.

Step 6: Make the SAS Publication Channel Available to Content Subscribers

Depending on who has access permission to the publication channel, content subscribers can use one of several methods to make SAS Publication Channel appear in the SAS Information Delivery Portal. Those methods include the following:

- The group content administrator can edit a collection portlet in order to add the publication channel to the portlet. A portal administrator can share a portlet with any group, including the PUBLIC group. Group content administrators can share the portlet with the group for which they are an administrator. Group members can search for and add the portlet to their pages.
- All subscribers can use the Publication Channel Subscriptions portlet to display the SAS publications channels that they subscribe to. This provides a convenient way to view content published to the channels. Subscribers can add this portlet to multiple pages.

Executing SAS Stored Processes from the SAS Information Delivery Portal

What Is a Stored Process?

A SAS Stored Process is a specialized SAS program that is stored in a central location, and which can be executed from the SAS Information Delivery Portal at your request. Stored processes give portal users the ability to run SAS reports dynamically in order to obtain the most current available data. The benefits of stored processes include centralized code management, increased security, and ad hoc reporting capabilities.

Developers in your organization create stored processes for portal users and assign permissions for running the stored processes. For details about creating stored processes, see the *SAS 9.2 Stored Processes: Developer's Guide*.

How Stored Processes Are Executed from the SAS Information Delivery Portal

The SAS Stored Process Web application is required in order to run stored processes from the SAS Information Delivery Portal. The SAS Stored Process Web application was deployed with the SAS Web Infrastructure Platform when you installed and configured the portal.

All portal users who have the appropriate permissions can run a stored process by clicking its icon or link in the portal. Users might first have to search for a stored process before they can run it. As with other portal objects, users can bookmark a stored process, add a stored process to a collection portlet, and publish it to a Package. If users add a Stored Process Navigator portlet to their portal views, they can explore the stored processes that have been defined in metadata (if they have the appropriate permissions). After running the stored process, users can bookmark the results or e-mail the results to other users.

When a user runs a stored process, by default, an Execution Options form is displayed, enabling the user to filter the output contents and to specify particular options for running the stored process. (Developers can choose not to display this form for stored processes that they create. Developers can also create their own custom input form.)

Stored processes fall into two broad categories that affect how the stored process is executed:

- streaming output: If a stored process was defined to stream output to the viewer, the results of the stored process are displayed in the portal immediately after execution.
- non-streaming output: If a stored process does not stream output to the viewer, the results are packaged for later viewing. Users have several options for viewing the stored process output.

The next section provides more information about non-streaming stored processes.


Characteristics of Non-Streaming Stored Processes

Depending on how the developer defined the stored process, a non-streaming stored process can produce transient package output, permanent package output, or no output (this latter type serves no useful purpose for users, but might provide some utility for administration). For descriptions of these output types, see the *SAS 9.2 Stored Processes: Developer's Guide*.

The SAS Information Delivery portal provides several options for locating and viewing the results of non-streaming stored processes. To accommodate the package format of non-streaming stored processes, the portal depends on additional software

that is not required for streaming stored processes. The following list summarizes the dependencies, options, and behaviors of non-streaming stored processes:

- Users can run the stored process in background mode. Background processing enables users to continue working in the portal while the stored process executes. After a stored process finishes running in the background, the results of the stored process are added to the WebDAV and can be viewed later in the Result Navigator portlet by the user. (Users might need to add the Results Navigator portlet to their portal views before they can see the package.)

Note: Background processing is one of the default options that is available in the Execution Options input form. Your developers can create their own input form, however. The developer is responsible for prompting or setting execution options such as background processing. 

- A stored process that runs in the background generates an alert upon successful execution. The alert is displayed in the Stored Process Alerts portlet. From the Stored Process Alerts portlet, users can track, view, and remove the stored process. For more information, see “About Stored Process Alerts” on page 348.
- Depending on how the stored process was created, permanent package output can be published to the SAS Content Server’s WebDAV repository.
- If the stored process package is a permanent package that is associated with a SAS publication channel, the package output is added automatically to the publication channel. (Users must add the channel to their portal views before they can see the package.) The SAS Information Delivery Portal must be installed in order to access publication channels, or to subscribe to the channels with portal as the delivery transport.


Main Tasks for Administering Stored Processes

The developer who creates a stored process designates the server on which the stored process runs, registers the stored process in metadata, and assigns access permissions for the stored process. Check with the developer to obtain information about the stored process as appropriate. For example, if you plan to share the stored process with portal users, you will want to know which group to share it with. You might also want to know the purpose of the stored process and what type of output it produces.

Here are the main tasks for administering stored processes. Perform the tasks that are appropriate for your environment:

- Share a page that contains a stored process: You can log on to the SAS Information Delivery portal, add the stored process to a collection portlet, and share that page with portal users. You might need to search for the stored process before it becomes available to you.

For instructions about sharing a page or adding items to a portlet, see the online Help that is provided with the portal.

Note: All portal users can add stored processes to collection portlets in their personal portal views. 

- Add a publication channel to the portal: If the stored process publishes a package to a SAS publication channel, you can create the SAS publication channel (if it has not already been created). You can also define subscriber profiles and set up subscriptions for portal users. For instructions about how to create subscriber profiles and set up subscriptions, see the online Help for the portal.

After the publication channel is created, users can log on to the portal and add the channel to their portal views. When users run the stored process, the results are added automatically to the channel. Note that the Publication Channel

Subscriptions portlet is refreshed only when the **Refresh** button is clicked in the portlet, or when you navigate to the page that contains the portlet. The group content administrators can add the channel to a page, and share that page with their respective groups.

- Ensure access to the SAS Content Server WebDAV repository: If the stored process publishes a package to a SAS Content Server WebDAV repository, make sure you have set up the appropriate SAS users and groups to enable users to access the package. For details, see “Planning for Portal Users and Groups” on page 278.
- Provide a Stored Process Alerts portlet to portal users: You might want to provide a Stored Process Alerts portlet to some or all or to all users. The Stored Process Alerts portlet displays an alert for any background stored process when the stored process is executed. By adding the Results Navigator portlet to the portal, users can access the results for the Stored Process directly. To provide a Stored Process Alerts portlet to a group of users, create a page template and define a Stored Process Alerts portlet in the page template. For details about creating a page template, see “Adding, Editing, and Removing Page Templates” on page 304. You can also add stored processes to this same page template by defining a collection portlet in the page template, and defining the stored processes as collection data for that portlet. Or, you can add a Stored Process Navigator portlet to the page template.

Note: All portal users can add Stored Process Alerts, collection, and Stored Process Navigator portlets to their portal views. △

- Share stored process results: You can log on to the portal, run a stored process, and share the results with others by publishing or e-mailing the resultant package. (All portal users who have access permission to the stored process can perform this task.)

Example stored processes are installed automatically if you install the portal. The samples are delivered in the Stored Process Web application through the **sas.storedprocess9.2 ear** file. Samples are available in the *SAS-installation-directory\SASWebInfrastructurePlatform\9.2\Static\wars\sas.storedprocess\input\Products\SAS_Intelligence_Platform\Samples* directory. You can log on to the portal and search for SAS stored processes.

For more information about stored process metadata, see the online Help in SAS Management Console. See also the *SAS Stored Processes: Developer's Guide*.

About Stored Process Alerts

Stored process alerts are used to notify you that certain types of SAS stored processes have finished executing and that the results are ready to view. Stored process alerts are displayed in the Stored Process Alerts portlet on a portal page. All users in your organization with access to the portal can add the Stored Process Alerts portlet to their personal portal views. You can also provide a Stored Process Alerts portlet to a group of users by adding the portlet to a page template that you share with the respective group.

Stored processes that run in the background generate alerts upon execution. After you run a background stored process, you can click the alert message in your Stored Process Alerts portlet to see the results of the stored process. (If the stored process was defined with no output, the alert is not linked.) From the Stored Process Alerts portlet, you can also remove the results of the stored process by deleting the respective alert.

Stored processes that generate alerts require that WebDAV or personal repositories be available.

Some SAS products can generate additional types of alert notifications. The alerts for those products are described in their product documentation.

For instructions about adding a Stored Process Alerts portlet or managing alerts, see the online Help that is provided with the portal.

Adding SAS Information Maps

Overview of Adding SAS Information Maps

SAS Information Maps are user-friendly metadata definitions of physical data sources that enable your business users to query a data warehouse to meet specific business needs. The Information Delivery Portal enables authorized users to search for and view SAS Information Maps that exist in the SAS Metadata Repository. When users view a SAS Information Map in the portal, the portal uses the Visual Data Explorer to display the data associated with the information map. (The Visual Data Explorer is provided with the portal.) Portal users must have Read and ReadMetadata permissions to the information map.

SAS Information Maps that exist in the SAS Metadata Repository have already been created and administered by an information map administrator.

The following sections describe how to add a SAS Information Map to the portal environment.

Step 1: Control Access the SAS Information Map

Determine who is authorized to access the SAS Information Map in order to determine which SAS users or groups will be allowed to view the SAS Information Map from the SAS Information Delivery Portal. Take any necessary steps to implement additional authorization for the page or the portlet that will contain the SAS Information Map. For general information about access control, see “Understanding Portal Authorization” on page 280.

Step 2: Make the SAS Information Map Available in the SAS Information Delivery Portal

Here are some ways to make a SAS Information Map appear in the SAS Information Delivery Portal:

- The group content administrator can edit a collection portlet in order to add an information map to the portlet. The portal administrator can share the portlet with any group, including the PUBLIC group. A group content administrator can share the portlet with the group for which he is an administrator. The group members can search for and add the portlet to their pages.
- After adding the information map to a portlet, the group content administrator can add the portlet to a page that has been shared or that you intend to share with a group. Depending on the page's share type, group members will either see the page the next time that they log on, or group members can search for and add the page.
- Users can use the Information Map Navigator to browse the metadata repository for information maps. The Information Map Viewer portlet enables users to view the Data Exploration bookmarks of the information map that is using VDE.

- Users can use the portal or SAS Publishing Framework to publish an information map to a SAS publication channel. Authorized users can subscribe to the SAS publication channel and add the Publication Channel Subscriptions portlet to their portal. Users can also add the publication channel to a collection portlet.

For details about any of these methods, see the portal's online Help.

Adding SAS Reports

Overview of Adding SAS Reports

If you have installed the appropriate software, you can view SAS Reports in the SAS Information Delivery Portal.

A SAS Report is a visual representation of data models and the results of analysis and summarization of the data from SAS procedural output. A SAS report is stored in the SAS Report Model format. The SAS Information Delivery Portal enables authorized users to search for and view SAS Reports that exist in the SAS Metadata Repository.

When users view a report in the portal, the portal uses either the SAS Web Report Studio interface or the SAS Web Report Viewer to display the report.

Reports that exist in the SAS Metadata Repository have already been created and administered by a report administrator. SAS Web Report Studio enables the report administrator to create reports in the SAS Report model format. SAS updates the metadata repository with the metadata for the report.

The following sections explain how to add a SAS Report to the portal environment.

Step 1: Control Access to the SAS Report

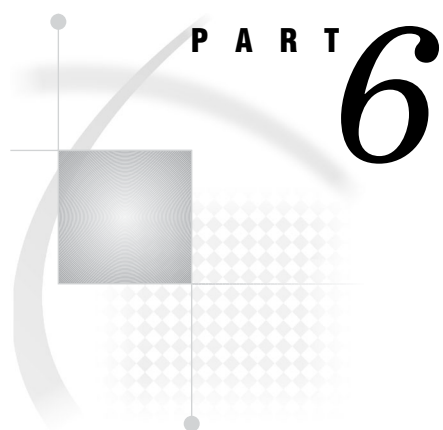
Determine who is authorized to access the SAS report in order to determine which SAS users or groups will be allowed to view the SAS report from the SAS Information Delivery portal. Take any necessary steps to implement additional authorization (access control) for the page (that will contain the portlet with the SAS report). For information about access control, see "Access Management" in the *SAS Intelligence Platform: Security Administration Guide*.

Step 2: Make the SAS Report Available to Portal Users

You can use one of several methods to make the SAS report appear in the SAS Information Delivery portal. These methods include the following:

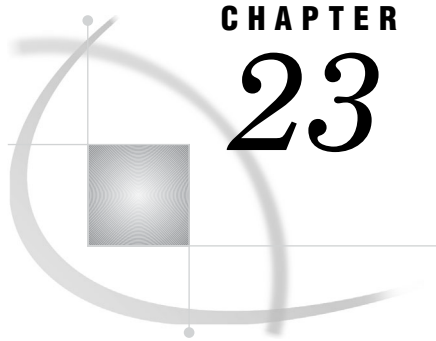
- The portal administrator or a group content administrator can edit a collection portlet in order to add a report to the portlet. The portal administrator can share the portlet with a group, including the PUBLIC group. Group content administrators can share the portlet with the group for which they are an administrator. Group members can search for and add the portlet to their pages.
- After adding the report to a Collection portlet, the portal administrator or a group content administrator can add the portlet to a page that has been shared or that you intend to share with a group. Depending on the page's share type, group members will either see the page the next time that they log on, or group members can search for and add the page.

For more information, see the portal's online Help.



SAS BI Portlets Administration

Chapter 23 **Administering SAS BI Portlets** 353



CHAPTER 23

Administering SAS BI Portlets

<i>Introduction to SAS BI Portlets</i>	353
<i>SAS Collection Portlet</i>	354
<i>SAS Navigator Portlet</i>	354
<i>SAS Report Portlet</i>	354
<i>SAS Stored Process Portlet</i>	354
<i>Using the SAS BI Portlets from a Portal</i>	354
<i>Using SAS BI Portlets with the SAS Information Delivery Portal</i>	354
<i>Using SAS BI Portlets with the WebSphere Portal</i>	354
<i>Configuring SAS BI Portlets for the WebSphere Portal</i>	355
<i>Modify JVM Arguments in the WebSphere Administrative Console</i>	356
<i>Set up the PFS JAAS Configuration in the WebSphere Administrative Console</i>	357
<i>Add the WebSphere Portal Users to the Metadata Server</i>	357
<i>Deploy the SAS BI Portlets in the WebSphere Portal</i>	358
<i>Assign Permissions to the SAS BI Portlets in the WebSphere Portal</i>	358
<i>Add the SAS BI Portlets to the WebSphere Portal Page</i>	359
<i>Removing SAS BI Portlets from the WebSphere Portal Server</i>	359
<i>Delete the SAS BI Portlets from the WebSphere Portal</i>	359
<i>Remove Custom JVM Arguments from the WebSphere Administrative Console</i>	359
<i>Remove the PFS JAAS Configuration in the WebSphere Administrative Console</i>	360

Introduction to SAS BI Portlets

The October 2009 Release provides SAS BI portlets that are based on JSR 168 and available in the SAS Enterprise BI Server offering. These portlets are seamlessly integrated into the SAS Information Delivery Portal that runs on JBoss, WebLogic, or WebSphere Web application servers. 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.

Use of SAS BI Portlets with the WebSphere Portal requires the completion of additional configuration and deployment steps before these portlets can be accessed within the WebSphere Portal. See “Configuring SAS BI Portlets for the WebSphere Portal” on page 355.

The **sas.biportlets4.2.ear** file, which is associated with the SAS BI Portlets, is located in the *SAS-configuration-directory \Lev1\Web\Staging\exploded* directory.

The suite of SAS BI portlets that are compliant with JSR 168 include the following:

- SAS Collection Portlet. See “SAS Collection Portlet” on page 354.
- SAS Navigator Portlet. See “SAS Navigator Portlet” on page 354.
- SAS Report Portlet. See “SAS Report Portlet” on page 354.
- SAS Stored Process Portlet. See “SAS Stored Process Portlet” on page 354.

SAS Collection Portlet

The SAS Collection portlet enables users to create a list of heterogeneous SAS content items that can be accessed by launching a content viewer. The content items are a subset of items supported by the SAS Information Delivery Portal. Portlets, pages, and page templates are not supported. Publication Framework packages are not supported directly. However, users can display packages indirectly by using a channel. A federated search interface enables users to remove or add items.

SAS Navigator Portlet

The SAS Navigator Portlet enables users to navigate repository folders in the metadata server and locate SAS content items such as reports and stored processes. Users can also access WebDAV folders and their contents available on the SAS Content Server. In an edit mode, users can customize the folder location in the tree, and control the types of content items displayed in this portlet.

SAS Report Portlet

The SAS Report Portlet allows users to display SAS reports in static HTML format. Using this portlet, users can drill within SAS Web Report Studio to take advantage of all reporting capabilities such as editing, sorting, and linking reports.

SAS Stored Process Portlet

The SAS Stored Process Portlet enables users to display stored process output. In the edit mode, users can manage parameters that are used when running the stored process.

Using the SAS BI Portlets from a Portal


Using SAS BI Portlets with the SAS Information Delivery Portal

During the installation of the October 2009 Release, SAS BI Portlets are installed, configured, and deployed to your Web application server. Users can access these portlets from within the SAS Information Delivery Portal. For instructions on how to add a portlet to a page in the SAS Information Delivery Portal, see the online Help for the portal.

Using SAS BI Portlets with the WebSphere Portal

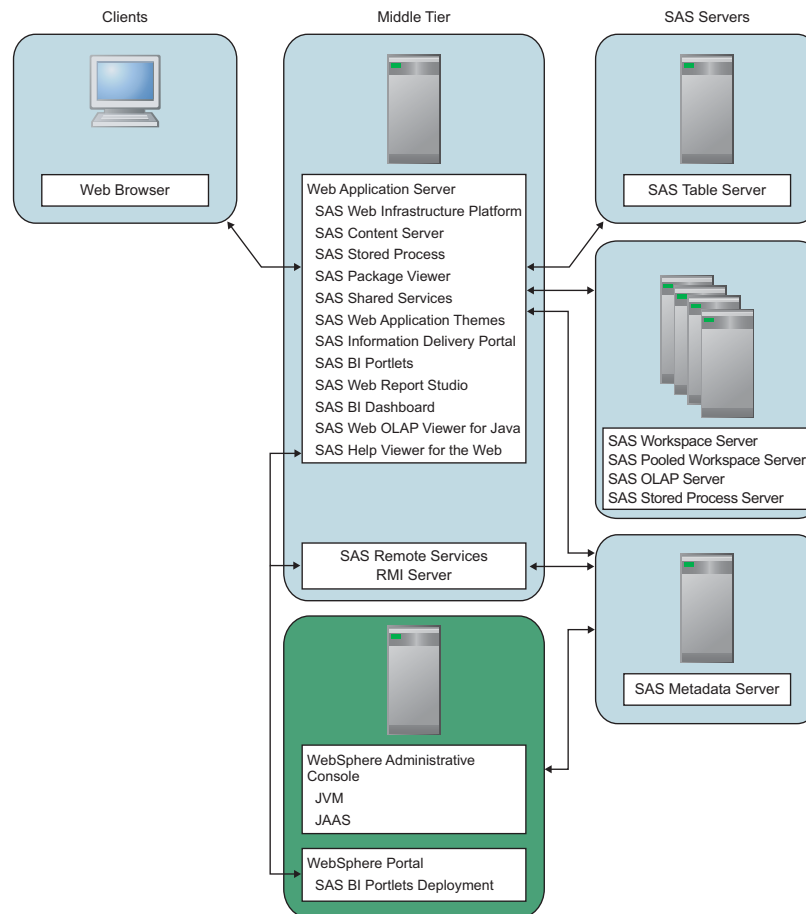
During installation of the October 2009 Release, if you selected the Typical or Custom installation paths in the SAS Deployment Wizard, you are given the choice to select the WebSphere Portal. Then, the Wizard prompts you to confirm or specify the host name, port number, and protocol used with the WebSphere Portal. After a SAS 9.2 installation and availability of WebSphere server, complete the additional configuration

and deployment steps to enable the availability of SAS BI Portlets in the WebSphere Portal. See “Configuring SAS BI Portlets for the WebSphere Portal” on page 355.

Note: SAS BI Portlets use content viewers provided by the SAS Information Delivery Portal. Therefore, SAS Information Delivery Portal must be installed and configured to enable the use of SAS BI Portlets from the WebSphere portal. 

The following figure shows the deployment of SAS BI Portlets in the WebSphere Portal.

Display 23.1 Deployment of SAS BI Portlets in the WebSphere Portal



Configuring SAS BI Portlets for the WebSphere Portal

To configure and deploy the SAS BI portlets to the WebSphere Portal server, follow these steps.

- 1 Modify the JVM arguments in the WebSphere Administrative Console. See Chapter 23, “Administering SAS BI Portlets,” on page 353.
- 2 Set up the PFS JAAS configuration in the WebSphere Administrative Console. See “Set up the PFS JAAS Configuration in the WebSphere Administrative Console” on page 357.

- 3 Create Metadata Accounts for WebSphere Portal Users in SAS Management Console. See “Add the WebSphere Portal Users to the Metadata Server” on page 357.
- 4 Deploy the SAS BI Portlets in the WebSphere Portal. See “Deploy the SAS BI Portlets in the WebSphere Portal” on page 358.
- 5 Assign permissions to SAS BI Portlets in the WebSphere Portal. See “Assign Permissions to the SAS BI Portlets in the WebSphere Portal” on page 358.
- 6 Add the SAS BI Portlets to the WebSphere Portal page. See “Add the SAS BI Portlets to the WebSphere Portal Page” on page 359.

Modify JVM Arguments in the WebSphere Administrative Console

Before you modify the JVM arguments in the WebSphere Administrative Console, access the **environment.properties** file located in the *SAS-configuration-directory\Lev1\Web\Applications\RemoteServices* directory. Make a note of the multicast address and port number specified in that file. You will need that information when specifying the JVM arguments in the WebSphere Administrative Console. For information about multicast IP address and multicast UDP port number, see “Key Multicast Properties” on page 170.

The JVM arguments that you specify in the WebSphere Administrative Console can also be obtained from your **instructions.html** file.

To modify JVM arguments in the WebSphere Administrative Console, follow these steps:

- 1 Start the WAS Administrative Console and log on to the console.
- 2 Expand **Servers ► Application Servers** and select **WebSphere Portal**.
- 3 On the **Configuration** tab, under Server Infrastructure, click on **Java and Process Management**.
- 4 Select **Process Definition**.
- 5 Under Additional Properties, click **Java Virtual Machine**.
- 6 In the **Generic JVM Arguments** field, enter the JVM arguments for the multicast address and port. Make sure that the multicast address matches the value specified in the **environment.properties** file located in the *SAS-configuration-directory/Lev1/Web/Applications/RemoteServices*. To avoid errors when entering the syntax for the JVM arguments, enter the syntax in a text file. Then, paste the typed text into the **Generic JVM Arguments** field. Substitute values for the multicast address and port and save your changes:

```
-Xgcpolicy:gencon -Dcom.sun.management.jmxremote
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.awt.headless=true -Dsas.container.identifier=websphere
-Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true
-Djava.net.preferIPv4Stack=true
-Djava.net.preferIPv6Addresses=false
-Dmulticast_udp_ttl=32
-Dmulticast.address=IP address used for the multicast address
-Dmulticast.port=port number
-Dcom.sas.log.config.url=file:///SAS-configuration-directory
/Lev1/Web/Common/LogConfig/
```

- 7 Save your changes.

Set up the PFS JAAS Configuration in the WebSphere Administrative Console

To set up the PFS JAAS configuration in the WebSphere Administrative Console, follow these steps:

- 1 Start the WAS Administrative Console and log on to the console.
- 2 Expand **Security ► Secure administration, applications, and infrastructure**.
- 3 Click to expand **Java Authentication and Authorization Service**, and click **Application logins**.
- 4 Click **PFS**.
- 5 On the **Configuration** tab, under General Properties, enter **PFS** in the field for **Alias**.
- 6 Under Additional Properties, click **JAAS Login Modules**.
- 7 In the **JAAS login modules** panel, under Preferences, click **New**.
- 8 In the field for **Module class name**, enter **com.sas.services.security.login.TrustedLoginModule**, and click **Apply**.
- 9 In the panel for **com.sas.services.security.login.TrustedLoginModule**, under Additional Properties, click **Custom Properties**.
- 10 Click **New**.
- 11 In the fields for **Name** and **Value**, enter each of the following pairs of values, and save your changes after entering each pair of values:

Name	<i>host name</i>
Value	<i>IP address or host name</i>
Name	port
Value	8561
Name	repository
Value	Foundation
Name	domain
Value	DefaultAuth
Name	trusteduser
Value	sastrust@saspw
Name	<i>user name</i>
Value	<i>password</i>
Name	debug
Value	false
- 12 Restart the WebSphere Portal Server.

Add the WebSphere Portal Users to the Metadata Server

In order to log on to a SAS client, a user must have an account that can provide access to the metadata server. This user account also enables each user to log on to the WebSphere portal to access the SAS BI portlets. This user account should not include a password.

Users of the SAS BI portlets must be defined in the metadata server used by the SAS BI portlets. These users should belong to the appropriate group in the metadata. There are several ways to add users to the metadata server. You can access User Manager plug-in within SAS Management Console and add users. Another method is to batch

import users from a provider such as LDAP into the SAS metadata. The group to which the users are added affects the functions performed by the users.

Deploy the SAS BI Portlets in the WebSphere Portal

To deploy the SAS BI portlets to the WebSphere Portal, follow these steps:

- 1 Log on to the WebSphere Portal Administrative Console.
- 2 On the **Administration** tab, under **Portlet Management**, select **Web Modules**.
- 3 In the **Manage Web Modules** panel, click **Install**.
- 4 Click **Browse**, navigate to the *SAS-configuration-directory\Lev1\Web\Staging* directory, and select **sas.biportlets4.2.war** file.
- 5 Click **Next** and enter the following values:
Enterprise Application display name: SASBIPortlets
Context root: /SASBIPortlets
- 6 Click **Finish**. Then, click **Cancel**.
- 7 Search for the **sas.biportlets4.2.war** file in order to verify that it was installed.

Assign Permissions to the SAS BI Portlets in the WebSphere Portal

To assign permissions to the SAS BI Portlets and enable users to access these portlets, follow these steps:

- 1 Log on to the WebSphere Portal Administration Console.
- 2 On the **Administration** tab, under **Portlet Management**, select **Portlets**.
- 3 In the **Manage Portlets** panel, search for the SAS BI Portlets.
- 4 For each of the SAS BI Portlets, change the resource permissions for the Editor and User:
 - a Click the Key icon to display the Resource Permissions page.
 - b Click the **Edit Role** icon for the Editor.
 - c In the Resource Permissions page for the Editor, click **Add** to display the **Add Role Members** panel.
 - d Select the check box for **Users and User Groups**, and click **OK** to return to the Resource Permissions page for the portlet. Note that when you select the check box for **Users and User Groups**, check boxes get selected for **All Authenticated Portal Users**, **All Portal User Groups**, and **Anonymous Portal User**.
 - e Click on the portlet name to return to the Resource Permissions page for the portlet.
 - f In the Resource Permission page for the portlet, click the Edit Role icon for the User.
 - g Click **Add** to display the **Add Role Members** panel.
 - h Select the check box for **Users and User Groups**, and click **OK** to return to the Resource Permissions page for the portlet.
 - i Click on the portlet name to return to the Resource Permission page for the portlet.
 - j Click **Apply**.
 - k Click **Done**.
- 5 Repeat these steps to assign permissions for the Editor and User associated with each of the SAS BI Portlets.

Add the SAS BI Portlets to the WebSphere Portal Page

To view and use the SAS BI Portlets, add the portlets to a user's Web page in the WebSphere Portal. For instructions about how to add and use portlets, see the online Help for WebSphere Portal.

Removing SAS BI Portlets from the WebSphere Portal Server

If you need to remove the SAS BI Portlets from the WebSphere Portal and WebSphere Web application server, complete these tasks:

- 1 Delete the SAS BI Portlets from the WebSphere Portal. See “Delete the SAS BI Portlets from the WebSphere Portal” on page 359.
- 2 Remove Custom JVM Arguments From the WebSphere Administrative Console. See “Remove Custom JVM Arguments from the WebSphere Administrative Console” on page 359.
- 3 Remove the PFS JAAS Configuration in the WebSphere Administrative Console. See “Remove Custom JVM Arguments from the WebSphere Administrative Console” on page 359.

Delete the SAS BI Portlets from the WebSphere Portal

To delete the SAS BI Portlets from the WebSphere Portal, follow these steps:

- 1 Log on to the WebSphere Portal Administrative Console.
- 2 In the **Administration** tab, under **Portlet Management**, select **Web Modules**.
- 3 Search for **sas.biportlets4.2.war** file to display the filename.
- 4 Click on the Delete Web Module icon for the **sas.biportlets4.2.war** file.
- 5 Click **Yes** in response to the message about the file deletion.
- 6 If necessary, refresh the page by navigating to a different page and returning back to the page to verify that the file has been deleted.

Remove Custom JVM Arguments from the WebSphere Administrative Console

To remove customizations that were applied to WebSphere for the SAS BI Portlets, follow these steps in the WebSphere Administrative Console:

- 1 Start the WebSphere Administrative Console and log on to the console.
- 2 Expand **Servers** ► **Application Servers** and select **WebSphere Portal**.
- 3 On the **Configuration** tab, under Server Infrastructure, click **Java and Process Management**.
- 4 Select **Process Definition**.
- 5 Under Additional Properties, click **Java Virtual Machine**.
- 6 In the **Generic JVM Arguments** field, remove the JVM Arguments that apply to the SAS BI Portlets:

```
-Xgcpolicy:gencon -Dcom.sun.management.jmxremote
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
```

```

-Djava.awt.headless=true -Dsas.container.identifier=websphere
-Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true
-Djava.net.preferIPv4Stack=true
-Djava.net.preferIPv6Addresses=false
-Dmulticast_udp_ttl=32
-Dmulticast.address=IP address used for the multicast address
-Dmulticast.port=port number
-Dcom.sas.log.config.url=file:///SAS-configuration-directory
/Levl/Web/Common/LogConfig/

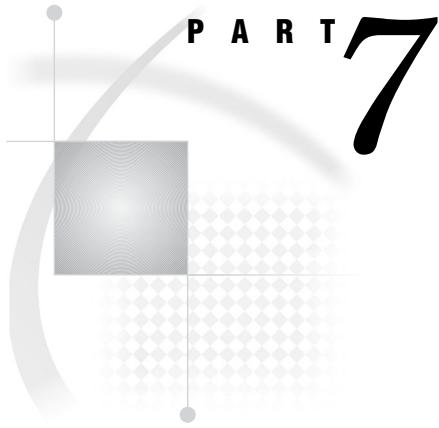
```

- 7 Save your changes.

Remove the PFS JAAS Configuration in the WebSphere Administrative Console

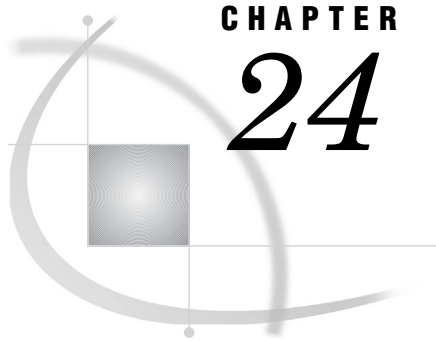
To remove the PFS JAAS configuration in the WebSphere Administrative Console, follow these steps:

- 1 Start the WebSphere Administrative Console and log on to the console.
- 2 Expand **Security ► Secure administration, applications, and infrastructure**.
- 3 Click to expand **Java Authentication and Authorization Service**, and click **Application logins**.
- 4 Select the check box for **PFS** and click **Delete**. The PFS JAAS configuration is deleted.
- 5 Restart the WebSphere Portal Server.



SAS Business Intelligence Dashboard Administration

Chapter 24 **Administering SAS BI Dashboard** 363



CHAPTER 24

Administering SAS BI Dashboard

<i>Overview of SAS BI Dashboard</i>	363
<i>Accessing SAS BI Dashboard</i>	364
<i>Main Tasks for Administering SAS BI Dashboard</i>	364
<i>Setting Up Indicator Alerts for SAS BI Dashboard Users</i>	365
<i>Examine or Configure Settings for the E-Mail Server</i>	365
<i>Configure E-mail Properties for Indicator Alerts</i>	365
<i>Understanding the Data Source XML (DSX) Files</i>	366
<i>Specifying the Location of SAS Data Sets for SAS BI Dashboard</i>	367
<i>Improving the Performance of SAS BI Dashboard</i>	367
<i>Performance Requirements</i>	367
<i>Guidelines for Data Caching</i>	368
<i>Configure a Data Cache</i>	368
<i>Configure Upper Limit for the Data Model Cache Size</i>	370
<i>Configure Cache for Images Used Frequently by SAS BI Dashboard</i>	370
<i>Configure the Pooling of Dashboard JDBC Connections</i>	371
<i>Configuring Alert Latency with Event Generation Framework</i>	372
<i>Seamless Access to SAS BI Dashboard From SAS Information Delivery Portal</i>	374
<i>Enabling the Display of Custom Repository Folders in SAS BI Dashboard</i>	374
<i>Managing Security for SAS BI Dashboard</i>	375
<i>Predefined Administration Role for SAS BI Dashboard</i>	375
<i>Manage Users in SAS BI Dashboard Groups</i>	375
<i>Key Aspects of Security for SAS BI Dashboard</i>	376
<i>Enable the Display of Custom Repository Folders in SAS BI Dashboard</i>	376
<i>Verify or Set Permissions for SAS BI Dashboard Folders</i>	377
<i>Configuration for Dashboard Portlets That Are Shared</i>	377
<i>About Shared Dashboard Portlets</i>	377
<i>Enforce Portlet Security</i>	378
<i>Configure Portlet Security for SAS BI Dashboard Users</i>	378
<i>Removing the SAS BI Dashboard Configuration</i>	379

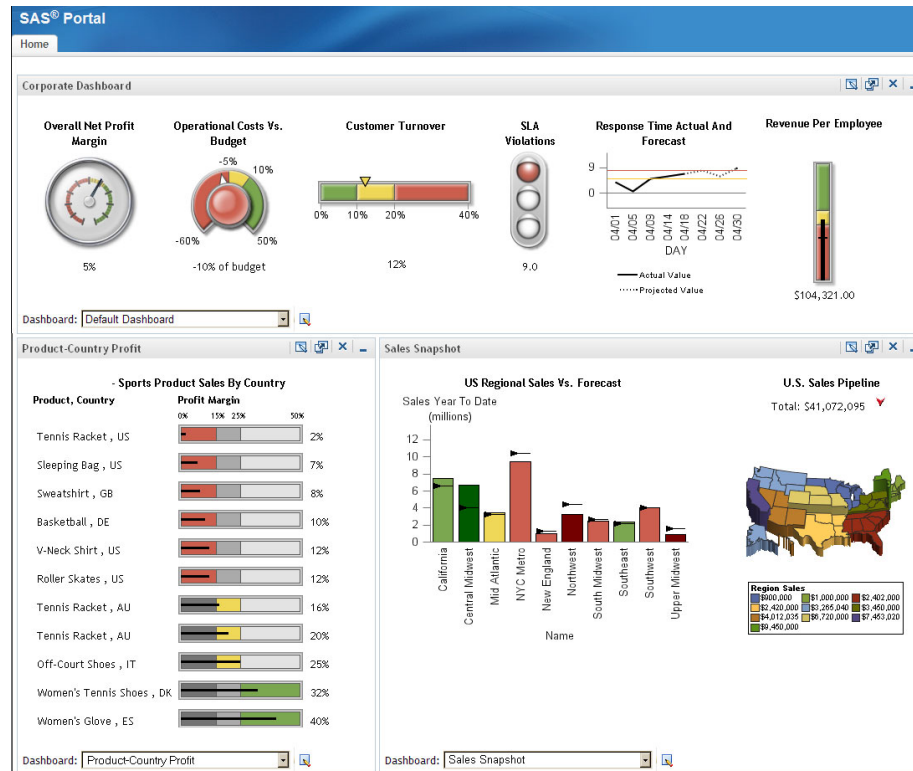
Overview of SAS BI Dashboard

A dashboard is a container that is nested within a portlet and that contains one or more indicators. An indicator is a composite of one or more related objects. Each indicator has a data source, one or more gauges, hyperlinks to additional information, and range settings for the gauges.

Dashboards display critical information in such a way that the information can be interpreted and monitored at a glance. Dashboards can also contain links to other pertinent information, important summary and highlights, and personalized information such as weather, news, and stock news.

Here is an example dashboard that contains several indicators.

Display 24.1 Example Dashboard Portlet



SAS BI Dashboard enables users to create their own dashboards from a variety of data sources including information maps and SAS data sets. Users can link dashboards to SAS business intelligence objects or to external URLs, and users can customize the visualization of the data in a number of ways.

SAS BI Dashboard uses the portal's remote portlet architecture. This means that the SAS Information Delivery Portal hosts a dashboard portlet that interacts with a remote Web application, which in turn handles the functionality. For more information about remote portlets, see *Developing Portlets for the SAS Information Delivery Portal*.

Accessing SAS BI Dashboard

You can access SAS BI Dashboard by logging in directly to the SAS Information Delivery Portal. For information about using SAS BI Dashboard, click **Manage Dashboards** in the dashboard portlet, and then click the **Help** menu that appears in the portal's banner. The **Manage Dashboards** link is available if these conditions are true:

- ☐ the Manage Dashboards link is enabled
- ☐ you belong to the BI Dashboard: Administration role

Main Tasks for Administering SAS BI Dashboard

The administrative tasks that are specific to SAS BI Dashboard are as follows:

- Configure e-mail properties to enable alerts. See “Setting Up Indicator Alerts for SAS BI Dashboard Users” on page 365.
- Configure data source XML files in order to specify the data sources that are created by data designers in your organization. See “Specifying the Location of SAS Data Sets for SAS BI Dashboard” on page 367. (This administrator’s guide does not describe how to create data sources.)
- (Optional) Improve the performance of SAS BI Dashboard. See “Improving the Performance of SAS BI Dashboard” on page 367.
- Implement security in order to manage user access to dashboards objects. See “Managing Security for SAS BI Dashboard” on page 375.

Setting Up Indicator Alerts for SAS BI Dashboard Users

SAS BI Dashboard users and administrators can receive indicator alerts either by e-mail or through the Alerts portlet. When the requirements for generating an alert are fulfilled, these administrators and users receive the alerts. Alerts are generated if the e-mail server is configured, the e-mail properties are configured for each user, and the users exist in the SAS metadata.

For information about how to create indicator alerts and specify the delivery methods (Alerts Portlet or e-mail), see the online Help for SAS BI Dashboard.

Examine or Configure Settings for the E-Mail Server

In order to enable indicator alerts by e-mail, the port and host name for an SMTP e-mail server must be configured. This information is specified in the SAS Deployment Wizard. The SAS Deployment Wizard uses this e-mail server as the default for the SAS Application Server to provide e-mail services to various SAS clients.

You can examine, add, or modify e-mail settings for SAS Application Servers by either of these methods:

- Edit the e-mail settings in the configuration file **sasv9_usermods.cfg** (located in the *SAS-configuration-directory\Lev1\SASApp* directory). See “Managing Workspace Servers and Stored Process Servers” in the *SAS Intelligence Platform: Application Server Administration Guide*.
- In SAS Management Console, navigate to **Application Management ► Configuration Manager**. Right-click on **SharedServices9.2** and select **Properties** to display the Shared Services 9.2 Properties dialog box. If you want to make any changes, specify the property values for **Email.Host** and **Email.Port** on the **Advanced** tab.

Configure E-mail Properties for Indicator Alerts

Before you create indicator alerts by e-mail for SAS BI Dashboard users, you must specify the e-mail type and e-mail address for each user. If the e-mail type and e-mail address are not specified, users will not receive any alerts. To configure the e-mail type and e-mail address, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Environment Management ► User Manager**.

In the window pane on the right side, select a user and right-click to select **Properties**.

- 2 On the **General** tab, click **New**.
- 3 In the Email Properties dialog box, enter the information for **Type** and **Address**.
An example of an entry for type is e-mail, and the address is a valid e-mail address.

Understanding the Data Source XML (DSX) Files

During initial configuration, SAS BI Dashboard created a sample library directory structure on the host machine and created several DSX files. These DSX files provide a central location in which you can specify data sources or optimize performance.

By default, the DSX files reside in *SAS-configuration-directory\Lev1\AppData\SASBIDashboard4.2\dataSourceDefs* directory on the middle-tier machine where SAS BI Dashboard is installed.

The following table summarizes the DSX files:

Table 24.1 Summary of the DSX Files

Filename	Description
dboard_sas.dsx	<p>Defines a directory in metadata containing SAS data sets that can be read by the SAS Business Intelligence Dashboard using SQL queries.</p> <p>The SAS data sets are located in the <i>SAS-configuration-directory\Lev1\AppData\SASBIDashboard4.2\sas-datasets</i>. The SASDATA refers to a pre-defined library that maps to the directory <i>SAS-configuration-directory\Lev1\SASApp\Data</i>.</p> <p>You can modify this file to configure performance settings (caching).</p>
infomap.dsx	<p>Enables the SAS BI Dashboard to use SAS Information Maps. Data modelers can specify information maps in order to read data.</p> <p>The only change you might normally make to this file is to configure performance settings (caching).</p>
omr.dsx	<p>Enables the SAS BI Dashboard to use tables registered in the SAS metadata repository. Data modelers can specify a table and columns in order to read data.</p> <p>The only change you might normally make to this file is to configure performance settings (caching).</p>
spm.dsx	<p>This file is used if SAS Strategic Performance Management has been installed at your site.</p>

Related Tasks:

- To specify a dashboard data source, see “Specifying the Location of SAS Data Sets for SAS BI Dashboard” on page 367.
- To improve dashboard performance, see “Improving the Performance of SAS BI Dashboard” on page 367.

Specifying the Location of SAS Data Sets for SAS BI Dashboard

In order to enable the SAS BI Dashboard to read data from existing SAS data sets, you must specify the location of the SAS data sets in a DSX file.

For a description of the DSX files, including their location, see “Understanding the Data Source XML (DSX) Files” on page 366.

To allow SAS data sets to be read by the SAS BI Dashboard, you can do any of the following:

- Add data sets to the default library that is referenced in **dboard_sas.dsx**. By default, this library is created in *SAS-configuration-directory\Lev1\SASApp\Data* directory.
- To reference a different library, in **dboard_sas.dsx**, change the **<LibRefs>** element so that it specifies the name and location of your library. For example:

```
<LibRefs></LibRefs>
<CommonLibRef>SASDATA</CommonLibRef>
```

You can provide multiple statements, each separated by a semicolon.

- Make a copy of **dboard_sas.dsx** and then reference a different library in the new file.

If you make a copy of the file, follow these steps:

- 1 Change the **ID** attribute of the **<DataSourceDef>** element. The ID attribute must match the name that you give the file, minus the .dsx extension.

For example, if your new file is named **myData.dsx**, then the file contains the following element:

```
<DataSourceDef id="myData"
providerClass="com.sas.bi.dashboard.provider.JdbcProvider">
```

- 2 Add the **<text>** element inside the **<LocalizedText>** element. For example:

```
<LocalizedText id="name">
<text>My Dashboard Library</text>
</LocalizedText>
```

In this example, the text “My Dashboard Library” is assigned to the myData data source.

The name that you specify here appears in the list of available data sources when dashboard developers create or edit data models.

- 3 Change the **<LibRefs>** element so that it specifies the name and location of your library.

You can provide multiple statements, each separated by a semicolon.

- 4 Store the new file in the same directory where **dboard_sas.dsx** resides.

After you make changes to any DSX file, you must restart the BI Dashboard.

Improving the Performance of SAS BI Dashboard

Performance Requirements

Dashboards present the following performance challenges:

- Dashboards typically render data from disparate sources and make multiple queries.
- Some sites need real-time dashboards that obtain their data at the time the dashboard is requested.
- Some sites must serve large numbers of concurrent users.

By default, dashboards obtain their data at the time the dashboard is requested. This default configuration can cause scalability problems and is unnecessary if the underlying data changes infrequently.

To meet different performance requirements, SAS Business Intelligence Dashboard provides two main optimization mechanisms: a data cache and, for JDBC data sources, control over how JDBC connections are made.

For greater performance and scalability, SAS BI Dashboard uses an in-memory Least Recently Used (LRU) cache. SAS BI Dashboard serves data to a data model from a cache, if the underlying data is not stale. If the underlying data is stale, then SAS BI Dashboard queries the underlying data, updates the cache, and returns the fresh data model. In addition, a background thread tries to keep the cache current by updating cached data models before they become stale. If configured properly and allowed enough memory, all data models can achieve 100% cache hit rates and no queries are ever made to the underlying data source during a user's dashboard request. If enough memory is unavailable, then the least recently used data models are dropped from the cache when the cache reaches its memory limit.

Guidelines for Data Caching

For greater performance and scalability, you can enable data caching for SAS BI Dashboard. When data caching is configured, SAS BI Dashboard serves data to a data model from a cache (if the underlying data is not stale), and all queries are run by the SAS Trusted User for all users. If your site has a large number of users with dashboards that retrieve large amounts of data, then increasing the data model cache size might improve performance.

Before enabling data caching, evaluate the following:

- In your environment, is it acceptable to have a single user (in this case, the SAS Trusted User) run the queries for the indicators that you want to cache? If the answer is yes, you can enable data caching.
- Do you enforce BI Row-Level permissions for information maps? If you do, data caching cannot be enabled for SAS BI Dashboard. BI Row-Level permissions require individual users to be able to execute requests to specific locations in a security associations table. SAS BI Dashboard does not allow queries from individual users, because only the SAS Trusted User executes all queries.

Configure a Data Cache

By default, caching is not enabled. To enable caching, you specify caching properties in the DSX file for each data source that you want cached. For a description of the DSX files, including their location, see “Understanding the Data Source XML (DSX) Files” on page 366.

To enable caching for a data source, follow these steps:

- 1 Remove the comment delimiters that surround the `<DefaultTimingCacheDirective>` element in its corresponding DSX file. For example, if you want to enable caching for information maps, then remove the

comment delimiters from **<DefaultTimingCacheDirective>** in the **<infomap.dsx>** file.

- 2 After you make changes to the DSX files, you must restart the Web application server before your changes take effect.

Here are descriptions of the **<DefaultTimingCacheDirective>** attributes:

Table 24.2 Attributes in the **<DefaultTimingCacheDirective>** Element

Attribute	Description
cacheDisplayValueForRefresh	Specifies the minimum amount of time that must elapse before the data source can be refreshed. The dashboard can refresh the data only after a cached data model reaches the age indicated by this number. (You specify the unit of measure in the cacheDisplayMultiplierForRefresh attribute.) Enter the number in quotation marks. You can enter "0" for this value to achieve near real-time data updates. This value is recommended when you have a small number of data sources. With this value, the data is never more than slightly out-of-date, regardless of the stale value. However, the underlying query server (workspace server) might be overloaded if you have a large number of data sources.
cacheDisplayMultiplierForRefresh	Specifies the unit of measure for the value that is entered in cacheDisplayValueForRefresh. Valid values are "SECONDS," "MINUTES," and "HOURS." Enter the value in quotation marks.
cacheDisplayValueForStale	Specifies how much time can pass before the data source becomes invalid or stale. If a cached model is older than this number, then it is invalid and a query will run during the next user's request. (You specify the unit of measure in the cacheDisplayMultiplierForStale attribute). If this value is specified to be higher than the value for cacheDisplayValueForRefresh, the data available for the dashboard will be fresh and the result would be a 100% cache hit rate. Enter the number in quotation marks.
cacheDisplayMultiplierForStale	Specifies the unit of measure for the value that is specified for cacheDisplayValueForRefresh and cacheDisplayValueForStale. Valid values are "SECONDS," "MINUTES," and "HOURS." Enter the value in quotation marks.

Here is example code with sample values for the attributes:

```
<DefaultTimingCacheDirective
  cacheDisplayValueForRefresh=' '5.0' '
```

```

cacheDisplayValueForStale='15.0'
cacheDisplayMultiplierForRefresh='MINUTES'
cacheDisplayMultiplierForStale='MINUTES'

```

In this example, the data is never older than 15 minutes. The background systems are not overloaded because a query executes only once every five minutes. The upper limit for the cache size is approximately 2 MB.

The values you specify for the `cacheDisplayValueForRefresh` and `cacheDisplayValueForStale` attributes depend on how often the data changes, and the extent to which your users need fresh data. If the values for these attributes are significantly large, fewer data retrievals are required to keep the cache updated. Reducing the frequency of the actual retrievals can help alleviate the overhead for the middle-tier servers.

Configure Upper Limit for the Data Model Cache Size

The `bid.maxDataModelCacheSize` property specifies the approximate upper limit of the data model cache size. By default, this property's value is set to 30 MB. When additional data models are cached after this size is met, the least recently used data models are dropped from the cache.

To modify the maximum cache size for the `bid.maxDataModelCacheSize` property, follow these steps:

- 1 Log on to SAS Management Console as the SAS administrator.
- 2 On the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► BI Dashboard 4.2**.
- 3 Right-click and select **Properties** to display the BI Dashboard 4.2 Properties dialog box.
- 4 Click the **Advanced** tab.
- 5 For the property name `bid.maxDataModelCacheSize`, click on the property value, and then enter a value in bytes.
- 6 Click **OK** to exit the SAS BI Dashboard 4.2 Properties dialog box.
- 7 To enable this property to go into effect, restart the Web application server.

Configure Cache for Images Used Frequently by SAS BI Dashboard

The `bid.PdvOutputCacheSize` property specifies the cache for images that are frequently used by the dashboard. Images are cached and reused for frequent display in the dashboard's gauges and indicators. By default, this property's value is set to 10 MB. When additional images are cached after this size is met, the least recently used images are dropped from the cache.

To modify the maximum cache size for the `bid.PdvOutputCacheSize` property, follow these steps:

- 1 Log on to SAS Management Console as the SAS administrator.
- 2 On the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► BI Dashboard 4.2**.
- 3 Right-click and select **Properties** to display the BI Dashboard 4.2 Properties dialog box.
- 4 Select the **Advanced** tab.
- 5 For the property name `bid.PdvOutputCacheSize`, click on the property value, and then enter a value in bytes.
- 6 Click **OK** to exit the SAS BI Dashboard 4.2 Properties dialog box.

- 7 To enable this property to go into effect, restart the Web application server.

Configure the Pooling of Dashboard JDBC Connections

SAS 9.2 supports three types of pooling:

- server-side pooling
- client-side pooling
- JDBC connection pooling for SAS BI Dashboard

Server-side pooling is the process by which the SAS Object Spawner maintains pools of Workspace Servers available for clients. The usage of servers in this pool is governed by the authorization rules set on the servers in the SAS metadata. Client-side pooling increases the efficiency of connections to workspace servers.

By default, SAS BI Dashboard uses server-side pooling. If several other applications and servers are requesting the same workspace servers concurrently, then SAS BI Dashboard and other applications might have to wait until a server in the pool is available. You can increase the number of servers in the pool, if doing so does not impact the memory or CPU resources on the middle-tier machines.

For more information about pooling, see the *SAS Intelligence Platform: Application Server Administration Guide*.

JDBC connection pooling allows SAS BI Dashboard to control its JDBC connections and determine how many workspace servers are spawned. With SAS BI Dashboard, you configure how JDBC connections are opened and managed so that real-time SQL queries can execute quickly without consuming excessive system resources. A single dashboard can have one or more separate indicators that point to different data and execute different SQL queries.

Regardless of whether you use data caching, you can configure pooled JDBC connections in order to improve performance. While caching is a preferable optimization mechanism for scalability, caching might require more memory than desired or it might not meet your data freshness requirements.

By default, BI Dashboard uses JDBC connection pooling. Perform the following procedure only if you want to change the values that are used for pooling. No action is required in order to use pooling with the default values.

To configure pooling for a data source, follow these steps:

- 1 Add pooling attributes to the **<DataSourceDef>** element in the corresponding DSX file. For a description of the DSX files, including their location, see “Understanding the Data Source XML (DSX) Files” on page 366.
- 2 After you make changes to the DSX file, you must restart the Web application server to enable the changes to take effect.

For descriptions of attributes, see the following table.

Table 24.3 Attributes and Descriptions

Attribute	Description
maxPoolSize	Specifies a maximum number of pooled connections. A high setting consumes more system resources, but might be necessary when you expect a large number of users. The default value is 20.
maxWaitForPooledConnection	Specifies the number of milliseconds to wait for a pooled connection before returning an error that the connection failed. The default value is 5000.
lingerTime	Specifies the number of milliseconds to hold a connection open after finishing a query. In dashboards, it is common to execute several queries within a single HTTP request. For that reason, it is important for connections to persist so that multiple connections do not have to be re-established within a single HTTP request. The default value is 300000.
alwaysConnectAsAdminUser	Specifies that clients always connect as the administrator user that is specified in the BIDashboard.config file. When the value is true, this setting results in a smaller number of pool connections because the same connection is used repeatedly. The default value is false.

Here is an example of pooling attributes that were added to the `<DataSourceDef>` element in the DSX file:

```
<DataSourceDef id='dboard_sas'
  providerClass='com.sas.bi.dashboard.provider.JdbcProvider'
  maxPoolSize='5'
  maxWaitForPooledConnection='60000'
  lingerTime='6000'
  alwaysConnectAsAdminUser='true'>
  ...
</DataSourceDef>
```

Configuring Alert Latency with Event Generation Framework

When a SAS BI Dashboard user or modeler sets up an alert, Event Generation Framework regularly polls BI Dashboard indicators, determines whether an event qualifies for an alert, and generates an alert for the user. An alert notification is generated when an indicator meets the criteria configured for the alert.

Alerts are generated via e-mail or they are posted as an event to the Alerts portlet.

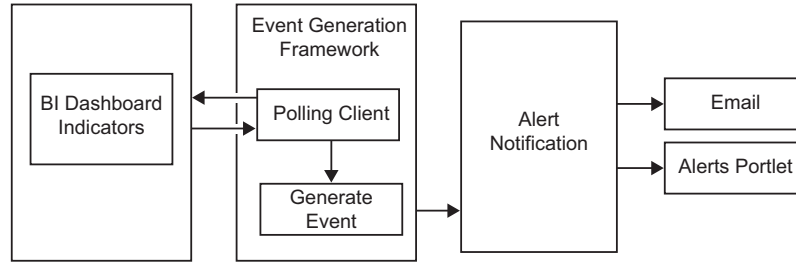
For information about how to configure SAS BI Dashboard indicators, ranges, and threshold for alerts, see the *SAS BI Dashboard 4.2: Users Guide*. Also, for information

about how to select the frequency and mode of output for alert notifications, see *SAS BI Dashboard 4.2: Users Guide*.

Some businesses might receive an alert once a quarter or once a month. Businesses that require operational data can receive alerts as frequently as once a minute or more often.

The following figure shows how the Event Generation Framework interacts with the SAS BI Dashboard indicators to poll for an event, trigger the event, and notify users through an alert notification service.

Figure 24.1 Event Generation Framework



Constant polling can impact the efficiency of SAS BI Dashboard. In order to minimize the impact of constant polling on performance of SAS BI Dashboard, you can customize and set parameters for alert latency. Alert latency is the time lag between when a data change occurs that triggers an alert and the time a user receives that alert.

To ensure optimum efficiency with alert latency, you can customize the time intervals for the `EventSourceCandidateRefHandler` and the `EventCandidateRequestHandler`. The `EventSourceCandidateRefHandler` is responsible for polling SAS BI Dashboard for a list of indicators that need to be queried.

The **heartbeatinterval** parameter determines how often the `EventSourceCandidateRefHandler` polls for the list. When the `EventSourceCandidateRefHandler` receives a list, the system pauses for the number of milliseconds specified by the **heartbeatinterval** parameter before asking for a list again.

The `EventCandidateRequestHandler` is responsible for requesting a small number of indicators at a time. The **throttlepause** parameter determines how often the `EventCandidateRequestHandler` requests the small number of indicators. After a request is completed, the system pauses for the number of milliseconds specified by the **throttlepause** parameter before issuing the next request. The `EventSourceCandidateRefHandler` puts less strain on the SAS BI Dashboard. Therefore, this call can be made more frequently with less performance impact.

The default value for **heartbeatinterval** is 30000 milliseconds. The **throttlepause** parameter is set to a default value of 60000 milliseconds. Depending on your business requirements, you can modify these values. Some businesses might choose to have an aggressive value of 30 seconds or specify a value equal to a few minutes. Other businesses might choose a higher value equal to several hours (for example, three hours or six hours). The values are always specified in milliseconds.

The **BIDashboardIndicatorRequestor** file located in the `SAS-configuration-directory\Lev1\AppData\SASBIDashboardEventGen4.2\candidateRequestHandlerConfigs` directory specifies the values for the **heartbeatinterval** and **throttlepause** parameters. The following example shows the default value for **throttlepause** in the **BIDashboardIndicatorRequestor** file:

```
<?xml version="1.0" encoding="UTF-8"?>
<EventCandidateRequestHandlerConfig id="BIDashboardIndicatorRequest"
```

```

classname="com.mgb.bi.dashboard.egfhandlers.IndicatorRequestHandler"
throttlepause="60000"
serviceEndpointUrl=''/services/IndicatorService"
endpointSoftwareComponentName="BI Dashboard 4.2"
requestorType="mgb.bid.indicator"
defaultEventCandidateType="mgb.bid.indicator"

```

Seamless Access to SAS BI Dashboard From SAS Information Delivery Portal

SAS BI Dashboard can be started from SAS Information Delivery Portal without requiring users to log on to the application. The current logon information from the portal is provided to SAS BI Dashboard. By default, this feature is not enabled. To allow portal users to access SAS BI Dashboard without logging into the application, you must add the **bid.portletShowManageLink** property and set it to **true**.

To enable SAS Information Delivery Portal users to access SAS BI Dashboard without logging into it, follow these steps:

- 1 Log on to SAS Management Console as the SAS administrator.
- 2 On the **Plug-ins** tab, navigate to **Application Management ► Configuration Manager ► BI Dashboard 4.2**.
- 3 Right-click and select **Properties** to display the BI Dashboard 4.2 Properties dialog box.
- 4 Select the **Advanced** tab.
- 5 For the property name **bid.portletShowManageLink**, click on the property value **false**, and then enter **true**.
- 6 Click **OK** to exit the SAS BI Dashboard 4.2 Properties dialog box.
- 7 To enable this property to go into effect, restart the Web application server.

Enabling the Display of Custom Repository Folders in SAS BI Dashboard

If you create custom repository folders, add them to the Foundation Services in SAS Management Console. As a result, you can view and access the custom repository folders within SAS BI Dashboard. For information about different repositories and administrative tasks associated with repositories, see “Creating, Registering, Moving, Copying, Renaming, and Deleting SAS Metadata Repositories” in the *SAS 9.2 Intelligence Platform System Administration Guide*.

To enable the display of custom repository folders in SAS BI Dashboard, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Environment Management ► Foundation Services Manager ► SAS BI Dashboard 4.2 Local Services ► Core ► Information Service**.
- 2 Right-click and select **Properties** to display the Information Properties dialog box.
- 3 Click on the **Service Configuration** tab.
- 4 Click **Configuration**.
- 5 Click on the **Repositories** tab, and select **New**.

- 6 In the New Information Service Repository dialog box, follow the instructions on the wizard pages. As you answer the wizard's prompts, be sure to specify a unique name for the repository and select the check box for **AutoConnect**.
- 7 To enable this property to go into effect, restart the Remote Services and your Web application server.

Managing Security for SAS BI Dashboard

You can use metadata layer permissions to manage access to dashboard objects such as dashboards, indicators, models, and ranges. This topic documents the requirements and describes the predefined groups that you can choose to use to manage access.

Predefined Administration Role for SAS BI Dashboard

SAS BI Dashboard includes a predefined role, BI Dashboard:Administration. In order to manage SAS BI Dashboard, administrators must meet the following criteria:

- Be assigned to the BIDashboard:Administration role. When SAS Deployment Wizard completes installation, the BI Administration role is added to the SAS BI Dashboard administrators group by default. If you create a different group for administrators, the BI Administration role must be added to that group.
- Explicitly have ReadMetadata and WriteMetadata permissions to folders. This is necessary in order for administrators to create, read, modify, and delete objects.

Manage Users in SAS BI Dashboard Groups

By default, two groups are available for SAS BI Dashboard:

- BI Dashboard Users
- BI Dashboard Administrators

You can use dashboard groups to manage access to dashboard objects. You are not required to use the BI Dashboard Users or the BI Dashboard Administrator groups. You can create your own groups that meet your organizational needs. Typically, you grant access to data designers so that they can create the dashboards, indicators, data models, and ranges using the graphical interface. You typically limit access for other users who need only to see dashboards on the portal page. You can manage user access by adding users to the appropriate group and then by assigning permissions to the groups on the BI Dashboard folder.

These default groups determine which dashboard objects users can access and manipulate as follows:

Table 24.4 Predefined Default User Groups and Their Access to Dashboard Objects

Group	Type of Access
BI Dashboard Users	Members of this group can view dashboards in portlets.
BI Dashboard Administrators	Members of this group can view dashboards in portlets and change the dashboard layout. Members also have access to a Manage Dashboards Application either by direct access to the BI Dashboard application or via the link in the portlet (if the value for the portlet ShowManageLink property is true). After they click this link, members can create, edit, and delete dashboard objects.

Key Aspects of Security for SAS BI Dashboard

The following list summarizes some key points that apply to SAS BI Dashboard security.

- For SAS applications including SAS BI Dashboard and the SAS Information Delivery Portal, authentication is through the SAS Logon Manager.
- The ability to render, create, edit, and delete dashboard objects is controlled by the permissions on the objects in the metadata. Users who can render data are able to view dashboards, but they have no ability to create new dashboards.
- In order to create the metadata folders and objects, SAS BI Dashboard uses a metadata account to connect to the metadata server. By default, this account is the SAS Trusted User, which gets explicit permissions.
- When a user wants to view a dashboard, the user's permissions for the dashboard, indicators, ranges, and data models are verified. If a user has permission to view a dashboard, but is not granted permission to read any of the indicators in the dashboard, an empty dashboard is displayed. If a user has permission to read an indicator, but does not have permission to read the data point model, the indicator does not render for that user.
 - If data caching is not enabled, and a user does not have Read permissions on an underlying information map, cube, or data set, then the query fails and an error message is returned. If data caching is enabled, the queries are run by the SAS Trusted User for all users.
 - If an information map uses row-level permissions, then only the data that is readable by a particular user appears in a dashboard indicator when that user is logged on to the portal.
- To ensure performance, object permissions are established at the beginning of the user's session. If a user has Read permission on an indicator at the beginning of the session, that permission applies to the entire session even if the administrator changes the permission in SAS Management Console during the user's session.

Enable the Display of Custom Repository Folders in SAS BI Dashboard

If you create custom repository folders, add them to the Foundation Services in SAS Management Console. As a result, you can view and access the custom repository folders within SAS BI Dashboard. For information about different repositories and administrative tasks associated with repositories, see "Creating, Registering, Moving,

Copying, Renaming, and Deleting SAS Metadata Repositories” in the *SAS 9.2 Intelligence Platform System Administration Guide*.

To enable the display of custom repository folders in SAS BI Dashboard, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Environment Management ► Foundation Services Manager ► SAS BI Dashboard 4.2 Local Services ► Information Service**.
- 2 Right-click and select **Properties** to display the Information Properties dialog box.
- 3 Click on the **Service Configuration** tab.
- 4 Click **Configuration**.
- 5 Click on the **Repositories** tab, and select **New**.
- 6 In the New Information Service Repository dialog box, follow the instructions on the wizard pages. As you answer the wizard’s prompts, be sure to specify a unique name for the repository and select the check box for **AutoConnect**.
- 7 To enable this property to go into effect, restart the Foundation Services and your Web application server.

Verify or Set Permissions for SAS BI Dashboard Folders

Permissions must be set on the BI Dashboard folders in metadata in order to enforce protections on dashboard objects. Some permissions were set during initial configuration. For some groups, such as PUBLIC and SASUSERS, you must set permissions manually.

To verify or set permissions on any of the Dashboards, Data Point Models, Indicator Definitions, or Ranges folders, go to SAS Management Console. On the **Folders** tab, navigate to **\System\Applications\SAS BI Dashboard\BI Dashboard 4.2**. Select the appropriate folder, right-click to select **Properties**, and then verify or assign permissions to the folder.

Permissions	Tasks
Read, ReadMetaData	Render an object or view the properties of an object.
Create (permission for parent folder)	Create an object.
Delete, ReadMetaData	Delete an object.

Configuration for Dashboard Portlets That Are Shared

About Shared Dashboard Portlets

Shared portlets are appropriate for users who need only to view dashboards. These users cannot manipulate portlet content in any way. Like other portlets, dashboard portlets can be shared with a group that is defined in metadata. To share a portlet, you must be a group content administrator or a sastrust user for the respective group. For more information about sharing portlets, see “Sharing Content in the Portal” on page 283.

Normally, when you share a portlet with a group, members of the group have read-only access to the portlet. Dashboard portlets, however, require some additional configuration. In order to enforce read-only access, set the value of the **bid.enforcePortletSecurity** property to true in the BI Dashboard Properties dialog box within SAS Management Console Configuration Manager.

Enforce Portlet Security

The **bid.enforcePortletSecurity** property determines whether users can edit a portlet that is shared. The following table describes the security options for Dashboard portlets.

Table 24.5 Values for the bid.enforcePortletSecurity Property

Property Value	Description
true (default)	This setting, which is the default setting, ensures that users have read-only capability to a portlet that is shared. Users who have access to the shared portlet cannot select which dashboard is displayed in the portlet, cannot edit the dashboard, and cannot edit the portlet. They can, however, personalize the indicators in a dashboard. This setting takes effect even when security is not enabled.
false	If enforcePortletSecurity is set to false, then the portlet layout can be changed by any user. Even when the portlet is shared, users who can access the shared portlet can select which dashboard is displayed in the portlet and can edit the dashboard. Any change they make to the dashboard display is visible to all users who can access the shared portlet. (Users cannot, however, edit the portlet's properties unless they are group content administrators for the group with which the portlet is shared.)

Configure Portlet Security for SAS BI Dashboard Users

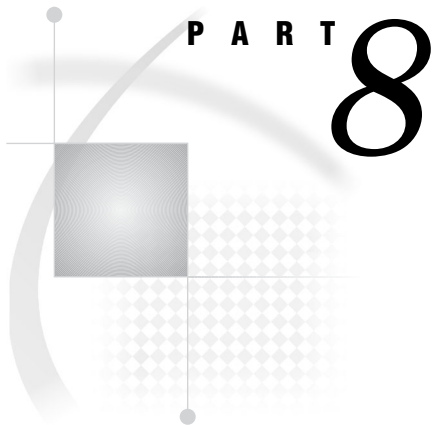
To configure portlet security for SAS BI Dashboard users, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, navigate to **Application Management ► Configuration Manager ► BI Dashboard 4.2**.
- 2 Right-click and select **Properties** to display the BI Dashboard 4.2 Properties dialog box.
- 3 Click on the **Advanced** tab.
- 4 Click **Add** to display the Define New Property dialog box.
- 5 Enter the property name and property value and click **OK** in the Define New Property dialog box:

Property Name: `bid.enforcePortletSecurity`
Property Value: `true`
- 6 Click **OK** to exit the BI Dashboard 4.2 Properties dialog box.
- 7 To enable this property to go into effect, restart your Web application server.

Removing the SAS BI Dashboard Configuration

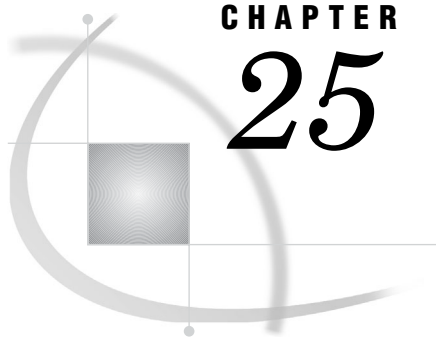
You can remove the SAS BI Dashboard configuration by following the steps for removing a portlet configuration. For information about removing a portlet configuration, see “Removing Portlet Configurations” on page 326.



SAS Web OLAP Viewer Administration

Chapter 25 **Configuring SAS Web OLAP Viewer for Java** 383

Chapter 26 **Customizing SAS Web OLAP Viewer for Java** 387



CHAPTER

25

Configuring SAS Web OLAP Viewer for Java

<i>Introduction to SAS Web OLAP Viewer for Java</i>	383
<i>Main Tasks for Administering SAS Web OLAP Viewer for Java</i>	383
<i>Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java</i>	384
<i>Configure Logging for SAS Web OLAP Viewer for Java</i>	384
<i>Redeploy SAS Web OLAP Viewer for Java</i>	384
<i>Additional Documentation for SAS Web OLAP Viewer for Java</i>	385

Introduction to SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java provides a Web interface for viewing and exploring OLAP data. SAS Web OLAP Viewer for Java provides an easy-to-use interface from which you can select a data source, view the data, and customize your view with features such as sorting and filtering. You cannot use SAS Web OLAP Viewer to make changes to information maps or to physical data.

SAS Web OLAP Viewer for Java can be run separately, or it can be launched from the SAS Information Delivery Portal.

Main Tasks for Administering SAS Web OLAP Viewer for Java

The following list summarizes the administrative tasks that are specific to SAS Web OLAP Viewer for Java:

- Ensure that SAS Web OLAP Viewer for Java can read your OLAP data.
Make sure that your OLAP data meets the requirements for rendering in SAS Web OLAP Viewer for Java. For details, see “Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java” on page 384.
- Configure the SAS Web OLAP Viewer for Java log contexts.
Use the logs to track and audit user actions for performance and security reasons. For details, see “Configure Logging for SAS Web OLAP Viewer for Java” on page 384.
- Enable ESRI maps.
The ESRI map component is a feature of SAS Web OLAP Viewer for Java that enables you to plot your OLAP data onto an interactive ESRI geographical map. With an ESRI map, you can zoom, subset, expand the map regions, and get detailed values. If you want to use this feature, then you must enable the ESRI map component. For details, see Appendix 1, “Configuring the ESRI Map Component,” on page 397.

- Customize the display.

You can customize the default display for the viewer, specify a default data source, and perform other customization tasks. For details, see Chapter 26, “Customizing SAS Web OLAP Viewer for Java,” on page 387.

Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java does not render SAS OLAP cubes directly. Instead, SAS Web OLAP Viewer for Java renders multi-dimensional SAS Information Maps that have been created from OLAP cubes. Information maps can be created in two ways:

- You can create information maps in SAS Information Map Studio.
- If a user attempts to view an OLAP cube directly, then SAS Web OLAP Viewer for Java generates an information map for that cube at run time.

In order for an information map to be displayed in SAS Web OLAP Viewer for Java, the information map must meet all of these criteria:

- The OLAP cube and the information map must exist in the same foundation repository that the user of SAS Web OLAP Viewer for Java is accessing.
- Users of SAS Web OLAP Viewer for Java must have ReadMetadata and Read permission for the information map.

Note: Users must have Read permission for an information map in order to access data through that information map. This requirement is explained in the **Instructions.html** file that was provided when you installed and configured the SAS software. △

Configure Logging for SAS Web OLAP Viewer for Java

The SAS 9.2 Intelligence Platform uses a standard logging facility to perform logging for SAS servers. Logging is managed through the Logging Service window within the SAS Management Console. For an overview and guidelines about logging, see “Administering Logging for SAS Servers” in the *SAS Intelligence Platform: System Administration Guide*.

For an overview of logging as it applies to SAS Web applications, see “Administering Logging for SAS Web Applications” on page 105.

You can access the Logging Service Configuration window in SAS Management Console. In SAS Management Console, navigate to **Plug-ins ► Environment ► Foundation Services Manager ► SASWeb OLAP Viewer4.2LocalServices ► LoggingServices** and right-click to display the Logging Service Properties window.

You can use SAS Web OLAP Viewer for Java log files to help manage performance, track security enforcement, and analyze specific situations. You can record events such as application use, failed attempts to log on, and other events.

Various contexts and outputs are created by default. You can control the level of logging messages by changing the logging priority for a particular context. Priority levels include DEBUG, INFO, WARN, ERROR, and FATAL. The default level is WARN. The DEBUG level is useful for troubleshooting, but is also very verbose and is not recommended for a production environment.

Redeploy SAS Web OLAP Viewer for Java

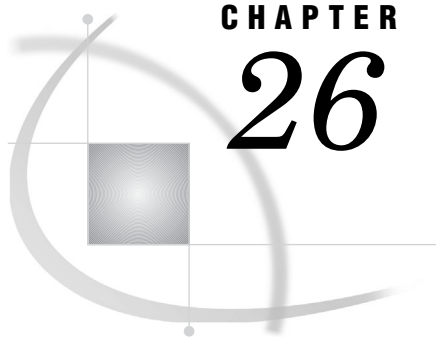
After initial installation, if you make configuration changes, then you should rebuild and redeploy SAS Web OLAP Viewer for Java. Redeploy SAS Web OLAP Viewer for

Java if the application is unconfigured and reconfigured, or if restrictive policy files are implemented or modified.

For information about rebuilding SAS Web applications with the SAS Deployment Manager, and manually redeploying SAS Web applications to a Web application server, see “Using the SAS Deployment Manager” on page 94.

Additional Documentation for SAS Web OLAP Viewer for Java

- SAS Web OLAP Viewer for Java online Help provides task instructions and information about the user interface.
- For information about middle-tier administration, see Chapter 3, “Best Practices for Configuring Your Middle Tier,” on page 23.
- For information about multicast security, single sign-on, Secure Sockets Layer, and restrictive policy files, see Chapter 4, “Middle-Tier Security,” on page 39.



CHAPTER

26

Customizing SAS Web OLAP Viewer for Java

<i>Customizing SAS Web OLAP Viewer for Java</i>	387
<i>Changes That Can Be Made to WebOLAPViewerConfig.xml</i>	388
<i>Specify a Default Data Source</i>	388
<i>Specify an Information Map</i>	388
<i>Specify a Data Exploration</i>	388
<i>Customize the Default Display for Viewers</i>	389
<i>Specify the Column Layout</i>	390
<i>Specify the Default Panel</i>	390
<i>Customize the Header and Footer Styles</i>	390
<i>Customize the Main Header</i>	391
<i>Specify a Main Footer</i>	391
<i>Specify an Alternate Header and Footer for Exporting and for Printing</i>	392
<i>Customize the Open Dialog Box</i>	392
<i>Specify a Default Initial Path</i>	393
<i>Customize Available File Types</i>	393
<i>Disable the Logoff Link</i>	393

Customizing SAS Web OLAP Viewer for Java

You can configure SAS Web OLAP Viewer for Java with custom properties for tables, plots, charts, and ESRI maps, column layout, header and footer styles, and more. You can also configure the interface to display a default information map or data exploration.

SAS Web Application Themes contains definitions for themes that are used by SAS Web OLAP Viewer for Java and other SAS Web applications. For information about themes, see Chapter 12, “Administering SAS Web Application Themes,” on page 153.

To customize SAS Web OLAP Viewer for Java, follow these steps:

- 1 Modify the **WebOLAPViewerConfig.xml** file, which is located in the *SAS-installation-directory\SASWebOlapViewerforJava\4.2\Static\wars\webolapviewer\WEB-INF* directory. For information about modifying the **WebOLAPViewerConfig.xml** file, see “Changes That Can Be Made to WebOLAPViewerConfig.xml” on page 388.
- 2 After you modify the **WebOLAPViewerConfig.xml** file, your changes will not take effect until you rebuild and redeploy SAS Web OLAP Viewer for Java. For more information, see “Redeploy SAS Web OLAP Viewer for Java” on page 384.

The following sections describe the changes that you can make to the **WebOLAPViewerConfig.xml** file.

Changes That Can Be Made to WebOLAPViewerConfig.xml

Specify a Default Data Source

You can specify a default data source that are displayed when SAS Web OLAP Viewer for Java opens. The default data source can be either an information map or a data exploration.

Note: If you specify both a default data exploration and a default information map, the data exploration is used. △

Specify an Information Map

To display a default information map, customize the **pathURL** attribute of the **<InformationMap>** element.

Use the following format to specify a fully qualified path to the information map:

```
SBIP://METASERVER/path-to-map
```

You can specify a custom query for the information map by defining **<DataItem>** elements within the **<Rows>**, **<Columns>**, and **<Slicer>** elements. If you do not specify a custom query, a default query is generated for the information map.

You can also specify one or more predefined filters for the information map by using the **<Filters>** element (located outside of the **<InformationMap>** element).

For example, the following code specifies a default information map, a custom query, and a filter:

```
<InformationMap pathURL=
  "SBIP://METASERVER/ReportStudio/Maps/SampleMap" emptyQuery=false>
  <Rows>
    <DataItem label="Geography"/>
    <DataItem label="Sum of Sales"/>
    <DataItem label="Average Sales"/>
  </Rows>
  <Columns>
    <DataItem label="Product"/>
  </Columns>
  <Slicer></Slicer>
</InformationMap>
<Filters>
  <Filter label="myFilter">
</Filters>
```

The **emptyQuery** property enables you to specify that an empty query be displayed in the user interface. When you change the value to **true**, a default query is not generated for the information map. End users have to create their initial query in the user interface. Note, if **emptyQuery** is set to true and a query is also defined, the query is ignored; the **emptyQuery** property overrides the specified query.

Specify a Data Exploration

To specify a default data exploration, modify the **activeDataExplorationPathURL** attribute of the **<DataExplorations>** element.

Use the following format to specify a fully qualified path to a data exploration:

```
SBIP://METASERVER/path-to-exploration
```

You can specify a particular bookmark from the data exploration by using the **activeBookmarkName** attribute. If you do not specify a bookmark, the default bookmark for the data exploration is displayed.

The following sample code specifies a default data exploration and bookmark:

```
<DataExplorations activeDataExplorationPathURL=
  "SBIP://METASERVER/Users/sasadm/SampleDE"
  activeBookmarkName="SampleBookmark">
</DataExplorations>
```

Customize the Default Display for Viewers

The **<Viewers>** element determines which viewers are displayed initially in the content area. The **layoutStyle** attribute controls the layout of these viewers. In addition, the number of visible rows and columns can be defined for the table viewers. If a data exploration has been specified, then its viewers take precedence over those defined here. By default, only the table is displayed, and the default layout style is one column.

To specify the default display for the data viewers, modify the elements that correspond to each viewer as shown in the following table:

Table 26.1 Viewer Elements

Element	Viewer
<Table>	Table viewer
<BarChart>	Bar chart viewer
<ColorMappedTable>	Color-mapped table viewer
<PieChart>	Pie chart viewer
<ScatterPlot>	Scatter plot viewer
<LineChart>	Line chart viewer
<BarLineChart>	Bar-line chart viewer
<TileChart>	Tile chart viewer
<ESRIMap>	ESRI map viewer
<AppliedFilters>	Applied filters viewer
<DrillPath>	Drill path viewer

The data viewer properties are located within the **<Viewers>** element, and the **<AppliedFilters>** and **<DrillPath>** elements are located outside of the **<Viewers>** element.

For all of the viewer elements, the **visible** attribute specifies whether the viewer is displayed or hidden by default. If the value for **visible** is **true**, then the viewer is displayed by default. If the value for **visible** is **false**, then the viewer is hidden by default.

For the table and color-mapped table, you can also specify number of rows and columns that are displayed by default.

For example, the following code specifies that the color-mapped table displays 10 columns and 25 rows. The bar chart viewer, color-mapped table viewer, applied filters viewer, and drill path viewer are displayed, and the other viewers are hidden.

```
<Viewers>
  <Table numberOfColumns="5" numberOfRows="20" visible="false"/>
  <BarChart visible="false"/>
  <ColorMappedTable numberOfColumns="10" numberOfRows="25"
    visible="false"/>
  <PieChart visible="false"/>
  <ScatterPlot visible="false"/>
  <LineChart visible="false"/>
  <BarLineChart visible="false"/>
  <TileChart visible="false"/>
  <ESRIMap visible="false"/>
</Viewers>

<AppliedFilters visible="true"/>
<DrillPath visible="true"/>
```

Specify the Column Layout

You can specify the column layout for your data viewers by setting the **layoutStyle** attribute of the **<Viewers>** element. Specify one of the following values:

Table 26.2 Column Layout Attributes

Attribute	Description
ONE_COLUMN_LAYOUT	Specifies that the viewers are arranged in a single column.
TWO_COLUMN_LAYOUT	Specifies that the viewers are arranged in two columns.

Specify the Default Panel

The **<StartPanel>** element controls which panel is initially displayed. The panel is the portion of the application that is displayed along the left side of the browser. By default, the Query Panel is displayed.

You can specify the default panel by using the **<StartPanel>** element. Specify one of the following values:

Table 26.3 <StartPanel> Element Attributes

Attribute	Description
BOOKMARK_PANEL	Specifies that the Bookmarks panel is displayed by default.
NAVIGATION_PANEL	Specifies that the Navigator panel is displayed by default.
QUERY_PANEL	Specifies that the Query panel is displayed by default.

Note: An invalid value results in no panel being displayed. △

Customize the Header and Footer Styles


By default, the header consists of the SAS banner. There is no default footer.

Customize the Main Header

You can customize the default page header by editing the **<Header>** element. You can edit the following elements within **<Header>**:

Table 26.4 <Header> Elements

Element	Description
<BackGroundImage>	The background image for the header. The default image is a blue SAS background.
<Image>	A logo image that is displayed in the top right corner of the header. The default image is a SAS logo. You can specify either a URL or the name of an image that is stored in the images subdirectory of your SAS Web OLAP Viewer for Java deployment.
<Text>	The main text for the header. This element specifies any static text that you want to show in the banner. The default text is “SAS Web OLAP Viewer.”
<SecondaryText>	Secondary text that follows the value of <Text> . This text can display the name of the information map or data exploration that is currently open. The useDynamicText attribute specifies whether the value of SecondaryText is generated from the name of the data source that you are viewing.
<TemplateName>	An HTML file that provides the structure of the header. The template file must reside in the templates subdirectory of the installation. If no file is specified, the default template is VisualDataExplorerHeader.html.

Note: Images are stored in the **images** subdirectory under the SAS Web OLAP Viewer for Java deployment in the servlet container. 

Specify a Main Footer

You can specify a page footer by customizing the **<Footer>** element. There is no default footer.

You can edit the following elements:

Table 26.5 <Footer> Elements

Element	Description
<BackGroundImage>	The background image for the footer.
<Image>	A logo image that is displayed in the top right.
<Text>	The main text for the footer.

Element	Description
<SecondaryText>	Secondary text that follows the value of <Text>. The <code>useDynamicText</code> attribute specifies whether the value of <SecondaryText> is generated from the name of the data source that you are viewing.
<TemplateFileName>	An HTML file that provides the structure of the footer. The template file must reside in the <code>templates</code> subdirectory of the installation. If no file is specified, then the default template is <code>VisualDataExplorerFooter.html</code> .

Specify an Alternate Header and Footer for Exporting and for Printing

You can specify an alternate header and footer that are used when you export to Excel or create a PDF file for printing.

To specify an alternate header and footer, customize the following elements within the <AlternateHeader> and <AlternateFooter> elements:

Table 26.6 Alternate Header and Footer Elements

Element	Description
<Image>	An image for the alternate header or footer.
<SecondaryImage>	A second image for the alternate header or footer.
<Text>	The main text for the alternate header or footer.
<SecondaryText>	Secondary text for the alternate header or footer. The <code>useDynamicText</code> attribute is available only in the <AlternateHeader> element, and it specifies whether the value of <SecondaryText> is generated from the name of the data source that you are viewing. The <code>newLine</code> attribute specifies whether the value of <SecondaryText> is displayed on a new line beneath the value of <Text>.

For the previous elements, you can specify the following attributes:

Table 26.7 Attributes for Alternate Header and Footer Elements

Attribute	Description
<code>hAlignment</code>	Specifies the horizontal alignment for the item. Specify either left, center, or right.
<code>vAlignment</code>	Specifies the vertical alignment for the item. Specify either top, middle, or bottom.
<code>column</code>	Specifies which column the item is placed in.
<code>row</code>	Specifies which row the item is placed in.

Customize the Open Dialog Box

You can customize the behavior of the Open dialog box by using the <FileOpen> element.

Specify a Default Initial Path

You can specify a default initial path by using the **<InitialPath>** element in the **WebOLAPViewerConfig.xml** file. The default initial path is used when you are not currently viewing a data source.

Use the following format to specify the initial path:

```
SBIP://METASERVER/path-name(Folder)
```

The **(Folder)** string at the end of the path is required.


For example, the following code fragment specifies a default initial path:

```
<FileOpen>
  <InitialPath>
    SBIP://METASERVER/ReportStudio/Maps (Folder)
  </InitialPath>
  . . .
</FileOpen>
```

Customize Available File Types

You can specify the file types that are available from the Open dialog box by using the **<FileTypes>** element.

The **<Cubes>** element specifies whether cubes are available, and the **<OLAPMaps>** element specifies whether information maps are available. For each element, the display attribute specifies whether the data type is available from the Open dialog box.

Note: Data explorations are always available from the Open dialog box. 

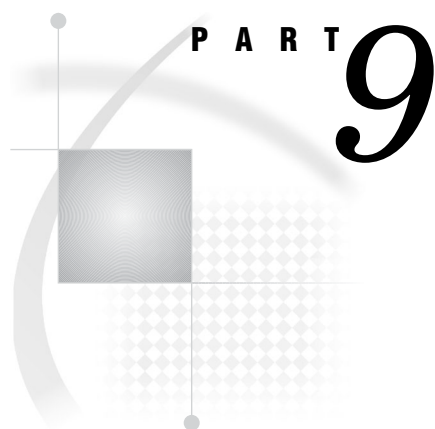
For example, the following code specifies that information maps are available and that cubes are not available:

```
<FileOpen>
  <InitialPath></InitialPath>
  <FileTypes>
    <Cubes display="false"/>
    <OLAPMaps display="true"/>
  </FileTypes>
</FileOpen>
```

Disable the Logoff Link

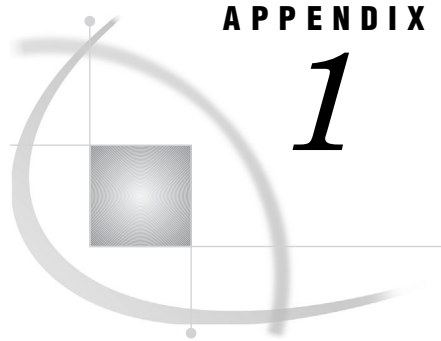
By default, SAS Web OLAP Viewer for Java displays a **Log Off** link in the banner, but you can disable the link. You might want to disable the logoff link when SAS Web OLAP Viewer for Java is accessed from SAS Information Delivery Portal, and you prefer that users log off from the portal. You can disable the logoff link and replace it with a link that returns the user to the portal.

To disable the logoff link, in the **WebOLAPViewerConfig.xml** file, change the **<LogoffButton visible="true" url="logoff.do"/>** element to **<LogoffButton visible="false" url="logoff.do"/>**.



Appendixes

<i>Appendix 1</i>	Configuring the ESRI Map Component	<i>397</i>
<i>Appendix 2</i>	Configuring the SAS Environment File	<i>403</i>
<i>Appendix 3</i>	Recommended Reading	<i>405</i>



APPENDIX

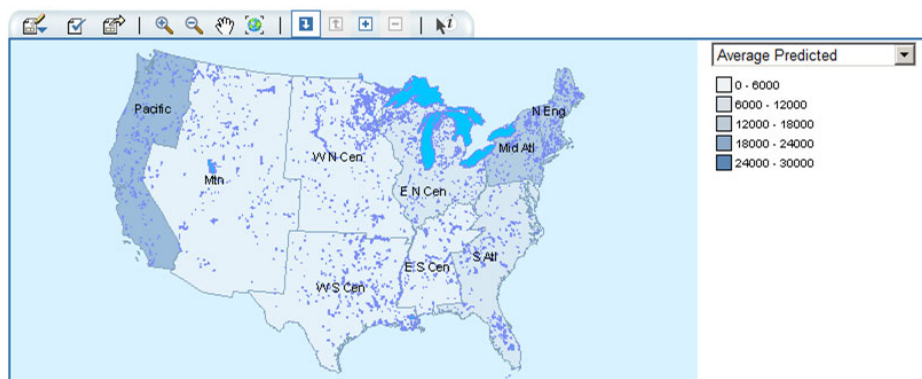
1

Configuring the ESRI Map Component

<i>About the ESRI Map Component</i>	397
<i>Create an ESRI ArcGIS Server Definition</i>	398
<i>Facilitate Authentication to the ESRI ArcGIS Server</i>	398
<i>Creating Geographic Map Services</i>	399
<i>About Geographic Map Services</i>	399
<i>Prerequisite: The Map Service Manager Plug-in</i>	399
<i>Is the Plug-in Installed on Your Desktop?</i>	399
<i>Is the Plug-in Visible to You?</i>	399
<i>Increase Visibility of the Plug-in</i>	399
<i>Create a Geographic Map Service: Basic Method</i>	400
<i>Create a Geographic Map Service: Alternate Method</i>	400
<i>About the Alternate Method for Creating a Map Service</i>	400
<i>Create a Map Service XML File</i>	400
<i>Import the Map Service XML File</i>	401
<i>Including Geographic Information in Cubes</i>	401

About the ESRI Map Component

The ESRI map component is a SAS feature that enables you to plot your OLAP data onto an interactive geographic map. For example, the following image shows a geographic map in SAS Web Report Studio:



The following table lists the SAS products that can use the ESRI map component and documents the software requirements for each product:

Table A1.1 Requirements for the ESRI Map Component

SAS Product	Required Third-Party Software
SAS Information Delivery Portal	ESRI ArcGIS Server 9.2 for the Java platform, with Service Pack 5 or later (the server does not have to run on the same machine as your SAS software).
SAS Web OLAP Viewer	
SAS Web Report Studio	
SAS Enterprise Guide	ESRI ArcGIS 9.2 engine run time, installed on each machine where a participating SAS Enterprise Guide client is running. Each ESRI map document must be specified in ArcGIS as a network location that SAS Enterprise Guide can access.

Create an ESRI ArcGIS Server Definition

To create an ESRI ArcGIS server definition, follow these steps:

- 1 Log on to SAS Management Console as someone who has the Server Manager capability (for example, sasadm@saspw).
- 2 On the **Plug-ins** tab, right-click **Server Manager** and select **Actions ► New Server**.
- 3 In the New Server Wizard, under **Resource Templates ► Servers ► Content Servers**, select **ESRI Map Server**. Click **Next**.
- 4 Enter a name for the server and click **Next**.
- 5 Enter software version information and click **Next**.
- 6 Next to the **Authentication Domain** drop-down list, click **New** and create a new authentication domain (for example, **ESRIauth**). Assign the server to the new authentication domain.

Note: You can enable users to access the ArcGIS Server under their own accounts if that server accepts the same credentials that users supply when they log on. To do this, assign the server to the DefaultAuth authentication domain and add each user's host account to the Windows ESRI group (**agusers**). Because this approach relies on reuse of cached credentials, it is not compatible with Web authentication. △

- 7 Enter the host name for the machine where the ArcGIS server is running. Click **Next**.
- 8 Click **Finish**.

Facilitate Authentication to the ESRI ArcGIS Server

Each user who uses the ESRI map component must be able to host authenticate to the Windows machine where the ArcGIS server runs. To facilitate authentication, follow these steps:

Note: This is one of several possible approaches. △

- 1 On the machine where the ArcGIS server is running, create a new Windows account named **esriuser**. Make this user a member of the **agusers** group (this Windows group is created during the ArcGIS installation).
- 2 Store the user ID and password for the **esriuser** account in the metadata as follows:

- a Log on to SAS Management Console as someone who has user administration capabilities (for example, sasadm@saspw).
- b On the **Plug-ins** tab, right-click **User Manager** and select **New ► Group**.
- c On the **General** tab, enter the name **ESRI Users**.
- d On the **Members** tab, move all of the identities that need to access the server to the **Current Members** list box.
- e On the **Accounts** tab, click **New**. In the New Login Properties dialog box:
 - i Enter the user ID in qualified form (for example, **machine\esriuser**) and password for the esriuser.
 - ii Assign the login to the authentication domain of the ArcGIS server (for example, **ESRIauth**).

All of the members of the **ESRI Users** group can now access the ArcGIS server.

Note: If you want to provide access for all SAS users, add the esriuser login to the SASUSERS group (instead of adding the login to a new group as described in this example). △

Creating Geographic Map Services

About Geographic Map Services

The ESRI map component uses a metadata object called a map service to determine how ESRI data corresponds to a particular SAS OLAP cube. You must create a separate map service for each map document that you want to associate with a SAS OLAP cube.

Prerequisite: The Map Service Manager Plug-in

Is the Plug-in Installed on Your Desktop?

The Map Service Manager plug-in to SAS Management Console is present on only those clients where the ESRI SAS Management Console plug-in is installed.

Is the Plug-in Visible to You?

The visibility of the Map Service Manager plug-in is determined by roles. In the initial configuration, the **Map Service Manager** is visible to only unrestricted users and members of the SAS Administrators group. For some sites, this level of visibility is sufficient.

Increase Visibility of the Plug-in

If you want to enable other users to define map services, you must increase the visibility of the plug-in. To make the plug-in widely available, follow these steps:

Note: This is one of several possible approaches. △

- 1 Log on to SAS Management Console as someone who has user administration capabilities (for example, sasadm@saspw).
- 2 On the **Plug-ins** tab in SAS Management Console, select **User Manager**.
- 3 In the display area, deselect the **Show Users** and **Show Groups** check boxes. Right-click the **Management Console: Content Management** role and select **Properties**.
- 4 On the **Capabilities** tab, under **Management Console 9.2 ► General**, select the **Access Unregistered Plug-ins** check box.

With this broad approach, all members of the **Management Console: Content Management** role can see the **Map Service Manager** (if that plug-in is installed on their desktop), along with any other unregistered plug-ins that they have installed. In the initial configuration, the SASUSERS group is a member of the **Management Console: Content Management** role, so all registered users are affected.

A more specific approach is to register the plug-in in SAS Management Console (under **Tools ► Plug-in Manager**). This gives the Map Service Manager capability its own check box on all roles, which enables you to directly manage access to this particular plug-in. For example, you might choose to create a new role that provides only the Map Service Manager capability and selectively assign members to that role.

Create a Geographic Map Service: Basic Method

To create geographic map service using the basic method, follow these steps:

- 1 On the **Plug-ins** tab in SAS Management Console, right-click **Map Service Manager** and select **Actions ► New Map Service**.
- 2 In the New Map Service Wizard, enter a name and select a map server. Click **Next**.
Note: When you click **Next**, SAS Management Console attempts to connect to the ArcGIS server. If the connection fails, see “Facilitate Authentication to the ESRI ArcGIS Server” on page 398. △
- 3 From the **Configuration** drop-down list, select the map document that you want to use. Click **Next**.
- 4 In the **Layers** selection box, select the layers that you want to associate with OLAP data. Click **Next**.
- 5 For each layer, select one or more fields that you want to associate with OLAP data. Click **Next**.
- 6 Click **Finish**.

Create a Geographic Map Service: Alternate Method

About the Alternate Method for Creating a Map Service

Use this method if you will use the ESRI map component through only SAS Enterprise Guide and you do not have access to the ArcGIS server. In this method, you use a text editor to create a map service XML file and then use a wizard to import that file into the metadata.

Create a Map Service XML File

In a text editor, create a new XML file and use the following template to create a map service:

```

<?xml version="1.0" encoding="utf-8" ?>
<EsriExtensionOutput>
  <MapService name="Service-Name" document="network-path-to-map-document">
    <Layer name="layer-1" alias="" uniqueId="field-1, field-2"/>
    <Layer name="layer-2" alias="" uniqueId="field-1"/>
  </MapService>
</EsriExtensionOutput>

```

The **<MapService>** element defines your map service. You can specify the following attributes:

Name= specifies the map service name.

Document= specifies the ESRI map document as a network location that all of your SAS Enterprise Guide users can access.

Each **<Layer>** element defines a layer with the map service. You can specify the following attributes:

Name= specifies the layer name.

Alias= specifies an alias for the layer. This attribute is optional.

UniqueId= specifies one or more fields that you want to associate with your OLAP data.

When you have finished creating the map service, save the document as an XML file.

Import the Map Service XML File

To import map service metadata from an XML file, follow these steps:

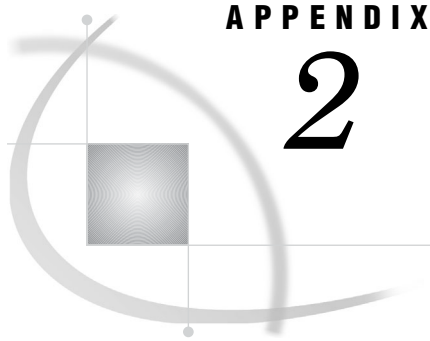
- 1 On the **Plug-ins** tab in SAS Management Console, right-click the Map Service Manager and select **Actions ► Import Map Service**.
- 2 In the Import Map Service Wizard, specify a name and then click **Browse** to locate and select the XML file that you want to import.
- 3 Click **Next**. If there is a problem with the structure of your XML file, then an error message appears. If there is no problem, then the wizard reads the information from your file.
- 4 In the **Layers** field, select the layers that you want to use. By default, all of the layers are selected. Click **Next**.
- 5 Verify the information from your XML file, and then click **Finish** to create your new map service.

Including Geographic Information in Cubes

Each cube must contain information about how its columns correspond to fields in the ESRI data. You can add this information to an existing cube or incorporate this information when you create a new cube.

The process involves designating a dimension as containing geographic information (by assigning the dimension a type of **GEO**) and then assigning ESRI spatial map information to levels within that dimension. For an example, see "Specifying an ESRI GIS Map For a Cube Dimension" in the chapter "Cube Building and Modifying Examples" in the *SAS OLAP Server: User's Guide*.

Note: In order to add geographic information to an existing cube, you must use SAS OLAP Cube Studio. The OLAP procedure does not support adding the GEO type to an existing dimension. △



APPENDIX

2

Configuring the SAS Environment File

About the SAS Environment File 403

Configuring the SAS Environment File 403

About the SAS Environment File

A SAS environment file defines the available set of SAS environments for SAS client applications, and resides within the Web Infrastructure Platform. If your application or solution uses a single environment, the SAS Logon Manager application provides a servlet that contains the information needed for a single default environment. This servlet cannot be modified to serve more than one environment. Instead, the **sas-environment.xml** file should be used to configure multiple environments.

Customize and deploy the **sas-environment.xml** to an HTTP server, if either of the following is true:

- At your site, applications or solutions are used in multiple environments such as development, test, and production environments. Your users should be able to select a SAS application or solution from multiple environments.
- SSL is configured at your site.

Configuring the SAS Environment File

The **sas-environment.xml** is located in the *SAS-configuration-directory\Lev1\Web\Common* directory.

Because Web application servers are likely to be rebooted, it is not recommended that this file be placed in a Web application server. Instead, place the customized file on an HTTP server.

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-environment-9.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.sas.com/xml/schema/sas-environment-9.2
    http://www.sas.com/xml/schema/sas-environment-9.2/
    sas-environment-9.2.xsd">
  <environment name="Red">
    <desc>test server Red for SAS Financial Management Studio</desc>
    <service-registry>http://red.na.sas.com:8080/SASWIPClientAccess/
remote/ServiceRegistry</service-registry>_
  </environment>
  <environment name="Blue" default="true">
```

```

        <desc>test server Blue for SAS Financial Management Studio</desc>
        <service-registry>http://blue.na.sas.com:7001/SASWIPClientAccess/
remote/ServiceRegistry</service-registry>
    </environment>
</environment>

```

The Service Registry, which is specified in the file, enables desktop client applications to determine the location of required services on the middle tier, and obtain a list of services available in the environment. Note that this **sas-environment.xml** file resides on an HTTP server, but the configuration in the file refers to the Web application servers and their port numbers.

If SSL is configured at your site, specify the https protocol and the SSL port number for the Service Registry.

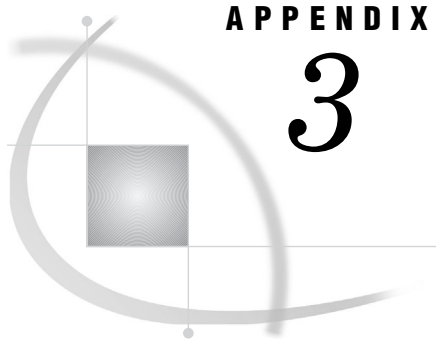
If your site has multilingual users, you can configure the **sas-environment.xml** file to include localized descriptions. In the next example, the Blue environment is specified in German:

```

<environment name="Blue"
    <desc>test2 Blue</desc>
    <desc xml:lang="de">Blau</desc>
    <service-registry>http://blue.na.sas.com:7001/SASWIPClientAccess
/remote/ServiceRegistry</service-registry>
</environment>

```

When the customized **sas-environment.xml** file is available for multiple environments, refer to the documentation for your SAS application or Solution for instructions about how to enable the availability of these environments for the users.



APPENDIX

3

Recommended Reading

Recommended Reading 405

Recommended Reading

Here is the recommended reading list for this title:

- “Special Considerations for Customers Upgrading to SAS 9.2”
- *SAS BI Dashboard: User’s Guide*
- *SAS Information Delivery Portal: Introduction*
- *SAS Integration Technologies: Overview*
- *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 Web Report Studio: User’s Guide*

For a complete list of SAS publications, go to **support.sas.com/bookstore**. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative at:

SAS Publishing Sales
 SAS Campus Drive
 Cary, NC 27513
 Telephone: 1-800-727-3228
 Fax: 1-919-531-9439
 E-mail: **sasbook@sas.com**
 Web address: **support.sas.com/bookstore**

Customers outside the United States and Canada, please contact your local SAS office for assistance.

Glossary

alert

an automatic notification of an electronic event that is of interest to the recipient.

archive

in the Publishing Framework, a package that is compressed and saved to a directory. The archive contains the contents of a package, plus metadata that is necessary for extracting the contents.

attribute

a characteristic that is part of the standard metadata for an object. Examples of attributes include the object's name, creation date, and modification date.

authentication

the process of verifying the identity of a person or process within the guidelines of a specific authorization policy.

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.

authorization

the process of determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

available page

a shared page that users of the SAS Information Delivery Portal can find using the search tool and can choose to add to their personal portals.

background

a mode of computer processing that does not require user interaction and which allows users to perform multiple tasks on the computer concurrently. In the SAS Information Delivery Portal, some stored processes run in the background so that you can perform other portal tasks during processing.

banner

a colored, rectangular area that appears at the top of some Web pages. Banners typically contain titles and navigation links.

base path

the location, relative to a WebDAV server's URL, in which packages are published and files are stored.

batch mode

a method of running SAS programs in which you prepare a file that contains SAS statements plus any necessary operating system control statements and submit the file to the operating system. Execution is completely separate from other operations at your terminal. Batch mode is sometimes referred to as running in the background.

bind

to create an association among two or more entities for a particular scope of time and place. For example, an association could be created between two or more programming objects, between a variable name and an object, between a symbolic address and a real machine address, or between a client and a server.

bookmark

a stored view for an information map. Bookmarks enable you to save and restore changes to the default view for an information map.

cache

a small, fast memory area that holds recently accessed data. The cache is designed to speed up subsequent access to the same data.

channel

a virtual communication path for distributing information. In SAS, a channel is identified with a particular topic (just as a television channel is identified with a particular radio frequency). Using the features of the Publishing Framework, authorized users or applications can publish digital content to the channel, and authorized users and applications can subscribe to the channel in order to receive the content. See also *publish* and *subscribe*.

client-side pooling

a configuration in which the client application maintains a collection of reusable workspace server processes. See also *puddle*.

cluster

a group of machines that participate in load balancing. Each machine in the cluster runs an object spawner that handles client requests for connections.

content administrator

See *group content administrator*.

content mapping

the correspondence of the SAS metadata folder structure to a content repository system. Frequently this term is used for report repository content mapping, which maps SAS metadata folders to a WebDAV (such as the SAS Content Server) repository or to a local file system. Report repository content mapping is configured when you install and configure your system.

context

the set of facts or circumstances that surround a situation or event. In Java applications, context generally refers to a collection of settings and attributes that describe a container or service that is currently executing.

credentials

the user ID and password for an account that exists in some authentication provider.

custom portlet

a portlet in the SAS Information Delivery Portal that does not fit in any of the portal's standard portlet categories (collection, navigation, bookmarks, and alert). Some custom portlets simply display data, text, or graphics, and other custom portlets have interactive features.

default page

a shared page that is automatically added to the portals of all users who belong to the group with which the page was shared. You can remove a default page from your personal portal.

delivery transport

in the Publishing Framework, the method of delivering a package to the consumer. Supported transports include e-mail and WebDAV. Although not a true transport, a channel also functions as a delivery mechanism.

deploy

to implement software in a distributed environment. Deployment typically involves installing, configuring, and testing software over a computing network.

development environment

a computing environment in which application developers use software tools to write, compile, and debug programs. See also testing environment and production environment.

encryption

the act or process of converting data to a form that only the intended recipient can read or use.

Extensible Markup Language

a markup language that structures information by tagging it for content, meaning, or use. Structured information contains both content (for example, words or numbers) and an indication of what role the content plays. For example, content in a section heading has a different meaning from content in a database table. Short form: XML.

foundation repository

in the SAS Open Metadata Architecture, a 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. Only one foundation repository should be defined on a metadata server.

foundation services

See SAS Foundation Services.

group

a collection of users who are registered in a SAS metadata environment. A group can contain other groups as well as individual users.

group content

content that a group of portal users can access. SAS Information Delivery Portal users who are designated as group content administrators can convert their personal content to group content. Group content can be edited and deleted only by the group content administrator who created it.

group content administrator

a portal user who is authorized to share pages, portlets, and other portal content items with all portal users or with other users in a group. After an item is shared, only the group content administrator can edit or delete the item.

group page

a page that has been shared with a particular group of portal users. The label Shared, followed by the name of the group, appears in the upper-right corner of group pages.

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.

HTML fragment

an HTML file that does not include opening and closing HTML tags, HEAD tags, or BODY tags and which can be displayed successfully in the cell of an HTML table.

HTTP server

a server that handles an HTTP request from a client such as a Web browser. Usually the client's HTTP request indicates that the client wants to retrieve information that is pointed to by a URL. An example of a popular HTTP server is the Apache HTTP Server from the Apache Software Foundation. See also Web server.

IFRAME

See inline frame.

information map

a collection of data items and filters that provides a user-friendly view of a data source. When you use an information map to query data for business needs, you do not have to understand the structure of the underlying data source or know how to program in a query language.

inline frame

a browser feature that enables an HTML page to be displayed within its own rectangle anywhere on another HTML page. Inline frames are created by using the HTML IFRAME tag. When necessary, inline frames contain horizontal and vertical scrollbars to enable users to view all of the page's contents within the frame.

Integrated Object Model

the set of distributed object interfaces that make SAS software features available to client applications when SAS is executed as an object server. Short form: IOM.

Integrated Object Model server

a SAS object server that is launched in order to fulfill client requests for IOM services. Short form: IOM server.

IOM

See Integrated Object Model.

IOM server

See Integrated Object Model server.

Java Development Kit

a software development environment that is available from Sun Microsystems, Inc. The JDK includes a Java Runtime Environment (JRE), a compiler, a debugger, and other tools for developing Java applets and applications. Short form: JDK.

Java RMI

See remote method invocation.

Java Virtual Machine

a program that interprets Java programming code so that the code can be executed by the operating system on a computer. The JVM can run on either the client or the server. The JVM is the main software component that makes Java programs portable across platforms. A JVM is included with JDKs and JREs from Sun Microsystems, as well as with most Web browsers. Short form: JVM.

JavaServer page

a type of servlet that enables users to create Java classes through HTML. Short form: JSP.

JDK

See Java Development Kit.

JSP

See JavaServer page.

JVM

See Java Virtual Machine.

LDAP

See Lightweight Directory Access Protocol.

Lightweight Directory Access Protocol

a protocol that is used for accessing directories or folders. LDAP is based on the X.500 standard, but it is simpler and, unlike X.500, it supports TCP/IP. Short form: LDAP.

link

(1) a portal content item that can be accessed using a URL; (2) a character string in a portal that you can click to initiate an action.

load balancing

for IOM bridge connections, a program that runs in the object spawner and that uses an algorithm to distribute work across object server processes on the same or separate machines in a cluster.

local portlet

a portlet that (1) is deployed within the same Web application that displays the portlet, (2) executes inside the portlet container, and (3) consumes the computing resources (for example, CPU, memory, and disk storage) of the server machine on which the portal Web application runs. See also remote portlet.

localhost

a keyword to specify the address of local computer that is currently in use. If a client uses localhost as the server address, then the client connects to a server that runs on the local computer.

logging context

a collection of attributes and settings that define a particular way in which the Logging Service is to be used. The logging context specifies where and in what format logging calls will be written. See also Logging Service.

Logging Service

one of the SAS Foundation Services. This service enables applications to (1) send run-time messages to one or more output destinations, including consoles, files, and socket connections; (2) configure and control the format of information that is sent to a particular destination, either by using static configuration files or by invoking run-time methods that control logging output; and (3) perform remote logging, which involves sending log messages that are generated in one Java Virtual Machine (JVM) to another JVM. See also SAS Foundation Services.

logical server

in the SAS Metadata Server, the second-level object in the metadata for SAS servers. A logical server specifies one or more of a particular type of server component, such as one or more SAS Workspace Servers.

login

a SAS copy of information about an external account. Each login includes a user ID and belongs to one SAS user or group. Most logins do not include a password.

metadata

data about data. For example, metadata typically describes resources that are shared by multiple applications within an organization. These resources can include software, servers, data sources, network connections, and so on. Metadata can also be used to define application users and to manage users' access to resources. Maintaining metadata in a central location is more efficient than specifying and maintaining the same information separately for each application.

metadata 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.

metadata object

a set of attributes that describe a table, a server, a user, or another resource on a network. The specific attributes that a metadata object includes vary depending on which metadata model is being used.

metadata repository

a collection of related metadata objects, such as the metadata for a set of tables and columns that are maintained by an application. A SAS Metadata Repository is an example.

metadata server

a server that stores information about servers, users, and stored processes and that provides this information to one or more client applications.

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.

navigation portlet

a portlet that displays content items in a hierarchical (tree) arrangement of folders and subfolders. Examples of this content might include stored processes, information maps, files that are stored in WebDAV repositories, and SAS reports.

OLAP

See online analytical processing.

online analytical processing

a software technology that enables users to dynamically analyze data that is stored in cubes. Short form: OLAP.

package

a container for data that has been generated or collected for delivery to consumers by the SAS Publishing Framework. Packages can contain SAS files (SAS catalogs; SAS data sets; various types of SAS databases, including cubes; and SAS SQL views), binary files (such as Excel, GIF, JPG, PDF, PowerPoint and Word files), HTML files (including ODS output), reference strings (such as URLs), text files (such as SAS programs), and viewer files (HTML templates that format SAS file items for viewing). Packages also contain metadata such as a description, an abstract, and user-specified name/value pairs.

PAR file

See portlet archive file.

parameter

a data item that is passed to a routine.

permanent package

a container for content that was produced by a SAS program or by a third-party application, and that is written to a specific location. Permanent packages remain in existence even after the stored process completes execution and the client disconnects from the server. See also *transient package*.

permission

the type of access that a user or group has to a resource. The permission defines what the user or group can do with the resource. Examples of permissions are *ReadMetadata* and *WriteMetadata*.

personal content

content that a portal user creates for his or her own use. As a portal user, you can create your own pages, your own portlets, and your own links. After you create these items, you can access them from the portal, edit them, remove them from your personal portal, use the Search tool to find them, or delete them permanently. Other portal users (other than a portal administrator) cannot access your personal content.

personal portal

a portal that has been personalized for or by a specific user.

personalization

the process of customizing a Web application or page to meet the needs and preferences of an individual user.

plug-in

a file that modifies, enhances, or extends the capabilities of an application program. The application program must be designed to accept plug-ins, and the plug-ins must meet design criteria specified by the developers of the application program. In SAS Management Console, a plug-in is a JAR file that is installed in the SAS Management Console directory to provide a specific administrative function. The plug-ins enable users to customize SAS Management Console to include only the functions that are needed.

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*, and *server-side pooling*.

pooling

the act or process of creating a pool. See also *pool*, *client-side pooling*, and *server-side pooling*.

portal

a Web application that enables users to access Web sites, 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.

portlet archive file

an archive (zipped) file with the suffix '.par' which includes all of the elements needed to deploy a new portlet (or group of portlets) into the SAS Information Delivery Portal, or into other applications that have been developed with the Web Infrastructure Kit. The elements in a PAR file can include a portlet deployment descriptor, JavaServer Pages (JSPs), custom Java classes, and associated resources

such as images, resource bundles, HTML files, and style sheets. Short form: PAR file. See also portlet.

portlet deployment descriptor

an XML file that specifies the actions of a portlet, as well as the portlet's initialization, path, access control, and search information. See also portlet archive file.

pre-installation checklist

a checklist that enumerates the tasks a customer must perform before installing the business intelligence platform. The primary task is to create a set of operating system user accounts on the metadata server host. See also metadata server.

production environment

a computing environment in which previously tested and validated software is used (typically on a daily basis) by its intended consumers. See also development environment and testing environment.

publication channel

an information repository that has been established using the SAS Publishing Framework and that can be used to publish information to users and applications. See also publish.

publish

to deliver electronic information, such as SAS files (including SAS data sets, SAS catalogs, and SAS data views), other digital content, and system-generated events to one or more destinations. These destinations can include e-mail addresses, message queues, publication channels and subscribers, WebDAV-compliant servers, and archive locations.

Publishing Framework

a component of SAS Integration Technologies that enables both users and applications to publish SAS files (including data sets, catalogs, and database views), other digital content, and system-generated events to a variety of destinations. The Publishing Framework also provides tools that enable both users and applications to receive and process published information.

puddle

a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers. See also client-side pooling.

remote method invocation

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. Short form: RMI. See also Java Virtual Machine.

remote portlet

a portlet that executes outside of the portal container. Remote portlets enable data from external applications to be incorporated into a Web application. When a user interacts with a remote portlet, the remote portlet appears to be the same as a local portlet. See also local portlet and portlet.

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.

report

See SAS report.

repository

a location in which data, metadata, or programs are stored, organized, and maintained, and which is accessible to users either directly or through a network. See also metadata repository, SAS Metadata Repository, and WebDAV repository.

resource

any object that is registered in a metadata repository. For example, a resource can be a server, a stored process, or a login.

result type

the kind of output that is produced by a stored process. Result types include none, streaming, permanent package, and transient package.

RMI

See remote method invocation.

SAS application server

a server that provides SAS services to a client. In the SAS Open Metadata Architecture, the metadata for a SAS application server specifies one or more server components that provide SAS services to a client.

SAS batch server

in general, 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 data set

a file whose contents are in one of the native SAS file formats. There are two types of SAS data sets: SAS data files and SAS data views. SAS data files contain data values in addition to descriptor information that is associated with the data. SAS data views contain only the descriptor information plus other information that is required for retrieving data values from other SAS data sets or from files that are stored in other software vendors' file formats.

SAS 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, activity logging, event management, information publishing, and stored process execution. See also service.

SAS log

a file that contains a record of the SAS statements that you enter as well as messages about the execution of your program.

SAS Management Console

a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Repository

one or more files that store metadata about application elements. Users connect to a SAS Metadata Server and use the SAS Open Metadata Interface to read metadata from or write metadata to one or more SAS Metadata Repositories. The metadata types in a SAS Metadata Repository are defined by the SAS Metadata Model.

SAS OLAP Server

a SAS server that provides access to multidimensional data. The data is queried using the multidimensional expressions (MDX) language.

SAS publication channel

See publication channel.

SAS report

a report that has been stored in the SAS Report Model format. A SAS report might be available for viewing in the portal if your organization has installed SAS Web Report Studio.

SAS Report Model

an XML specification that defines a standard reporting format and provides common reporting functions for SAS applications.

SAS Stored Process

a SAS program that is stored on a server and which can be executed as requested by client applications. SAS Stored Processes can be used with either a SAS Workspace Server or a SAS Stored Process Server.

SAS Stored Process Server

a SAS IOM server that is launched in order to fulfill client requests for SAS Stored Processes. See also IOM server.

SAS Stored Process Web Application

a Web application that enables you to execute stored processes and have the results returned to a Web browser.

SAS table

another term for SAS data set. See also SAS data set.

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 IOM server that is launched in order to fulfill client requests for IOM workspaces.

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.

servlet

a Java program that runs on a Web server. Servlets can be considered a complementary technology to applets, which run in Web browsers. Unlike applet code, servlet code does not have to be downloaded to a Web browser. Instead, servlets send HTML or other appropriate content back to a browser or to another type of Web-based client application.

session

a period of activity that starts when a visitor first accesses a particular Web site and that ends when the visitor has not performed any actions at that Web site within a specified time interval (usually 30 minutes). A session ID is associated with each session, and the activity that occurs during the session is recorded in a Web server log file.

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, User Services, and Logging Services use the session context to facilitate resource management and to pass information among services. See also context and bind.

single sign-on

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.

stored process

See SAS Stored Process.

streaming result

a type of output that is generated by a stored process. In a streaming result, the content that the stored process generates is delivered to the client through an output stream. The output stream is generally accessible to the stored process as the `_WEBOUT` fileref. See also result type.

subscribe

to sign up to receive electronic content that is published to a SAS publication channel.

subscriber profile

a set of personal preferences for subscribing to SAS publication channels. A subscriber profile includes the method by which you want published information to

be delivered and filtering criteria (in the form of name/value pairs) to limit the types of information that you receive. You can create multiple subscriber profiles if you want to subscribe to channels in different ways.

subscription

the association of a subscriber with a group or a channel.

syndication channel

a channel that provides syndicated, continuously updated Web content from a content provider.

testing environment

a computing environment in which application developers typically use real-life data and scenarios to test software that has been migrated from a development environment. See also development environment and production environment.

theme

a collection of specifications (for example, colors, fonts, and font styles) and graphics that control the appearance of an application.

transient package

a container for content that was produced by a SAS program or by a third-party application for immediate use, and that is not saved. After the client program disconnects from the server, the transient package disappears. See also permanent package.

trust

to accept the authentication or verification that has been performed by another software component. See also trust relationship and 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.

Unicode Transformation Format 8

a method for converting 16-bit Unicode characters to 8-bit characters. This format supports all of the world's languages, including those that use non-Latin 1 characters. Short form: UTF-8.

Uniform Resource Locator

a character string that is used by a Web browser or other software application to access or identify a resource on the Internet or on an intranet. The resource could be a Web page, an electronic image file, an audio file, a JavaServer page, or any other type of electronic object. The full form of a URL specifies which communications protocol to use for accessing the resource, as well as the directory path and filename of the resource. Short form: URL.

unrestricted user

a special user of a metadata server who can access all metadata on the server (except for passwords, which an unrestricted user can overwrite but cannot read). An unrestricted user can also perform administrative tasks such as starting, stopping, pausing, and refreshing the metadata server. You are an unrestricted user if your user ID is listed in the adminUsers.txt file and is preceded by an asterisk.

URL

See Uniform Resource Locator.

URL display portlet

a portlet that accesses a specific URL and displays the returned information inside the portlet's borders. If the URL points to a complete HTML page, then the portlet can be set up to display the URL contents inside an inline frame (IFRAME). If the URL points to an HTML fragment that is allowed by the portal's security policies, then the portlet can display the URL contents without an IFRAME. See also portlet, inline frame (IFRAME), and HTML fragment.

user context

a context that contains information about the user who is associated with an active session. The user context contains information such as the user's identity, profile, and active repository connections. See also context.

UTF-8

See Unicode Transformation Format 8.

Web Distributed Authoring and Versioning

an emerging industry standard, based on extensions to HTTP 1.1, that enables users to collaborate in the development of files and collections of files on remote Web servers. Short form: WebDAV. See also delivery transport.

Web server

a server machine and software that enable organizations to share information through intranets and through the Internet.

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 read and edit them. See also Web Distributed Authoring and Versioning.

XML

See Extensible Markup Language.

Index

A

- access control
 - for publication channels 345
- accessibility features 3
- ACTIVEX device driver 208
- administrative tasks 4
- alerts 14, 117
 - default delivery type 117
 - indicator alerts 365, 372
 - stored process alerts 348
- anonymous access 40
- anonymous Web user 143
- Apache HTTP Server
 - cache control for static content 37
 - serving SAS Themes Web application static content 36
- application scope 152
- application server
 - updating JBoss configuration 146
 - updating WebLogic configuration 148
 - updating WebSphere configuration 147
- archive permission statement 343
- ARM (application response measurement) 57
 - configuring 57
 - enabling for SAS Logon Manager 57
- audit log file 109
- auditing 109
 - for user authentication actions 110
 - for Web applications 109
 - relational tables for 110
- authenticated users 73
- authentication 33, 142
 - See also* Web authentication
 - auditing for user authentication actions 110
 - credentials for MBean access 88
 - ESRI ArcGIS server 398
 - SAS Anonymous Web User with 40
 - SAS authentication for Java 144
 - token for multicast security 172
- authentication requests 63
- authorization
 - See also* portal authorization
 - for custom-developed portlets 326
 - for custom Web applications 332
 - for packages 341
 - for publication channels 345
 - for SAS Content Server 127
 - for syndication channels 340
- AVAILABLE page attribute 299

B

- banner images 179, 181
 - for SAS Web Report Studio 209
- banner properties
 - hiding 187
- banner window
 - customizing titles 187
- batch generation tool
 - migrating to report output generation tool 257
- batch mode
 - report output generation tool 249
- batch processing reports 249
- BEA WebLogic Server
 - See* WebLogic
- bind address 173
- branding 155
- browser window
 - customizing titles 187
- burst mode
 - report output generation tool 249

C

- cache
 - configuring data cache 368
 - data caching guidelines 368
 - for images 370
 - query cache 193
- cache control
 - time-out values for static content (Apache) 37
- capabilities
 - SAS Information Delivery Portal 262
 - SAS Web Report Studio 201
 - scheduling and distributing reports 238
- cascading style sheets (CSS) 154
 - custom report styles and 222
 - formats for custom report styles 224
 - migrating 167
 - supported format properties 224
- channels
 - See also* publication channels
 - adding syndication channels to portal 337
 - deleting packages 83
 - permissions for publishing reports to 216
 - RSS channels 337
- class loader settings 103
- clear-Text 142
- client access
 - enabling for JMX 88

- client-side pooling 54
 - clustering 30
 - for Web application servers 29, 34
 - setting up for Java 150
 - codebase 50
 - colors
 - changing in themes 161
 - column layout
 - SAS Web OLAP Viewer for Java 390
 - comment management 14, 117, 118
 - conditional highlighting images
 - for SAS Web Report Studio 209
 - configuration
 - alert latency 372
 - ARM capabilities 57
 - audit log file 109
 - auditing for Web applications 109
 - cache for images 370
 - cluster of Web application servers 29, 34
 - content mapping 217
 - custom logoff message 111
 - data cache 368
 - data model cache size 370
 - data sources for middle tier 54
 - group content administrator 282
 - HTTP sessions 34
 - JBoss application server 146
 - logging for SAS Web OLAP Viewer for Java 384
 - logging for SAS Web Report Studio 189
 - middle-tier 23
 - PFS JAAS 357, 360
 - pooling of dashboard JDBC connections 371
 - properties for SAS Web Report Studio 66
 - reconfiguring Web application server 101
 - removing configuration content 94
 - removing portal configuration 273
 - restrictive policy files 45
 - sample middle-tier deployment scenarios 24
 - SAS BI Portlets, for WebSphere Portal 355
 - SAS environment file 403
 - SAS Web Report Studio 181, 183
 - SAS Web Report Studio analysis of properties 184
 - SAS Web Report Studio properties 183
 - scheduling server 239
 - shared between middle and server tiers 53
 - shared dashboard portlets 377
 - SharedServices DSN 55
 - SMTP mail server for middle tier 53
 - Web application server, to enable JMX client access 88
 - Web services for Java 137
 - Web services for .NET 136
 - WebLogic application server 148
 - WebSphere application server 147
 - Configuration Manager 64
 - configuring SAS Web Report Studio properties 183
 - deleting Web services 135
 - example 66
 - properties for SAS Web Report Studio 66
 - summary of steps for 66
 - connection parameters
 - for HTTP and HTTPS sessions 70
 - content
 - See also* portal content
 - See also* SAS Content Server
 - See also* static content
 - adding, for use by report creators 207
 - content mapping 64
 - configuring with WebDAV location 217
 - content subscribers
 - making publication channels available to 345
 - crosstabulation tables 227
 - cubes
 - including geographic information in 401
 - viewing 384
 - custom-developed portlets 265, 313, 324
 - adding permission statements 326
 - adding resource data to repository 325
 - adding to SAS Information Delivery Portal 326
 - authorization for 326
 - deploying in SAS Information Delivery Portal 325
 - designing and coding 325
 - custom logoff message 111
 - custom report styles 221, 222
 - CSS formats for 224
 - disclaimer text for graphs and tables 232
 - display filters 231
 - fonts for PDF reports 233
 - graphs 228
 - property names and values for 222
 - steps for supplying 222
 - synchronized objects container 231
 - tables 226
 - custom reports 182
 - custom repository folders 374, 376
 - custom themes
 - See* themes
 - custom Web applications 265
 - See also* Web applications
 - adding metadata to repository 330
 - adding permission statements to policy files 332
 - adding to portal 329
 - authorization for 332
 - creating user or group permissions tree 330
 - designing and coding 329
 - making available to portal 332
 - updating or removing from portal 333
 - custom window titles 187
 - customized page deployment 298
- ## D
- dashboards 363
 - See also* SAS BI Dashboard
 - performance challenges of 367
 - data caching 368
 - data exploration 388
 - data model cache size 370
 - data sets
 - specifying location, for SAS BI Dashboard 367
 - data source XML (DSX) files 366
 - data sources 54
 - configuring for middle tier 54
 - configuring SharedServices DSN 55
 - default for SAS Web OLAP Viewer for Java 388
 - for middle tier 54
 - SAS Web Report Studio 180, 207
 - DAVTree utility 79
 - adding resources to WebDAV 80
 - advanced features 82
 - connecting to a WebDAV location 80
 - copying or moving files in WebDAV 82
 - editing text files in WebDAV 82

- starting 80
- DBMS credentials 181
 - providing interactively 185
- debugging
 - Package Clean-Up utility 86
 - SAS Web Report Studio 189
- DEFAULT page attribute 299
- Default theme 155
- demilitarized zone (DMZ) 31
- Deploy as Web Service Wizard
 - overwriting Web services 152
- deployment
 - customized page deployment 298
 - EAR files 19
 - hot 150
 - manually deploying SAS OnlineDoc 102
 - portlets 321
 - redeploying SAS Web OLAP Viewer for Java 384
 - redeploying SAS Web Report Studio 196
 - redeploying Web applications 97
 - sample middle-tier scenarios 24
 - SAS BI Portlets, in WebSphere Portal 358
 - SAS Deployment Manager 94
 - themes 155, 156
 - themes, in test environment 164
 - themes, on different Web application server 165
- directives 77
- disclaimer text 179, 182
 - adding to graphs and tables 232
- display filters
 - custom report styles 231
- distributed reports 236
- distribution library
 - verifying permissions for 241
 - viewing for recipient lists 241
- DMZ (demilitarized zone) 31
- documentation 16
 - as portal component 264
 - manually deploying OnlineDoc 102
 - SAS Information Delivery Portal 275
 - SAS Web OLAP Viewer for Java 385
 - SAS Web Report Studio 182
- DSX files 366

E

- e-mail
 - configuring SMTP server 53
 - sending to users 73
 - transport restriction 343
- EAR files
 - deploying in correct order 19
 - exploded, in development environment 103
 - for rebuilding Web applications 97
 - rebuilding 95
 - redeploying 97
- encryption 32
- environment
 - See also* middle-tier environment
 - restoring to use default restrictive policy files 48
- environment file 403
 - configuring 403
- ESRI ArcGIS server
 - authentication 398
 - defining 398

- ESRI maps 383, 397
 - authentication to ESRI ArcGIS server 398
 - creating geographic map services 399
 - defining ESRI ArcGIS server 398
 - including geographic information in cubes 401
 - Map Service Manager plug-in 399
 - software requirements 397
- Event Generation Framework
 - configuring alert latency 372
- events
 - reporting in Key User Action Log 192
- exploded directories 95
- exploded EAR files
 - in development environment 103
- exploded format 150
- extract mode
 - report output generation tool 249

F

- files
 - adding to portal 328
 - adding to SAS Content Server 126
 - as portal content 279
 - deleting 126
 - permissions for files in SAS Content Server 344
 - permissions for WebDAV files 124
- filters
 - display filters 231
 - report filter values 184
- firewalls 31
- folders
 - creating 126
 - deleting 126
 - displaying custom repository folders in SAS BI Dashboard 374, 376
 - displaying legacy path folders after migration 199
 - modifying 64
 - permission tree folders 288
 - permissions for folders in SAS Content Server 344
 - permissions for SAS BI Dashboard folders 377
 - permissions for WebDAV folders 124
 - SAS Web Report Studio 178, 197
 - SAS Web Report Studio content 198
- fonts
 - for PDF reports 233
 - for SAS Web Report Studio 210
- footer style
 - SAS Web OLAP Viewer for Java 390
- forcing users to log off 74

G

- generated Web services 135, 143
- geographic information
 - including in cubes 401
- geographic map services
 - creating 399
- geographical maps feature 181
- global properties
 - setting for SAS applications 67
- graph data styles 228
- graphics
 - changing in themes 162
- graphs
 - adding disclaimer text to 232

- custom report styles 228
- WebDAV graph portlets 317
- group content administrator 278
 - assigning 280
 - configuring 282
- groups
 - associating portlets with 323
 - planning for portal groups 278

H

- header style
 - SAS Web OLAP Viewer for Java 390
- heap size 33
- Help 17
- Home page template 302
- hot deployment 150
- HTTP servers
 - load balancing for 32
 - proxy plug-in between Web application server and 36
 - serving static content 35
 - static content deployed in reverse proxy 27
- HTTP sessions
 - configuring 34
 - connection parameters for 70
 - time-out interval 112
- HTTP transport-level security 143
- https protocol
 - for SAS Content Server 44
 - for SAS Information Delivery Portal 43
- HTTPS sessions
 - connection parameters for 70

I

- IBM WebSphere Application Server
 - See* WebSphere
- images 154
 - cache for 370
 - changing in themes 162
 - for SAS Web Report Studio 209
 - migrating 168
- importing
 - legacy reports 213
 - map service XML file 401
 - reports 212
- indicator alerts 365
 - configuring alert latency 372
 - e-mail properties for 365
 - settings for 365
- information maps 179
 - adding to portal 349
 - default for SAS Web OLAP Viewer for Java 388
 - for SAS Web Report Studio 207
 - relational, implementing row-level security 212
- IOM Spawners 90
- IP multicasting 173

J

- Java
 - additional tasks for Web services 150
 - configuring Web services for 137
 - enabling application scope 152
 - enabling Java2 Security 150
 - Java Development Kit (JDK) requirement 150

- overwriting Web services 152
- running exploded in WebLogic application server 150
- SAS authentication for 144
- securing Web services for 144
- setting up clustering 150
- Web authentication for 144
- Java classes 264
- Java Development Kit (JDK) 10, 150
- Java Mail Session 53
- Java Management Extensions
 - See* JMX (Java Management Extensions)
- Java2 Security 150
- JavaBeans 264
- JBoss Application Server 9
 - creating restrictive policies for 47
 - disabling restrictive policy handling for 49
 - example policy files 46
 - exploded EAR files 105
 - message-level security 145
 - redeploying Web applications 98
 - updating configuration 146
- JConsole
 - managing SAS resources 89
- JDBC connections
 - pooling, for SAS BI Dashboard 371
- JGroups 173
- JMX (Java Management Extensions) 87
 - enabling client access 88
 - JConsole 89
 - managing SAS resources 87
 - MBeans 87, 90
- JSPs 264
- JVM arguments
 - modifying in WebSphere Administrative Console 356
 - removing custom arguments from WebSphere Administrative Console 359

K

- Key User Action Log 190
 - output 191
 - reporting events in 192

L

- languages
 - fonts for PDF reports 233
- legacy path folders
 - displaying after migration 199
- legacy reports
 - importing 213
- links
 - adding to portal 327
- list tables 226
- load balancing 32
- local portlets 322
- LocalProperties.xml file 187
- log files
 - changing location of 107
 - configuring audit log file 109
- Log On button
 - enabling on timeout pages 63
- logging 179, 181
 - changing logging levels 107
 - for Web applications 105
 - logging contexts 107, 108

- Package Clean-Up utility 86
- SAS Information Delivery Portal 274
- SAS Web OLAP Viewer for Java 384
- SAS Web Report Studio 189
- service settings for Web applications 105
- logging off
 - forcing users to log off 74
- login sessions
 - system maintenance tools for managing 74
- logoff link
 - disabling for SAS Web OLAP Viewer for Java 393
- logoff message
 - configuring custom message 111
- Logon Manager 63

M

- Maintenance Restart Wizard 75
- maintenance tools
 - for managing user login sessions 74
- manually refreshed reports 239
- Map Service Manager plug-in 399
- map service XML file
 - creating 400
 - importing 401
- MBeans 87, 90
 - accessing 87
 - authentication credentials for access 88
 - Server MBean 91
 - ServerFactory MBean 90
 - Spawner MBean 90
- message-level security 144
 - JBoss 145
 - WebLogic 145
 - WebSphere 145
- metadata
 - deleting themes from 167
 - for adding portal content 294
 - for custom Web applications 330
 - for syndication channels 338
 - modifying theme metadata 166
 - portal permission trees in 288
 - resource data for custom portlets 325
 - summary of portal content and metadata 294
- metadata server
 - adding WebSphere Portal users to 357
- middle tier 3
 - configuration shared with server tier 53
 - configuring 23
 - configuring data sources for 54
 - configuring SMTP mail server for 53
 - relational databases with 56
 - sample deployment scenarios 24
 - SAS Table Server with 55
 - security 39
- middle-tier environment 7
 - SAS Content Server 13
 - SAS Foundation Services 14
 - SAS Shared Services 14
 - SAS Web Infrastructure Platform 10
 - starting Web applications 17
 - third-party software components 9
 - Web applications 14
- migrating themes 167
 - cascading style sheets (CSS) 167
 - images 168

- theme descriptors 168
- theme templates 168
- migration
 - displaying legacy path folders 199
 - from batch generation tool to report output generation tool 257
- monitoring users 73
- multicast options 169
- multicast security 40, 169, 172
 - authentication token for 172

N

- naming themes 163
- .NET
 - configuring Web services for 136
 - securing Web services for 143
- NIC cards 173
- non-streaming stored processes 346

O

- ODS (Output Delivery System)
 - importing legacy reports 213
- OLAP
 - See SAS Web OLAP Viewer for Java
- OLAP cubes
 - including geographic information in 401
 - viewing 384
- online documentation
 - See documentation
- online Help 17
- Open dialog box
 - available file types 393
 - customizing for SAS Web OLAP Viewer for Java 392
 - default initial path 393
- output
 - Key User Action Log 191
 - stored processes 208
- Output Delivery System (ODS)
 - importing legacy reports 213

P

- Package Clean-Up utility 82
 - arguments 85
 - changing prompt behavior 84
 - deleting packages 83
 - deleting specific packages 84
 - examples 86
 - listing packages 84
 - logging and debugging 86
 - syntax for deleting packages 83
- Package Viewer 264
- packages 340
 - adding to portal 340
 - as portal content 279
 - authorization for 341
 - creating 341
 - deleting 83
 - deleting specific packages 84
 - listing 84
 - making available 341
 - publishing 341
- page templates 301
 - adding to portal 304

- deleting from portal 310
- editing 310
- editing or removing pages created from 309
- features 301
- Home page template 302
- overview 301
- pages 296
 - adding and sharing 303
 - administrators 298
 - attributes 299
 - customized deployment 298
 - editing 301, 303
 - editing or removing pages created from template 309
 - personal 300
 - rank values 298
 - removing from portal 304
 - shared 286, 300, 303
- panel
 - default for SAS Web OLAP Viewer for Java 390
- passwords
 - PUBLIC access for SAS Web Report Studio 185
 - updating 94
- PDF reports
 - fonts for 233
- performance
 - query cache and 193
 - SAS BI Dashboard 367
 - SAS Web Report Studio 192
- permission tree folders 278
 - creating 288
 - overview 288
 - removing 289
 - verifying 289
- permission trees 278, 288
 - creating for Web applications 330
 - for syndication channels 338
- permissions
 - assigning to SAS BI Portlets in WebSphere Portal 358
 - custom portlets and Web applications 50
 - custom Web applications 332
 - customizing for socket access 49
 - for folders and files in SAS Content Server 344
 - portal administration 269
 - publication channels 343
 - publishing reports to channels 216
 - remote custom portlets 326
 - report output generation tool 252
 - SAS BI Dashboard folders 377
 - syndication channels 337
 - verifying for distribution library 241
 - verifying for permission tree folders 289
 - verifying Trusted User to Content Server directories 218
 - WebDAV folders and files 124
- PERSISTENT page attribute 299
- personal content 281
- personal pages 300
- PFS JAAS configuration
 - removing in WebSphere Administrative Console 360
 - setting up in WebSphere Administrative Console 357
- plug-ins 64
- pooling 54, 179
 - of dashboard JDBC connections 371
- pooling workspace server 182
- port number
 - for SAS Content Server 44
 - for SAS Information Delivery Portal 43
- portal 264
 - See also* portal authorization
 - See also* portal content
 - See also* SAS Information Delivery Portal
 - administration of 267
 - customizing appearance 272
 - definition of 262
 - file locations 274
 - logging for 274
 - permissions for administration 269
 - providing content 270
 - recommended users for administration 269
 - reconfiguring 273
 - redistributing 273
 - removing configuration 273
 - routine maintenance 272
 - security for 271
 - setting up portal views 271
 - sharing content 283
 - using SAS BI Portlets from 354
 - verifying operation 273
- portal administrator 277
- portal authorization 277
 - analyzing and grouping users 280
 - analyzing and uploading content 279
 - assigning group content administrator 280
 - configuring group content administrator 282
 - content requiring authorization 282
 - for SAS publication channels 287
 - implementation methods 281
 - overview 280
 - planning for users and groups 278
 - portal permission trees 288
 - sharing portal content 283
- portal content 278, 293
 - custom-developed portlets 324
 - custom Web applications 329
 - example 333
 - executing stored processes from portal 346
 - files 328
 - hiding portlets from users 322
 - information maps 349
 - links 327
 - metadata requirements 294
 - packages 340
 - page templates 301, 304
 - pages 296, 303
 - portlet deployment 321
 - portlets 312
 - publication channels 342
 - SAS application server requirements 293
 - SAS reports 350
 - SAS Web OLAP Viewer for Java 333
 - SAS Web Report Studio 333
 - steps for adding portlets 316
 - summary of content and metadata 294
 - syndication channels 337
 - WebDAV graph portlets 317
- portal permission trees 288
- portal users and groups 278
- portal views 271, 298
 - page attributes and 299
- portlet templates 313
- portlets 297, 312
 - See also* custom-developed portlets

- See also SAS BI Portlets
 - access permissions for custom portlets 50
 - associating with groups 323
 - configuring shared dashboard portlets 377
 - deploying 321
 - editable 313
 - execution of local and remote 322
 - hiding from users 322
 - overview 312
 - predefined 315
 - remote 326
 - steps for adding 316
 - that cannot be shared 286
 - updating remote portlets for SSL 44
 - WebDAV graph portlets 317
 - pre-generated reports 236
 - batch processing 249
 - configuring a scheduling server 239
 - distributed reports 236
 - manually refreshed reports 239
 - processed outside of Web Report Studio 249
 - recipient list for report distribution 241
 - report output generation tool 249
 - report scheduling versus report distribution 237
 - role capabilities for scheduling and distribution 238
 - scheduled reports 236
 - security considerations 216
 - verifying permissions for distribution library 241
 - predefined portlets 315
 - predefined roles
 - SAS Web Report Studio 201
 - SAS Web Report Viewer 201, 205
 - preferences 62, 264
 - product-specific branding 155
 - production environment
 - moving themes to 164
 - prompts
 - Package Clean-Up utility 84
 - prompts file
 - as input for reports 256
 - properties
 - CSS formats 224
 - custom report styles 222
 - editing LocalProperties.xml file 187
 - for SAS Web Report Studio 66
 - global properties for SAS applications 67
 - hiding banner properties 187
 - SAS Application Infrastructure 67
 - SAS Web Report Studio 179, 183, 184
 - proxy configurations
 - configuring HTTP sessions in environments with 34
 - proxy plug-ins
 - between Web application server and HTTP server 36
 - Public Key Cryptography 32
 - PUBLIC users 181
 - publication channels 342
 - adding subscribers to 345
 - adding to portal 342
 - adding to SAS Metadata Repository 345
 - archive permission statement 343
 - as portal content 279
 - authorization for 287, 345
 - creating 345
 - e-mail transport restriction 343
 - for folders and files in SAS Content Server 344
 - making available to content subscribers 345
 - subscriber profiles 343
 - WebDAV 343
 - Publishing Framework plug-in 287
 - publishing packages 341
 - publishing reports
 - permissions for 216
- ## Q
- query cache 193
 - disabling 196
 - host access to directory 194
 - location of library 195
 - quiesce the system 76
- ## R
- Rampart security 144
 - rebuilding themes 163
 - rebuilding Web applications 95
 - EAR file names 97
 - EAR files 95
 - exploded directories 95
 - rebuilding one or more 95
 - when to rebuild 95
 - recipient lists 241
 - considerations for creating 248
 - creating 243
 - creating with SQL procedure 247
 - enabling report distribution with 242
 - SAS Web Report Studio 179
 - viewing distribution library for 241
 - redeploying Web applications 97
 - JBoss 98
 - redeploying EAR files 97
 - WebLogic 98
 - WebSphere 101
 - redemption
 - EAR files 97
 - SAS Web OLAP Viewer for Java 384
 - SAS Web Report Studio 196
 - refreshing reports manually 239
 - relational databases
 - with middle tier 56
 - relational information maps
 - implementing row-level security 212
 - row-level security and 182
 - relational tables
 - for auditing 110
 - remote portlets 322, 326
 - updating for SSL 44
 - remote services
 - updating files for Web authentication 149
 - report distribution
 - compared with report scheduling 237
 - enabling with recipient lists 242
 - required role capabilities for 238
 - setting up recipient lists 241
 - verifying permissions for distribution library 241
 - with report output generation tool 253
 - report filters
 - maximum values for 184
 - report output generation tool 249
 - distributing reports with 253
 - examples 255
 - logging directory permissions for 252

- migrating from batch generation tool 257
- report scheduling
 - compared with report distribution 237
 - configuring a scheduling server 239
 - required role capabilities for 238
- reports
 - See also* custom report styles
 - See also* pre-generated reports
 - See also* SAS Web Report Studio
 - access to 214
 - adding content for report creators 207
 - adding SAS reports to portal 350
 - batch processing 249
 - changing access to 216
 - customizing 182
 - distributed 236
 - fonts for PDF reports 233
 - importing 212
 - importing legacy reports 213
 - permissions for publishing to channels 216
 - protecting data in temporary files 219
 - protecting WebDAV server content 217
 - resources 180
 - SAS Report Portlet 354
 - SAS Stored Process Portlet 354
 - scheduled 236
 - scheduling and distributing 182
 - security considerations 214
 - viewing 179
- repository folders
 - displaying custom folders in SAS BI Dashboard 374, 376
- resources
 - adding to WebDAV repository 80
 - managing SAS resources with JConsole 89
 - managing SAS resources with JMX tools 87
 - report resources 180
- restrictive policy files 45
 - creating for JBoss 47
 - creating for WebSphere 48
 - customizing permissions for socket access 49
 - disabling for JBoss 49
 - disabling for WebSphere 49
 - example files for JBoss and WebSphere 46
 - permissions for custom portlets and Web applications 50
 - permissions for publishing reports to channels 216
 - restoring default policies 48
- reverse proxy
 - static content deployed in 27
- Rich Site Summary (RSS) channels 337
- roles
 - capabilities for scheduling and distributing reports 238
 - predefined, for SAS BI Dashboard 375
 - predefined, for SAS Web Report Studio 201
 - predefined, for SAS Web Report Viewer 201, 205
 - SAS Web Report Studio 179
- row-level security 182, 217
 - implemented by relational information maps 212
- RSS channels 337

S

- SAS Anonymous Web User
 - SAS authentication with 40
- SAS Application Infrastructure properties 67

- SAS application server
 - as portal content 279
 - requirements for adding portal content 293
- SAS applications
 - global properties for 67
- SAS authentication 142
 - for Java 144
 - SAS Anonymous Web User with 40
- SAS BI Dashboard 16, 23, 363
 - accessing 364
 - administration tasks 364
 - alert latency 372
 - configuring shared portlets 377
 - data caching 368
 - data set locations 367
 - displaying custom repository folders 374, 376
 - DSX files 366
 - folder permissions 377
 - indicator alerts 365, 372
 - performance 367
 - pooling of JDBC connections 371
 - predefined administration role 375
 - seamless access from SAS Information Delivery Portal 374
 - security 375
 - security, key points for 376
 - user groups 375
- SAS BI Portlets 353
 - adding to user's Web page in WebSphere Portal 359
 - assigning permissions to, in WebSphere Portal 358
 - configuring for WebSphere Portal 355
 - deleting from WebSphere Portal 359
 - deploying in WebSphere Portal 358
 - removing from WebSphere Portal server 359
 - SAS Collection Portlet 354
 - SAS Navigator Portlet 354
 - SAS Report Portlet 354
 - SAS Stored Process Portlet 354
 - using from a portal 354
 - using with SAS Information Delivery Portal 354
 - using with WebSphere Portal 354
- SAS BI Web Services for Java 10
- SAS Collection Portlet 354
- SAS Comment Manager 118
- SAS Content Server 10, 13, 121
 - adding files to 126
 - Administration Console 122
 - authorization for 127
 - https protocol and port number 44
 - permissions for folders and files in 344
 - portal authorization and 278
 - SAS Information Delivery Portal and 264
 - verifying SAS Trusted User access to directories 218
 - verifying SAS Trusted User permissions to directories 218
 - verifying user access to directories 219
- SAS Content Server Administration Console 122
 - accessing 122
 - adding files to SAS Content Server 126
 - creating folders 126
 - deleting folders or files 126
 - interface 123
 - permissions for WebDAV folders and files 124
- SAS Default theme 155
- SAS Deployment Manager 94
 - accessing 94

- auditing for Web applications 109
- exploded EAR files in development environment 103
- HTTP session time-out interval 112
- logging for Web applications 105
- manually deploying SAS OnlineDoc 102
- rebuilding Web applications 95
- removing configuration content 94
- updating passwords 94
- SAS Documentation Web application 16, 264
- SAS environment file 403
 - configuring 403
- sas-environment.xml 403
 - configuring 403
- SAS Foundation Services 12, 14
- SAS Help Viewer for the Web 17
- SAS Help Viewer Metadata Configuration 17
- SAS Information Delivery Portal 16, 23, 261
 - See also* portal
 - See also* portal authorization
 - See also* portal content
 - additional documentation 275
 - capabilities 262
 - components 263
 - executing stored processes from 346
 - features 263
 - https protocol and port number 43
 - logging for 274
 - one-way SSL for 43
 - predefined portlets provided with 315
 - seamless access to SAS BI Dashboard 374
 - two-way SSL for 44
 - using SAS BI Portlets with 354
- SAS Intelligence Platform 7
 - SAS Web Report Studio and 178
- SAS Logon Manager 10, 63
 - enabling ARM for 57
- SAS Mail Service 53
- SAS Management Console 64
 - assigning default theme from 165
 - Configuration Manager 64
 - modifying folders 64
 - modifying theme metadata 166
- SAS Metadata Repository
 - adding publication channels to 345
 - adding resource data, for custom portlets 325
- SAS Navigator Portlet 354
- SAS OnlineDoc for the Web 17
 - accessing 103
 - manually deploying 102
 - WebSphere class loader settings 103
- SAS Preferences Manager 10, 62
- SAS Preferences Web application 264
- SAS programs
 - converting to stored processes 208
- SAS Remote Services Application
 - heap size for 33
 - restarting 44
- SAS Report Portlet 354
- SAS reports
 - adding to portal 350
- SAS resources
 - managing with JConsole 89
 - managing with JMX tools 87
- SAS servers 90, 91
- SAS Shared Services 14, 117
 - default alert notification delivery type 117
 - SAS Comment Manager 118
- SAS Shared Web Assets 10
- SAS Stored Process Server Web application 264
- SAS Stored Process Web application 10
- SAS Table Server 55
 - SharedServices database on 56
 - with middle tier 55
- SAS Themes Web application
 - serving static content 35
 - serving static content with Apache HTTP Server 36
- SAS Trusted User
 - verifying access to Content Server directories 218
 - verifying permissions to Content Server directories 218
- SAS Web Administration Console 10, 72
 - accessing 73
 - forcing users to log off 74
 - monitoring users 73
 - sending e-mail to users 73
 - system maintenance tools for user login sessions 74
 - users appearing in 73
 - viewing information about Web applications 76
- SAS Web Application Themes
 - See* themes
- SAS Web Infrastructure Platform 10, 61
 - Configuration Manager 64
 - connection parameters for HTTP and HTTPS sessions 70
 - global properties for SAS applications 67
 - SAS Logon Manager 63
 - SAS Management Console 64
 - SAS Preferences Manager 62
 - SAS Web Administration Console 72
- SAS Web Infrastructure Platform Services 10, 13
- SAS Web OLAP Viewer for Java 15, 23, 383
 - adding to portal 333
 - additional documentation 385
 - administration tasks 383
 - column layout 390
 - configuring logging for 384
 - customizing 387
 - default data source 388
 - default display for viewers 389
 - default panel 390
 - disabling logoff link 393
 - header and footer styles 390
 - Open dialog box 392
 - redeploying 384
 - SAS Information Delivery Portal and 264
 - viewing OLAP cubes 384
- SAS Web Report Studio 15, 23, 177
 - access to reports 214
 - adding content for report creators 207
 - adding to portal 333
 - additional documentation 182
 - banner images 179, 181
 - banner properties, hiding 187
 - capabilities for 201
 - changing access to reports 216
 - configuring 181, 183
 - configuring analysis of properties 184
 - configuring properties 66, 183
 - content storage folders 198
 - customizing banner and browser window titles 187
 - customizing reports 182
 - data sources 207
 - DBMS credentials 185
 - debugging 189

- disclaimer text 179
- displaying legacy path folders after migration 199
- editing LocalProperties.xml 187
- folders 197
- fonts 210
- fonts for PDF reports generated by 233
- images for 209
- importing legacy reports 213
- importing reports 212
- information maps 179, 207
- Key User Action Log 190, 191, 192
- logging 179, 181, 189
- main administration tasks 180
- new administration features in 4.2 178
- passwords for PUBLIC users 185
- performance 192
- permissions for publishing reports to channels 216
- pooling 179
- pre-generated reports 236
- predefined roles 201
- preparing report resources 180
- prerequisites for administering 180
- properties 179
- protecting data in temporary files created by 219
- protecting report content in WebDAV server 217
- query cache 193
- recipient lists 179
- redeploying 196
- relational information maps with row-level security 212
- report filter values 184
- roles 179
- row-level security 217
- SAS Information Delivery Portal and 264
- SAS Intelligence Platform and 178
- SAS Web Report Viewer 178
- scheduling and distributing reports 182
- security for pre-generated reports 216
- security implementation 181
- stored processes for 207
- user folders 178
- viewing reports 179
- working area 179
- SAS Web Report Viewer 178
 - predefined roles 201, 205
- scheduled reports 236
- scheduling server
 - configuring 239
- Secure Sockets Layer
 - See* SSL (Secure Sockets Layer)
- security
 - access to reports 214
 - ESRI ArcGIS server 398
 - HTTP transport-level 143
 - Java2 150
 - message-level 144
 - middle tier 39
 - multicast 40, 169, 172
 - portal 271
 - pre-generated reports 216
 - Rampart 144
 - restrictive policy files 45
 - row-level 212, 217
 - SAS Anonymous Web User 40
 - SAS BI Dashboard 375
 - SAS BI Dashboard, key points 376
 - SAS Web Report Studio 181
 - shared dashboard portlets 377
 - Single Sign-On 41
 - SSL 41
 - transport-level 145
 - Web services 142
 - Web services for Java 144
 - Web services for .NET 143
 - WS-Security message-level 143
 - Server MBean 91
 - server-side pooling 54
 - server tier
 - configuration shared with middle tier 53
 - ServerFactory MBean 90
 - servers
 - interactions 18
 - required for Web applications 17
 - starting in correct order 18
 - servlets 264
 - session affinity 31
 - session scope 152
 - session time-out interval 112
 - shared dashboard portlets 377
 - shared pages 286, 300, 303
 - SharedServices database 55
 - backing up and restoring 56
 - on SAS Table Server 56
 - SharedServices DSN 55
 - configuring 55
 - sharing portal content 283
 - items that contain other items 286
 - portlets that cannot be shared 286
 - shared pages 286
 - sharing with multiple groups of users 287
 - types of changes for shared content 285
 - types of users that can share 285
 - when content can be shared 287
 - Single Sign-On (SSO) 41
 - SMTP mail server
 - configuring for middle tier 53
 - socket access
 - customizing permissions for 49
 - sources
 - See* data sources
 - Spawner MBean 90
 - SQL procedure
 - creating recipient lists 247
 - SSL (Secure Sockets Layer) 32
 - enabling 142
 - for Web applications 41, 42
 - one-way, for SAS Information Delivery Portal 43
 - setup for Web application server 42
 - two-way, for SAS Information Delivery Portal 44
 - updating remote portlets 44
 - verifying connections 44
 - SSO (Single Sign-On) 41
 - static content 35
 - Apache cache control for 37
 - deployed in reverse proxy 27
 - serving in SAS Themes Web application 35
 - serving SAS Themes Web application content with Apache HTTP Server 36
 - serving with HTTP servers 35
 - storage folders
 - for SAS Web Report Studio content 198
 - stored process alerts 348

- stored processes 264, 346
 - converting SAS programs to 208
 - executing from portal 346, 347
 - for SAS Web Report Studio 207
 - non-streaming 346
 - output style 208
 - SAS Stored Process Portlet 354
- subscriber profiles 343
- subscribers
 - adding to publication channels 345
 - making publication channels available to 345
- synchronized objects container
 - custom report styles 231
- syndication channels
 - adding metadata to repository 338
 - adding to portal 337
 - authorization for 340
 - making available to portal 340
 - permission statements 337
 - updating or removing from portal 340
 - user or group permissions tree for 338
- system maintenance tools
 - for managing user login sessions 74
- system users 73

T

- tables
 - adding disclaimer text to 232
 - custom report styles 226
- tasks 4
- templates
 - See* page templates
 - See* portlet templates
- temporary files
 - protecting, when created by Web Report Studio 219
- test environment
 - deploying themes in 164
- testing themes 164
- text files
 - editing in WebDAV 82
- theme descriptors 154
 - migrating 168
- theme templates 154
 - changing 163
 - migrating 168
- themes 154, 221
 - assigning as default theme 165
 - cascading style sheets (CSS) 154
 - changing colors 161
 - changing graphics 162
 - changing theme templates 163
 - components 154
 - creating and deploying 155
 - creating work area for 157
 - Default theme 155
 - defining and deploying 156
 - deleting from metadata 167
 - deploying in test environment 164
 - deployment on different Web application server 165
 - designing 156
 - images and 154
 - migrating 167
 - migrating cascading style sheets (CSS) 167
 - migrating images 168
 - migrating theme descriptors 168

- migrating theme templates 168
- modifying metadata 166
- moving to production environment 164
- naming 163
- rebuilding 163
- SAS Information Delivery Portal and 264
- testing 164
- third-party software 9
- time-out interval 112
- timeout pages
 - enabling Log On button on 63
- titles
 - customizing window titles 187
- transport-level security 145
- tuning Web application servers 34

U

- UpdateDefaultTheme.sas program 165
- UpdateTheme.sas program 166
- users
 - adding WebSphere Portal users to metadata server 357
 - analyzing and grouping portal users 280
 - appearing in SAS Web Administration Console 73
 - auditing for authentication actions 110
 - authenticated 73
 - forcing users to log off 74
 - hiding portlets from 322
 - in SAS BI Dashboard groups 375
 - monitoring with SAS Web Administration Console 73
 - planning for portal users 278
 - recommended for portal administration 269
 - sending e-mail to 73
 - sharing portal content 285
 - system maintenance tools for managing login sessions 74
 - system users 73
 - verifying access to Content Server directories 219

V

- Visual Data Explorer 264

W

- Web application servers 9
 - bind address and JGroups 173
 - configuring a cluster of 29, 34
 - deploying themes on different server 165
 - enabling JMX client access 88
 - proxy plug-in between HTTP server and 36
 - reconfiguring 101
 - restarting 44
 - SSL setup for 42
 - tuning 34
 - Web applications deployed in single server 24
- Web application themes
 - See* themes
- Web applications 14
 - See also* custom Web applications
 - access permissions for foundation service-enabled 50
 - auditing for 109
 - changing location of log files 107
 - changing logging levels 107
 - configuring custom log off messages 111
 - deployed across Web application server cluster 29
 - deployed in single Web application server 24

- deploying EAR files in correct order 19
- directives 77
- exploded EAR files in development environment 103
- HTTP session time-out interval 112
- logging contexts 108
- logging for 105
- manually deploying SAS OnlineDoc 102
- prerequisites for administering 4
- rebuilding 95
- redeploying 97
- required servers 17
- SAS BI Dashboard 16
- SAS Deployment Manager and 94
- SAS documentation for the Web 16
- SAS Information Delivery Portal 16
- SAS Web Administration Console 72
- SAS Web OLAP Viewer for Java 15
- SAS Web Report Studio 15
- settings 76
- Single Sign-On 41
- SSL for 41, 42
- starting 17
- starting servers in correct order 18
- themes 154
- viewing information about 76
- Web authentication 33, 143, 144, 182
 - See also* authentication
 - for Java 144
 - message-level security 144
 - transport-level security 145
 - updating JBoss application server configuration 146
 - updating remote services files 149
 - updating WebLogic application server configuration 148
 - updating WebSphere application server configuration 147
- Web Service Maker 135, 143
- Web services
 - additional Java tasks 150
 - configuring for Java 137
 - configuring for .NET 136
 - deleting 135
 - generated 135, 143
 - overwriting 152
 - security for 142
 - security for Java 144
 - security for .NET 143
 - XMLA 143
- Web Services Enhancements for Microsoft .NET (WSE) 143
- webanon account 40
- WebDAV
 - See also* DAVTree utility
 - adding resources to repository 80
 - configuring content mapping 217
 - content management with DAVTree utility 79
 - copying or moving files 82
 - deleting packages 83
 - editing text files 82
 - graph portlets 317
 - permissions for folders and files 124
 - protecting report content 217
 - publication channels 343
- WebLogic 9
 - exploded EAR files 105
 - message-level security 145
 - redeploying Web applications 98
 - running exploded in application server 150
 - updating application server configuration 148
- WebOLAPViewerConfig.xml 388
- WebSphere 9
 - class loader settings 103
 - creating restrictive policies for 48
 - disabling restrictive policy handling for 49
 - example policy files 46
 - exploded EAR files 105
 - message-level security 145
 - redeploying Web applications 101
 - updating application server configuration 147
- WebSphere Administrative Console
 - modifying JVM arguments in 356
 - removing custom JVM arguments from 359
 - removing PFS JAAS configuration 360
 - setting up PFS JAAS configuration 357
- WebSphere Portal
 - adding SAS BI Portlets to user's Web page in 359
 - adding users to metadata server 357
 - assigning permissions to SAS BI Portlets in 358
 - configuring SAS BI Portlets for 355
 - deleting SAS BI Portlets from 359
 - deploying SAS BI Portlets in 358
 - using SAS BI Portlets with 354
- WebSphere Portal server
 - removing SAS BI Portlets from 359
- windows
 - customizing titles 187
- work area
 - creating for themes 157
 - SAS Web Report Studio 179
- WS-Security message-level security 143

X

XMLA Web services 143

Your Turn

We welcome your feedback.

- ☐ If you have comments about this book, please send them to **`yourturn@sas.com`**. Include the full title and page numbers (if applicable).
- ☐ If you have comments about the software, please send them to **`suggest@sas.com`**.

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn



**THE
POWER
TO KNOW®**