



THE  
POWER  
TO KNOW.

# **SAS<sup>®</sup> 9.1.3**

# **Intelligence Platform**

## **Web Application**

## **Administration Guide**

## **Third Edition**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2008. *SAS® 9.1.3 Intelligence Platform: Web Application Administration Guide, Third Edition*. Cary, NC: SAS Institute Inc.

**SAS® 9.1.3 Intelligence Platform: Web Application Administration Guide, Third Edition**

Copyright © 2008, SAS Institute Inc., Cary, NC, USA

ISBN-13: 978-1-59994-834-8

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 printing, July 2008

1st electronic printing, July 2008

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/pubs](http://support.sas.com/pubs) 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>	<i>ix</i>
Overview	<b>ix</b>
Support for IBM WebSphere Application Server 6.1	<b>ix</b>

## **PART 1**    **Getting Started**    **1**

<b>Chapter 1</b> △ <b>Before You Begin</b>	<b>3</b>
Introduction to This Guide	3
Accessibility Features in the SAS Intelligence Platform Products	3
Prerequisites for Administering the Web Applications	4
High-Level Overview of Administrative Tasks	5
<b>Chapter 2</b> △ <b>Working In the Middle-Tier Environment</b>	<b>7</b>
Understanding the Middle-Tier Environment	7
Starting the Web Applications	13
Redeploying the Web Applications	14
Change the HTTP Session Timeout Interval	14

## **PART 2**    **Middle-Tier Administration**    **17**

<b>Chapter 3</b> △ <b>Setting Up and Managing Middle-Tier Security</b>	<b>19</b>
Planning Your Middle-Tier Security Implementation	20
Understanding Single Sign-On	24
Changing to Trusted Web Authentication	32
Configuring the Web Applications for Secure Sockets Layer (SSL)	42
Adding Permissions to Policy Files	45
<b>Chapter 4</b> △ <b>Best Practices for Configuring Your Middle Tier</b>	<b>57</b>
Overview of Middle Tier Configuration	58
Tuning the Java Virtual Machine	58
Tuning the J2EE Application Server or Servlet Container	64
Tuning WebSphere 6.0.2 or 6.1	66
Sample Middle-Tier Deployment Scenarios	70
Configuring a Cluster of J2EE Application Servers	84
Configuring an HTTP Server to Serve Static Content for SAS Web Applications	86
Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server	90
Configuring Apache Cache Control for Static Content	96

## **PART 3**    **SAS Web Report Studio Administration**    **99**

<b>Chapter 5</b> △ <b>Introduction to SAS Web Report Studio Administration</b>	<b>101</b>
Introduction to SAS Web Report Studio	101

Prerequisites for Administering SAS Web Report Studio	101
Main Tasks for Administering SAS Web Report Studio	102
Additional Documentation for SAS Web Report Studio	104
<b>Chapter 6 △ Configuring SAS Web Report Studio</b>	<b>105</b>
SAS Web Report Studio Configuration Files and Tools	105
Enabling Interaction with Other SAS Applications	109
Configuring the SAS Web Report Studio Logs	109
Improving the Performance of SAS Web Report Studio	113
Re-Create and Redeploy SAS Web Report Studio	116
<b>Chapter 7 △ Managing SAS Web Report Studio Content and Users</b>	<b>119</b>
Setting Up Storage for Reporting	119
Adding Content for Use by Report Creators	123
Setting up Users for SAS Web Report Studio	129
Managing Access to Reports	134
<b>Chapter 8 △ Customizing Reports</b>	<b>139</b>
Add Disclaimer Text to Graphs and Tables	139
Customizing Report Styles	140
<b>Chapter 9 △ Scheduling and Distributing Pre-generated Reports</b>	<b>153</b>
Overview: Scheduling and Distributing Pre-generated Reports	153
Main Administrative Tasks for Scheduling and Distributing Reports	156
Setting Up a Distribution Library and Recipient List	156

## **PART 4 SAS Web OLAP Viewer Administration 165**

<b>Chapter 10 △ Introduction to SAS Web OLAP Viewer for Java Administration</b>	<b>167</b>
Introduction to SAS Web OLAP Viewer for Java	167
Prerequisites for Administering SAS Web OLAP Viewer for Java	167
Main Tasks for Administering SAS Web OLAP Viewer for Java	168
Additional Documentation for SAS Web OLAP Viewer for Java	168
<b>Chapter 11 △ Configuring SAS Web OLAP Viewer for Java</b>	<b>169</b>
Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java	169
Upgrade Information Maps to the SAS Information Map Studio 3.1 Format	169
Configure Logging for SAS Web OLAP Viewer for Java	170
Improving the Performance of SAS Web OLAP Viewer for Java	171
Re-Create and Redeploy SAS Web OLAP Viewer for Java	171
<b>Chapter 12 △ Customizing SAS Web OLAP Viewer for Java</b>	<b>173</b>
Main Steps for Customizing SAS Web OLAP Viewer for Java	173
Changes That Can Be Made to WebOLAPViewerConfig.xml	174

## **PART 5 Portal Web Application Administration 181**

<b>Chapter 13</b>	<b>△ Overview of the Portal Web Application</b>	<b>183</b>
	Introduction to the Portal Web Application	183
	Understanding the SAS Web Infrastructure Kit and the SAS Information Delivery Portal	184
	Summary of Portal Features and Their Software Requirements	186
	Understanding the Portal Components	188
<b>Chapter 14</b>	<b>△ Introduction to Portal Administration</b>	<b>191</b>
	Prerequisites for Administering the Portal Web Application	191
	Who Can Administer the Portal Web Application	193
	Main Tasks for Administering the Portal Web Application	196
	Suggestions for Verifying Portal Operation	198
	Important Portal Administrative Files	199
	Loading Initial Metadata	200
	Administering the Public Kiosk	202
	Modifying the Logging Output Information and Location	204
	Additional Documentation for the Portal	207
<b>Chapter 15</b>	<b>△ Using the Portal Administration Tools</b>	<b>209</b>
	Overview of the Portal's Administration Tools	209
	Using the Portal Options Menu	210
	Re-Create and Redeploy the Portal Web Application	211
	Using <code>initPortalData</code> to Update Portal Permission Trees	212
	Using the Quiesce Portlet to Bring Down the Portal	213
	Using the SAS Portal Metadata Tool to Remove Portal Metadata	215
<b>Chapter 16</b>	<b>△ Administering Portal Authorization</b>	<b>219</b>
	Overview of Portal Authorization Tasks	219
	Planning for Portal Users and Groups	220
	Understanding Portal Authorization	222
	Configure a Group Content Administrator	224
	Sharing Content in the Portal Web Application	226
	Setting Up Authorization for Stored Processes and Publication Channels	229
	Implementing Authorization for the Xythos WebFile Server	231
	Managing Portal Permission Trees in Metadata	233
<b>Chapter 17</b>	<b>△ Adding Content to the Portal</b>	<b>237</b>
	Overview of Adding Content	239
	Summary of Content That Can Be Added to the Portal	240
	Understanding Pages and Page Templates	242
	Adding, Editing, and Removing Pages	249
	Adding, Editing, and Removing Page Templates	251
	Understanding Portlets	258
	Main Steps to Add a Portlet	262
	Adding WebDAV Graph Portlets	264
	Adding Custom-Developed Portlets	268

Understanding Portlet Deployment	270
Hiding Portlets from Users	272
Adding Links	274
Adding Files	275
Adding Web Applications	276
Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java	282
Adding Syndication Channels	286
Adding SAS Packages	290
Adding SAS Publication Channels	292
Adding and Administering SAS Stored Processes	294
Adding SAS Information Maps	299
Adding SAS Reports	300
<b>Chapter 18</b> △ <b>Administering SAS Business Intelligence Dashboard</b>	<b>303</b>
Overview of SAS Business Intelligence Dashboard	303
Main Tasks for Administering SAS Business Intelligence Dashboard	304
Understanding the Data Source XML (DSX) Files	304
Specify a JDBC Data Source for SAS Business Intelligence Dashboard	305
Improving the Performance of SAS Business Intelligence Dashboard	306
Managing User Security for SAS Business Intelligence Dashboard	309
<b>Chapter 19</b> △ <b>Customizing the Portal's Display</b>	<b>317</b>
Overview of Portal Customization	317
Changing the Default Preferences	318
Upgrading 9.1.2 Preferences to the 9.1.3 Preferences Format	327
Theme Deployment	328
Changing the Default Theme	332
Deleting Custom-Developed Themes	333
<b>Chapter 20</b> △ <b>Foundation Services and WebDAV Server Deployment</b>	<b>335</b>
Overview of the SAS Foundation Services That Are Used by the Portal	335
Service Deployment Configurations	336
SAS Foundation Service Deployment and Use	341
Run Remotely Deployed Services as a Windows Service	345
WebDAV Server Metadata	345
<b>Chapter 21</b> △ <b>Redistributing Portal Web Applications and Servers</b>	<b>347</b>
Overview of Redistributing Applications and Servers	347
Redistributing the SAS Services Application (and Java RMI Server)	348
Redistributing the SAS Stored Process Web Application	349
Redistributing the SAS Preferences Web Application	350
Redistributing the SAS Themes Web Application	351
Portal Configuration After Redistributing SAS Web Report Viewer	352
Portal Configuration After Redistributing SAS Web Report Studio	353
Using SAS Web Report Studio as the Default Report Viewer	353
Portal Configuration After Redistributing the SAS Metadata Server	354

**PART 6**   **Appendixes**   **357****Appendix 1** △ **Summary of the Required SAS Users and Groups**   **359**Overview of the Required SAS Users and Groups   **359**Users That Are Configured on the System   **359**Users and Groups That Are Defined in Metadata   **360****Appendix 2** △ **SAS Application Servers That Are Required for SAS Content**   **363**SAS Application Servers That Are Required for SAS Content   **363****Appendix 3** △ **Logon Formats for the Web Applications**   **365**Overview of Logon Formats   **365**Logon Formats for SAS Metadata Server Authentication   **365**Logon Format for Web (Trusted) Authentication   **366****Appendix 4** △ **Configuring the ESRI Map Component**   **369**Overview of the ESRI Map Component   **369**Software Requirements   **370**Define an ESRI Server   **371**Configure Security for the ESRI Server   **371**Define a Map Service   **372**Configure Your OLAP Cubes for ESRI Integration   **374****Appendix 5** △ **Recommended Reading**   **377**Recommended Reading   **377****Glossary**   **379****Index**   **393**





# What's New

---

## Overview

---

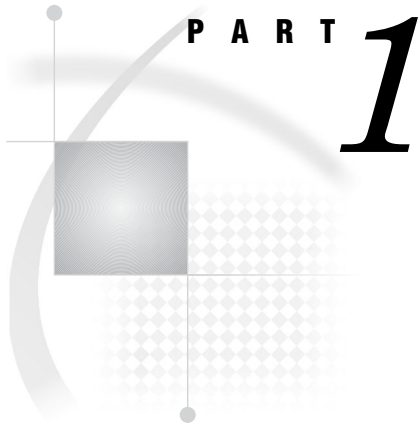
The SAS Intelligence Platform has expanded its support for the IBM WebSphere Application Server.

## Support for IBM WebSphere Application Server 6.1

---

In addition to supporting IBM WebSphere Application Server 5.1 and 6.0.2, the SAS Intelligence Platform now supports Version 6.1 of the application server. The *SAS Intelligence Platform: Web Application Administration Guide* includes information about improving the performance of WebSphere 6.1. See “Tuning WebSphere 6.0.2 or 6.1” on page 66.



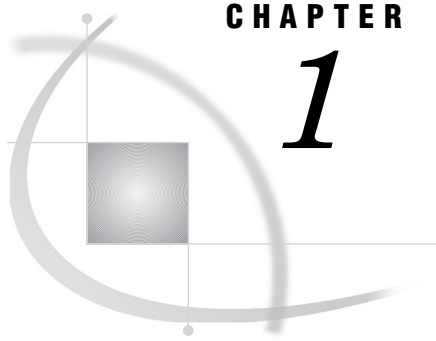


## Getting Started

*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>	4
<i>What You Should Know</i>	4
<i>What You Should Do</i>	4
<i>High-Level Overview of Administrative Tasks</i>	5

---

### 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.

The guide consolidates information that was previously located in the *SAS Intelligence Platform: Administration Guide* and the *Web Infrastructure Kit: Administrator's Guide*.

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/913administration>.

---

### 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](mailto: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 13. For a summary of the servers that are required for particular content, see “SAS Application Servers That Are Required for SAS Content” on page 363. For more details about the servers, 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. For additional documentation about the Web applications, see the following topics:
  - “Additional Documentation for SAS Web Report Studio” on page 104
  - “Additional Documentation for SAS Web OLAP Viewer for Java” on page 168
  - “Additional Documentation for the Portal” on page 207

---

### 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 configuration, as described in the *SAS Intelligence Platform: Installation and Configuration Guide*. If you are upgrading, then your initial installation should be a planned installation. Install and configure the SAS Web applications and the third-party software that they require.
- Your installation should include the standard, required SAS accounts that are described in the *SAS Intelligence Platform: Pre-installation Checklists*. These accounts are summarized in Appendix 1, “Summary of the Required SAS Users and Groups,” on page 359.
- 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 server, a servlet container, or a J2EE application server 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.
- 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 *SAS Intelligence Platform: Application Server Administration Guide*. You can also make configuration changes that are specified in Chapter 4, “Best Practices for Configuring Your Middle Tier,” on page 57.

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

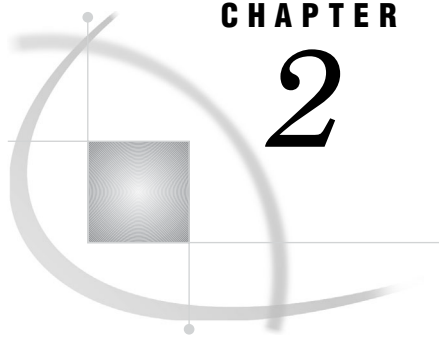
“Main Tasks for Administering the Portal Web Application” on page 196

“Main Tasks for Administering SAS Web Report Studio” on page 102

“Main Tasks for Administering SAS Web OLAP Viewer for Java” on page 168







## CHAPTER

## 2

## Working In the Middle-Tier Environment

---

<i>Understanding the Middle-Tier Environment</i>	7
<i>Overview of the Middle-tier Environment</i>	7
<i>SAS Foundation Services</i>	9
<i>Reporting Components</i>	10
<i>SAS Web Report Studio</i>	10
<i>SAS Web Report Viewer</i>	10
<i>SAS Query and Reporting Services</i>	11
<i>Portal Components</i>	11
<i>SAS Information Delivery Portal</i>	11
<i>SAS Web Infrastructure Kit</i>	11
<i>SAS Services Application / Remote Services</i>	12
<i>SAS Web OLAP Viewer for Java</i>	12
<i>WebDAV Server</i>	12
<i>Starting the Web Applications</i>	13
<i>Main Steps for Starting the Web Applications</i>	13
<i>Start-Up Order for Servers and Services</i>	13
<i>Redeploying the Web Applications</i>	14
<i>Change the HTTP Session Timeout Interval</i>	14

---

## Understanding the Middle-Tier Environment

### Overview of the Middle-tier Environment

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. These products run in a servlet container or Java 2 Enterprise Edition (J2EE) application server on the middle tier. They communicate with the user by sending data to and receiving data from the user's Web browser.

The middle-tier environment includes the following SAS software components:

- SAS Foundation Services
- Reporting components, which include the following:
  - SAS Web Report Studio
  - SAS Web Report Viewer
  - SAS Query and Reporting Services (a component of SAS Application Services)
- Portal components, which include the following:
  - SAS Information Delivery Portal, or a portal Web application that you develop using the SAS Web Infrastructure Kit

- SAS Web Infrastructure Kit
- SAS Services Application (deploys remote foundation services)
- SAS Web OLAP Viewer for Java

The middle-tier environment includes the following third-party software:

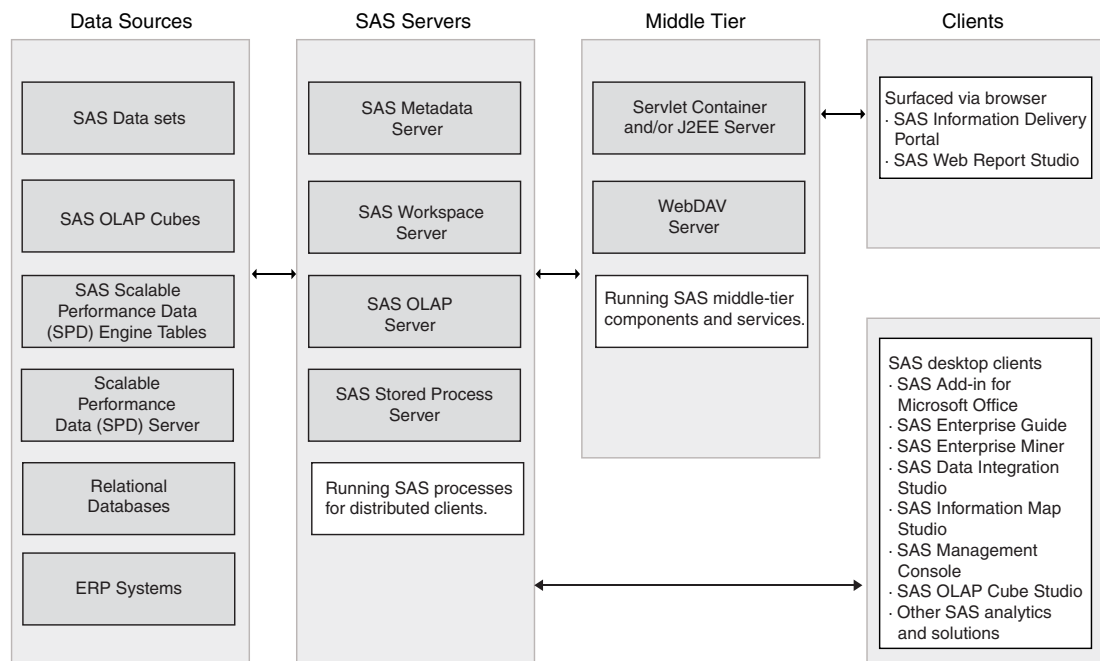
- servlet container or J2EE application server
- Java 2 Software Development Kit, Standard Edition (J2SE SDK)
- WebDAV (Web-Based Distributed Authoring and Versioning) server

For a basic description of these components, see the middle-tier section of the *SAS Intelligence Platform: Overview* document.

For information about the currently supported versions of the third-party products, see the SAS Third-Party Software Downloads page at <http://support.sas.com/thirdpartysupport>.

The following figure from the *SAS Intelligence Platform: Overview* document shows how the middle tier interacts with other tiers of the SAS Intelligence Platform.

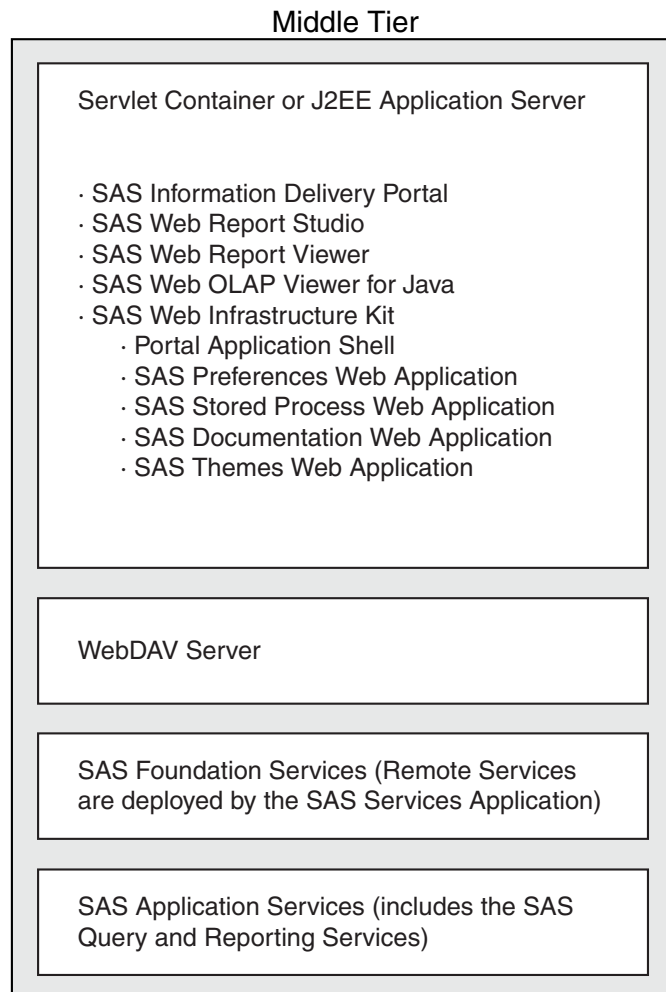
**Figure 2.1** Architecture of the SAS Intelligence Platform



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 servlet container, might be distributed on separate machines. For implementation scenarios and best practices, see Chapter 4, "Best Practices for Configuring Your Middle Tier," on page 57.

The following figure provides a more detailed view of the middle tier:

**Figure 2.2** Middle-Tier Architecture



The following sections elaborate on the information that is provided in the *SAS Intelligence Platform: Overview* document, and explain how the components operate in the middle tier.

---

## 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-based application programming interfaces provides core middleware infrastructure services that 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

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 Foundation Services. If you have correctly installed and configured the Web applications, then the foundation services will be defined in your SAS metadata repository.

You can verify this metadata in the Foundation Services Manager of SAS Management Console. Depending on which Web applications you have installed, you should see some or all of the following in SAS Management Console:

- ID Portal Local Services (used by the SAS Information Delivery Portal)
- Remote Services (used by the SAS Information Delivery Portal and any Web application that wants to achieve single sign-on with the portal)
- Query and Reporting Services (local services that are installed with SAS Web Report Studio)
- SAS Web OLAP Viewer Local Services (local services that are installed with SAS Web OLAP Viewer for Java)

If you have installed the SAS BI Web Services, then you will also have a Web Services deployment.

In addition, other applications and portlets might have deployed their own local service configurations.

The Remote Services play an important role in application federation, which enables users to access a variety of computing resources without being prompted repeatedly for their user IDs and passwords. Applications must use the Remote Services in order to achieve single sign-on with the portal. For more information about the remote services, see “SAS Services Application / Remote Services” on page 12.

## Reporting Components

### SAS Web Report Studio

SAS Web Report Studio is a query and reporting application that is specifically designed for general business users who want to view, author, and share reports on the Web.

SAS Web Report Studio runs within the servlet container, and requires the SAS Query and Reporting Services. SAS Web Report Studio does not require any of the portal components, such as the SAS Web Infrastructure Kit or the SAS Services Application. SAS Web Report Studio uses a local deployment of the SAS Foundation Services. This deployment is created during installation and configuration.

SAS Web Report Studio can be invoked from the SAS Information Delivery Portal. With additional configuration, SAS Web Report Studio can support single sign-on with the portal.

### SAS Web Report Viewer

SAS Web Report Viewer is a Web application that is used only for viewing reports. Web applications such as the SAS Information Delivery Portal use SAS Web Report Viewer to render reports.

## SAS Query and Reporting Services

SAS Query and Reporting Services, a component of the SAS Application Services, provides business-oriented query and reporting services to SAS Web Report Studio. SAS Query and Reporting Services must be deployed if you are using SAS Web Report Studio. (These services are different from the Query and Reporting Services that are described in “SAS Foundation Services” on page 9.)

---

## Portal Components

### 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.

The portal is an implementation of the SAS Web Infrastructure Kit.

### SAS Web Infrastructure Kit

The SAS Web Infrastructure Kit serves as the infrastructure for the SAS Information Delivery Portal, and must be deployed if you are using the SAS Information Delivery Portal.

The SAS Web Infrastructure Kit includes the following components:

- a SAS Portal Web Application Shell, which displays content in portlets and pages. It also provides logon and logoff capabilities, metadata searching, bookmarking, and content administration features. Developers can use this application shell to build their own portal instead of using the SAS Information Delivery Portal.

*Note:* For full portal capabilities, you must install the SAS Information Delivery Portal. For a summary of the differences between the two portals, see “Summary of Portal Features and Their Software Requirements” on page 186. △

- the SAS Stored Process Web Application, which is a Web application that enables stored processes to be run from the Web.
- the SAS Documentation Application, which is a Web application that manages SAS documentation.
- the SAS Services Application (including Remote Foundation Services), which is a Java application that manages services that are shared by SAS applications. The SAS Services Application must be running in order to use the SAS Portal Web Application Shell or the SAS Information Delivery Portal.
- the SAS Preferences and SAS Themes Web applications, which enable users to personalize their portal views.
- predefined portlets for content viewing and navigation.
- a portlet development kit, which includes an API and a set of best practices for developing custom portlets.
- administrative tools for deploying services, portlets, and additional Web applications.
- SAS Java components and Web infrastructure components.

## SAS Services Application / Remote Services

The portal also uses local and remote SAS Foundation Services. The Remote Services are registered with a remote Discovery Service. This registration enables the SAS Information Delivery Portal, the SAS Stored Process Web application, the SAS Preferences Web application, and other applications and portlets to locate and use the remotely deployed services. An application or portlet can use the Remote Services to access the portal's session context and bind to the portal's remote session service. The session context contains the status, condition, or content of the portal session.

The Remote Services are deployed by running the SAS Services Application that is included with the SAS Web Infrastructure Kit. The SAS Services Application runs in a separate Java Virtual Machine process.

For more information, see the following topics:

- Chapter 20, "Foundation Services and WebDAV Server Deployment," on page 335
- "Understanding Single Sign-On" on page 24

## SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java is a Web-based application for viewing and exploring SAS 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. You can configure SAS Web OLAP Viewer for Java to support single sign-on with the portal.

## WebDAV Server

The SAS Web applications use the WebDAV server in the following ways:

- SAS Web Report Studio uses a pre-defined storage directory structure that resides either in the file system or in a third-party WebDAV server. The directory structure parallels the arrangement of report objects in the SAS Metadata Repository. 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.
- The SAS Information Delivery Portal uses a WebDAV server to manage particular content. With the exception of reports, which can be stored on any type of WebDAV server, the portal Web application supports only Xythos WebFile Server (WFS) content (for SAS publication channels, files, and SAS Stored Process package output).

Xythos WFS is a WebDAV server that is configured by default to run in its own separate Tomcat servlet container. Xythos WFS requires a database, and can be configured with a PostgreSQL, IBM DB2, Oracle, or Microsoft SQL Server database.

For a list of features that are available depending on whether you install Xythos WFS, see "Summary of Portal Features and Their Software Requirements" on page 186.

---

## Starting the Web Applications

---

### Main Steps for Starting the Web Applications

To start the Web applications:

- 1 Start the necessary servers and services in the correct order. For the correct start-up order, see “Start-Up Order for Servers and Services” on page 13.
- 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 SAS configuration directory. The exact URL varies with the servlet container that you are using and the configuration that you have defined for your environment.

If you performed an Index installation rather than a Planned installation, then the URL for a Web application can be found in the configuration file for that application. For example, for the SAS Information Delivery Portal, you can find the URL in the **wik\_readme.html** document.

- 3 Log on to the Web application. For instructions on logging on to a Web application, refer to the online Help that is provided with the application.

For a description of logon formats based on the authentication provider that you are using, see Appendix 3, “Logon Formats for the Web Applications,” on page 365.

---

### Start-Up Order for Servers and Services

To ensure proper operation of your portal Web application implementation, if you are starting your servers and services manually, you must start your SAS Metadata Server, Xythos WFS Server, SAS Servers, remote services, and servlet container in the appropriate order.

*Note:* The following list indicates the order in which the servers should be started, but does not explain how to start the servers. For instructions on starting the servers, see the **instructions.html** document. See also “Starting, Stopping, and Pausing Servers” in the *SAS Intelligence Platform: System Administration Guide*. The start-up order is presented here only for quick reference. △

- 1 If you are authenticating against an LDAP or Microsoft Active Directory server, start the LDAP or Microsoft Active Directory server.
- 2 Start the SAS Metadata Server.
- 3 If you installed a WebDAV server, start the WebDAV server.
- 4 Start the SAS Workspace Server and SAS Stored Process Server.
- 5 If you will be running the SAS Information Delivery Portal, start the SAS Services Application. The SAS Services Application deploys the remote foundation services. The SAS Services Application must be started and initialized before you start the servlet container in order for the portal to operate correctly.

*Note:* With SAS 9.1.3 and higher, the SAS Services Application can be run as a Windows service. For details, see “Run Remotely Deployed Services as a Windows Service” on page 345. △

- 6 Start the servlet container. If the servlet container is already running, then you must restart it.

For BEA WebLogic 8.1 SP2 and SP1 on UNIX Systems, and for BEA WebLogic 8.1 SP2 on Windows Systems, the SAS Information Delivery Portal Web applications should be started in the following order:

- a SASTheme\_default
- b SASPreferences
- c SASDoc
- d Portal
- e SASStoredProcess

---

## Redeploying the Web Applications

After initial installation and configuration, if you make changes to your middle-tier configuration, then you might be instructed to redeploy the middle-tier Web applications. For example, if you change the authentication model, then you are instructed to redeploy all middle-tier applications that you have installed. All of the procedures in the documentation explicitly state when you must redeploy the middle-tier applications.

Redeploying a Web application typically involves the following tasks.

- 1 running a configuration script
- 2 deploying the WAR file that is created by the configuration script to your servlet container

Each middle-tier application has its own deployment instructions. The following table indicates where to find complete instructions.

**Table 2.1** Deployment Instructions for Middle-Tier Applications

<b>Application</b>	<b>Location of the Deployment Instructions</b>
SAS Web Report Studio	<i>SAS-install-dir\SASWebReportStudio\3.1\deployment.html</i>
SAS Web Report Viewer	<i>SAS-install-dir\SASWebReportViewer\3.1\deployment.html</i>
SAS Information Delivery Portal	<i>SAS-install-dir\Web\Portal2.0.1\wik_readme.html</i> Some configuration changes require you to re-import the foundation services that the portal uses. For details, see “Re-Create and Redeploy the Portal Web Application” on page 211.
SAS Web OLAP Viewer for Java	<i>SAS-install-dir\SASWebOlapViewerforJava\3.1\config.pdf</i>
SAS BI Web Services for Java	<i>SAS-install-dir\Web\WebServicesforJava\1.0\xmla_readme.html</i>

---

## Change the HTTP Session Timeout Interval

By default, the Web applications use the session timeout interval that is specified in your servlet container configuration. You can specify a different timeout interval by modifying one or more **web.xml**\* files. The following table lists the file or files that should be modified for each Web application. If multiple files are listed for an application, then you should modify all the files that are listed. (Your deployment might not include all of these applications. Modify the files only for the applications that you are deploying.):



**Table 2.2** Files to Modify for the Timeout Interval

<b>Web Application</b>	<b>Files to Modify</b>
SAS Information Delivery Portal	<i>SAS-install-dir</i> \Web\Portal2.0.1\Portal\WEB-INF\web.xml.orig
SAS Web OLAP Viewer for Java	<p>the following two files:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>web.xmlhost.orig</b></li> <li><input type="checkbox"/> <b>web.xmltrusted.orig</b></li> </ul> <p>in this location:</p> <p><i>SAS-install-dir</i>\SASWebOLAPViewerforJava\3.1\SASWebOLAPViewer\WEB-INF</p>
SAS Web Report Studio	<p>the following files:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>web.xml.host.tomcat</b></li> <li><input type="checkbox"/> <b>web.xml.host.weblogic</b></li> <li><input type="checkbox"/> <b>web.xml.host.websphere</b></li> <li><input type="checkbox"/> <b>web.xml.trusted.tomcat</b></li> <li><input type="checkbox"/> <b>web.xml.trusted.weblogic</b></li> <li><input type="checkbox"/> <b>web.xml.trusted.websphere</b></li> </ul> <p>in this location:</p> <p><i>SAS-install-dir</i>\SASWebReportStudio\3.1\config\Source\Java\resources</p>
SAS Web Report Viewer	<p>the following files:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>web.xml.host.tomcat</b></li> <li><input type="checkbox"/> <b>web.xml.host.weblogic</b></li> <li><input type="checkbox"/> <b>web.xml.host.websphere</b></li> <li><input type="checkbox"/> <b>web.xml.trusted.tomcat</b></li> <li><input type="checkbox"/> <b>web.xml.trusted.weblogic</b></li> <li><input type="checkbox"/> <b>web.xml.trusted.websphere</b></li> </ul> <p>in this location:</p> <p><i>SAS-install-dir</i>\SASWebReportViewer\3.1\config\Source\Java\resources</p>

To specify a session timeout interval, perform the following steps:

- 1 Modify the following code in the appropriate files:

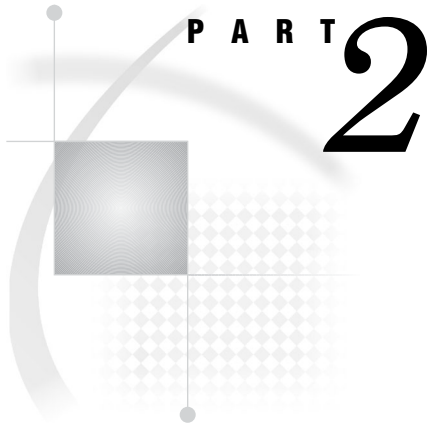
```
<session-config>
<session-timeout>timeout-interval</session-timeout>
</session-config>
```

In the previous code, *timeout-interval* specifies the timeout interval in minutes. As a recommendation, the number should be no smaller than 5.

When you are finished, save and close the file.

- 2 Redeploy the Web applications whose files you modified.
- 3 If the servlet container is running, stop and restart it.



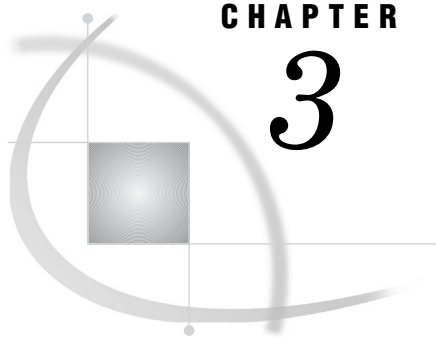


## **Middle-Tier Administration**

*Chapter 3* . . . . . **Setting Up and Managing Middle-Tier Security** 19

*Chapter 4* . . . . . **Best Practices for Configuring Your Middle Tier** 57





## CHAPTER

## 3

## Setting Up and Managing Middle-Tier Security

<i>Planning Your Middle-Tier Security Implementation</i>	20
<i>Overview: Planning Your Middle-Tier Security Implementation</i>	20
<i>Choosing an Authentication Provider</i>	20
<i>Planning User Accounts and Their Organization into Groups</i>	21
<i>Deciding Who Will Access Which Resources</i>	23
<i>Performing Additional Planning</i>	24
<i>Understanding Single Sign-On</i>	24
<i>What Is Single Sign-On?</i>	24
<i>Understanding Application Single Sign-On</i>	25
<i>About Stand-Alone Web Applications</i>	25
<i>Which SAS Web Applications Support Single Sign-On?</i>	25
<i>About Local and Remote Foundation Services</i>	25
<i>Summary of Single Sign-On</i>	26
<i>Sample Implementation</i>	26
<i>Implementation That Uses Metadata Server Authentication</i>	27
<i>Overview of Single Sign-On with Metadata Server Authentication</i>	27
<i>Requirements for Single Sign-On When Using Metadata Server Authentication</i>	28
<i>Sample Metadata Server Authentication Sequence</i>	28
<i>Implementation That Uses Trusted Web Authentication</i>	29
<i>Overview of Single Sign-On When Using Trusted Web Authentication</i>	29
<i>Requirements for Single Sign-On When Using Trusted Web Authentication</i>	30
<i>Sample Trusted Web Authentication Sequence</i>	30
<i>Changing to Trusted Web Authentication</i>	32
<i>Overview of Setting Up Web Authentication</i>	32
<i>About Web Authentication</i>	32
<i>Prerequisites for Setting Up Web Authentication</i>	32
<i>Step 1: Modify Configuration Files</i>	32
<i>Modify the Properties Files</i>	32
<i>Configure Security Constraints for the Web Realm</i>	34
<i>Implement Security Role Mapping for Your Servlet Container or J2EE Application Server</i>	36
<i>Step 2: Redeploy the Web Applications</i>	37
<i>Step 3: Update IBM WebSphere Application Server</i>	38
<i>Step 4: Add SAS Users to the Web Authentication Provider</i>	38
<i>Step 5: Modify SAS Users on the SAS Metadata Server</i>	39
<i>Step 6: Modify Xythos WebFile Server Configuration</i>	40
<i>Step 7: Restart the Servlet Container</i>	41
<i>Step 8: Define all Users</i>	41
<i>Step 9: Ensure That All Users Know How to Access the Web Applications</i>	41
<i>Configuring the Web Applications for Secure Sockets Layer (SSL)</i>	42
<i>Overview of SSL</i>	42

Set Up the SSL Environment for Your Servlet Container	42
Example for Importing Distributed Certificates	43
Add an Extra Argument for WebSphere on Solaris	43
Portal-Specific Configuration	44
About Portal-Specific Configuration	44
Step 1: Update the Themes and Preferences Metadata	44
Step 2: Update Remote Portlets for SSL	45
Step 3: Restart the SAS Services Application	45
Adding Permissions to Policy Files	45
Overview: Adding Permissions to Policy Files	45
Permissions That Are Provided by SAS	46
Overview of the SAS Policy Files	46
Policy Files with No Security Restrictions	46
Policy Files with Security Restrictions	46
Modifying the Java Policy File	47
Resources That Require Permissions in the Application's Policy File	48
Servers	48
Services (SAS Services Application)	49
Portal Content	49
Access Permissions for the Portal Components	50
Summary of Access Permissions for the Portal Components	50
CodeBase: Portal	50
CodeBase: SASPreferences	52
CodeBase: SASStoredProcess	52
CodeBase: SASServices	53
Access Permissions for Custom Portlets and Web Applications	53
About Access Permissions for Custom Portlets and Web Applications	53
CodeBase: <Remote Portlet or Web Application>	54
CodeBase: Portal	55
CodeBase: SASServices	55

---

## Planning Your Middle-Tier Security Implementation

---

### Overview: Planning Your Middle-Tier Security Implementation

Deployment of the middle tier requires careful planning in order to meet your organization's security requirements. 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.

The following sections summarize some of the security decisions that you will make. For in-depth information about these decisions, or for more information about security planning, see the *SAS Intelligence Platform: Security Administration Guide*.

---

### Choosing an Authentication Provider

Choose the authentication provider that you want to use to verify credentials that users submit.

You can choose from the following authentication providers:

- the authentication provider that the SAS Metadata Server uses. This is the default behavior. During a planned installation, the Web applications are configured to use the metadata server's authentication provider.

By default, the metadata server relies on its host operating system to authenticate users. You can configure the metadata server to use an LDAP (Lightweight Directory Access Protocol) server or a Microsoft Active Directory server instead.

- a Web server or servlet container (for trusted authentication).

You might choose to change to Web authentication if you want to take advantage of user accounts that are already established with an authentication provider for a servlet container, or if you want to minimize the number of user accounts that you have to create on the metadata server.

*Note:* If you configure Web authentication for SAS Information Delivery Portal, then you should also configure Web authentication for SAS Web Report Studio and SAS Web OLAP Viewer for Java (if they are included in your deployment). △

For a description of how authentication works, see “The Authentication Process” in the *SAS Intelligence Platform: Security Administration Guide*.

SAS Web applications can support single sign-on from one application to another. When Web applications share user context and session information, users can launch one Web application from within another Web application without having to log on to the second application. For details, see “Understanding Single Sign-On” on page 24.

*Related Tasks:*

- For instructions on configuring the metadata server to use LDAP or Active Directory authentication, see “Modifications to Support Alternative Authentication Mechanisms” in the “Customizing the Authentication Configuration” section in the *SAS Intelligence Platform: Security Administration Guide*.

The above topic in the *SAS Intelligence Platform: Security Administration Guide* also provides a general overview of the authentication providers, and offers help deciding which authentication provider to use.

- For instructions on configuring trusted Web authentication, see “Changing to Trusted Web Authentication” on page 32.

---

## Planning User Accounts and Their Organization into Groups

Determine the types of users who will access the Web applications, and how those users can be grouped into logical or organizational units. Then you can define groups in SAS metadata, define the users, and add the users to the appropriate groups.

The following steps can help you plan the users and groups for your deployment:

- 1 Decide which users will log on to the Web applications. These users will require an account on the authentication provider.
- 2 Determine whether the users also need an identity in the metadata repository. Consider the following:
  - By default, all users who can access the metadata server (PUBLIC users) can log on to SAS Web Report Studio and manipulate reports. After you set up security by assigning roles, then PUBLIC users can only view reports. In order to manipulate reports, users require a metadata identity and association with a role.
  - SAS Information Delivery Portal users must have a metadata identity in order to log on. PUBLIC users can access only the Public Kiosk.

- SAS Web OLAP Viewer for Java users must have a metadata identity in order to log on.
- 3 In addition to accessing the SAS Metadata Server, users often require access to one or more of the servers that are listed in this table:

**Table 3.1** Additional Authentication for Web Application Users

Server	Interaction
SAS Workspace Server	Access resources such as tables.
SAS OLAP Server	Access cube data and process MDX queries.
SAS Stored Process Server	Execute stored processes and collect resulting output.

*Note:* For a summary of the servers that are required for particular content, see “SAS Application Servers That Are Required for SAS Content” on page 363.  $\Delta$

The authentication processes and requirements for these servers are documented in detail in “Understanding Authentication in the SAS Intelligence Platform” in the *SAS Intelligence Platform: Security Administration Guide*. Here are key points to consider for a default configuration:

- In the simplest case, all of the SAS servers use the same host authentication provider. For example, in a single-machine deployment with a default configuration, a user needs only a local (or network) account in the operating system. Similarly, in a multi-machine deployment where all servers use the same host authentication provider, a user needs only a network account with that host authentication provider.
- In a more diverse environment, additional accounts, logins, and authentication domains are required. For example, if your stored process server is running on UNIX and your other servers are using Windows host authentication, then each user also needs an (individual or shared) account on the UNIX server and an additional (individual or group) login in the metadata. The additional login must include the credentials for the UNIX account. The additional login must be associated with the stored process server’s authentication domain.

Here are the main steps for planning additional credentials:

- a Identify the accounts and metadata logins you need to create to enable users to access additional servers.
  - b Decide whether you want to manage those additional credentials for each user individually, or whether you prefer to manage user credentials as shared accounts.
  - c Determine which authentication domains will be associated with the additional credentials.
- 4 Determine which groups you want to define in metadata, and which users should belong to those groups. You can manage security efficiently when you organize users into groups. When you add users to a group, you can give those users access to particular content that is restricted to other groups.

When you add users to groups, you can do the following:

- Add a user as a member of more than one group: You might find that the authorization (access) requirements of a group of users are not necessarily identical. In these cases, you can assign a user to more than one group to accommodate unique needs.



- Add a group as a member of another group: You might find that a larger group might have smaller groups as members. For example, a group of worldwide sales users might contain a group of regional sales users.

*Related Tasks:*

- 1 Create user accounts on the authentication provider that you are using.
- 2 Use SAS Management Console to create user and group metadata identities, including one or more logins for the users or groups.

For more information about creating users and groups, see “User and Group Management” in the *SAS Intelligence Platform: Security Administration Guide*.

- 3 If your deployment includes SAS Web Report Studio, then you can associate your users and groups with user roles. See “Using SAS Web Report Studio Roles” on page 130.

---

## Deciding Who Will Access Which Resources

Decide which users and groups will have access to which data and resources.

When a user requests access to a resource, an authorization decision is made based on an evaluation of all of the relevant access controls. In order to implement access controls (authorization), make the following decisions:

- Decide which permissions you want to assign to specific users and groups.

By default, when you first install the software, all members of the PUBLIC group have administrator permissions. (All users who can access the metadata server are members of the PUBLIC group.) To implement security, assign specific permissions to particular groups, and then restrict the permissions for the PUBLIC group.

- Decide which resources you want to protect with permissions set directly on the resource.

For example, if the SAS Information Delivery Portal displays SAS reports that contain employee salary information, you will want to limit access to those reports. If your deployment includes the SAS Information Delivery Portal, then you will set up authorization in metadata for the content that you make available to the portal. The method that you use to control access varies with the type of content. For a summary of content types and their respective access control methods, see “Summary of Content That Can Be Added to the Portal” on page 240.

For more information, see “Understanding Authorization” in the *SAS Intelligence Platform: Security Administration Guide*.

Finally, you should plan to apply Java security to the SAS Web applications, as well as to any Web applications or portlets that you develop and deploy. By default, the SAS Web applications have no security restrictions.

*Related Tasks:*

- Assign permissions to specific users and groups. For more information, see “Using the Metadata Authorization Layer” in the *SAS Intelligence Platform: Security Administration Guide*.
- Set permissions on the resources that you want to protect. For more information, see “Access Guidelines and Requirements” in the *SAS Intelligence Platform: Security Administration Guide*.
- For information about setting up Java security for the Web applications, see “Adding Permissions to Policy Files” on page 45.

---

## Performing Additional Planning

You might want to enable encryption by using SAS/SECURE software, or set operating system permissions in order to protect the configuration directories. These tasks are described in the *SAS Intelligence Platform: Security Administration Guide*.

You might also want to use Secure Sockets Layer (SSL) in order to provide network security, as described in “Configuring the Web Applications for Secure Sockets Layer (SSL)” on page 42.

---

## Understanding Single Sign-On

---

### What Is Single Sign-On?

*Single sign-on* is an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords.

The term "single sign-on" is used in SAS documentation to mean one of the following:

- SAS proprietary server single sign-on:

After users have been authenticated for an application, they can access workspace, stored process, and OLAP servers that run on different platforms, without being prompted for their user IDs and passwords. To accomplish SAS proprietary server single sign-on, you might need to configure additional credentials for users. For more information, see “Planning User Accounts and Their Organization into Groups” on page 21.

- Application single sign-on (Web applications only):

After users have been authenticated for an application, they can access other applications without being prompted for their user IDs and passwords. Only Web applications can participate in this type of single sign-on; desktop applications are not currently supported.

There are two types of application sign-on. Depending on the authentication provider that is configured for your environment, you can implement either of the following:

- Single sign-on that relies on SAS Metadata Server authentication: Participating Web applications use the authentication provider that the SAS Metadata Server uses. By default, the authentication provider is the host system, but you can configure the metadata server to use LDAP or Microsoft Active Directory. To accomplish single sign-on for SAS proprietary applications, the applications share metadata session information about the user through a deployment of the remote foundation services.

*Note:* To deploy the remote foundation services, you must start the SAS Services application. △

- Single sign-on that relies on trusted Web authentication: Participating Web applications use trusted Web authentication. To accomplish single sign-on, the applications obtain the user’s ID from the HTTP header that is included with the HTTP request. Typically, an application retrieves the ID through a servlet API call: `getRemoteUser()`.

*Notes:*

- This document describes only application single sign-on. It describes single sign-on that relies on trusted Web authentication and sign-on that relies on SAS Metadata Server authentication.
- Regardless of the authentication provider that you use, you must still configure authorization in order to safeguard sensitive data and resources. For an overview of authorization, see “Deciding Who Will Access Which Resources” on page 23.
- Once you have enabled single sign-on among Web applications, you must still provide credentials as needed so that users can access the SAS workspace, stored process, and OLAP servers.

---

## Understanding Application Single Sign-On

### About Stand-Alone Web Applications

A stand-alone Web application can be configured to participate in a single sign-on configuration with the portal Web application. With regard to the portal, a *stand-alone application* is any Web application that is not referenced in the **PortalContent.xml** file (located in the *SAS-install-dir/Web/Portal2.0.1/Portal/WEB-INF/content* directory). Because the application is not referenced in that file, the portal does not know how to invoke the application.

Any custom Web application that you develop is considered a stand-alone application. Before a stand-alone application can be launched from the portal Web application, the stand-alone application must be added to the portal environment. (For instructions, see “Adding Web Applications” on page 276).

For example, SAS Web Report Studio is not referenced in **PortalContent.xml**, so it must be added to the portal environment in order to be launched from the portal. However, SAS Web Report Viewer is referenced in **PortalContent.xml** along with an access point (“/SASWebReportViewer/logonFromPortal.do”). Therefore, SAS Web Report Viewer can be launched from the portal with no additional configuration.

### Which SAS Web Applications Support Single Sign-On?

The following SAS Web applications support single sign-on among applications:

- The portal Web application
- SAS Web Report Studio
- SAS Web Report Viewer
- SAS Web OLAP Viewer for Java
- SAS Stored Process Web application
- SAS Preferences Web application

With the exception of SAS Web Report Viewer, these applications are considered stand-alone applications.

For introductory information about these applications, see “Understanding the Middle-Tier Environment” on page 7.

### About Local and Remote Foundation Services

If you have correctly installed and configured the portal Web application, then the local and remote foundation services will be defined in your metadata repository. For an overview of these services, see “SAS Foundation Services” on page 9.

The Remote Services play an important role in application single sign-on. Web applications share metadata session information about the user through a deployment of the Remote Services, which are deployed by the SAS Services Application.

For related topics, see the following:

- Chapter 20, “Foundation Services and WebDAV Server Deployment,” on page 335
- For details about service deployments, see “Understanding Service Deployments” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/platserv/ps\\_servdep.html](http://support.sas.com/rnd/itech/doc9/admin_oma/platserv/ps_servdep.html).
- To learn how to use the foundation services in your own custom Web applications and portlets, see “Using SAS Foundation Services With the Portal” in the *SAS Web Infrastructure Kit: Developer’s Guide*, at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/webapps/dg\\_found.html](http://support.sas.com/rnd/itech/doc9/portal_dev/webapps/dg_found.html). See also the SAS Foundation Services class documentation at <http://support.sas.com/rnd/gendoc/bi/api/Foundation/overview-summary.html>.

## Summary of Single Sign-On

As noted previously, single sign-on can be achieved whether you are using trusted Web authentication or SAS Metadata Server authentication. In order for stand-alone Web applications to participate in a single sign-on configuration, the applications must share a common authentication provider and user context. The user context varies between Web authentication and SAS Metadata Server authentication.

The following table summarizes the similarities and differences between a single sign-on configuration that relies on SAS Metadata Server authentication, and one that relies on trusted Web authentication. The remainder of this documentation explains the summarized points in more detail. The points are summarized here for quick reference:

**Table 3.2** Single Sign-On Summary Based on Authentication Provider

SAS Metadata Server Authentication*	Trusted Web Authentication
Web applications must be launched from a participating application that has authenticated the user. For example, if a user logs on to the portal Web application, then the user can access SAS Web Report Studio without an additional logon only if SAS Web Report Studio is launched from the portal Web application.	Web applications can be launched independently of any other participating application. For example, if a user logs on to the portal Web application, then the user can launch SAS Web Report Studio from the Windows Start menu without being prompted for an additional logon.
The application that is launched obtains the user context from the shared remote services that are deployed by the SAS Services application. The <i>user context</i> contains the user’s active repository connections, identities, and profile.	The application that is launched obtains the user identity from the HTTP session. The section “Implementation That Uses Trusted Web Authentication” on page 29 explores some of the ways that applications can accomplish this.

Regardless of the authentication provider that you use, any stand-alone application that is launched from the portal Web application must be added to the portal environment. For instructions, see “Adding Web Applications” on page 276.

## Sample Implementation

For a sample implementation that uses SAS Metadata Server authentication, see “Sample: Web Application (HelloUserWikExample)” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/samples/webapp/dg\\_sample\\_webapp.html](http://support.sas.com/rnd/itech/doc9/portal_dev/samples/webapp/dg_sample_webapp.html).

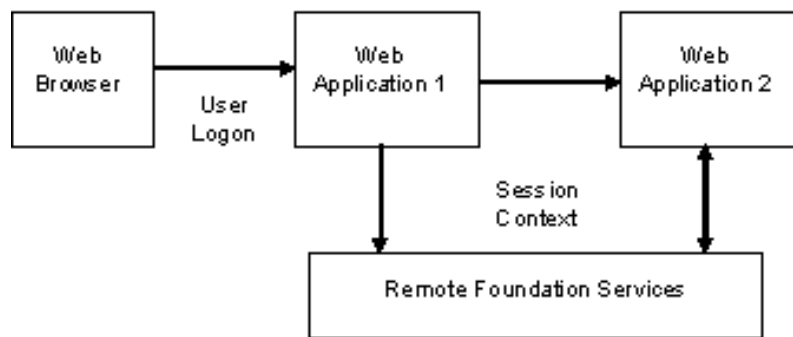
## Implementation That Uses Metadata Server Authentication

### Overview of Single Sign-On with Metadata Server Authentication

When you use metadata server authentication, Web applications must be launched from a participating application that has already authenticated the user. By default, the participating Web applications rely on the host system of the metadata server in order to authenticate users.

Here is a simplified illustration of single sign-on in an environment that uses metadata server authentication:

**Figure 3.1** Single Sign-On with Metadata Server Authentication



The figure depicts the following high-level sequence:

- 1 After a user logs on to Web Application 1, that application uses the remote services API to send the session context to the remote foundation services. The *session context* 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.
- 2 From within Web Application 1, the user launches Web Application 2. Web Application 1 passes the metadata session identifier to Web Application 2.
- 3 Web Application 2 uses the session identifier to obtain user information from SAS metadata. Web Application 2 first queries to determine if the Session ID is valid. If so, then User Context information is retrieved to create the local User Context in the Local Services for Web Application 2.

For a more detailed example, see "Sample Metadata Server Authentication Sequence" on page 28.

You can implement this scenario in order to do the following:

- launch your custom Web application from a collection portlet in the portal Web application
- launch a stored process or a report from the search results of the portal Web application
- launch a stored process or a report from your custom Web application
- launch the portal Web application from your custom Web application

Web Applications 1 and 2 can reside in different servlet containers, but must share the same remote foundation services and metadata repository. The remote foundation services are started by the SAS Services Application.

## Requirements for Single Sign-On When Using Metadata Server Authentication

Here are the requirements to implement single sign-on when you have configured metadata server authentication:

- All of the participating Web applications that require users to log on must use metadata server authentication. For your own custom Web applications, consult the responsible developers in your organization to verify that the applications support metadata server authentication.
- All of the participating Web applications must support single sign-on. This means that they must use the Foundation Services API to manage user and session information. Specifically, these things are required:
  - A local foundation services deployment configuration for the purpose of establishing connections, creating sessions, authenticating users, and logging messages.
  - API calls to the shared remote services for the purpose of obtaining the current user's context. (A privileged user is required to do this. The remote services use the privileged user to connect with the SAS Metadata Server.)
- Any stand-alone application that is launched from the portal Web application must be added to the portal environment.

## Sample Metadata Server Authentication Sequence

Here is a sample sequence of events for an environment that uses metadata server authentication. The sequence illustrates launching SAS Web Report Viewer from the portal Web application:

- 1 The SAS Services Application deploys the remote foundation services that participating Web applications must use. At startup, the portal Web application connects to the services.
- 2 A user logs on to the portal Web application and is authenticated via the metadata server.
- 3 Once the user is authenticated, the portal Web application uses the remote services to create a remote session context for the user. The portal uses the `SessionContextInterface` to create and persist a unique session key that can be shared with other applications. The portal then locks the session context. The lock persists for as long as the user is logged on.
- 4 The user selects a content item, and the portal Web application launches a Web application to view the content. For this example, the user searches for and selects a report that is displayed in SAS Web Report Viewer.
- 5 The request parameters that are sent to SAS Web Report Viewer include the logical `SessionContextInterface`. Using this interface, SAS Web Report Viewer invokes the remote Session service API to obtain the portal's current session context. SAS Web Report Viewer uses a privileged user identity in order to obtain this session context. (SAS Web Report Viewer uses `sastrust`. For your custom applications, you can use `sastrust`, `saswbadm`, or create a different user for this purpose. The requirement is that the privileged identity be a member of the SAS System Services group in metadata. )

- 6 Once SAS Web Report Viewer has obtained the remote session context, SAS Web Report Viewer locks the session context for its own use.
- 7 SAS Web Report Viewer instantiates a local session from the remote session context. From this local session, SAS Web Report Viewer uses the user context to obtain credentials for application server access (SAS Web Report Viewer code must access a SAS Workspace Server).

*Note:* SAS Web Report Viewer only needs to obtain credentials for this server if the server is not pooled.  $\Delta$

- 8 SAS Web Report Viewer is displayed with the user's context without prompting the user for authentication. In this example, SAS Web Report Viewer opens and displays the report. For your own custom Web applications, if the application is called by a portlet, then it should generate an HTML fragment for display within the portlet.
- 9 If the user logs off the portal Web application, the session is unlocked by the portal, but the lock remains in SAS Web Report Viewer. The last application to unlock the user session is responsible for releasing the global session. The session context will be deleted by the Remote Session service when all locks have been released.

---

## Implementation That Uses Trusted Web Authentication

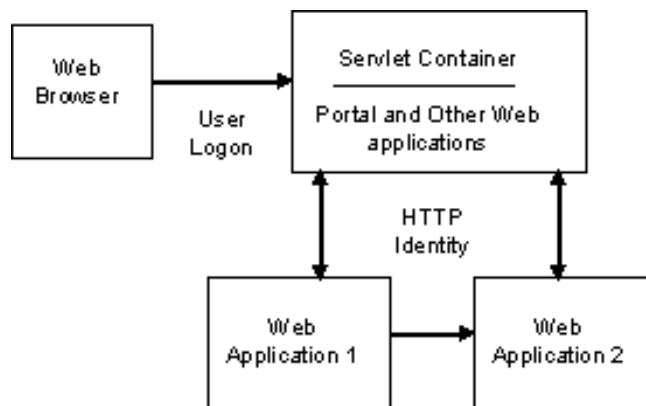
### Overview of Single Sign-On When Using Trusted Web Authentication

When you use trusted Web authentication, the J2EE application server, servlet container, or Web server is responsible for authenticating users. Web authentication can occur using one of the standard authentication mechanisms (Basic, Digest, and so on). After authentication, user credentials are persisted in the form of header name-value pairs that are included with each HTTP request.

The application that is launched must be able to access the remote user identity in the HTTP session. Typically, the application uses the container's API call `getRemoteUser()` to obtain a user identity. Alternatively, the application could parse the HTTP request header to obtain the credentials. There are multiple ways to implement single sign-on, and you should carefully choose the implementation that is best for your environment. For complete details, refer to the documentation that is provided by your servlet container or Web server.

Here is a simplified illustration of single sign-on in an environment that relies on a servlet container for authentication:

**Figure 3.2** Single Sign-On With Trusted Web Authentication



The figure depicts the following high-level sequence:

- 1 The user launches Web Application 1, and the servlet container authenticates the user.
- 2 The user launches Web Application 2. Web Application 2 obtains the authenticated identity of the user from the Web server. Web Application 2 can be launched separately, or it can be launched from Web Application 1.

For a detailed example, see “Sample Trusted Web Authentication Sequence” on page 30.

You can implement this scenario in order to do the following:

- launch any series of SAS web applications in consecutive order, such as the portal Web application, followed by SAS Web Report Studio, and then SAS Web OLAP Viewer for Java
- launch your custom Web application from a collection portlet in the portal Web application
- launch a stored process or a report from the search results of the portal Web application
- launch a stored process or a report from your custom Web application
- launch the portal Web application from your custom Web application

## Requirements for Single Sign-On When Using Trusted Web Authentication

These are the requirements to implement single sign-on when you have configured trusted Web authentication:

- All of the participating Web applications must support single sign-on. See “Which SAS Web Applications Support Single Sign-On?” on page 25. For your own custom Web applications, consult the responsible developers in your organization to verify that the applications support single sign-on.
- All of the participating Web applications that require users to log on must use trusted Web authentication. For your own custom Web applications, consult the responsible developers in your organization to verify that the applications support trusted Web authentication.
- All of the Web applications must either be deployed in the same servlet container, deployed in a cluster of containers, or served by the same Web server. When it authenticates users, the Web server or servlet container uses an HTTP authorization header that the browser supplies. The Web server or servlet container must be able to verify the user credentials that are passed in the header.
- If you use WebLogic, WebSphere, or Tomcat for authentication, then all of the Web applications must be in the same realm. A *realm* is a configuration mechanism that enables you to identify which portions of your site are accessible and which portions are restricted to some or all users. The realm configuration specifies the authentication provider and the criteria or user role for authorized access to the web application. The application’s configuration file (for example, web.xml for the portal Web application) associates the application with a realm, and optionally identifies one or more authorized groups (user roles) that are defined in the realm and that are authorized to access the Web application.
- Any stand-alone application that is launched from the portal Web application must be added to the portal environment.

## Sample Trusted Web Authentication Sequence

Here is a sample sequence of events for an environment that uses trusted Web authentication. The sequence illustrates launching SAS Web Report Studio from the portal Web application:



- 1 The SAS Services Application deploys the remote foundation services that participating Web applications must use. At startup, the portal Web application connects to the services.
- 2 When a user accesses the portal Web application, the user is prompted to log on to the Web server or the servlet container.
- 3 The Web server or servlet container verifies the user ID and password either by performing its own verification check or by forwarding the user credentials to an authentication provider, such as an LDAP server, which authenticates the user.
- 4 The authenticated user ID is returned back to the Web server or servlet container, which stores the ID in the HTTP session header. If you are using a J2EE application server or servlet container for authentication, the application's web.xml file associates the application with a realm, and optionally with one or more authorized groups that are defined in the realm. If the user is a member of any groups that are defined in the web.xml file, then the user is allowed to access the application. If you use an Web server for authentication, then you will not configure an XML file. The Web server authenticates the user and then passes the authenticated user name and password to the servlet container.
- 5 The portal obtains the user identity by calling `getRemoteUser()`. In order to verify the user's authorization to access portal content, the portal Web application uses the SAS Trusted User (sastrust) to establish a connection to the metadata server on behalf of the user. The trusted user connection is used to obtain a one-time use password from the metadata server. This password, along with the user ID obtained by calling `getRemoteUser()`, is used to establish a connection for the user to the metadata server. The metadata server locates the user's metadata identity. Based on the user's metadata identity, the portal Web application displays the user's portal pages.
- 6 Once the user identity is established, the portal requests a user context from its local User Service and writes this context to the remote User Service. The portal uses the remote service's API to do this. The user context will be used to access various SAS application servers. A portal user would access a SAS Workspace, SAS Stored Process, or SAS OLAP Server in order to run SAS programs and stored processes, or to render OLAP data.

*Note:* The user might require additional credentials in order to access these servers from the portal. For more information, see "Planning User Accounts and Their Organization into Groups" on page 21.  $\Delta$
- 7 When the user selects a content item, the portal launches a viewer application, which can be displayed either within the portal or outside the portal. In this example, the user selects SAS Web Report Studio, and the portal invokes the corresponding URL.
- 8 Because the servlet container already has the user's logon credentials, the user is not prompted to log on again. SAS Web Report Studio obtains the user's authenticated identity from the servlet container or Web server using the `getRemoteUser()` method. SAS Web Report Studio relies on the servlet container to parse the request HTTP request and supply information about the user.
- 9 SAS Web Report Studio makes a call to the remote services, and requests the user context that was returned by the `getRemoteUser()` method. SAS Web Report Studio locks the context, makes a local copy for its own use, and releases the lock. Credentials that are associated with the user context are used to connect to SAS application servers.

---

# Changing to Trusted Web Authentication

---

## Overview of Setting Up Web Authentication

### About Web Authentication

As an alternative to using the host operating system accounts to authenticate users, you can configure trusted Web authentication. When the Web applications use Web authentication, the Web applications obtain authentication information from a Web server or a servlet container, and set up a trust relationship with the SAS Metadata Server.

To authenticate users, you can use a servlet container or a J2EE application server. You can also use an HTTP server front end to authenticate users, and then forward the authenticated user credentials to the servlet container. For information about the supported servers, see <http://support.sas.com/documentation/configuration/thirdpartysupport/index.html>.

This topic describes the configuration settings that are required to set up the trust relationship between the Web authentication provider and the SAS Metadata Server. For information about the planning phase, see “Planning Your Middle-Tier Security Implementation” on page 20.

*Note:* Throughout this topic, discussion of Web authentication can be assumed to mean authentication by a servlet container, J2EE application server, or HTTP server. Any distinctions are explicitly mentioned and explained where they occur.  $\triangle$

### Prerequisites for Setting Up Web Authentication

Before you set up Web authentication, you should do the following:

- Verify that host authentication operates correctly in your environment. If host authentication does not work correctly in your environment, then Web authentication probably won't work correctly either.

One way to verify authentication is to log on to the Web applications and perform some tasks. Additionally, if you are using the SAS application and data servers, make sure that you can access those servers. For example, you might log on to the portal and search for and execute a stored process.

- If you plan to set up Web authentication for SAS Web Report Studio, then you must specify a surrogate metadata identity for public-only users. See “Designate a Surrogate Metadata Identity” on page 129.
- If you plan to use an HTTP server to authenticate users, then you must install and configure the HTTP server. Refer to the documentation for your HTTP server product.

The following sections describe the steps for configuring trusted Web authentication.

---

## Step 1: Modify Configuration Files

### Modify the Properties Files

Edit particular properties files for the Web applications as shown in the following table.

*Global Notes:*

- Before you modify any configuration file, it is recommended that you make a backup copy of the file first.
- Your deployment might not include all of the Web applications that are listed in these instructions. Modify the files only for the Web applications that you have installed.

**Table 3.3** Modifications to the Properties Files

Application, File to be Modified	Modifications
SAS Information Delivery Portal <i>SAS-install-dir</i> \ Web\Portal2.0.1\PortalConfigure\ install.properties	Set the properties as follows: <b>\$USER_DOMAIN\$=web</b> <b>\$AUTH_MECHANISM\$=trusted</b> <b>\$DAV_DOMAIN\$=web</b> <b>\$PORTAL_AUTH_MODULE\$=com.sas.portal.delegates.authentication.factory.</b> <b>BasicAuthentication</b> <b>\$SERVICES_WEB_DOMAIN\$=web</b> <b>\$SERVICES_WEB_PRIVILEGED_USER_ID\$=&lt;domain&gt;*\sastrust</b> <b>\$SERVICES_WEB_PRIVILEGED_USER_PASSWORD\$=&lt;sastrust password&gt;**</b>  In addition, for Windows hosts only, remove the domain qualifier from the following user ID values. (The qualifiers are required only for host authentication.) Here are examples of how they might appear after the change: <b>\$PORTAL_GUEST_ID\$=sasguest</b> <b>\$PORTAL_ADMIN_ID\$=saswbadm</b> <b>\$PORTAL_DEMO_ID\$=sasdemo</b>
SAS Web OLAP Viewer for Java <i>SAS-install-dir</i> \ <b>SASWebOlapViewerforJava\3.1\Configure\install.properties</b>	Set the properties as follows: <b>\$LOGON_DOMAIN\$=web</b> <b>\$AUTH_MECHANISM\$=trusted</b> <b>\$TRUSTED_USER_ID\$=&lt;domain&gt;\sastrust</b> <b>\$TRUSTED_USER_PASSWORD\$=&lt;sastrust password&gt;</b>
SAS Web Report Studio <i>SAS-install-dir</i> \ <b>SASWebReportStudio\3.1\wrs.config</b>	Set the properties as follows: <b>\$LOGON_DOMAIN\$=web</b> <b>\$AUTH_MECHANISM\$=trusted</b> <b>\$WEB_ADMIN_ID\$=saswbadm</b> <b>\$SERVICES_OMI_USER_ID\$=&lt;domain&gt;\sastrust</b> <b>\$SERVICES_OMI_USER_PASSWORD\$=&lt;password&gt;</b>  The value that you enter for <b>\$SERVICES_OMI_USER_ID\$</b> must match exactly (including capitalization) the login value that you have defined in metadata for the SAS Trusted User.

Application, File to be Modified	Modifications
SAS Web Report Viewer <i>SAS-install-dir</i> \ SASWebReportViewer\3.1\ wrv.config	Set the properties as follows: \$LOGON_DOMAIN\$=web \$AUTH_MECHANISM\$=trusted \$WEB_ADMIN_ID\$=saswbadm \$SERVICES_OMI_USER_ID\$=<domain>\sastrust \$SERVICES_OMI_USER_PASSWORD\$=<password>  The value that you enter for \$SERVICES_OMI_USER_ID\$ must match exactly (including capitalization) the login value that you have defined in metadata for the SAS Trusted User.
SAS BI Web Services for Java <i>SAS-install-dir</i> \ Web\WebServicesforJava\1.0\ Configure\install.properties	Set the properties as follows: \$AUTH_MECHANISM\$=trusted \$SERVICES_WEB_DOMAIN\$=web \$SERVICES_WEB_PRIVILEGED_USER_ID\$=sastrust \$SERVICES_WEB_PRIVILEGED_USER_PASSWORD\$= <sastrust password>
* Where the user name includes a <domain>, the domain refers to the host or domain qualifier that is required on a Windows system.	
** Where the property is a password, the password can be in clear text. However, for security reasons, you should enter a password that you have encoded using SAS. To obtain this encoded password, use PROC PWENCODE.	

*Note:* If any of the properties is not included in your properties file, then add the property and its value to the file. If you copy and paste any values from this document, then be sure to remove any unwanted spaces or characters. △

## Configure Security Constraints for the Web Realm

If you are using a servlet container or a J2EE application server for trusted authentication (as opposed to an HTTP server), then you must edit one or more XML files as described in this step.

- 1 Edit the portal's **web.xml.orig** file (located in *SAS-install-dir*\Web\Portal2.0.1\Portal\WEB-INF).

Comment the following **<error-page>** block by adding the begin- and end-comment elements "**<!--**" and "**-->**":

The commented block should look like this:

```
<!--
<error-page>
<error-code>401</error-code>
<location>/</location>
</error-page>
-->
```

- 2 In the Web XML files that are listed in the following table, make these changes:
  - a Uncomment the following block by removing the begin- and end-comment elements "**<!--**" and "**-->**":

```
<!--
<security-constraint>
<web-resource-collection>
<web-resource-name>Success</web-resource-name>
<url-pattern>/*</url-pattern>
```

```

<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>

<auth-constraint>
<role-name>webuserRole
</role-name>
</auth-constraint>
</security-constraint>

<login-config>
<auth-method>BASIC</auth-method>
<realm-name>default</realm-name>
</login-config>

<security-role>
<role-name>webuserRole</role-name>
</security-role>
-->

```

- b Edit the **role-name** and **realm-name** values if needed for your deployment.

The **role-name** value should match the identity group name that you have set on your authentication provider. The **role-name** value should be unique to the realm configuration that is defined for your servlet container. Your **role-name** value might differ from the one shown here. If you change the **role-name** value, make sure that your value is identical for the **<auth-constraint>** and **<security-role>** sections. In addition, make sure the value is the same in all files that you modify.

The **realm-name** value is displayed in the dialog box that portal users see when they are prompted to log on.

Make these changes in the files that are listed in the following table:

**Table 3.4** Files in Which You Uncomment the Security Block

Application	File to Modify
SAS Information Delivery Portal	<i>SAS-install-dir</i> \Web\Portal2.0.1\Portal\WEB-INF\web.xml.orig
SAS Stored Process Web Application	<i>SAS-install-dir</i> \Web\Portal2.0.1\SASStoredProcess\WEB-INF\web.xml.orig
SAS Web OLAP Viewer for Java	<i>SAS-install-dir</i> \SASWebOlapViewerforJava\3.1\SASWebOLAPviewer\WEB-INF\web.xmltrusted.orig
SAS Web Report Studio	<i>SAS-install-dir</i> \SASWebReportStudio\3.1\config\Source\Java\resources Edit one of the following, depending on your servlet container. web.xml.trusted.tomcat web.xml.trusted.weblogic web.xml.trusted.websphere

Application	File to Modify
SAS Web Report Viewer	<p><i>SAS-install-dir\SASWebReportViewer\3.1\config\Source\Java\resources</i></p> <p>Edit one of the following, depending on your servlet container.</p> <p><b>web.xml.trusted.tomcat</b></p> <p><b>web.xml.trusted.weblogic</b></p> <p><b>web.xml.trusted.websphere</b></p>
SAS BI Web Services for Java	<p><i>SAS-install-dir\Web\WebServicesforJava\1.0\SASXMLA\WEB-INF\web_trusted.xml.orig</i></p>

The security block might already be uncommented in some files. However, you may still need to edit the files if you are changing the role-name values.

## Implement Security Role Mapping for Your Servlet Container or J2EE Application Server

### *BEA WebLogic*

If you are using WebLogic for authentication, then you must edit the WebLogic XML file as follows:

- 1 Uncomment the following block by removing the begin- and end-comment elements "<!--" and "-->":

```
<!--
  <security-role-assignment>
    <role-name>webuserRole</role-name>
    <principal-name>webuserGroup</principal-name>
  </security-role-assignment>
-->
```

- 2 Change the role-name and principal-name values as needed for your deployment.

The principal-name corresponds to the WebLogic group that is associated with the specified WebLogic role (role-name). The default values are **webuserRole** and **webuserGroup**.

Make these changes in the files that are listed in the following table:

**Table 3.5** WebLogic Files That You Modify

Application	File to Modify
SAS Information Delivery Portal	<i>SAS-install-dir\Web\Portal2.0.1\Portal\WEB-INF\weblogic.xml</i>
SAS Stored Process Web Application	<i>SAS-install-dir\Web\Portal2.0.1\SASStoredProcess\WEB-INF\weblogic.xml</i>
SAS Web OLAP Viewer for Java	<i>SAS-install-dir\SASWebOlapViewerforJava\3.1\SASWebOLAPViewer\WEB-INF\weblogic.xmltrusted.orig</i>
SAS Web Report Studio	<i>SAS-install-dir\SASWebReportStudio\3.1\config\conf\weblogic.xml.orig</i>

Application	File to Modify
SAS Web Report Viewer	<i>SAS-install-dir</i> \SASWebReportViewer\ 3.1\config\conf\weblogic.xml.orig
SAS BI Web Services for Java	<i>SAS-install-dir</i> \Web\Portal2.0.1\ SASServicesConfig\WEB- INF\weblogic_trusted.xml.orig

The security role assignment block might already be uncommented in some files. However, you may still need to change the role-name and principal-name values in these files.

#### *IBM WebSphere*

If you are using WebSphere for authentication, then in the WebSphere administration console, configure role mapping for the role name that is being used for your deployment (**webuserRole** by default). See the WebSphere documentation for instructions.

#### *Apache Tomcat*

If you are using Tomcat for authentication, then modify the **tomcat-users.xml** file. See the Tomcat documentation for instructions.

## Step 2: Redeploy the Web Applications

After you modify a Web application's properties file, as described in the previous step, you must redeploy the application. To redeploy, follow these steps:

- 1 Re-create and redeploy the SAS Information Delivery Portal. Here is a summary of the steps:
  - a Run the portal's **configure\_wik** utility.
  - b Deploy the **Portal.war** and **SASStoredProcess.war** files to your servlet container.
  - c In the Foundation Services Manager of SAS Management Console, delete and re-import the local and remote foundation services. For details, see "Reimport the Service Deployment Configurations" on page 340.

For full instructions, see "Re-Create and Redeploy the Portal Web Application" on page 211.

- 2 Re-create and redeploy SAS Web OLAP Viewer for Java. Here is a summary of the steps:
  - a Run the SAS Web OLAP Viewer for Java **configure** utility.
  - b Deploy SAS Web OLAP Viewer for Java to your servlet container.
  - c Delete and re-import the SAS Web OLAP Viewer for Java services. The services are located at: *SAS-install-dir*\SASWebOlapViewerforJava\  
3.1\SASServicesConfig\sas\_services\_webolapviewer\_local\_omr.xml

For full instructions, see the deployment instructions for your platform that are described in the **config.pdf** file. This file is located in *SAS-install-dir*\SAS\SASWebOlapViewerforJava\3.1.

- 3 Re-create and redeploy SAS Web Report Studio and SAS Web Report Viewer. Here is a summary of the steps:
  - a Rebuild and redeploy SAS Web Report Studio and SAS Web Report Viewer. Here is a summary of the steps:

- b Run the configuration scripts (for example, **sas.wrs.config.bat** and **sas.wrv.config.bat** on a Windows machine).

If you are using WebLogic, then you must explode the new WAR file that is created.

- c Deploy the **SASWebReportStudio.war** and **SASWebReportViewer.war** files to your servlet container.
- d Re-import the foundation services that SAS Web Report Studio and SAS Web Report Viewer use.

In the Foundation Services Manager of SAS Management Console, delete the *Query and Reporting Services* node. Then, import the following:

```
SAS-install-dir\SASWebReportViewer\3.1\wrvpackaging\WEB-INF\pfsconfig\config\_sas_pfs_queryandreporting.xml
```

For full instructions, see “Re-Create and Redeploy SAS Web Report Studio” on page 116.

- 4 Re-create and redeploy SAS BI Web Services for Java.

You should perform this step after you have rebuilt and deployed the portal. Here is a summary of the steps:

- a Run the SAS BI Web Services for Java **configure\_xmla** utility, which is located at *SAS-install-dir\Web\WebServicesforJava\1.0\Config*.
  - b Deploy SAS BI Web Services for Java (**SASXMLA.war**) to your servlet container.
- 5 Restart the SAS Services Application if you are redeploying the SAS Information Delivery Portal.

*Note:* Do not restart the servlet container or J2EE application server at this time. You must add users to SAS metadata, as instructed later in this procedure, before you restart the servlet container.  $\Delta$

---

### Step 3: Update IBM WebSphere Application Server

If you use IBM WebSphere Application Server, then you should update the WebSphere JAAS configuration file to include the authentication information that is used by the SAS Web Infrastructure Kit Web applications.

To do this, concatenate the contents of the **login.config** file (located in the **SASServicesConfig** folder of the setup directory) to your WebSphere server’s JAAS configuration file (located in the WebSphere **AppServer/properties** directory). By default, the server’s JAAS configuration file is named **wsjaas.conf**.

(You performed this step initially during installation using the instructions in the **wik\_readme.html** file. You must now repeat the step because you are changing the authentication provider.)

---

### Step 4: Add SAS Users to the Web Authentication Provider

When you installed the middle-tier software, you created particular SAS users on the metadata server host and in SAS metadata. In order for these users to log on to the Web applications when Web authentication is used, you must add the users to the servlet container or HTTP server’s authentication provider. For example, if your HTTP server is configured to authenticate users that are defined in a database, then add the users to that database. When you add the users, for the user name and password, you must specify the user ID and password that you specified for the user when you ran the installation program.



The following table lists the users that you should add to the Web server's authentication provider:

**Table 3.6** SAS Users to be Added to the Authentication Provider

Type of User	User ID (Default)	Password
SAS Web Administrator	saswbadm	*****
SAS Demo User	sasdemo	*****

*Notes:*

- If you are using WebLogic, then in WebLogic you must add these users to the group that you created for the realm (the default is **webuserGroup**).
- These users are not required for portal operation. However, if you want the SAS Web Administrator and SAS Demo User to be able to log on to the portal Web application, then you must add them to the authentication provider. The SAS Web administrator will log on in order to administer the portal. The SAS Demo User is useful for testing your Web configuration.

## Step 5: Modify SAS Users on the SAS Metadata Server

You must add logins to the metadata identities for particular required SAS users so that those users can authenticate against the servlet container or HTTP server. You will need to add logins for each of these users:

- SAS Web Administrator (**saswbadm**)
- SAS Guest User (**sasguest**)
- Optionally, SAS Demo User (**sasdemo**)

You add logins to metadata identities in SAS Management Console. For instructions, see the User Manager Help in SAS Management Console. See also *SAS Intelligence Platform: Security Administration Guide* for detailed information about logins, user identities, and identity management.

The new logins that you add should have the following properties:

- User ID: Specify the same user ID that is specified in the initial login for this user. (The initial login is associated with the default authentication domain (**DefaultAuth**). For example, for the SAS Web Administrator you might specify **saswbadm**. On Windows systems, do *not* use a host- or domain-qualified user ID.
- Password: You do not need to specify a password.
- Authentication Domain: Specify the **web** authentication domain. (This is the value that you specified for the \$USER\_DOMAIN\$ or the \$LOGON\_DOMAIN\$ property in the application's properties file. See "Step 1: Modify Configuration Files" on page 32.) If the **web** authentication domain doesn't exist, then create a new authentication domain.

If you have defined additional logins for the user, then you should not change those logins. The additional logins are required to access the SAS application servers (SAS Workspace Server, SAS Stored Process Server, SAS OLAP Server).

The following table summarizes login information for all the SAS users and groups that are required for portal operation (not just the users that you modified in this step). The table shows example settings for their logins.

*Note:* Where user IDs include a <domain>, the domain refers to the host or domain qualifier that is required when the metadata server is on a Windows host. △

**Table 3.7** Summary of Login Information

Name	Example Credentials		Password
	Authentication Domain	User ID	
SAS Administrator	(none)	<domain>\sasadm	(none)
SAS Trusted User	DefaultAuth	<domain>\sustrust	(none)
SAS Guest User	web	sasguest	(none)
	—	—	—
SAS Web Administrator	DefaultAuth	<domain>\sasguest	*****
	web	saswbadm	(none)
SAS Demo User	—	—	—
	DefaultAuth	<domain>\saswbadm	*****
SAS General Servers (group)	web	sasdemo	(none)
	—	—	—
	DefaultAuth	<domain>\sasdemo	*****
	DefaultAuth	<domain>\sassrv	*****

*Notes about the table:*

- If the SAS Trusted User in your configuration does not have an authentication domain associated with its login, then you should add an authentication domain. The example here has a **DefaultAuth** authentication domain.
- The table does not show additional logins that you might have defined for these users so they can access a SAS Workspace Server, a SAS Stored Process Server, or a SAS OLAP Server.
- For the users that have two logins, the first login is required to log on to the portal Web application using Web authentication. (The SAS Guest User probably won't log on to the portal Web application, because the Public Kiosk is not used with Web authentication. However, the SAS Guest User still requires the **web** login in order for Web authentication to work correctly.) The second login is required to connect to any SAS application servers that are defined with the **DefaultAuth** authentication domain, and to provide authentication for SAS applications, such as SAS OLAP Cube Studio or SAS Information Map Studio, that do not support Web authentication. For the SAS Web Administrator and SAS Guest users, the second login is also required to load portal metadata.
- The SAS Administrator, SAS Trusted User, and SAS General Server users will not directly log on to the portal Web application. They therefore have only one login.

---

## Step 6: Modify Xythos WebFile Server Configuration

If you use Xythos WebFile Server, then you must modify the **saswfs.properties** file, which is located in the Xythos installation directory. Specify these two values as follows:

```
com.sas.wfs.domain.metadata=web
com.sas.wfs.domain.dav=web,DefaultAuth
```

After you make this change and save the **saswfs.properties** file, restart Xythos WebFile Server.

---

## Step 7: Restart the Servlet Container

You must restart the servlet container or J2EE application server in which the portal Web application, SAS Web OLAP Viewer for Java, SAS Web Report Studio, and SAS Web Report Viewer are running.

---

## Step 8: Define all Users

If you haven't already done so, define all users who will use the Web applications.

*Note:* Before you proceed to define all users, verify that the SAS users that you configured in previous steps can authenticate properly within your Web environment. For example, to verify authentication, you might log on to the portal Web application as the SAS Demo User (sasdemo) and perform some tasks. Additionally, if you are using the SAS application and data servers, make sure that you can access those servers from the portal. For example, you might search for and execute a stored process. △

To define users, do the following for each user:

- 1 Create an account for the user on the servlet container or the HTTP server's authentication provider. For example, if your HTTP server is configured to authenticate users that are defined in a database, then add the user to that database. If the HTTP server is configured to use LDAP authentication, then add the user to the LDAP directory.
- 2 Create a user identity for the user in SAS metadata. The user identity must have a login that is associated with the **web** authentication domain.

*Do not create duplicate identities in metadata.* If you have already created a user definition for any of these users as part of another installation, then do not create it again. Instead, modify the login definitions, if needed, with the **web** authentication domain.
- 3 Create additional logins as necessary in order for the user to access the SAS application servers (SAS Workspace Server, SAS Stored Process Server, SAS OLAP Server). You can alternatively use a shared account on the authentication provider, and add the user to a group in metadata that has login credentials for that account. For more information, see "Planning User Accounts and Their Organization into Groups" on page 21.

*Do not create duplicate logins.* If you have already created a login for any of these users or groups as part of another installation, then do not create it again. No two logins should have the same combination of User ID and authentication domain for any user or group identity.

---

## Step 9: Ensure That All Users Know How to Access the Web Applications

Make sure that all the users in your organization know how to access the Web applications (their URL) and know the proper logon format to use for Web authentication. For more information, see "Starting the Web Applications" on page 13.

*Note:* Although users don't directly log on to the SAS Information Delivery Portal, users can and should log off the portal when they finish using the portal. If they don't log off, their portal sessions continue to use system resources until the sessions time out. When users log off, these resources become available for other applications to use. For instructions on logging off, refer to the online Help that is provided with the portal. △

---

## Configuring the Web Applications for Secure Sockets Layer (SSL)

---

### 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 Netscape Navigator 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. The documentation doesn't explain information that is specific to your servlet container. You should consult the servlet container documentation for SSL implementation details.

*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.  $\Delta$

See "Secure Sockets Layer" on page 82 for things that you should consider when planning your SSL implementation.

The following sections describe the steps for configuring the portal Web applications for SSL.

---

### Set Up the SSL Environment for Your Servlet Container

If you haven't already done so, configure your servlet container for SSL. For details, see your servlet container's SSL documentation:

- For BEA WebLogic Server, see <http://e-docs.bea.com/wls/docs81/secmanage/ssl.html>
- For IBM WebSphere Application Server, see [http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec\\_adminsec.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_adminsec.html)
- For Apache Tomcat, see <http://tomcat.apache.org/tomcat-4.0-doc/ssl-howto.html>

After you have configured the servlet container, the Java Runtime 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 by default the portal invokes SAS Web Report Viewer in order to display the report. If the portal and SAS Web Report Viewer run on different machines, then the certificate must reside in the JRE for each

machine. Because the portal communicates directly with SAS Web Report Viewer, and SAS Web Report Viewer 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 Viewer.

---

## Example for Importing Distributed Certificates

If you redistribute any Web applications that communicate with each other, then you must import the certificate from the JRE that hosts the distributed application into the JRE that is used for the remaining applications.

*Note:* The procedure for moving certificates between JREs varies with the J2EE application server. Consult your vendor's documentation for import and export procedures. △

The following steps summarize what you might do for a typical Sun SSL implementation:

- 1 Export the certificate from the distributed application's keystore. (A keystore is the SSL term that is commonly used for a key repository.)
- 2 Import the certificate into the **cacerts** file for the target application's JRE. The **cacerts** file is located in the `JAVA_HOME/jre/lib/security` directory.

If you are storing certificates in a file other than the **cacerts** file, then use the following command-line option in order to specify the location of the file that you are using:

```
-Djavax.net.ssl.trustStore=trustedKeyStore
```

In the previous command, replace *trustedKeyStore* with the name and location of the file.

To apply the command to the JRE in which the Web application runs, use the command-line option when you start the servlet container.

To apply the command to the SAS Services Application, add the command-line option to either file below:

- If you use the **StartRemoteServices** script to start the SAS Services application, then add the command to that script. The script is located at `SAS-install-dir\Web\Portal2.0.1\SASServices\WEB-INF`.
- If you run the SAS Services Application as a Windows service, then add the command to the Java Service Wrapper that is provided with SAS Foundation Services. The file that you modify can be found at `SAS-config-dir\Lev1\web\Deployments\RemoteServices\WEB-INF\conf\wrapper.conf`. For more information, see "Run Remotely Deployed Services as a Windows Service" on page 345.

*Note:* Changes to the wrapper file will be overwritten if you later run the portal's **configure\_wik** script. △

For more information about the **cacerts** file or the command to set the **trustStore**, see <http://java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html>.

---

## Add an Extra Argument for WebSphere on Solaris

Perform this step if you have an IBM WebSphere Application Server that runs on a Solaris operating system.

To enable SSL, add a Java command-line argument to the following start-up commands:

- the command that starts IBM WebSphere Application Server
  - Use the WebSphere Administration console to add the argument to the execution arguments of the process definition for the WebSphere server.
- the command that starts the SAS Services Application (if your deployment includes the SAS Information Delivery Portal)
  - Modify the `StartRemoteServices` script to add the argument to the start-up command for SAS Services Application.

Here is the argument that you should add to the start-up commands:

```
-Djava.protocol.handler.pkgs=com.ibm.net.ssl.internal.www.protocol
```

## Portal-Specific Configuration

### About Portal-Specific Configuration

If your deployment includes the SAS Information Delivery Portal, then you must perform the following additional configuration:

- Update the metadata for the SAS Themes and SAS Preferences applications so that they can use the SSL protocol.
- Update the protocol that is specified in the each remote portlet's XML file so that the protocol specifies HTTPS.
- Restart the SAS Services Application.

In addition, if you redistribute any of the portal's components, then the JRE for each distributed component requires a certificate.

Here are the portal components that might be distributed:

- the portal Web application (the SAS Information Delivery Portal or the portal Web application shell that is included with the SAS Web Infrastructure Kit)
- the SAS Services Application
- the SAS Stored Process Web Application
- the SAS Themes Application
- the SAS Preferences Application
- the SAS Documentation Application
- all remote portlets
- the Xythos WebFile Server.

*Note:* Normally, users will not access the Xythos server directly, and you would not need to configure the Xythos server for SSL communication. Most user will access the Xythos server via the portal, which is often configured to reside behind a firewall and, therefore, does not require encryption. However, the Xythos administrator will likely access Xythos directly. To accommodate this case, you should consider configuring Xythos for SSL.  $\Delta$

The following sections describe the steps for configuring the portal for SSL.

### Step 1: Update the Themes and Preferences Metadata

If you are deploying the SAS Information Delivery Portal applications, then ensure that the SAS Themes and SAS Preferences applications can use the SSL protocol. To do this, modify and run the following programs:

- **UpdateThemeConnection.sas**
- **UpdatePreferencesConnection.sas**

Both files are located in the **OMR** subdirectory of the portal installation.

In each file, change the **hostName**, **port**, and **protocol** fields as applicable. Here is an example:

```
%let hostName=SSLHost.com;
%let port=443;
%let protocol=https;
```

All other fields should use the same values that were specified when the themes and preferences were initially loaded. After running the programs, check the SAS log to verify that they ran successfully.

For more information about **UpdateThemeConnection.sas**, see “Redistributing the SAS Themes Web Application” on page 351 . For more information about **UpdatePreferencesConnection.sas**, see “Redistributing the SAS Preferences Web Application” on page 350 .

## Step 2: Update Remote Portlets for SSL

If you have remote portlets, update the protocol that is specified in the portlet’s XML file so that it specifies HTTPS. For more information about updating the XML file, see “Creating a Deployment Descriptor” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/tasks/dg\\_portlet\\_descr.html](http://support.sas.com/rnd/itech/doc9/portal_dev/tasks/dg_portlet_descr.html).

## Step 3: Restart the SAS Services Application

Stop and restart the SAS Services Application in order to register the changes made by the **UpdateThemeConnection.sas** program.

---

# Adding Permissions to Policy Files

---

## Overview: Adding Permissions to Policy Files

To enable Web applications to access resources (servers and services) on either their own machine or other machines, the appropriate security permissions must be specified in the Java policy file for the Web application’s servlet container. In addition, to enable applications to share remote services, the appropriate security permissions must be specified in the policy file for the SAS Services application.

After installation, the Web applications run with minimal security restrictions. When you are ready to increase the level of security, you can modify the servlet container’s policy file with permissions from policy files that are provided by SAS.

If you create your own Web applications, then you will also want to specify permissions for those applications. This topic describes the permissions that you might specify.

For more information, see the following sections:

- “Permissions That Are Provided by SAS” on page 46
- “Modifying the Java Policy File” on page 47
- “Resources That Require Permissions in the Application’s Policy File” on page 48
- “Access Permissions for the Portal Components” on page 50

- “Access Permissions for Custom Portlets and Web Applications” on page 53

---

## Permissions That Are Provided by SAS

### Overview of the SAS Policy Files

The Web application installation provides two types of policy files:

- policy files with no security restrictions. During installation, the permissions in these files are added to the servlet container’s policy file.
- policy files with security restrictions. After installation, you can replace the existing permissions in the servlet container’s policy file with the more restrictive permissions that are in these files.

### Policy Files with No Security Restrictions

The Web applications provide the following files that have no security restrictions:

- The SAS Information Delivery Portal provides these files in its **SASServicesConfig** directory:

```
sas.wik.allpermissions.tomcat.policy
sas.wik.allpermissions.weblogic.policy
sas.wik.allpermissions.websphere.policy
sas.wik.allpermissions.sasservices.policy
```

The first three files contain the permissions for the components of the portal.

The **sas.wik.allpermissions.sasservices.policy** file contains the permissions for the SAS Services application. This file is specified for the **-Djava.security.policy** parameter in the *SAS-config-dir\Lev1\web\Deployments\RemoteServices\WEB-INF\StartRemoteServices* script (or in *SAS-config-dir\Lev1\web\Deployments\RemoteServices\WEB-INF\conf\wrapper.conf* if you run SAS Services as a Windows service).

- SAS Web Report Studio provides these files in its **live** and **tomcat** directories:

```
sas.wrs.allpermissions.tomcat.policy (tomcat directory)
sas.wrs.allpermissions.weblogic.policy (live directory)
sas.wrs.allpermissions.websphere.policy (live directory)
```

- Similarly, SAS Web Report Viewer provides these files in its **live** and **tomcat** directories:

```
sas.wrv.allpermissions.tomcat.policy (tomcat directory)
sas.wrv.allpermissions.weblogic.policy (live directory)
sas.wrv.allpermissions.websphere.policy (live directory)
```

- SAS Web OLAP Viewer for Java provides these files in its **SASServicesConfig** directory:

```
sas.webolapviewer.allpermissions.tomcat.policy
sas.webolapviewer.allpermissions.weblogic.policy
sas.webolapviewer.allpermissions.websphere.policy
```

### Policy Files with Security Restrictions

The Web applications provide the following files that have no security restrictions:



- The SAS Information Delivery Portal provides these files in its **SASServicesConfig** directory:
  - sas.wik.tomcat.policy**
  - sas.wik.weblogic.policy**
  - sas.wik.websphere.policy**
  - sas.wik.sasservices.policy**
- SAS Web Report Studio provides these files in its **live** and **tomcat** directories:
  - sas.wrs.tomcat.policy** (tomcatdirectory)
  - sas.wrs.weblogic.policy** (livedirectory)
  - sas.wrs.websphere.policy** (live directory)
- SAS Web Report Viewer provides these files in its **live** and **tomcat** directories:
  - sas.wrv.tomcat.policy** (tomcatdirectory)
  - sas.wrv.weblogic.policy** (livedirectory)
  - sas.wrv.websphere.policy** (live directory)
- SAS Web OLAP Viewer for Java provides these files in its **SASServicesConfig** directory:
  - sas.webolapviewer.tomcat.policy**
  - sas.webolapviewer.weblogic.policy**
  - sas.webolapviewer.websphere.policy**

---

## Modifying the Java Policy File

After your applications have been installed and are working properly, to increase the level of security, complete these steps:

- 1 Replace the non-secure permissions in the servlet container's policy file with the contents of the appropriate `sas.*` policy file.

For example, for the portal Web application, locate the permissions in the servlet container's policy file by searching for the following comment lines:

```
// =====
//           WIK / IDP / WebServices Code Permissions
//
// Note: Use of this file will result in your Web applications
//       running without security restrictions.
// =====
```

Once you find the applicable grant block, do the following:

- a Locate the following grant statement:
 

```
grant codeBase "file:${catalina.home}/webapps/Portal/-"
{
    permission java.security.AllPermission;
};
```
  - b Replace the statement with the corresponding grant codeBase statement from the appropriate `sas.wik.*` policy file. In this example, you would replace the statement with the corresponding grant codeBase statement that you find in the **sas.wik.tomcat.policy** file.
  - c Repeat the previous steps for the SAS Preferences Application codeBase, and for any application that is included with the portal Web application.
- 2 Similarly, replace the grant codeBase statements for all the Web applications that you have deployed.

- 3 For the SAS Services Application, which deploys the Remote Services, specify the **sas.wik.sasservices.policy** file for the `-Djava.security.policy` parameter in the `SAS-config-dir/Lev1/web/Deployments/RemoteServices/WEB-INF/StartRemoteServices` script (or in `SAS-config-dir/Lev1/web/Deployments/RemoteServices/WEB-INF/conf/wrapper.conf` if you run SAS Services as a Windows service).

For many configurations, it is sufficient to copy the permissions from the policy files that are provided by SAS. However, in some cases it is necessary to modify the permissions in your policy files more directly. For example, you must modify your policy files in the following situations:

- if you add a new portlet or Web application that communicates with SAS servers or with SAS Foundation Services
- if you add a syndication channel, or if you add a publication channel that publishes to an archive
- if you redistribute servers to different machines or change server port numbers

When you add permission statements that contain machine names, if you use a value of `localhost` for the machine name, Java will resolve the `localhost` value to the IP address `127.0.0.1`. Because the Java-resolved IP address is not the same as the machine's IP address, when you use the value `localhost`, you must specify two permissions statements: one statement for `localhost`, and one statement for the fully qualified machine name. When you specify a permission statement with a fully qualified machine name, you need to specify only one statement—a statement for the fully qualified machine name. For example, to add a permission statement for the SAS Service application's machine, add the following two lines to the policy file:

```
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";
```

---

## Resources That Require Permissions in the Application's Policy File

### Servers

For each application (Web or stand-alone) that needs to communicate with a SAS server, the Java policy files for the calling application need to include a permission to communicate with the SAS Server. If you add a new portlet or Web application that communicates with servers, if you add new servers to your portal Web application's server deployment, or if you redistribute servers to other machines, then you must update the appropriate policy file with permission statements for the new server machines. To add a permission statement for server access to a policy file, for each application's codebase, you must add a statement with the following format:

```
// SAS Stored Process, Workspace,
// or OLAP server - need one entry
// per machine
permission java.net.SocketPermission
    "host:1024-", "connect, resolve";
```

where *host* is the host name of your server. For example, in the Apache Tomcat's **catalina.policy** file, you must add a permission statement to each of the following codebases:

```

grant codeBase
  "file:${catalina.home}/webapps/Portal/-" {

  // SAS Stored Process, Workspace,
  // or OLAP server - need one entry
  // per machine
  permission java.net.SocketPermission
    "host:1024-", "connect, resolve";
  // -----
};
grant codeBase
  "file:${catalina.home}/webapps/SASStoredProcess/-" {
  // SAS Stored Process or Workspace server -
  // need one entry per machine
  permission java.net.SocketPermission
    "host:1024-", "connect, resolve";

```

## Services (SAS Services Application)

The SAS Services Application is used as a Java RMI server to enable access to remote services and session context sharing between applications. When an application (Web or stand-alone) must communicate with another application, it uses the SAS Services application to access a set of shared remote services; when the application and SAS Services application share remote services, the applications are both RMI endpoints. To enable RMI endpoints to communicate, the Java policy files for both applications must include a permission statement that enables communication with the other application end-point's machine. Add a permissions with the following format to both the policy file for the SAS Services application and the policy file for the foundation service-enabled remote portlet or Web application:

```

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

```

To understand the required permissions for the SAS Services application and the foundation service-enabled remote portlet or Web application, see "Access Permissions for Custom Portlets and Web Applications" on page 53.

## Portal Content

When you want to add the following content to the portal Web application, you must have been granted permissions for the content in the policy file for the portal Web application:

- URL Display portlet. For details about adding the appropriate permissions for URL Display portlets, see "Main Steps to Add a Portlet" on page 262 .
- syndication channels. For details about adding the appropriate permissions for syndication channels, see "Adding Syndication Channels" on page 286 .
- SAS publication channels. For details about adding the appropriate permissions for SAS publication channels, see "Adding SAS Publication Channels" on page 292 .

---

## Access Permissions for the Portal Components

### Summary of Access Permissions for the Portal Components

This section describes the permission statements that appear in the `sas.wik.tomcat.policy` file. The policy files for the other servlet containers and other applications are similar.

To secure resources within the portal Web application infrastructure, the appropriate permissions are required for the following applications:

- **Portal Web Application:** The portal Web application must be able to access the SAS Metadata Server, its local services, the Java remote method invocation (RMI) server for the SAS Services application's remote services, any other servers that it needs to access, and any foundation service-enabled applications (such as the SAS Themes Web application) which it calls. The portal Web application also requires permission to listen for calls from foundation service-enabled applications.
- **SAS Stored Process Web Application:** The SAS Stored Process Web application must have access to the SAS Metadata Server, its local services, the Java RMI server for the SAS Services application's remote services, and any other servers that it needs to access. The SAS Stored Process application also requires permission to receive calls from any calling application, such as the portal Web application.
- **SAS Preferences Web Application:** The SAS Preferences Web application must have access to the Java RMI server for the SAS Services application's remote services. The SAS Preferences Web application must also have access to the SAS Themes Web application.
- **SAS Services Application:** The SAS Services application must have access to the Java RMI server to register the remote services. In addition, it must have permissions for all applications that participate in SAS Foundation Service session sharing (access to remote-accessible services), and it must have permissions for the SAS Themes Web application.
- **Remote Portlets and Foundation-service Enabled Web Applications:** Remote portlets and any foundation-service enabled Web applications must have access to the SAS Metadata Server and to the Java RMI server for the SAS Services application's remote service. The remote portlet or Web application also requires permission to receive calls from any calling application. In addition, if a remote portlet or foundation-service enabled application calls the SAS Themes Web application, then it must have access to the SAS Themes Web application.

The following sections show the permission statements you must specify in each application to enable communication with its required servers and services.

If some of the applications are located on the same machine, there might be duplicate permission statements.

### CodeBase: Portal

Here are the permissions for the Portal CodeBase:

- Access to the SAS Metadata Server:

When running on localhost, also create an entry containing 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, also create an entry containing 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 portal 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
  <portal Web application's machine name>:1024-,
  "listen, connect, accept, resolve";

```

- 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
  <SAS Stored Process 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
  <remote portlet or 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 OLAP Server's machine name>:1024-,
  "connect, resolve";

```

- Access to the WebDAV server:

```

permission java.net.SocketPermission
  <WebDAV server's machine name>:8300,
  "connect, resolve";

```

- Access to the SAS Themes Application:

The values for `$CONTAINER_HOST$` and `$CONTAINER_PORT$` are specified in the `install.properties` file

```
// ----- Socket Access to Themes -----

permission java.net.SocketPermission
<${CONTAINER_HOST}><${CONTAINER_PORT}>:,
"connect, resolve";
```

## CodeBase: SASPreferences

Here are the permissions for the SAS Preferences application CodeBase:  
Access to the SAS Themes application:

The values for `${CONTAINER_HOST}` and `${CONTAINER_PORT}` are specified in the `install.properties` file.

```
// ----- Socket Access to Themes -----

permission java.net.SocketPermission
<${CONTAINER_HOST}><${CONTAINER_PORT}>:,
"connect, resolve";
```

## CodeBase: SASStoredProcess

Here are the permissions for the SAS Stored Process application CodeBase:

- Access to the SAS Metadata Server:

When running on localhost, also create an entry containing 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, also create an entry containing 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 SAS Stored Process 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
<SAS Stored Process 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 Stored Process Server's machine name>:1024-,
  "connect, resolve";

```

- Access to the WebDAV server:

```

permission java.net.SocketPermission
  <WebDAV server's machine name>:8300,
  "connect, resolve";

```

## CodeBase: SASServices

Here are the permissions for the SAS Services Application CodeBase:  
(In the policy file used by the SAS Services Application)

- Access to the Java RMI server and remote SAS Foundation Services

When running on localhost, also create an entry containing 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";

```

- Connections with application(s) that use SAS Foundation Service session sharing:

```

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

```

---

## 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 portal 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 to enable communication with its required servers and services.

## CodeBase: <Remote Portlet or Web Application>

Here are the permissions for the remote portlet or Web application's CodeBase:

- Access to the SAS Metadata Server:

When running on localhost, also create an entry containing 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, an entry is also required containing 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 SAS Themes Application:

The values for \$CONTAINER\_HOST\$ and \$CONTAINER\_PORT\$ are specified in the **install.properties** file.



```
// ----- Socket Access to Themes -----
permission java.net.SocketPermission
<${CONTAINER_HOST}><${CONTAINER_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";
```

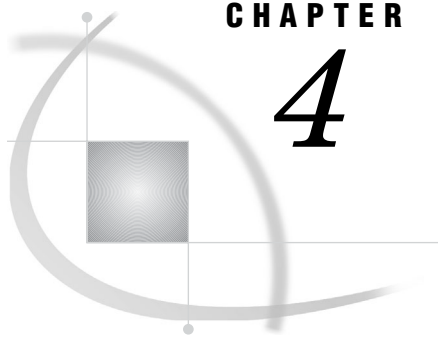
### **CodeBase: SAServices**

(In the policy file used by the SAS Services Application)

Connections with application(s) that utilize SAS Foundation Service session sharing:

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





## CHAPTER

## 4

# Best Practices for Configuring Your Middle Tier

<i>Overview of Middle Tier Configuration</i>	58
<i>Tuning the Java Virtual Machine</i>	58
<i>Which JVM Are You Using?</i>	59
<i>Where to Specify JVM Options</i>	59
<i>Quick Start Settings</i>	60
<i>Few Users, Sun JDK</i>	60
<i>More Users, Sun JDK</i>	61
<i>Few Users, IBM JDK</i>	61
<i>More Users, IBM JDK</i>	62
<i>Setting Just-in-Time Compiler and Memory Options</i>	62
<i>Selecting a Garbage Collector</i>	63
<i>Configuring the Garbage Collector</i>	64
<i>Tuning the J2EE Application Server or Servlet Container</i>	64
<i>Overview of Tuning the J2EE Application Server or Servlet Container</i>	64
<i>Detecting Changes in JavaServer Pages and Servlets</i>	64
<i>Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)</i>	65
<i>Improving the Performance of WebLogic on HP-UX</i>	66
<i>Tuning WebSphere 6.0.2 or 6.1</i>	66
<i>Overview of Tuning WebSphere 6.0.2 or 6.1</i>	66
<i>Java Virtual Machine (JVM) Arguments</i>	66
<i>About the JVM Arguments</i>	66
<i>Configure the JVM Arguments</i>	66
<i>WebSphere 6.0.2 JVM Arguments</i>	67
<i>WebSphere 6.1 JVM Arguments</i>	67
<i>Descriptions of the Arguments</i>	68
<i>Setting Web Container Properties</i>	69
<i>Add Required Custom Properties to the Web Container</i>	69
<i>Set Thread-Pool Properties</i>	69
<i>Set Tuning Values for the AIX Operating System</i>	69
<i>Sample Middle-Tier Deployment Scenarios</i>	70
<i>Overview of Middle-Tier Deployment Scenarios</i>	70
<i>Criteria for Choosing a Middle-Tier Configuration</i>	71
<i>Security</i>	72
<i>Availability</i>	73
<i>Performance and Scalability</i>	74
<i>Maintainability</i>	75
<i>Scenario 1: Web Applications Deployed in a Single J2EE Application Server</i>	75
<i>Scenario 2: Static Content Deployed in an HTTP Server Proxy</i>	77
<i>Static Content to Deploy for SAS Web Applications</i>	78
<i>Advantages and Disadvantages of Using Scenario 2</i>	78
<i>Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster</i>	79

<i>Understanding Clusters</i>	80
<i>Requirement for Session Affinity</i>	80
<i>Understanding Demilitarized Zones</i>	80
<i>Advantages and Disadvantages of Using Scenario 3</i>	81
<i>Additional Considerations for Planning a Deployment</i>	81
<i>Load-Balancing Software and Hardware for the HTTP Servers</i>	82
<i>Secure Sockets Layer</i>	82
<i>Middle-Tier (Trusted Web) Authentication</i>	83
<i>SAS Services Application Heap Size</i>	83
<i>Configuring a Cluster of J2EE Application Servers</i>	84
<i>Overview of Cluster Configuration</i>	84
<i>Additional Manual Steps for WebLogic</i>	84
<i>Configuring an HTTP Server to Serve Static Content for SAS Web Applications</i>	86
<i>Overview of Configuring an HTTP Server to Serve Static Content</i>	86
<i>Example: Setting Up Apache to Serve Static Content</i>	86
<i>Main Steps for Setting Up Apache to Serve Static Content</i>	86
<i>Deploy Static Content for SAS Web Report Studio and SAS Web Report Viewer</i>	87
<i>Deploy Static Content for SAS Information Delivery Portal</i>	88
<i>Deploy Static Content for SAS Web OLAP Viewer for Java</i>	90
<i>Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server</i>	90
<i>Sample: Proxy Setup for SAS Web Report Studio</i>	91
<i>Sample: Proxy Setup for SAS Web OLAP Viewer for Java</i>	92
<i>Sample: Proxy Setup for the Portal and Related Web Applications</i>	93
<i>Configuring Apache Cache Control for Static Content</i>	96

---

## Overview of Middle Tier Configuration

This chapter explains how you can configure middle-tier components of the SAS Intelligence Platform for better efficiency and performance.

The middle tier provides an environment for running the following SAS Web clients:

- SAS Web Report Studio
- SAS Web Report Viewer
- SAS Information Delivery Portal, or the portal Web application that is included in the SAS Web Infrastructure Kit
- SAS Web OLAP Viewer for Java

*Note:* In this chapter, all references to the SAS Information Delivery Portal can be assumed to include the portal Web application that the Web Infrastructure Kit includes.  $\triangle$

---

## Tuning the Java Virtual Machine

Your J2EE application server or servlet container's JVM can be started with a number of options that affect its behavior. For a quick overview of where to set these options and a list of generally applicable settings, see the following subsections:

- "Which JVM Are You Using?" on page 59
- "Where to Specify JVM Options" on page 59
- "Quick Start Settings" on page 60

For more details about the various JVM options, see the following subsections:

- “Setting Just-in-Time Compiler and Memory Options” on page 62
- “Selecting a Garbage Collector” on page 63
- “Configuring the Garbage Collector” on page 64
- “Tuning the J2EE Application Server or Servlet Container” on page 64
- “Detecting Changes in JavaServer Pages and Servlets” on page 64
- “Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)” on page 65

---

## Which JVM Are You Using?

The information in this section applies both to the JVM that is supplied by Sun Microsystems and to the JVM that is supplied by IBM. Note that these two JVMs use different parameters. Most installations that use BEA WebLogic should use Sun’s JVM and its corresponding parameters. Most installations that use IBM WebSphere should use IBM’s JVM and its parameters. At the time this is being written, two exceptions are known:

- 1 On IBM’s AIX operating system, only the JVM that is supplied by IBM should be used, even for BEA WebLogic.
- 2 On Sun’s Solaris operating system, only the JVM that is supplied by Sun should be used, even for IBM WebSphere.

---

## Where to Specify JVM Options

### *BEA WebLogic*

For WebLogic version 8.1, you can set your JVM options in one of two ways: by editing the script `startManagedWebLogic.extension` or by using the product’s administrative console. Use whichever approach you have used to set other server options. The paragraphs below explain how to use each approach.

*Note:* If you are not running the Node Manager, you must specify your JVM options using the script. △

To set your JVM options in the `startManagedWebLogic.extension` script, perform the following steps:

- 1 Change directories to `WebLogic-install-dir\user_projects\domains\domain`.
- 2 Open the script in a text editor.
- 3 Uncomment the line reserved for setting the `JAVA_OPTIONS` environment variable, and set this variable as explained later in this section.
- 4 Save your changes, and close the file.

To set the JVM options from the administrative console, perform the following steps::

- 1 From the BEA WebLogic console, in the left panel, expand the **Servers** node in the tree.
- 2 Select the server that you want to configure.
- 3 In the right pane, select the **Remote Start** tab. That screen has an **Arguments** field where you can insert your JVM options. After inserting your options, click **Apply**.
- 4 Restart the server so that the new settings will be in effect.

*IBM WebSphere*

The IBM WebSphere administrative console is used to set Java startup parameters. The following procedure will enable you to set the parameters for IBM WebSphere version 5.1. For more information, see the IBM WebSphere documentation.

*Note:* For tuning information that is specific to WebSphere 6.0.2 or 6.1, see “Tuning WebSphere 6.0.2 or 6.1” on page 66. △

- 1 From the IBM WebSphere console, on the left panel, select **Servers**.
- 2 Select **Application Servers**.
- 3 On the right will be displayed a list of your servers. Select the appropriate server.
- 4 On the resulting screen, select **Process Definition**, then **Java Virtual Machine**.
- 5 Enter your Java parameters into the **Generic JVM Arguments** field.

*Note:* Some parameters can be specified either in other boxes on this screen, or in the **Generic JVM Arguments** field as recommended here. Avoid placing the same information in both areas; this can have unpredictable results. △

*Apache Tomcat*

When the SAS Configuration Wizard runs, it creates a script that you can use to start your servlet container or J2EE application server. This script is called **startServletContainer.bat** or **startServletContainer.sh** and resides in the directory *path-to-config-dir\Lev1\web*. This script provides one place in which you can specify JVM options. For example, if you are using the Apache Tomcat servlet container on a Windows system, the contents of the **startServletContainer.bat** script will look something like this:

```
set JAVA_HOME=C:\jdk1.4.2_05
set CATALINA_HOME=C:\Tomcat4.1
set CATALINA_OPTS=-Xms512m -Xmx1024m -server -XX:-UseOnStackReplacement
-Djava.awt.headless=true

call "%CATALINA_HOME%\bin\catalina.bat" run -security
```

In this case, you can change the options that are used to start the JVM by changing the value of the environment variable CATALINA\_OPTS. You can add JVM options to other versions of this script in a similar manner.

---

## Quick Start Settings

If you want to start with a group of settings that will provide you with a convenient starting place, find the description in the following examples that matches your situation. Then, use the options that follow that description. (You can later fine tune these settings as necessary.)

### Few Users, Sun JDK

If you will have 10 or fewer concurrent users and are using a Sun JDK whose revision level is 1.4.2 or later, use these settings:

```
-server -Xms512m -Xmx512m -XX:NewSize=64m -XX:MaxNewSize=64m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
```

```
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

Notes:

- If you are running WebSphere 6.0.2 or 6.1, then do not use these settings. Use the settings that are described in “Tuning WebSphere 6.0.2 or 6.1” on page 66.
- The value of **-Xss** shown above is appropriate for Windows systems. On UNIX systems (including the HP-UX implementation of the Sun JVM), use the option **-Xss256k** instead.
- Do not use the option **-XX:-UseTLAB** on HP-UX Itanium systems. There is a known problem with its use.
- In version 1.4.2\_04 of the JVM and earlier versions, setting **-XX:MaxNewSize** to a value greater than the value of **-XX:NewSize** will not enable the Young Generation to grow larger than the latter value.

## More Users, Sun JDK

If you will have more than 10 concurrent users and are using a Sun JDK whose revision level is 1.4.2 or later, use these settings:

```
-server -Xms1280m -Xmx1280m -XX:NewSize=160m -XX:MaxNewSize=160m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

Notes:

- If you are running WebSphere 6.0.2 or 6.1, then do not use these settings. Use the settings that are described in “Tuning WebSphere 6.0.2 or 6.1” on page 66.
- The value of **-Xss** shown above is appropriate for Windows systems. On UNIX systems (including the HP-UX implementation of the Sun JVM), use the option **-Xss256k** instead.
- Do not use the option **-XX:-UseTLAB** on HP-UX Itanium systems. There is a known problem with its use.
- In version 1.4.2\_04 of the JVM and earlier versions, setting **-XX:MaxNewSize** to a value greater than the value of **-XX:NewSize** will not enable the Young Generation to grow larger than the latter value.

## Few Users, IBM JDK

If you will have 10 or fewer concurrent users and are using an IBM JDK whose revision level is 1.4.1 or later, use these settings:

```
-Xms256m -Xmx512m -Xss128k -Xoss128k
-Xpartialcompactgc -Xgcpolicy:optthruput
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If the JDK is installed on a UNIX system, change the value of the **-Xss** and **-Xoss** options to 256k.

*Note:* If you are running WebSphere 6.0.2 or 6.1, then do not use these settings. Use the settings that are described in “Tuning WebSphere 6.0.2 or 6.1” on page 66. △

## More Users, IBM JDK

If you will have more than 10 concurrent users and are using an IBM JDK whose revision level is 1.4.1 or later, use these settings:

```
-Xms640m -Xmx1280m -Xss128k -Xoss128k
-Xpartialcompactgc -Xgcpolicy:optthruput
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If the JDK is installed on a UNIX system, change the value of the **-Xss** and **-Xoss** options to 256k.

*Note:* If you are running WebSphere 6.0.2 or 6.1, then do not use these settings. Use the settings that are described in “Tuning WebSphere 6.0.2 or 6.1” on page 66.  $\triangle$

---

## Setting Just-in-Time Compiler and Memory Options

A number of JVM options affect which compiler the JVM uses and the amount of memory that the JVM uses for such things as the object heap.

### *Sun-Based JVM*

You should always use the Just-in-Time compiler if one is available. You enable the JIT compiler with the **-server** option. The following list includes the relevant memory settings and their respective options:

- Minimum heap size (**-Xms**)
- Maximum heap size (**-Xmx**)
- Thread stack size (**-Xss**)
- Minimum new generation size (**-XX:NewSize**)
- Maximum new generation size (**-XX:MaxNewSize**)
- Maximum permanent generation size (**-XX:MaxPermSize**)

Minimum and maximum heap size, **-Xms** and **-Xmx** respectively, define the total amount of memory that the JVM has at its disposal. To eliminate heap growth overhead, set both of these options to the same value. The recommended value for heap size depends on which applications will be running and how much load will be supported. For 10 or fewer concurrent users, set the heap size to 512 MB (**-Xms512m -Xmx512m**). For more than 10 users, set the heap size to 1280 MB (**-Xms1280m -Xmx1280m**).

Although a 1280 MB heap is the maximum possible heap size on a 32-bit Windows system, most versions of UNIX allow the JVM to address up to 3 GB of memory. If your J2EE application server is running on UNIX and you anticipate a load of hundreds of users, you might want to scale the implementation. However, large heap sizes can cause longer garbage collection times, so it is important to expand the heap only as much as necessary. We recommend an upper limit of 1.5GB.

*Note:* If this maximum heap size is affecting the number of users that you can accommodate, you should consider creating a cluster of servers.  $\triangle$

Thread stack size defines the default amount of memory allocated to each native thread spawned by the JVM. Keeping this value as low as possible allows the JVM to dedicate more process memory to the heap, which in turn increases scalability. The recommended setting is 128 KB for Windows systems and 256 KB for Solaris systems. You might need to adjust this number for other platforms or for additional load. An additional optimization is possible for Sun JVMs running on SPARC processors. You can include the **-XX:-UseTLAB** option, which tells the JVM to minimize thread stack usage. For all other Sun JVMs, this is the default behavior.



Finally, you can size different memory regions, or generations, appropriately to ensure efficient garbage collector performance. You use the minimum new generation size (**-XX:NewSize**), maximum new generation size (**-XX:MaxNewSize**), and maximum permanent generation size (**-XX:MaxPermSize**) to control the sizes of the various memory regions. The new generation should receive about 12.5% of total memory, up to a maximum of about 256 MB. Both the minimum and maximum new generation settings should be set to the same value to avoid generation growth overhead. Assuming a 1280 MB heap, the calculation and associated options would look like this:

1280 x 12.5% = 160 MB (**-XX:NewSize=160m -XX:MaxNewSize=160m**)

The permanent generation is used to store loaded Java class definitions and extremely long-lived objects. Given the variety of components, services, and frameworks in use by SAS applications, the maximum permanent generation size should be set at 128 MB (**-XX:MaxPermSize=128m**).

### *IBM JVM*

The following list includes the relevant memory settings and their respective options:

- Minimum heap size (**-Xms**)
- Maximum heap size (**-Xmx**)
- Native code thread stack size (**-Xss**)
- Java code thread stack size (**-Xoss**)

Minimum and maximum heap size, **-Xms** and **-Xmx** respectively, define the total amount of memory that the JVM has at its disposal. To allow the JVM to manage memory efficiently, set the minimum heap size below the maximum heap size. The recommended value for heap size depends on which applications will be running and how much load will be supported. A good rule of thumb is to set the minimum heap size to one half of the maximum heap size, but do not set it lower than 256 MB. For 10 or fewer concurrent users, set the heap sizes to 256 MB and 512 MB (**-Xms256m -Xmx512m**). For more than 10 users, set the heap sizes to 640 MB and 1280 MB (**-Xms640m -Xmx1280m**).

Although a 1280 MB heap is the maximum possible heap size on a 32-bit Windows system, most versions of UNIX allow the JVM to address up to 3 GB of memory. If your J2EE application server is running on UNIX and you anticipate a load of hundreds of users, you might want to scale the implementation. However, large heap sizes can cause longer garbage collection times, so it is important to expand the heap only as much as necessary. We recommend an upper limit of 1.5 GB.

*Note:* If this maximum heap size is affecting the number of users that you can accommodate, you should consider creating a cluster of servers.  $\Delta$

Native code and Java code thread stack size defines the default amount of memory allocated to each native thread spawned by the JVM. Keeping the values as low as possible allows the JVM to dedicate more process memory to the heap, which in turn increases scalability. The recommended setting for **-Xss** and **-Xoss** is 128 KB for Windows systems and 256 KB for UNIX systems.

---

## Selecting a Garbage Collector

The following subsections explain how to specify a garbage collector.

### *Sun-Based JVM*

You should use the concurrent mark and sweep collector, which you enable by specifying the **-XX:+UseConcMarkSweepGC** option. The concurrent mark sweep collector

is implemented in the old generation and, by default, its use automates the parallel copying collector in the young generation.

#### *IBM JVM*

You control the IBM JVM's garbage collection behavior using the **-Xgcpolicy** switch. The **optthroughput** policy provides the default garbage collection behavior for the IBM JVM, which is optimized for response time. If you see long pauses in response time, you might want to investigate the use of the **optavgpause** policy.

## Configuring the Garbage Collector

After you have chosen a garbage collector, configure the collector appropriately.

#### *Sun-Based JVM*

You can also influence how often the garbage collector runs by using the three options mentioned below. The **-XX:+DisableExplicitGC** option prevents Java code from invoking the garbage collector. In addition, there are two Java properties that you can set to control distributed garbage collection. (This is important because most SAS Java applications use Remote Method Invocation, which in turn uses distributed garbage collection.) Setting the **sun.rmi.dgc.client.gcInterval** and **sun.rmi.dgc.server.gcInterval** command line properties to 3600000 will reduce the frequency of full collections on the host machines..

#### *IBM JVM*

Select the **-Xpartialcompactgc** parameter, which spreads the time of compacting the memory across multiple collections, rather than waiting until memory is completely fragmented. If this parameter is not selected, garbage collection times can be excessive, especially under heavy load.

## Tuning the J2EE Application Server or Servlet Container

### Overview of Tuning the J2EE Application Server or Servlet Container

In addition to specifying Java Virtual Machine options, you can improve the performance of SAS Web Report Studio, SAS Information Delivery Portal, and SAS Web OLAP Viewer for Java by configuring other aspects of your servlet container or J2EE 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.

The following sections explain how to change these settings on the Java application servers that are supported by the SAS Intelligence Platform.

### Detecting Changes in JavaServer Pages and Servlets

Most servlet containers and J2EE application servers perform a periodic check of compiled class files and source files to determine whether a servlet or JavaServer Page has been edited. This behavior is only appropriate while applications are under

development. In your production environment, you should disable these features. The following subsections explain how to do this for the supported J2EE application servers and Apache Tomcat.

#### *BEA WebLogic Server*

BEA WebLogic does not require tuning for JSP changes for use with SAS Web Report Studio and SAS Information Delivery Portal. The **weblogic.xml** file that is shipped with both applications performs all necessary tuning functions.

#### *IBM WebSphere*

IBM WebSphere does not require tuning for JSP changes or the number of threads available.

#### *Apache Tomcat*

- 1 Open the file *Tomcat-install-dir\conf\web.xml*.
- 2 Add the following XML to the `<init-param>` block for the servlet with the servlet name `jsp`:
 

```
<init-param>
  <param-name>reloading</param-name>
  <param-value>>false</param-value>
</init-param>
<init-param>
  <param-name>development</param-name>
  <param-value>>false</param-value>
</init-param>
```
- 3 Restart the Tomcat server.

---

## Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)

Because each user request requires a server thread to service it, the server must be able to start a sufficient number of threads to service the expected load. The following sections explain how to control the number of available of threads on the different supported servers.

#### *WebLogic Server*

- 1 Log on to the administration console.
- 2 Open the Servers node of the tree.
- 3 Right-click a server definition.
- 4 Select the **View Execute Queues** menu option.
- 5 Select the **Configure a new Execute Queues** menu option.
- 6 Enter **sas.portal.default** as the queue name.
- 7 Click **Create**.

*Note:* The default **Thread Count** value of 25 should be sufficient for most loads. The BEA Web site (<http://e-docs.bea.com/platform/docs81/index.html>) contains specific information about tuning execute queues.  $\Delta$

#### *Apache Tomcat*

- 1 Open the file *Tomcat-install-dir\conf\server.xml*.

- 2 Locate the **Connector** element for the server's listening port. By default, this is port 8080.
- 3 Change the value of the **maxProcessors** attribute to a number greater than 75 (the default).
- 4 Restart the server.

*Note:* The default value of 75 provides good performance for most loads. You should need to change this setting only for very heavy loads.  $\Delta$

---

## Improving the Performance of WebLogic on HP-UX

If you are running BEA WebLogic Server on a HP-UX system, use the HPjconfig utility to determine the optimal settings for your kernel parameters. You might need to change some of the values to improve performance and to prevent Out of Memory errors.

For more information about the HPjconfig utility, see <http://www.hp.com/products1/unix/java/java2/hpjconfig/index.html>.

---

## Tuning WebSphere 6.0.2 or 6.1

---

### Overview of Tuning WebSphere 6.0.2 or 6.1

This section contains performance tuning settings that are specific to version 6.0.2 and 6.1 of the IBM WebSphere Application Server. This section assumes that you have installed and configured WebSphere by using the instructions that are available from the SAS third-party support Web page at:

<http://support.sas.com/resources/thirdpartysupport/v913sp4/index.html#appsrv>

---

### Java Virtual Machine (JVM) Arguments

#### About the JVM Arguments

This topic contains the recommended platform-specific JVM arguments to use in your deployment. These arguments were determined from a multi-user suite of benchmark and endurance tests that were run against SAS enterprise applications. These recommendations apply only to the 32-bit versions of the JVM and are not appropriate for the 64-bit versions.

#### Configure the JVM Arguments

To configure the JVM arguments for WebSphere 6.0.2 or 6.1, perform these steps:

- 1 In the WebSphere administrative console, select **Servers**  $\blacktriangleright$  **Application servers**  $\blacktriangleright$  **<serverName>**. In this menu sequence, **<serverName>** corresponds to the name of the WebSphere server that you are configuring.
- 2 Select **Java and Process Management**  $\blacktriangleright$  **Process Definition**  $\blacktriangleright$  **Java Virtual Machine**.
- 3 In the **Generic JVM Argument** field, set the generic JVM arguments that are appropriate for the version of WebSphere that you are using and the operating

*Note:* If you copy and paste these arguments, then be sure to remove any unwanted newline characters. △

## WebSphere 6.0.2 JVM Arguments

The SAS Intelligence Platform supports running WebSphere 6.0.2 with the Java Development Kit 1.4.2.

The following sections list the JVM arguments that are appropriate for the different supported operating systems.

*Windows (W32):*

```
-Xms640m -Xmx1472m -Xss128k -Xoss128k -Xpartialcompactgc
-Xgcpolicy:optthruput -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Xk15000 -Dsun.rmi.dgc.ackTimeout=1
-Dcom.ibm.websphere.threadpool.clearThreadLocal=TRUE
```

*AIX:*

```
-Xms640m -Xmx1840m -Xss256k -Xoss256k -Xpartialcompactgc
-Xgcpolicy:optthruput -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Xk15000 -Xloratio0.2 -Dsun.rmi.dgc.ackTimeout=1
-Dcom.ibm.websphere.threadpool.clearThreadLocal=TRUE
```

*Solaris (SPARC):*

```
-Xms640m -Xmx1840m -Xss256k -XX:+UseConcMarkSweepGC
-XX:NewSize=256m -XX:MaxNewSize=512m -XX:MaxPermSize=256m
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsun.rmi.dgc.ackTimeout=1
-Dcom.ibm.websphere.threadpool.clearThreadLocal=TRUE
```

## WebSphere 6.1 JVM Arguments

The SAS Intelligence Platform supports running WebSphere 6.1 with the Java Development Kit 5.0. One difference between JDK 5.0 and JDK 1.4.2 (used by WebSphere 6.0.2) is the selection of garbage collection algorithms. Both versions of the JDK offer the monolithic Java heap model, and both versions support the garbage collection policies: `-Xgcpolicy:optavgpause` and `-Xgcpolicy:optthruput`. However, with the **`-Xgcpolicy:gencon`** setting, JDK 5.0 offers one more garbage collection choice. The new garbage collection policy enables a generational Java heap that is conducive to a better spatial and temporal location of Java objects. This behavior is consistent with the SAS enterprise business intelligence suite of Java applications, though not necessarily with all Java applications.

The following sections list the JVM arguments that are appropriate for the different supported operating systems.

*Windows (W32):*

```
-Xms640m -Xmx1472m -Xss128k -Xmsol28k
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.awt.headless=true
-Dsun.rmi.dgc.ackTimeout=1
-Dcom.ibm.websphere.threadpool.clearThreadLocal=TRUE
```

*AIX:*

```
-Xms640m -Xmx1840m -Xss256k -Xmso256k -Xpartialcompactgc
-Xgcpolicy:gencon -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.awt.headless=true
-Dsun.rmi.dgc.ackTimeout=1
-Dcom.ibm.websphere.threadpool.clearThreadLocal=TRUE
```

*Solaris (SPARC):*

```
-Xms1840m -Xmx1840m -Xss256k -XX:+UseConcMarkSweepGC
-XX:NewSize=256m -XX:MaxNewSize=256m -XX:MaxPermSize=256m
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.awt.headless=true
```

**Descriptions of the Arguments**

Most of these arguments are described in “Tuning the Java Virtual Machine” on page 58. The following arguments are specific to using WebSphere 6.0.2 and 6.1:

- **-xk** (JDK 1.4) specifies the maximum number of classes that are contained within the kCluster. This argument instructs the JVM to allocate space for *nnnn* class blocks (in this case, 15000) and to avoid potential fragmentation issues.
- **-Xgcpolicy:gencon** (JDK 5) enables two major subspaces in the Java heap. These memory areas use the optimum locality of an allocated Java object.
- **-Dsun.rmi.dgc.ackTimeout** specifies the length of time (in milliseconds) that the server-side remote method invocation (RMI) runtime refers to a remote object (or a reference to a remote object) that has been returned from the current virtual machine as part of the result of a remote method call. The length of time that is specified stays in effect until the RMI receives positive acknowledgment from the client that the remote reference has been fully received and processed.
- **-Dcom.ibm.websphere.threadpool.clearThreadLocal**, when set to TRUE, enables WebSphere to clear the thread local allocation buffers (TLAB) when a thread is returned to the thread pool.
- **-Xloratio** (JDK 1.4) specifies the percentage of the Java real-time heap that is reserved for large objects. Large objects are equal to or greater than 64 KB. The reserved portion of the Java heap is never used for objects that are less than 64 KB. In this case, the value is 0.2, which corresponds to 20 percent of the real-time heap. This setting enables the JVM to avoid potential fragmentation issues that can occur for large objects.

*Note:* You must be running a JVM with a build date of 20050209 or later in order to support the Large Object Area feature.  $\triangle$

- **-XX:+UseConcMarkSweepGC** is a Sun JVM argument that enables the concurrent mark sweep garbage collection algorithm. This type of collector runs several stages in parallel with application threads, thereby avoiding long pauses during garbage collection activity.

---

## Setting Web Container Properties

### Add Required Custom Properties to the Web Container

You must set two Web container custom properties on each application server where your SAS Web applications are deployed. One property forces WebSphere to use synchronous TCP/IP channel write types, as required by the SAS Web applications. Another property prepends a forward slash to all URL resource references, as required by the SAS Web applications.

To configure the custom properties for WebSphere 6.0.2 or 6.1, perform these steps:

- 1 In the WebSphere administrative console, select **Servers ► Application servers ► <serverName>**. In this menu sequence, <serverName> corresponds to the name of the WebSphere server that you are configuring.
- 2 Select **Web Container Settings ► Web Container ► Custom Properties**.
- 3 Select **New**.
- 4 Add the **prependSlashToResource** property with a value of **true**, and click **Apply**.
- 5 Select **New**.
- 6 Add the **com.ibm.ws.webcontainer.channelwritetype** property with a value of **sync**.
- 7 Click **OK** and save your changes.

### Set Thread-Pool Properties

To set the recommended thread-pool properties for WebSphere 6.0.2 or 6.1, perform these steps. These properties apply regardless of the operating system:

- 1 In the WebSphere administrative console, select **Servers ► Application servers ► <serverName>**. In this menu sequence, <serverName> corresponds to the name of the WebSphere server that you are configuring.
- 2 Select **Thread Pools ► Web Container**.
- 3 Change the minimum size value from **10** to **35**.
- 4 Change the maximum size value from **50** to **35**.
- 5 Click **OK** and save your changes.

---

## Set Tuning Values for the AIX Operating System

To configure the recommended settings when WebSphere 6.0.2 or 6.1 runs on the AIX operating system, perform these steps. These steps enable large-page support and ensure that user resource limits are set to maximum levels:

- 1 As the root user, enter the following commands to reserve a large-page memory area of 4 GB with a page size of 16MB. Then, reboot the system:

```
vmo -r -o lgpg_regions=256 -o lgpg_size=16777216
bosboot -ad /dev/ipldevice
reboot -q
```

- 2 After you reboot, enter the following command to enable large page support on AIX:

```
vmo -p -o v_pinshm=1
```

- 3 As the root user, enter the following command. This command sets the capabilities for the account under which WebSphere runs:

```
chuser capabilities=CAP_BYPASS_RAC_VMM,CAP_PROPAGATE <WebSphereUserID>
```

In the command, replace <WebSphereUserID> with the name of the account under which WebSphere runs.

- 4 Add the **-x1p** Java option to the WebSphere generic JVM options. Perform these steps:
  - a In the WebSphere administrative console, select **Servers ► Application servers ► <serverName>**. In this menu sequence, <serverName> corresponds to the name of the WebSphere server that you are configuring.
  - b Select **Java and Process Management ► Process Definition ► Java Virtual Machine**.
  - c In the **Generic JVM Argument** field, add **-x1p**. **-x1p** enables large-page support in Java.
- 5 Enter the following commands:

```
ulimit -f unlimited
ulimit -m unlimited
```

The **-f** option in the **ulimit** command sets the maximum size of the files that are created. The **-m** option sets the maximum resident set size.

---

## Sample Middle-Tier Deployment Scenarios

---

### Overview of Middle-Tier Deployment Scenarios

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

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:

- “Criteria for Choosing a Middle-Tier Configuration” on page 71
- “Scenario 1: Web Applications Deployed in a Single J2EE Application Server” on page 75
- “Scenario 2: Static Content Deployed in an HTTP Server Proxy” on page 77
- “Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster” on page 79
- “Additional Considerations for Planning a Deployment” on page 81

The scenarios that are presented here range from simple to complex. Scenario 1 represents the deployment that results from running the SAS Software Navigator and the SAS Configuration Wizard to install and configure middle-tier components. 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 same SAS server tier, which consists of a SAS Metadata Server that resides on a dedicated machine, and additional systems that run various



SAS application servers, including SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers.

---

## Criteria for Choosing a Middle-Tier Configuration

Before you can design a middle-tier configuration that is suitable for your organization, you must understand your organization's requirements. For example, the size of your organization and the level of security that your organization requires are two factors that you should consider. There are additional considerations that must be made as well in order to provide a robust, scalable, and secure middle-tier environment.

In addition to defining clear requirements, you should identify the priorities of those requirements and determine whether there are any conflicting requirements. For example, you might want a high level of security, but high security can be expensive to maintain and can restrict the scalability of the system. Establishing priorities for potentially competing objectives will facilitate your decision-making process. The information in the following tables can help you identify and clarify requirements.

*Note:* The implementation considerations in the following tables are all described in the remaining sections of this topic.  $\Delta$

## Security

Here are some things to consider when you plan your security implementation:

**Table 4.1** Security Criteria

Category	Questions to Ask	Implementation Considerations
Physical access	<p>How will clients physically access your application?</p> <p>Will they need external access through the Internet—that is, from outside your corporate firewall?</p> <p>Will access come from workstations and other client devices that reside on your intranet?</p> <p>Will external access require or support access through a virtual private network (VPN) ?</p>	<p>Whether and how to use firewalls to protect internal components.</p> <p>Whether and where to use the Secure Sockets Layer (SSL) protocol.</p>
Authentication and authorization	<p>Is it acceptable for all members of your organization to look at your organization's data and reports?</p> <p>Do any resources require access control?</p> <p>Do you have more than one type of user?</p> <p>Do you need to limit access to certain types of users?</p> <p>Will people outside your organization have access to some of your data?</p> <p>What is the business cost that you will incur if your sensitive data is accessed by the wrong people?</p>	<p>Whether there is a need to authenticate users in order to support authorization.</p>
Threats	<p>What threats concern you the most?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> External unauthorized entry?</li> <li><input type="checkbox"/> Internal unauthorized access to sensitive data?</li> <li><input type="checkbox"/> Packet sniffing to gain unauthorized information?</li> <li><input type="checkbox"/> Denial of service attacks?</li> <li><input type="checkbox"/> Modification or falsification of sensitive data?</li> </ul>	<p>Whether and how to use firewalls to protect internal components.</p> <p>Whether and where to use SSL.</p>

## Availability

Here are some things to decide about the level of availability that you provide to Web application users:

**Table 4.2** Availability Criteria

<b>Category</b>	<b>Questions to Ask</b>	<b>Implementation Considerations</b>
Planned downtime	<p>Does the deployment require some sort of fault isolation so that one system failure won't cripple all usage?</p> <p>How much downtime can your application tolerate?</p> <p>Will you have planned periodic outages, or will the system be expected to be up continuously?</p>	<p>Whether to use a J2EE application server cluster and bring down only one server at a time.</p> <p>Whether to use clustered HTTP proxy servers.</p>
Unplanned outages	<p>How fast must the system be brought back up after an unplanned outage?</p> <p>What is the business cost associated with unplanned failure?</p>	<p>Whether to use a J2EE application server cluster, so that if a server goes down, the remaining servers in the cluster can continue with limited functionality.</p> <p>Whether to use clustered HTTP proxy servers.</p>

## Performance and Scalability

Here are some considerations related to how well your Web applications handle the load of expected users:

**Table 4.3** Performance and Scalability Criteria

Category	Questions to Ask	Implementation Considerations
Responsiveness	<p>What kind of response time is acceptable to your expected users?</p> <p>What is the business cost associated with delays?</p>	<p>Whether to use a J2EE application server cluster to maintain an even load across all servers.</p> <p>Whether to use clustered HTTP proxy servers.</p> <p>Whether to use load-balancing hardware or software.</p>
Number of users	<p>Does the deployment need to support large numbers (hundreds) of concurrent users?</p> <p>Will requests come in as a steady stream, or will you have a burst of traffic at certain points in the business cycle, such as a particular time of the day or a particular day of the month?</p>	<p>Whether to use a J2EE application server cluster.</p> <p>Whether to use clustered HTTP proxy servers.</p> <p>Whether to use load-balancing hardware or software.</p>
Extensibility	<p>Do you anticipate growth in usage over time?</p> <p>How quickly must your production application environment adapt to change?</p>	<p>Whether to use an HTTP server as a proxy to the J2EE application server.</p> <p>Whether to use a J2EE application server cluster.</p> <p>Whether to use clustered HTTP proxy servers.</p> <p>Whether to use load-balancing hardware or software.</p>

## Maintainability

These considerations help you quantify how difficult it will be to maintain the system:

**Table 4.4** Maintainability Criteria

Category	Questions to Ask	Implementation Considerations
Maintenance	<p>Who will maintain your system?</p> <p>Will the maintenance staff consist of one person or a team of people?</p> <p>What level of monitoring will take place?</p> <p>Will automated processes be required in order to deploy and configure components of the system?</p>	Whether you can maintain a particular level of security, scalability, and availability.
Upgrades	<p>How will new components and updates be introduced?</p> <p>Will there be a test environment for staging modifications and updates?</p> <p>How much manual work is involved with upgrades?</p>	Whether upgrades will require more resources than you have.

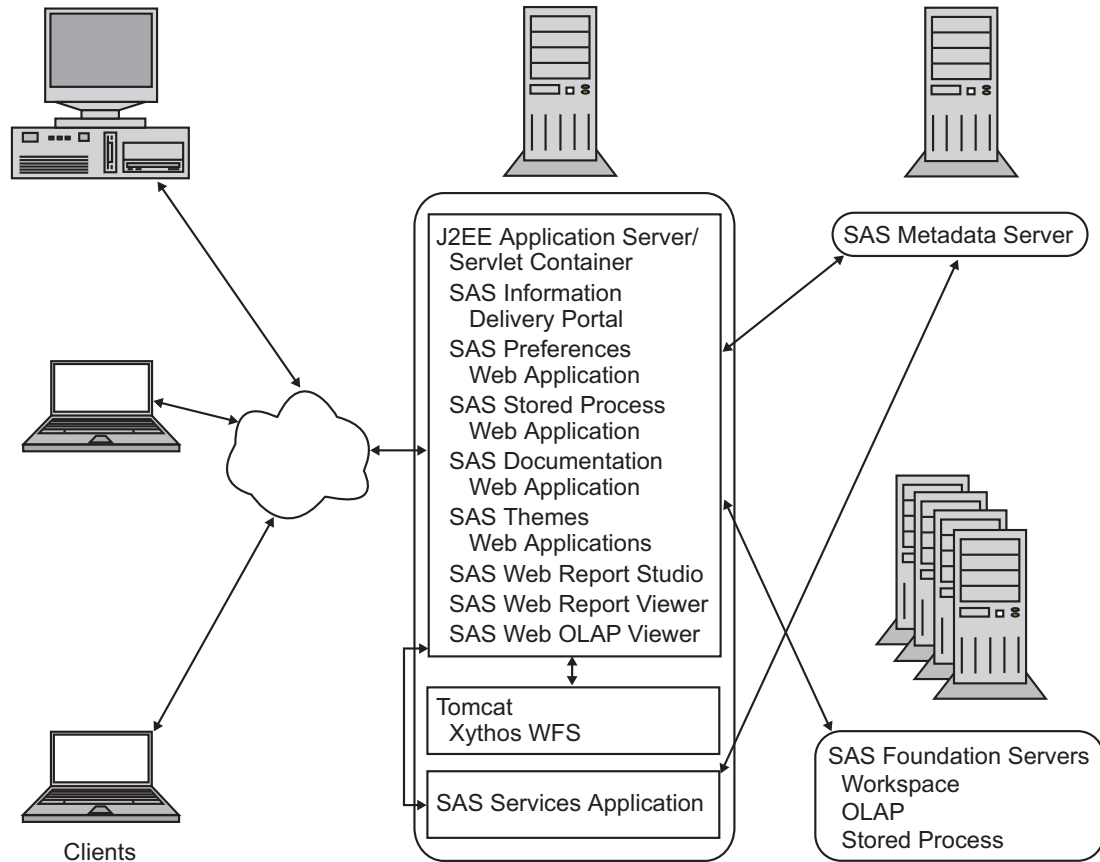
## Scenario 1: Web Applications Deployed in a Single J2EE Application Server

Scenario 1 illustrates the most basic setup, one in which all of the SAS middle-tier components are installed on a single system.

*Note:* This scenario represents the deployment that results from running a planned Advanced installation of the middle tier using the SAS Software Navigator. If you want more sophisticated configurations, such as those presented in Scenarios 2 and 3, then you must manually modify your deployment after you run the SAS tools.  $\Delta$

The following figure illustrates the configuration for Scenario 1.

Figure 4.1 Scenario 1: Middle Tier on a Single System



After installation, the system contains the following software:

- J2EE application server or servlet container (BEA WebLogic, IBM WebSphere, or Apache Tomcat)
- The following SAS Web applications, which run in the J2EE application server or servlet container:
  - SAS Information Delivery Portal (or a portal Web application that you develop using the SAS Web Infrastructure Kit)
  - SAS Preferences Web application
  - SAS Themes Web applications
  - SAS Stored Process Web application
  - SAS Documentation Web application
  - SAS Web Report Viewer
  - SAS Web Report Studio
  - SAS Web OLAP Viewer
- WebDAV content services, which run in a separate Tomcat container
 

The Xythos WebFile Server (WFS) is a WebDAV server that is configured by default to run in its own separate Tomcat servlet container. Xythos WFS requires a database, but this detail is not shown in the diagram. Xythos WFS can be configured with a PostgreSQL, IBM DB2, Oracle, or Microsoft SQL Server database.
- The SAS Services application, which runs in a separate Java Virtual Machine process

This type of deployment requires little additional configuration or tuning when you use an appropriately designed SAS planning file.

Following are the advantages and disadvantages of using this scenario:

- Advantages:
  - is easy to set up and configure
  - is least expensive of the three scenarios
  - is ideal for a development or test environment where frequent changes might be required
- Disadvantages:
  - does not adequately support large numbers (hundreds) of concurrent users
  - is less secure and scalable than the other scenarios
  - has no provision for improving performance
  - has no provision for serving users during planned or unplanned downtime

---

## Scenario 2: Static Content Deployed in an HTTP Server Proxy

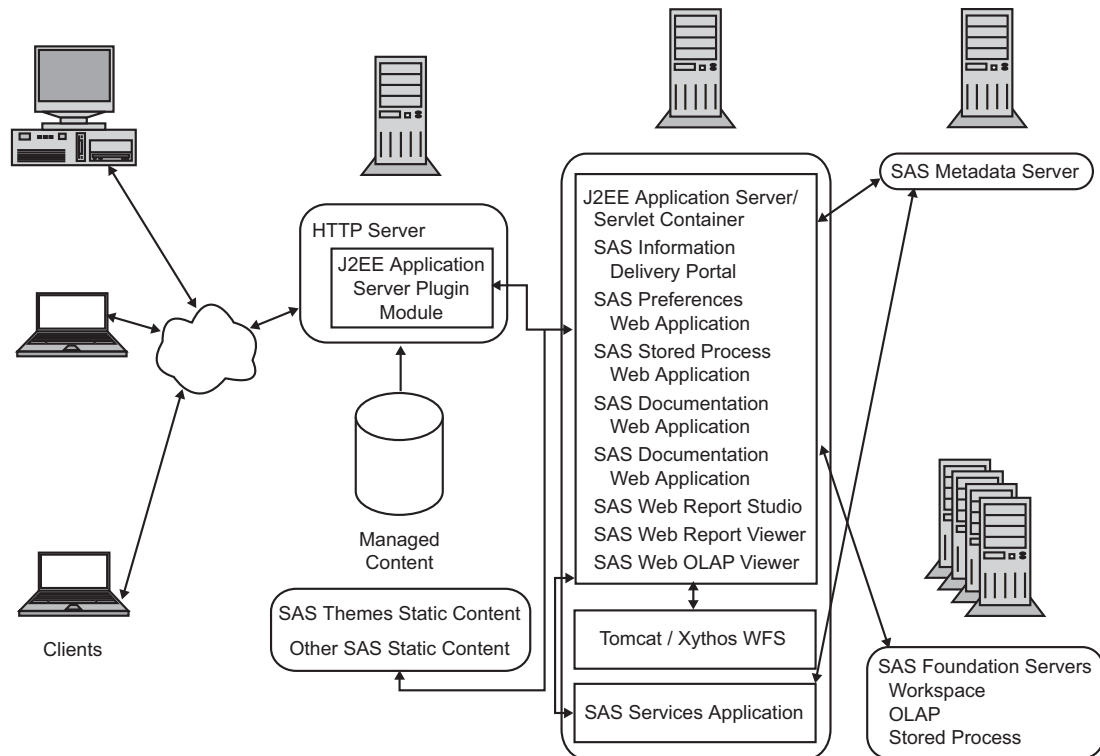
This configuration delivers static HTML content to clients from a separate HTTP server, such as Apache HTTP Server or Microsoft Internet Information Services (IIS), rather than from the J2EE application server or servlet container.

When a browser requests static content (for example, an HTML file) from a server, the server simply returns the requested document to the browser, and the browser displays the document. Dynamic content requests, however, involve some type of data manipulation that the server must perform before returning the requested page.

By design, an HTTP server is ideally suited for handling static content efficiently. Allowing an HTTP server to handle as much static content as possible enables the J2EE application server to concentrate on dynamic, more resource-intensive content.

The following figure illustrates the configuration for Scenario 2.

Figure 4.2 Scenario 2: Using a Proxy HTTP Server



In a typical configuration, the HTTP server acts as a proxy that handles all client requests for static content, and forwards requests for dynamic content to the J2EE application server or servlet container. The J2EE application server provides a module or plug-in that enables communication with the HTTP server. In this configuration, the J2EE application server is not directly exposed to clients.

## Static Content to Deploy for SAS Web Applications

SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal include static content in the form of HTML files, images, cascading style sheets, XML files, and scripts. The SAS Information Delivery Portal also contains two or more SAS Themes applications, which consist entirely of static files that control the portal's design and appearance (you must unpack the applications before you can deploy the static content). For information about deploying this static content, see "Configuring an HTTP Server to Serve Static Content for SAS Web Applications" on page 86.

*Note:* If you deploy SAS static content in an HTTP server, then you must be sure to redeploy this content if you later install a SAS software upgrade that includes new files for the static content.  $\Delta$

## Advantages and Disadvantages of Using Scenario 2

Following are the advantages and disadvantages of using this scenario:

- Advantages:
  - provides faster response time for static content
  - stores all configuration details about Web applications in a central location (the HTTP server), so it is easier to move Web applications to different machines



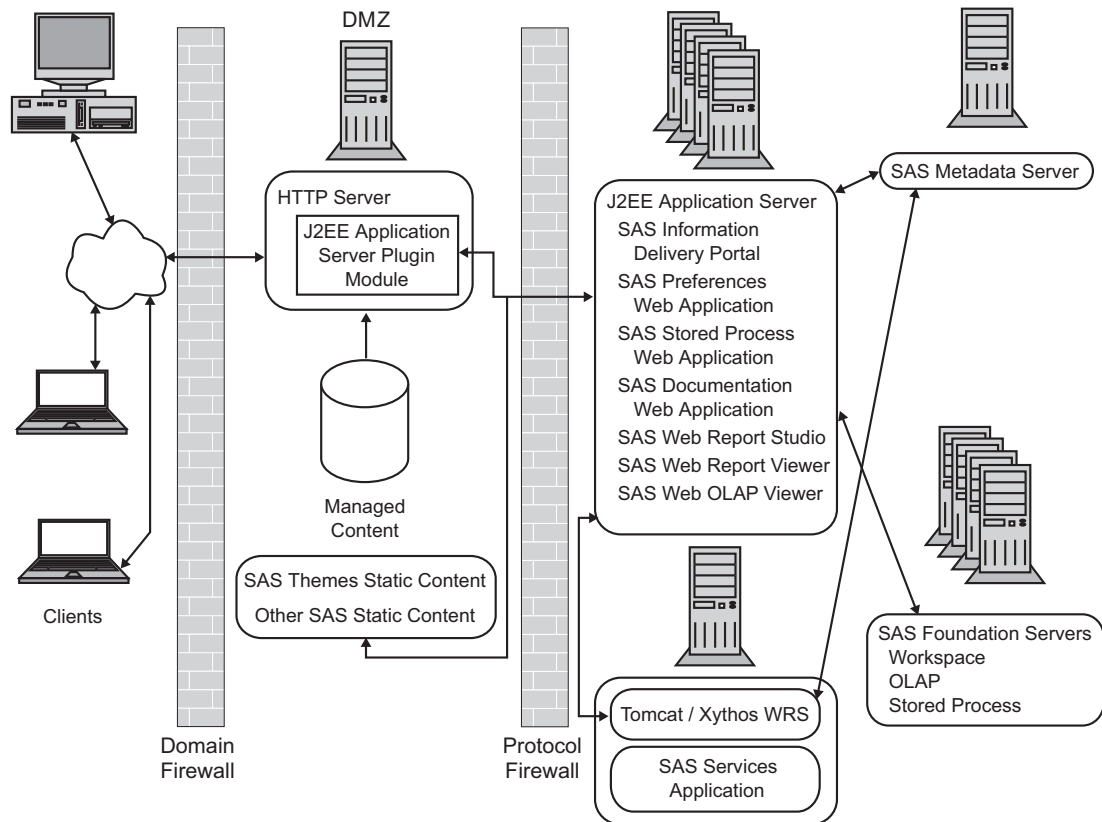
- protects the J2EE application server better because it is not exposed directly to clients
- can be scaled to support a cluster of application servers and a demilitarized zone (described in the next scenario)
- Disadvantages:
  - requires multiple network hops and multiple servers for the proxy model to handle each dynamic request and each response
  - is not configured using the SAS Configuration Wizard; must be manually maintained, and must be manually upgraded in the event of a new SAS release

### Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster

Scenario 3 includes a cluster of J2EE application servers in a deployment that implements a secure demilitarized zone (DMZ).

The following figure illustrates the configuration for Scenario 3. Note that the middle-tier software is now distributed across multiple machines. The entire column of nodes (J2EE application server and SAS Web applications, SAS Services application, and Xythos) represents the middle-tier.

**Figure 4.3** Scenario 3: Clustered J2EE Application Servers and a Demilitarized Zone



*Note:* As indicated in the figure, if you configure a cluster of J2EE application servers, then you must deploy all the SAS Web applications to all nodes of the cluster.  $\Delta$

In the figure, note that the SAS Services Application resides on a host that is separate from the cluster of J2EE application servers. This separation serves only to illustrate that the SAS Services Application is not replicated along with the other Web applications that are deployed across the cluster. The SAS Services Application could just as well reside on any of the hosts in the cluster. If you choose to deploy the SAS Services Application on a separate host, be aware that this deployment requires additional configuration after the initial installation. For instructions, see “Redistributing the SAS Services Application (and Java RMI Server)” on page 348.

## Understanding Clusters

In order to provide greater scalability, availability and robustness, WebLogic and WebSphere support some form of clustering. With clustering, multiple J2EE 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 an HTTP proxy server for static content (see “Scenario 2: Static Content Deployed in an HTTP Server Proxy” on page 77).

Individual J2EE application server instances in a cluster can coexist on the same machine, or they can be distributed across a group of server machines. Each distribution mode has advantages and disadvantages, and you should evaluate these carefully when designing your deployment.

A different approach to load distribution involves merely deploying individual Web applications on separate, non-clustered J2EE 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, J2EE 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.

*Note:* The Apache Tomcat version that is used in the current release of the SAS Intelligence Platform provides only limited support for clustering. For this reason, we recommend that you use WebLogic or WebSphere to implement clustering.  $\Delta$

## Requirement for Session Affinity

For SAS Web Report Studio and SAS Information Delivery Portal to be deployed into a clustered environment, the J2EE application servers must implement session affinity. *Session affinity* is an association between a J2EE application server and a client that requests an HTTP session with that server. With session affinity, once a client has been assigned to a session with an application server, the client remains with that server for the duration of the session. This association is known in the industry by several terms, including session affinity, server affinity, and sticky sessions. By default, session affinity is enabled in WebSphere and WebLogic.

*Why session affinity is required:* Although WebSphere and WebLogic 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 J2EE 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 buffers the servers from

clients, whether those clients reside within the organization's computing infrastructure 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. Only a few ports are typically allowed to be opened through this firewall. Often, only HTTP and HTTPS connections using the standard ports 80 and 443 respectively can be opened. 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 named the Protocol Firewall sits between these non-secure machines and the secure server network. The Protocol Firewall might allow a few additional ports to be open, but the range of machines allowed to make connections is typically restricted to the servers that reside in the DMZ.

The DMZ usually contains HTTP servers, proxy servers, and load-balancing proxy software. The DMZ should not contain your application servers or any SAS servers that handle important business logic, data, or metadata.

If your applications will be accessed by clients through the Internet, then you should include a DMZ as part of your deployment in order to safeguard critical information. To be extra safe, you might want to implement a DMZ even when user workstations and client devices connect to secure servers within your organization's intranet.

### Advantages and Disadvantages of Using Scenario 3

Following are the advantages and disadvantages of using this scenario:

□ Advantages:

- provides a level of fault isolation that is not possible with a single J2EE application server. If a node fails in a cluster, only those users on that node will be affected by the failure. For example, if you have an even distribution of users on a three-node cluster, only one third of the users would be affected by the loss of a single node.
- provides a higher level of security for environments where external clients connect over the Internet, because the back-end servers are isolated from clients by the DMZ.
- provides better overall performance because of clustering.

□ Disadvantages:

- is more expensive than the other scenarios.
- is not configured using the SAS Software Navigator and SAS Configuration Wizard; must be manually maintained, and must be manually upgraded in the event of a new SAS release.
- is unsuited to the version of Tomcat that is used by the current release of SAS Intelligence Platform because that version does not adequately support a clustered environment. You should use WebSphere or WebLogic.

*Note:* While SAS does not support Tomcat in the clustered configuration that is described here, you can set up a DMZ and HTTP server proxy to work with a single Tomcat server. △

---

## Additional Considerations for Planning 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 J2EE application servers are clustered to balance the load and to provide increased availability. While this scenario provides redundancy for the application servers, the HTTP servers remain as potential bottlenecks and single points of failure. You can also distribute HTTP traffic across multiple HTTP servers 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 HTTP servers.

A number of vendors sell load-balancing software and hardware products for HTTP servers, including IBM, Cisco, and Nortel Networks. 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 will probably 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/private key pair. Each of your servers that handles encrypted communications will need to manage 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 could 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-sniffing by Internet users.
  - Encrypt all traffic that moves between the client and your HTTP proxy server that resides in the DMZ.
  - Always encrypt credit card numbers, social security numbers, and any other sensitive information.

To achieve strong security, encrypt links all the way to the J2EE application server. If you are concerned about internal packet sniffing, you could encrypt everything. However, total encryption comes with a cost, as explained in the remaining considerations below.

- 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.

For more information about setting up SAS Information Delivery Portal and other Web applications to use SSL, see “Configuring the Web Applications for Secure Sockets Layer (SSL)” on page 42.

## Middle-Tier (Trusted Web) Authentication

By default, the SAS Web applications authenticate users on the metadata server. You can configure the Web applications to authenticate on the middle tier instead. When users log on to a Web application that is configured to authenticate on the middle tier, a Web server or a servlet container handles the initial authentication. First, the Web server’s authentication provider verifies the user’s identity. Then, the Web application uses a trusted user connection to access the metadata server.

There are several places where middle-tier authentication (sometimes called Web authentication) can occur. The servlet container, an HTTP server, or a proxy server can take responsibility for authentication. In general, it is more secure to perform authentication in the DMZ, as close to the client as possible. It’s best to filter out all unauthorized requests before they reach your secure servers. If a proxy HTTP server is used, it can authenticate users and pass the user credentials on to the J2EE application server or servlet container as part of the HTTP request packet.

Performing authentication in the DMZ also facilitates single sign-on. Most likely, your organization will have several applications behind a common set of proxy and HTTP servers. By having a common server handle authentication, users will not need to re-authenticate for each back-end application.

You incur some limitations when you use middle-tier authentication. For example, the SAS Information Delivery Portal’s Public Kiosk is not available when you authenticate users on the middle tier. In addition, you cannot customize authentication based on your own business logic. If either of these are important capabilities, then you might prefer to use the default host method of authentication.

For more information, see the following topics:

- For a description of how trusted authentication works, see “Initial Authentication on a Web Application Server” in the “Understanding Authentication” section in the *SAS Intelligence Platform: Security Administration Guide*.
- For instructions on setting up the middle-tier applications to use trusted authentication, see “Changing to Trusted Web Authentication” on page 32.
- For information about achieving a single sign-on approach to authentication, see “Understanding Single Sign-On” on page 24.

## SAS Services Application Heap Size

SAS Information Delivery Portal uses the SAS Services Application to pass session and user context to content viewers, remote portlets, and Web applications launched from the portal (for example, SAS Web Report Viewer). This Java application enables the user to pass seamlessly through to the target without the requirement for a separate logon. By default, the SAS Services Application’s JVM options are set to handle a moderate number of concurrent users, typically about 750 to 1000. For the application to support a larger number of concurrent users, you must increase its minimum and maximum heap size (`-Xms` and `-Xmx`, respectively). The application will need approximately 128 MB of additional memory for every additional 750 to 1000 users.

To increase the minimum and maximum heap size of the SAS Services Application, add the following properties to the end of the `install.properties` file, which is located in the `SAS-install-dir\Web\Portal2.0.1\PortalConfigured` directory. The values shown below will increase the minimum and maximum heap size to 256 MB.

- `$SERVICES_REMOTE_JVM_INIT_HEAP$=256`
- `$SERVICES_REMOTE_JVM_MAX_HEAP$=256`

After updating the properties file, redeploy the portal and the SAS Services Application. For instructions, see “Re-Create and Redeploy the Portal Web Application” on page 211.

---

## Configuring a Cluster of J2EE Application Servers

---

### Overview of Cluster Configuration

Cluster configuration varies widely between J2EE application server vendors. You should 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 J2EE Application Server Cluster” on page 79.

It’s 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 J2EE application server. For this configuration, you would set the address of the single-node cluster equal to the address of the HTTP server.

The following table can help you find vendor documentation about cluster configuration:

**Table 4.5** Vendor Documentation for Cluster Configuration

Product	Location of the Documentation
BEA WebLogic Server	<a href="http://e-docs.bea.com/wls/docs81/cluster/load_balancing.html#1044135">http://e-docs.bea.com/wls/docs81/cluster/load_balancing.html#1044135</a>
IBM WebSphere Application Server	<a href="http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html">http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html</a>

*Note:* SAS currently does not support Apache Tomcat clustering .  $\Delta$

The next section describes additional manual steps that you should perform if you are using WebLogic. No manual steps are necessary if you are using IBM WebSphere.

---

### Additional Manual Steps for WebLogic

If you are using BEA WebLogic, then after configuring the basic cluster, you must perform a few additional manual steps before the cluster is ready to use with the SAS Web applications.

Following are the steps to configure a WebLogic cluster for the SAS Information Delivery Portal. These steps assume that Apache has been configured as the cluster front-end. If you are using some other HTTP server, then alter the last step below as applicable:

- 1 Add the following lines to the bottom of the WebLogic `httpd.conf` file (located in the Apache installation):

```
<IfModule mod_weblogic.c>
  <Include conf/weblogic.conf />
</IfModule>
```

- 2 Create a file named `weblogic.conf` in the same directory as `httpd.conf`. Add the following lines to `weblogic.conf`:

```
WebLogicCluster commaSeparatedListOfServers
WLogFile pathToLogFile
Debug NONE
<Location /Portal>
  SetHandler weblogic-handler
  CookieName sas.portal.sessionid
</Location>
```

- 3 Save and close both files.
- 4 Make sure that each of the cluster nodes points to the Apache proxy server's host name and port. This is done by using the WebLogic Administrator Console (**Protocols ► HTTP ► Advanced Options**).

Configure a WebLogic cluster for SAS Web Report Studio, SAS Web Report Viewer, and SAS Web OLAP Viewer for Java in a similar way. Their respective location and cookie names are as follows (note that the capitalization is different for "sessionID"):

```
<Location /SASWebReportStudio>
  SetHandler weblogic-handler
  CookieName sas.wrs.sessionID
</Location>
```

```
<
Location /SASWebReportViewer>
  SetHandler weblogic-handler
  CookieName sas.wrv.sessionID
</Location>
```

```
<Location /SASWebOLAPViewer>
  SetHandler weblogic-handler
  CookieName sas.swov.sessionid
</Location>
```

---

# Configuring an HTTP Server to Serve Static Content for SAS Web Applications

---

## Overview of Configuring an HTTP Server to Serve Static Content

Your middle-tier deployment can use an HTTP server to handle requests for static content and to forward requests for dynamic content to your J2EE application server or servlet container. This configuration makes efficient use of the HTTP server, and enables your servlet container to devote its resources to dynamic content. The performance benefits are particularly notable for large-scale deployments that include a cluster of servlet containers. For an overview of this configuration, see “Sample Middle-Tier Deployment Scenarios” on page 70.

In order to offload static content to the HTTP server, all incoming traffic must be routed through the HTTP server, which then passes requests for dynamic content to the servlet container. This section provides an example of how to deploy static content on the Apache HTTP server for the following SAS Web applications:

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

There is more than way one to configure Apache, and your configuration might differ from what is presented here.

The example also illustrates the use of a cluster of J2EE application servers. This configuration helps ensure that the application that runs in the container correctly sets the address of the HTTP server as the address for all of the application’s pages. The cluster configuration also provides greater computation capacity, redundancy, and other benefits as described in “Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster” on page 79. As noted in scenario 3, Apache Tomcat currently does not support clustering. Therefore, this example would not work with Tomcat.

*Note:* Although these instructions apply to Apache HTTP Server, you can use any server that is compliant with HTTP 1.1 to deliver static content. For general information about supported third-party software, see <http://support.sas.com/documentation/configuration/thirdpartysupport/index.html>  $\triangle$

---

## Example: Setting Up Apache to Serve Static Content

### Main Steps for Setting Up Apache to Serve Static Content

Follow these steps in order to deploy static content and to configure Apache to handle that content. These steps assume that you have already installed and configured the SAS middle tier:

- 1 Verify that the SAS middle-tier components operate properly in your J2EE application server or your servlet container. One way to verify proper operation is to start SAS Web Report Studio or SAS Information Delivery Portal (preferably both, if both are installed) and log on. For instructions on starting either Web application, see the `instructions.html` document that you used during configuration of the Web application.



- 2 If you have not already done so, install Apache HTTP Server and verify that it runs properly. For more information, see the *SAS Intelligence Platform: Installation Guide*.
- 3 Install and configure the proxy plug-in that enables your J2EE application server to interact with Apache. See the documentation that is provided for BEA WebLogic Server, IBM WebSphere Application Server, or Apache Tomcat. For general information and examples, see “Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server” on page 90.
- 4 Optionally, you can configure a cache timeout value for the static content in order to reduce the requests that the browser sends to the HTTP server to check for updated content. For more information, see “Configuring Apache Cache Control for Static Content” on page 96.
- 5 Configure a cluster of J2EE application servers or servlet containers (even if the cluster contains only one node). You must set up the cluster in order to route browser requests to Apache instead of to your J2EE application server. In effect, this configuration hides the J2EE application server from public access, since all public requests will target Apache. For instructions on configuring a cluster, see “Configuring a Cluster of J2EE Application Servers” on page 84.
- 6 Stop and restart Apache HTTP Server.
- 7 Before you deploy static content, verify proper interaction between Apache HTTP Server and the SAS Web applications (SAS Web Report Studio or SAS Information Delivery Portal) that reside in the J2EE application server. For example, use the URL address and port of Apache HTTP Server to access SAS Web Report Studio. (Conversely, you should *not* be able to access SAS Web Report Studio using the URL of the J2EE application server in which SAS Web Report Studio is deployed.)
- 8 Deploy the static content to Apache HTTP Server as follows:
  - If you are deploying content for SAS Web Report Studio or SAS Web Report Viewer, then see “Deploy Static Content for SAS Web Report Studio and SAS Web Report Viewer” on page 87.
  - If you are deploying content for SAS Information Delivery Portal, then see “Deploy Static Content for SAS Information Delivery Portal” on page 88.
  - If you are deploying content for SAS Web OLAP Viewer for Java, then see “Deploy Static Content for SAS Web OLAP Viewer for Java” on page 90.
- 9 Stop and restart Apache HTTP Server.
- 10 If you deployed content for SAS Information Delivery Portal, then the configuration changes that you made in a previous step require that you stop and restart the SAS Services application. For instructions, see the **instructions.html** document that you used during configuration of the SAS Information Delivery Portal.
- 11 Stop and restart the J2EE application server or the servlet container.

## Deploy Static Content for SAS Web Report Studio and SAS Web Report Viewer

SAS Web Report Studio and SAS Web Report Viewer use static content for images, styles, themes, and scripts.

*Note:* The deployment that is described here is only one of several steps that are required to set up Apache to serve static content. You must also configure the proxy plug-in that enables your J2EE server to interact with Apache. In addition, you must configure a cluster of J2EE application servers (even if the cluster contains only one node). To view all of the required steps, see “Main Steps for Setting Up Apache to Serve Static Content” on page 86. △

To deploy this static content to Apache HTTP Server, follow these instructions:

- 1 In the Apache **htdocs** directory, create a directory to hold SAS Web Report Studio contents. For example, create a directory named **SASWebReportStudio**.
- 2 Locate the following four directories that contain SAS Web Report Studio static content:
  - images**
  - scripts**
  - styles**
  - themes**

These directories reside in the **wrspackaging** directory of your SAS Web Report Studio installation. For example, on Windows the directory might be **C:\Program Files\SAS\SASWebReportStudio\3.1\wrspackaging**.

- 3 Copy the four directories and all their contents to the directory that you created in Apache.
- 4 Deploy the static content for SAS Web Report Viewer in a similar way. Create a **SASWebReportViewer** directory in Apache, and then copy the four same directories from the SAS Web Report Viewer installation.

Apache reads these files in the location that you created. For example, if the URL for SAS Web Report Studio is **http://www.yourdomain.com/SASWebReportStudio/**, then the static content for all images is served from **http://www.yourdomain.com/SASWebReportStudio/images/**.

*Note:* After you deploy static content to Apache, you must be sure to redeploy this content if you later install a SAS Web Report Studio or SAS Web Report Viewer upgrade that includes new files for the static content.  $\triangle$

## Deploy Static Content for SAS Information Delivery Portal

SAS Information Delivery Portal uses static content for images, styles, themes, and scripts.

*Note:* The deployment that is described here is only one of several steps that are required to set up Apache to serve static content. You must also configure the proxy plug-in that enables your J2EE server to interact with Apache. In addition, you must configure a cluster of J2EE application servers (even if the cluster contains only one node). To view all of the required steps, see “Main Steps for Setting Up Apache to Serve Static Content” on page 86.  $\triangle$

To deploy static content to Apache HTTP Server, follow these instructions:

- 1 In the Apache **htdocs** directory, create the following directories (use the spelling and capitalization shown below):
  - SASTheme\_default**
  - SASTheme\_winter**
  - sasweb\graph**
  - Portal\images**
  - Portal\scripts**
  - Portal\styles**
  - Portal\themes**
  - SASStoredProcess\images**
  - SASStoredProcess\scripts**
  - SASStoredProcess\themes**

- 2 Locate the **Portal** directory in your portal installation directory. For example, on Windows the default portal installation directory is **C:\Program Files\SAS\Web\Portal2.0.1**.  
Copy all subdirectory files as follows:
  - Copy the contents of **Portal\images** to the **htdocs\Portal\images** directory that you created.
  - Copy the contents of **Portal\scripts** to the **htdocs\Portal\scripts** directory that you created.
  - Copy the contents of **Portal\styles** to the **htdocs\Portal\styles** directory that you created.
  - Copy the contents of **Portal\themes** to the **htdocs\Portal\themes** directory that you created.
- 3 Locate the **SASStoredProcess** directory in your portal installation directory.  
Copy all subdirectory files as follows:
  - Copy the contents of **SASStoredProcess\images** to the **htdocs\SASStoredProcess\images** directory that you created.
  - Copy the contents of **SASStoredProcess\scripts** to the **htdocs\SASStoredProcess\scripts** directory that you created.
  - Copy the contents of **SASStoredProcess\themes** to the **htdocs\SASStoredProcess\themes** directory that you created.
- 4 Copy the contents of the **sasweb\graph** directory, located in the portal installation directory, into the **htdocs\sasweb\graph** directory that you created.
- 5 Deploy the static theme contents as follows:
  - a Unpack the **SASTheme\_default.war** file, located in the portal installation directory, into the **htdocs\SASTheme\_default** directory that you created.  
*Note:* It is recommended that you use the Java **jar** command to unpack the **WAR** file. △
  - b Modify and run the appropriate SAS program so that the URL for the theme points to Apache HTTP Server:
    - If you have already deployed this theme to your servlet container, then modify and run **UpdateThemeConnection.sas**. For instructions, see “Redistributing the SAS Themes Web Application” on page 351.
    - If you are deploying this theme for the first time, then modify and run **LoadThemeConnection.sas**. For instructions, see “Theme Deployment” on page 328.
  - c Repeat the previous two steps for the **SASTheme\_winter.war** file. Unpack the file into the **htdocs\SASTheme\_winter** directory that you created; then modify and run the appropriate SAS program for this theme.  
*Note:* If you have previously deployed any additional custom themes that your organization has developed, repeat the above steps for those themes as well. △
  - d Optionally, remove the themes from the servlet container. To remove the themes, manually delete the theme files (Tomcat) or use the administrator console (WebLogic and WebSphere) to remove the themes. One reason to remove the themes is to avoid possible confusion if you upgrade to a new release or service pack. If the themes reside on both the servlet container and the HTTP server, then it's possible that you might accidentally update the themes in the wrong location.

*Note:* After you deploy static content to Apache, you must be sure to redeploy this content if you later install a SAS portal upgrade that includes new files for the static content.  $\Delta$

## Deploy Static Content for SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java uses static content for images, styles, scripts, and templates.

*Note:* The deployment that is described here is only one of several steps that are required to set up Apache to serve static content. You must also configure the proxy plug-in that enables your J2EE server to interact with Apache. In addition, you must configure a cluster of J2EE application servers (even if the cluster contains only one node). To view all of the required steps, see “Main Steps for Setting Up Apache to Serve Static Content” on page 86.  $\Delta$

To deploy this static content to Apache HTTP Server, follow these instructions:

- 1 In the Apache **htdocs** directory, create a directory to hold SAS Web OLAP Viewer for Java contents. For example, create a directory named **SASWebOLAPViewer**.
- 2 Locate the following directories that contain SAS Web OLAP Viewer for Java static content:
  - images**
  - scripts**
  - styles**
  - templates**

These directories reside in the **SASWebOLAPViewer** directory of your SAS Web OLAP Viewer for Java installation.

- 3 Copy the directories and all their contents to the directory that you created in Apache.

After you deploy static content to Apache, you must be sure to redeploy this content if you later install a SAS Web OLAP Viewer for Java upgrade that includes new files for the static content.

---

## Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server

BEA WebLogic Server, IBM WebSphere Application Server, and Apache Tomcat servlet container 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 J2EE application server or servlet container. In this scenario, the HTTP server handles all the static content and relies on the J2EE application server for dynamic content.
- To forward requests and distribute those requests among a cluster of J2EE application servers using a load-balancing algorithm. For this functionality, we recommend that you use WebLogic or WebSphere rather than Tomcat.

The plug-in enables the HTTP server to behave as a proxy server, which typically filters requests and passes requests that meet the filter requirements to the J2EE application server. Many proxy servers use a local cache of Web pages to respond to requests.

This chapter does not contain the instructions for configuring the plug-ins. The configuration instructions vary greatly depending on your particular architecture. For the best information, see the vendor's documentation. The following table can help you find that documentation:

**Table 4.6** Vendor Documentation for the Proxy Plug-In

Product	Plug-In Configuration File	Location of the Documentation
BEA WebLogic Server	<code>mod_wl</code>	<a href="http://e-docs.bea.com/wls/docs81/plugins/overview.html">http://e-docs.bea.com/wls/docs81/plugins/overview.html</a>
IBM WebSphere Application Server	<code>plugin-cfg</code>	<a href="http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsv_plugin.html">http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsv_plugin.html</a>
Apache Tomcat	<code>mod_jk</code>	<a href="http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/jk.html">http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/jk.html</a>

**CAUTION:**

When using WebLogic with an Apache HTTP server, you must explicitly enable the pooling of connections between the WebLogic server and the Apache plug-in. You enable pooling by setting the `KeepAliveEnabled` parameter to *ON* in the Apache plug-in configuration. By default, that parameter is set to *OFF*. (The vendor documentation is misleading about this.) If `KeepAliveEnabled` is set to *OFF*, then you can experience session failures under heavy load conditions.  $\Delta$

The following sections provide two sample configurations for Apache HTTP Server. Both configurations assume that all static content has been copied to the appropriate subdirectory of the Apache `htdocs` directory. See “Configuring an HTTP Server to Serve Static Content for SAS Web Applications” on page 86.

## Sample: Proxy Setup for SAS Web Report Studio

This sample shows the configuration for an Apache HTTP Server that proxies SAS Web Report Studio static content to a WebLogic cluster.

```
WebLogicCluster host1.yourdomain.com:7101,host2.yourdomain.com:7101
WLLogFile c:\apps\Apache2\logs\proxy_debug.log
Debug OFF

# Instruct Apache to forward WRS requests to BEA
<Location /SASWebReportStudio>
    CookieName sas.wrs.sessionID
    SetHandler weblogic-handler
</Location>

# Override static content (images)
<Location /SASWebReportStudio/images>
    SetHandler default-handler
</Location>
```

```

# Override static content (styles)
<Location /SASWebReportStudio/styles>
  SetHandler default-handler
</Location>

# Override static content (scripts)
<Location /SASWebReportStudio/scripts>
  SetHandler default-handler
</Location>

# Override static content (themes)
<Location /SASWebReportStudio/themes>
  SetHandler default-handler
</Location>

```

---

## **Sample: Proxy Setup for SAS Web OLAP Viewer for Java**

This sample shows the configuration for an Apache HTTP Server that proxies SAS Web OLAP Viewer for Java static content to a WebLogic cluster.

```

#Configuremod_wl
# http://e-docs.bea.com/wls/docs81/plugins/apache.html
WebLogicCluster host1.yourdomain.com
WLogicPort 7101
Debug OFF

# Instruct Apache to forward requests to BEA
<Location /SASWebOLAPViewer>
  CookieName sas.swov.sessionid
  SetHandler weblogic-handler
</Location>

# Override static content (images)
<Location /SASWebOLAPViewer/images>
  SetHandler default-handler
</Location>

# Override static content (styles)
<Location /SASWebOLAPViewer/styles>
  SetHandler default-handler
</Location>

# Override static content (scripts)
<Location /SASWebOLAPViewer/scripts>
  SetHandler default-handler
</Location>

# Override static content (templates)
<Location /SASWebOLAPViewer/templates>
  SetHandler default-handler
</Location>

```

---

## Sample: Proxy Setup for the Portal and Related Web Applications

This sample shows the configuration for an Apache HTTP Server that proxies static content to a WebLogic cluster. This sample includes static content for SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal.

```
# Configure mod_wl
# http://e-docs.bea.com/wls/docs81/plugins/apache.html

LoadModule weblogic_module "C:/bea/weblogic81/server/bin/mod_wl_20.so"

WebLogicCluster host1.yourdomain.com:7101,host2.yourdomain.com:7101
WLLogFile c:\apps\Apache2\logs\proxy_debug.log
Debug OFF

# Instruct Apache to forward Portal requests to WebLogic
<Location /Portal>
    CookieName sas.portal.sessionid
    SetHandler weblogic-handler
</Location>

# The following entries override the location of static
# content to Apache. Enable them only if you have manually
# copied the static content to the appropriate
# subdirectories in Apache.

<Location /Portal/images>
    SetHandler default-handler
</Location>

<Location /Portal/scripts>
    SetHandler default-handler
</Location>

<Location /Portal/styles>
    SetHandler default-handler
</Location>

<Location /Portal/themes>
    SetHandler default-handler
</Location>

# Instruct Apache to forward Stored Process requests to WebLogic
<Location /SASStoredProcess>
    CookieName sas.stp.sessionid
    SetHandler weblogic-handler
</Location>

# The following entries override the location of static
# content to Apache. Enable them only if you have manually
# copied the static content to the appropriate
```

```
# subdirectories in Apache.

<Location /SASStoredProcess/images>
    SetHandler default-handler
</Location>

<Location /SASStoredProcess/scripts>
    SetHandler default-handler
</Location>

<Location /SASStoredProcess/themes>
    SetHandler default-handler
</Location>

# Instruct Apache to forward Preferences requests to WebLogic
<Location /SASPreferences>
    CookieName sas.preferences.webapp.sessionid
    SetHandler weblogic-handler
</Location>

# Instruct Apache to forward SASDoc requests to WebLogic
<Location /SASDoc>
    SetHandler weblogic-handler
</Location>

# By default, Apache will use the default-handler for
# content manually copied to the appropriate
# subdirectories of the Apache htdocs directory.
# The contents of sasweb, SASTheme_default, and SASTheme_winter
# should be unpacked and copied into these subdirectories.
# If instead, they are deployed as a WAR file into
# WebLogic, a Location entry must be added using the
# weblogic-handler.

# Instruct Apache to forward SAS Web Report Viewer
# requests to WebLogic
<Location /SASWebReportViewer>
    CookieName sas.wrv.sessionID
    SetHandler weblogic-handler
</Location>

# The following entries override the location of static
# content to Apache. Enable them only if you have manually
# copied the static content to the appropriate
# subdirectories in Apache.

<Location /SASWebReportViewer/images>
    SetHandler default-handler
</Location>

<Location /SASWebReportViewer/styles>
    SetHandler default-handler
</Location>
```



```

<Location /SASWebReportViewer/scripts>
    SetHandler default-handler
</Location>

<Location /SASWebReportViewer/themes>
    SetHandler default-handler
</Location>

# Instruct Apache to forward SAS Web Report Studio
# requests to WebLogic
<Location /SASWebReportStudio>
    CookieName sas.wrs.sessionID
    SetHandler weblogic-handler
</Location>

# The following entries override the location of static
# content to Apache. Enable them only if you have manually
# copied the static content to the appropriate
# subdirectories in Apache.

<Location /SASWebReportStudio/images>
    SetHandler default-handler
</Location>

<Location /SASWebReportStudio/styles>
    SetHandler default-handler
</Location>

<Location /SASWebReportStudio/scripts>
    SetHandler default-handler
</Location>

<Location /SASWebReportStudio/themes>
    SetHandler default-handler
</Location>

# Instruct Apache to forward SAS Web OLAP Viewer
# requests to WebLogic
<Location /SASWebOLAPViewer>
    CookieName sas.swovj.sessionid
    SetHandler weblogic-handler
</Location>

# The following entries override the location of static
# content to Apache. Enable them only if you have manually
# copied the static content to the appropriate
# subdirectories in Apache.

<
Location /SASWebOLAPViewer/images>
    SetHandler default-handler
</Location>

<Location /SASWebOLAPViewer/styles>

```

```

        SetHandler default-handler
    </Location>

    <Location /SASWebOLAPViewer/scripts>
        SetHandler default-handler
    </Location>

    <Location /SASWebOLAPViewer/templates>
        SetHandler default-handler
    </Location>

```

---

## Configuring 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 timeout 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 resent 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 SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal is not modified often, most of these requests are unnecessary.

When you specify a cache timeout 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 timeout 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 timeout period elapses within the session.

You can configure Apache HTTP Server to set cache timeout values for static content regardless of whether Apache HTTP Server is configured to serve that static content or is merely a front-end proxy to your J2EE application server.

*Note:* Although these instructions apply to Apache HTTP Server, you can use any server that is compliant with HTTP 1.1 and that supports timeout values for static content.  $\triangle$

The configuration instructions can vary greatly depending on your particular architecture. In general, you must complete these steps:

- 1 Create a configuration file for the Apache HTTP Server, or edit an existing configuration file. If you configured Apache HTTP Server to serve static content, then you can add the cache timeout settings to that configuration file.
- 2 In the configuration file, provide directives to load the cache timeout module:
 

```

LoadModule expires_module modules/mod_expires.so

LoadModule headers_module modules/mod_headers.so

```
- 3 Specify the timeout value for each static content location.

The following example shows the configuration for an Apache HTTP Server that serves SAS Web Report Studio static content with a cache timeout value and proxies traffic to a WebLogic cluster. This example is based on the example shown previously (see “Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server” on page 90). The new statements for timeout values are shown with a highlighted background.

(If Apache HTTP Server does not serve the static content, then your configuration file would be similar, but would exclude the **SetHandler** statements in the # **Override static content** blocks.)

```
# The following two entries must be uncommented to enable
# cache timeout headers to be sent by Apache
LoadModule expires_module modules/mod_expires.so
LoadModule headers_module modules/mod_headers.so

# The following section enables the expiration directives
<IfModule mod_expires.c>
ExpiresActive On
</IfModule>

WebLogicCluster host1.yourdomain.com:7101,host2.yourdomain.com:7101
WLLogFile c:\apps\Apache2\logs\proxy_debug.log
Debug OFF

# Instruct Apache to forward WRS requests to BEA
<Location /SASWebReportStudio>
    CookieName sas.wrs.sessionID
    SetHandler weblogic-handler
</Location>

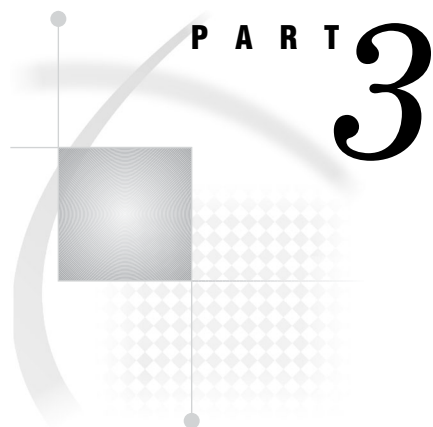
# Override static content (images)
<Location /SASWebReportStudio/images>
    SetHandler default-handler
    # set static timeout
    ExpiresDefault 'access plus 60 minutes'
</Location>

# Override static content (styles)
<Location /SASWebReportStudio/styles>
    SetHandler default-handler
    # set static timeout
    ExpiresDefault 'access plus 60 minutes'
</Location>

# Override static content (scripts)
<Location /SASWebReportStudio/scripts>
    SetHandler default-handler
    # set static timeout
    ExpiresDefault 'access plus 60 minutes'
</Location>

# Override static content (themes)
```

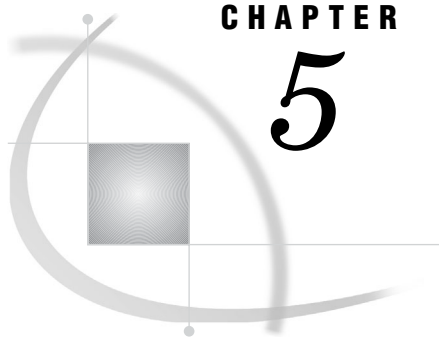
```
<Location /SASWebReportStudio/themes>
  SetHandler default-handler
  # set static timeout
  ExpiresDefault 'access plus 60 minutes'
</Location>
```



## **SAS Web Report Studio Administration**

<i>Chapter 5</i> .....	<b>Introduction to SAS Web Report Studio Administration</b>	<i>101</i>
<i>Chapter 6</i> .....	<b>Configuring SAS Web Report Studio</b>	<i>105</i>
<i>Chapter 7</i> .....	<b>Managing SAS Web Report Studio Content and Users</b>	<i>119</i>
<i>Chapter 8</i> .....	<b>Customizing Reports</b>	<i>139</i>
<i>Chapter 9</i> .....	<b>Scheduling and Distributing Pre-generated Reports</b>	<i>153</i>





## CHAPTER

## 5

## Introduction to SAS Web Report Studio Administration

<i>Introduction to SAS Web Report Studio</i>	101
<i>Prerequisites for Administering SAS Web Report Studio</i>	101
<i>Main Tasks for Administering SAS Web Report Studio</i>	102
<i>Overview: Main Tasks for Administering SAS Web Report Studio</i>	102
<i>Prepare Report Resources</i>	102
<i>Configure SAS Web Report Studio</i>	103
<i>Implement Security for SAS Web Report Studio</i>	103
<i>Perform Additional SAS Web Report Studio Administration</i>	103
<i>Additional Documentation for SAS Web Report Studio</i>	104

### Introduction to SAS Web Report Studio

SAS Web Report Studio is a query and reporting application that is specifically designed for general business users who want to view, author, and share reports on the Web.

SAS Web Report Studio runs within the servlet container, and requires the SAS Query and Reporting Services. SAS Web Report Studio does not require any of the portal components, such as the SAS Web Infrastructure Kit or the SAS Services Application. SAS Web Report Studio uses a local deployment of the SAS Foundation Services. This deployment is created during installation and configuration.

SAS Web Report Studio can be invoked from the SAS Information Delivery Portal. With additional configuration, SAS Web Report Studio can support single sign-on with the portal.

*Note:* SAS Web Report Viewer is a separate application that can be installed along with SAS Web Report Studio. SAS Web Report Viewer is used only for viewing reports. Web applications such as the SAS Information Delivery Portal use SAS Web Report Viewer to render reports. Where applicable, this documentation includes instructions for configuring SAS Web Report Viewer.  $\Delta$

### Prerequisites for Administering SAS Web Report Studio

This documentation assumes that you have successfully installed and configured SAS Web Report Studio. There are two methods by which you might have performed your installation:

- A planned installation (personal or advanced) uses information from a planning document as input to the SAS Software Navigator and the SAS Configuration Wizard. For this type 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 Configuration Wizard. In addition, you can verify that SAS Web Report Studio has been deployed properly by checking that the instructions that are provided in the `deployment.html` file have been completed.

- An installation using the Software Index is performed without the use of plans and SAS project directories. For this type of installation, you should follow the steps that are described in `installation..html` and in `deployment.html`.

*Note:* The `installation..html` and `deployment.html` files both reside in your installation directory.  $\Delta$

For a comprehensive overview of installation, see the *SAS Intelligence Platform: Installation and Configuration Guide*. For instructions on using the SAS Web Report Studio interface, see the product's online Help.

---

## Main Tasks for Administering SAS Web Report Studio

---

### Overview: 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 will need to 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.

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 haven't 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. For details, see “Setting Up Storage for Reporting” on page 119.

- Add content for use by report creators.

Make data sources (information maps), stored processes, images, fonts, and imported reports available to users of SAS Web Report Studio. See “Adding Content for Use by Report Creators” on page 123.

- Enable ESRI maps.

If you want to display your OLAP data in interactive ESRI geographical maps, then you must enable the ESRI feature. See Appendix 4, “Configuring the ESRI Map Component,” on page 369.



---

## Configure SAS Web Report Studio

Here are some configuration tasks that you might perform:

- Understand your SAS Web Report Studio configuration.
  - Modify, update, or troubleshoot the configuration files and the settings in the BI Manager and Web Report Studio plug-ins to SAS Management Console. For details, see “SAS Web Report Studio Configuration Files and Tools” on page 105.
- Configure the SAS Web Report Studio log files.
  - The log files help you to track and audit user actions for performance and security reasons. For details, see “Configuring the SAS Web Report Studio Logs” on page 109.
- Improve the performance of SAS Web Report Studio.
  - Manage memory, take advantage of server pooling capabilities, and understand the performance trade-offs between content servers. For details, see “Improving the Performance of SAS Web Report Studio” on page 113.

---

## Implement Security for SAS Web Report Studio

For general security tasks, see “Planning Your Middle-Tier Security Implementation” on page 20. 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 appropriate roles. For details, see “Setting up Users for SAS Web Report Studio” on page 129.
- Manage access to reports
  - Restrict access to reports in accordance with your security goals. See “Managing Access to Reports” on page 134.
- 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. For details, see “Configure a Pooling Workspace Server to Enforce Row-Level Security” in the *SAS 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 113. △
- Set up trusted Web authentication.
  - Optionally, you can configure SAS Web Report Studio to use trusted Web authentication. For instructions, see “Changing to Trusted Web Authentication” on page 32.

---

## Perform Additional SAS Web Report Studio Administration

You might also want to do the following:

- Customize reports.
  - Add disclaimer text to reports and to add custom report styles. For details, see Chapter 8, “Customizing Reports,” on page 139.
- Set up the scheduling and distribution of reports.

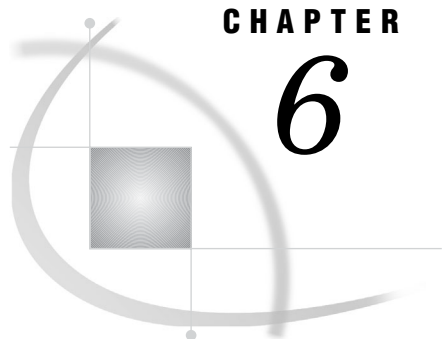
Schedule the creation of pre-generated reports so they will render quickly, and distribute reports to users. For details, see Chapter 9, “Scheduling and Distributing Pre-generated Reports,” on page 153.

---

## 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.
- *SAS Web Report Studio: User’s Guide*, which is available from within SAS Web Report Studio (the Help link).
- The **deployment.html** file, located in the SAS Web Report Studio installation directory, contains configuration and deployment information.
- Chapter 4, “Best Practices for Configuring Your Middle Tier,” on page 57 contains information that is associated with middle-tier administration.
- Chapter 3, “Setting Up and Managing Middle-Tier Security,” on page 19 contains information about authentication, single sign-on, Secure Sockets Layer, and other security related administration.



## CHAPTER

## 6

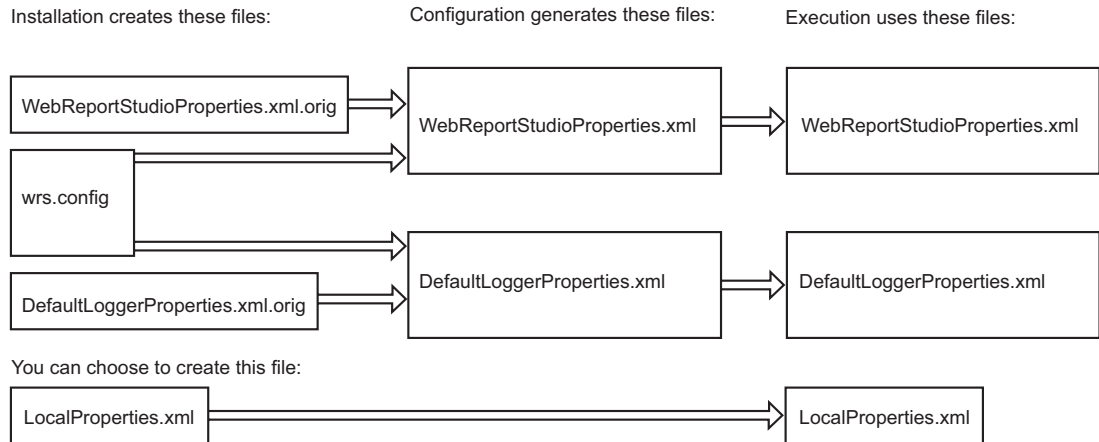
## Configuring SAS Web Report Studio

<i>SAS Web Report Studio Configuration Files and Tools</i>	<b>105</b>
<i>Administrative Files for SAS Web Report Studio</i>	<b>105</b>
<i>Create a LocalProperties.xml File</i>	<b>107</b>
<i>How SAS Web Report Studio Uses BI Manager</i>	<b>107</b>
<i>Report Studio Configuration Plug-In</i>	<b>108</b>
<i>Set Maximum Values for Report Filters</i>	<b>108</b>
<i>Enabling Interaction with Other SAS Applications</i>	<b>109</b>
<i>Configuring the SAS Web Report Studio Logs</i>	<b>109</b>
<i>Overview of SAS Web Report Studio Log Files</i>	<b>109</b>
<i>Change the General Purpose Log File's Logging Level</i>	<b>110</b>
<i>Configure Debug Logging Dynamically</i>	<b>110</b>
<i>Manage the Key User Action Log File</i>	<b>111</b>
<i>Understanding Key User Action Log Output</i>	<b>111</b>
<i>Suggested Procedure for Reporting Events in the Key User Action Log</i>	<b>112</b>
<i>Improving the Performance of SAS Web Report Studio</i>	<b>113</b>
<i>Suggestions for Improving the Performance of SAS Web Report Studio</i>	<b>113</b>
<i>Using the Query Cache</i>	<b>114</b>
<i>Overview of the Query Cache</i>	<b>114</b>
<i>Security Considerations for the Query Cache Library</i>	<b>115</b>
<i>Change the Location of the Query Cache Library</i>	<b>115</b>
<i>Disable the Query Cache</i>	<b>116</b>
<i>Re-Create and Redeploy SAS Web Report Studio</i>	<b>116</b>

## SAS Web Report Studio Configuration Files and Tools

### Administrative Files for SAS Web Report Studio

The configuration and properties files for SAS Web Report Studio are located on the middle-tier server on which SAS Web Report Studio is deployed. The following figure depicts the files that an administrator should be familiar with.

**Figure 6.1** SAS Web Report Studio Configuration and Properties Files

The following list describes the depicted files:

- When SAS Web Report Studio is installed, these files are created:

**WebReportStudioProperties.xml.orig**

Contains most application properties. This file is located in *SAS-install-dir\SASWebReportStudio\3.1\config\*. Changes that you make to this file take effect after you deploy a new WAR file.

This file can be overwritten by subsequent installation activities. For this reason, it is recommended that you store application properties settings in a centralized location that is not affected by subsequent installation activities. To do this, create and use a **LocalProperties.xml** file. For instructions, see “Create a LocalProperties.xml File” on page 107. Settings in the **LocalProperties.xml** file override conflicting settings in the **WebReportStudioProperties.xml** file.

**wrs.config**

Contains attributes such as connection parameters, authentication providers, and location of the application log. This file is located in *SAS-install-dir\SASWebReportStudio\3.1\*. Changes that you make to this file take effect after you deploy a new WAR file.

**DefaultLoggerProperties.xml.orig**

Contains attributes such as the format, locations, names, and contents of the SAS Web Report Studio logs. This file is located in *SAS-install-dir\SASWebReportStudio\3.1\config\*. Changes that you make to this file take effect after you deploy a new WAR file. This file can be overwritten by subsequent installation activities.

- Each time you configure SAS Web Report Studio to create a new application WAR file, these files are written to the **WEB-INF** directory of your Web application server:

**WebReportStudioProperties.xml**

Contains most application properties. You should modify this file only for the purposes of making temporary changes, because this file is overwritten by subsequent installation and configuration activities. To make changes in a clustered environment, you must locate and modify multiple instances of this file.

**DefaultLoggerProperties.xml**

Contains log configuration settings. You should modify this file only for the purposes of making temporary changes, because this file is overwritten by

subsequent installation and configuration activities. To make changes in a clustered environment, you must locate and modify multiple instances of this file.

- When SAS Web Report Studio executes, it uses the properties files that are in the **WEB-INF** directory (and your local properties file if you choose to create one).

---

## Create a LocalProperties.xml File

The **WebReportStudioProperties.xml** file can be overwritten by subsequent installation activities. It is recommended that you store application properties settings in a centralized location that is not affected by subsequent installation. To do this, create and use a **LocalProperties.xml** file.

Settings in the **LocalProperties.xml** file override conflicting settings in the **WebReportStudioProperties.xml** file.

To create a **LocalProperties.xml** file, complete these steps:

- 1 Locate the sample file in the **customer** subdirectory of the installation directory. The sample file is named **LocalProperties.xml.sample**.
- 2 Make a copy of this file in the **customer** directory, and name the copy **LocalProperties.xml**.

*Note:* If you plan to use SAS Web Report Viewer to view reports, create a **LocalProperties.xml** file for SAS Web Report Viewer in a similar way. Locate the **LocalProperties.xml.sample** file in the SAS Web Report Viewer **customer** subdirectory, make a copy of the file, and name your new file **LocalProperties.xml**. △

Changes that you make to **LocalProperties.xml** take effect after you restart your servlet container.

---

## How SAS Web Report Studio Uses BI Manager

Beginning with SAS Foundation Services 1.2, BI Manager is available in SAS Management Console to help you manage reports.

*Note:* If you have not upgraded to the SAS Foundation Services 1.2 release, then you can use Business Report Manager to register and manage reports. BI Manager replaces Business Report Manager. For more information about using BI Manager or Business Report Manager, see the Help in SAS Management Console. △

The BI Manager enables administrators to perform these tasks:

- specify the account that SAS Web Report Studio uses to connect to your WebDAV content server.
- coordinate report storage by associating a top-level reporting folder in the metadata repository with a top-level report content area in the external content server.
  - In the metadata repository, the default name for the top-level reporting folder is **ReportStudio**. You can have more than one top-level reporting folder. The top-level reporting folder is sometimes referred to as the root folder.
  - In the content server, the top-level report content area is a directory location. If you are using the file system as your content server, the directory is simply a file system address. If you are using a WebDAV server, the directory is a content base path (**/sasdav/wrs**).
- manage the report storage containers. However, the BI Manager does not enable you to control access to the report folders. To set permissions on a reporting folder,

navigate to the folder in SAS Management Console under **Environment Management ► Authorization Manager ► By Application ► BIP Tree ► ReportStudio** and then access the **Authorization** tab in the properties dialog box for that folder.

- import reports and other report content into the metadata repository.
- distribute reports among different metadata repositories by exporting and then importing a BI package that contains the reports along with their metadata definitions.
- deploy reports as jobs for scheduling.

This plug-in is added to your local copy of SAS Management Console when SAS Foundation Services is installed on the computer where you are working. For detailed instructions on using this plug-in, select **Environment Management ► BI Manager** and then select Help from the main menu bar in SAS Management Console.

---

## Report Studio Configuration Plug-In

The Report Studio Configuration plug-in for SAS Management Console provides a graphical interface to the **wrs.config** file, which is documented in “Administrative Files for SAS Web Report Studio” on page 105. This plug-in is added to your local copy of SAS Management Console when SAS Query and Reporting Services is installed on the computer where you are working.

---

## Set Maximum Values for Report Filters

Two values in the **LocalProperties.xml** file determine the following:

- The maximum number of filter values that can be displayed when report creators define a filter.

Here is the corresponding element block, along with the default value:

```
<webreportstudio.max.filter.choices>1000
</webreportstudio.max.filter.choices>
```

- The maximum number of filter values that can be displayed when report viewers query for available filter values.

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 a **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.

Here is the corresponding element block, along with the default value:

```
<webreportstudio.max.prompt.choices>1000
</webreportstudio.max.prompt.choices>
```

To configure the maximum number of prompt values, in the **LocalProperties.xml** file, specify the number that you want for the appropriate element. You might need to add the element to the file. If you don't have a **LocalProperties.xml** file, you can create one. See “Create a LocalProperties.xml File” on page 107.

For more information about dynamic prompt values, or for instructions on creating a filter, see the product Help and Documentation.

## Enabling Interaction with Other SAS Applications

Interaction between SAS Web Report Studio and other SAS products, such as SAS Enterprise Guide, requires the creation of metadata in the SAS Metadata Repository. Depending on how you installed the software, you might have created some of this metadata during your initial configuration.

To enable interaction with other SAS applications, perform the following steps:

- Run a program named **LoadDefaultPreferences.sas**.

Run **LoadDefaultPreferences.sas** on the same system on which SAS Web Report Studio is installed. **LoadDefaultPreferences.sas** can be executed only one time, but it is safe to issue a run command multiple times (the program simply aborts if you try to rerun it).

- Create a SAS profile.
- Enable a Web service that provides WebDAV access.

For instructions on performing all of these steps, see the **deployment.html** file in the SAS Web Report Studio installation directory. (See the “Installing Metadata” and “Enabling Report Web Services” sections in the file.)

If you don’t run **LoadDefaultPreferences.sas**, or if you don’t create the profile, an error message is added to your log file when you log on to SAS Web Report Studio. The context for this error message is **com.sas.apps.citation.model.pfs.PFSHelper**. The error is not serious, but you can eliminate the error message by performing the steps mentioned above.

## Configuring the SAS Web Report Studio Logs

### Overview of SAS Web Report Studio Log Files

You can use the SAS Web Report Studio log files to help you manage performance, track security enforcement, and analyze specific situations. SAS Web Report Studio records events in two log files. By default, both log files are created in the *SAS-config-dir*\Lev1\web\Deployments\WebReportStudio\logs directory.

The following table summarizes the log files:

**Table 6.1** Log Files

Log Context and Default File Name	Description
General Purpose Log (WebReportStudio.log)	Logs events such as serious errors, application restarts, and users logging on.
Key User Action Log (WebReportStudio_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 “Understanding Key User Action Log Output” on page 111.

*Note:* There are similar log files for SAS Web Report Viewer in the directory. △

---

## Change the General Purpose Log File's Logging Level

For the General Purpose Log, you can change the amount of information that is recorded. The log has four levels of warnings: DEBUG, INFO, WARN, and ERROR. By default, the log level is set to WARN, which means that only WARN and ERROR 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. (You can also set the level to DEBUG, but the amount of output is very high, and this can negatively affect performance and consume file space. If you need to debug a problem, it is recommended that you dynamically change the log output temporarily. See “Configure Debug Logging Dynamically” on page 110.)

To enable INFO level messages in the General Purpose Log:

- 1 Open the **DefaultLoggerProperties.xml.orig** properties file, which is located in *SAS-install-dir\SASWebReportStudio\3.1\config*.
- 2 In the **<LoggingContext>** block for "com.sas.apps.citation," change the priority value from **WARN** to **INFO**.
- 3 If you are using SAS Web Report Viewer to render reports, then make similar changes for SAS Web Report Viewer. Open the SAS Web Report Viewer's **DefaultWRVLoggerProperties.xml.orig** properties file, and edit the priority value.
- 4 After you make these changes and save the files, you must redeploy SAS Web Report Studio and, if applicable, SAS Web Report Viewer before your changes take effect. See “Re-Create and Redeploy SAS Web Report Studio” on page 116.

*Note:* It is recommended that you make a copy of the **DefaultLoggerProperties.xml.orig** file. Subsequent installation or upgrade activity can overwrite this file.  $\triangle$

---

## Configure Debug Logging Dynamically

The previous topic discusses how to change the logging level for debugging. However, the procedure that was specified 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. You must be logged on as a WRS Administrator (member of the WRS Administrator group) to use this functionality.

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

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 also can delete files after the number of archived files reaches a particular limit.

To manage the Key User Action Log:

- 1 Edit the **<rollover>** element in the **<sas.wrs.keyUserActionLog>** block of your **LocalProperties.xml** file. Each archive event is considered a rollover. The **<rollover>** element contains attributes that enable you to specify the maximum number of rollovers, the maximum size of the log file, and the schedule for performing rollovers.
 

If your **LocalProperties.xml** file does not contain the **<sas.wrs.keyUserActionLog>** element block, then you can add it to the file by copying the block from the **WebReportStudioProperties.xml.orig** file. For instructions on creating a **LocalProperties.xml** file, see “Create a LocalProperties.xml File” on page 107.
- 2 If you are using SAS Web Report Viewer to render reports, then make similar changes for SAS Web Report Viewer. Open the SAS Web Report Viewer’s **LocalProperties.xml** file, and edit the **<rollover>** element.
- 3 Periodically move or delete outdated archived log files.

## Understanding 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 6.2** 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
User started a system.	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>8/15/05</date><time>4:13PM/
time</time><code>9</code><description>System Startup</description></event>

<event><javaDate>1124136826633</javaDate><date>8/15/05</date><time>4:13PM/
</time><user>saswbadm</user><code>0</code><description>Logon</description>
</event>

<event><javaDate>1124136878587</javaDate><date>8/15/05</date><time>4:14PM/
</time><user>dolson</user><code>0</code><description>Logon</description></event>

<event><javaDate>1124136923432</javaDate><date>8/15/05</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>8/15/05</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>8/15/05</date><time>4:16PM/
</time><user>dolson</user><code>2</code><description>Logoff</description>
</event>
```

## Suggested Procedure for Reporting 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. Here is a suggested procedure for reporting the data:

- 1 Import the `WebReportStudio_KeyActions.log` data into a SAS data set. Following 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 DATASETS procedure to import the XML file into a SAS data set.

Here is an example:

```
proc datasets library=myxml;
```

For more information, see SAS Help and Documentation.

- 2 In SAS Information Map Studio, create an information map based on the data set that you created in the previous 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. 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 optionally 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, you should do these things:

- Convert your workspace server to use pooling, as described in “Convert a Workspace Server to Pooling” in the *SAS Intelligence Platform: Application Server Administration Guide*. Setting up a pool of workspace server processes eliminates the need to start a new process for each user request.
- 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 “Changing a Workspace Server’s Launch Command” in the *SAS Intelligence Platform: Application Server Administration Guide*.
- Configure your middle tier as recommended in Chapter 4, “Best Practices for Configuring Your Middle Tier,” on page 57. That chapter includes information about setting up your J2EE application server to use the correct Java Virtual Machine options, creating a cluster of servers, and using an HTTP server to handle requests for static pages.
- Make appropriate use of pre-rendered reports, such as manually refreshed reports and batch reports. Use report scheduling to control when batch reports are generated. For example, you can schedule reports to be generated on a nightly, weekly, or monthly basis. For more information, see Chapter 9, “Scheduling and Distributing Pre-generated Reports,” on page 153.
- Use the query cache, which is enabled by default. You can optionally change the location of the cache, and you can disable caching. For more information, see “Using the Query Cache” on page 114.
- Consider the performance, security, and flexibility trade-offs between using a file-based content server and using a WebDAV content server. Using a file-based content server can result in faster response times because the report read and write requests do not pass through an HTTP/DAV server. Although several

variables influence how much of an improvement can be realized, a performance increase of 10–15% is typical.

However, this performance increase comes at the cost of flexibility. A DAV-based content server provides access to content without direct operating system support or shared network areas. This type of access is especially important if an installation requires that content be accessible by tools or applications running on several machines or on widely dispersed machines. In such a diverse environment, a DAV-based content server is most likely a necessity. If content is accessed only from one machine or a very small number of machines, sharing the content space may not be an issue, and a file-based content server with the resulting performance increase is the better choice.

*Note:* If you use Xythos WFS as your WebDAV content server, then you can improve the performance by changing the document store location to external storage in a file system location. The SAS installation instructions for Xythos WFS follow this recommended approach.  $\triangle$

## 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-report 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.  $\triangle$

During installation, the query cache is enabled and is associated with a SAS library. After installation, you can optionally do the following:

- change the location of the query cache library
- disable the query cache

Using the query cache will likely increase performance if any of your reports has any of the following characteristics:

- a large number of joins from many tables
- many BY groups
- many report objects
- data sources other than SAS (for example, Oracle 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 table(s) 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.

## Security Considerations for the Query Cache Library

During installation and configuration, a query cache library was created at *SAS-config-dir\Lev1\SASMain\Data\wrstemp*. By default, all users have read and write permissions on this library.

If you set up workspace server pooling, then you can implement tighter security and grant full permissions only to the user IDs that you specified for the puddle login definitions in your pool. To use the query cache, make sure each puddle login definition has access permissions (read and write) for the query cache library.

*Note:* With pooled workspace servers, a user group is granted rights to use the puddles in the pool. In order to use the pooled workspace server connections for query caching, the SAS Web Administrator (saswbadm) must be a member of the puddle user group. Otherwise, a standard workspace server session will be launched whenever the query cache is used.  $\Delta$

If you have not configured pooling, then each requesting user's individual (or shared) account will need read and write permissions for the library in order to access the tables.

Regardless of whether you use pooling, you must manually grant the SAS Web Administrator (saswbadm) full permissions, including Delete, in order for old tables to be removed automatically.

If you change the location of the query cache library, then be sure to grant users access to the new library as described here.

## Change the Location of the Query Cache Library

The default location for the library that is used for the query cache is *SAS-config-dir\Lev1\SASMain\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). Backups are unnecessary because the cache files are temporary.

*Note:* Do not use the Work or Saswork library for this feature. The query cache won't function correctly if you use Work or Saswork.  $\Delta$

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.

To change the query cache library, follow these steps:

- 1 Create a library where the query cache can temporarily store common data tables. To create the library, follow these steps:

- a Create an **autoexec.sas** file that assigns the library. For example, the file might contain the following:

```
/* Libname assigned for temporary tables for the query cache */
libname optlib 'C:\SASServers\wrstemp';
```

- b Save the **autoexec.sas** file on the workspace server machine.
- c In SAS Management Console, navigate under the Server Manager to the lowest level of the workspace server definition.
- d Select **Properties** ► **Options**.

- e In the **Command** field, append a pointer to the **autoexec.sas** file that you saved on the server. For example, add this string to the command:

```
-autoexec c:\servers\autoexec.sas
```

- f Click **OK**.

*Note:* Make sure that the required users are granted operating system permissions for the directory that is associated with the library. For details, see “Security Considerations for the Query Cache Library” on page 115.  $\Delta$

- 2 Edit the following values in the **wrs.config** file:

```
$RENDERER_OPTIMIZER_LIBNAME$=
```

```
$RENDERER_OPTIMIZER_SERVER$=
```

For example:

```
$RENDERER_OPTIMIZER_LIBNAME$=optlib
```

```
$RENDERER_OPTIMIZER_SERVER$=Pooled Workspace Server - Logical  
Workspace Server
```

- 3 Redeploy SAS Web Report Studio and, if it is installed, SAS Web Report Viewer. For instructions, see “Re-Create and Redeploy SAS Web Report Studio” on page 116.

## Disable the Query Cache

To disable the query cache, in the **LocalProperties.xml** file, set the **<wrs.activateReportOptimizer>** property to **false**. You might need to add the property to the file. If you don't have a **LocalProperties.xml** file, you can create one. See “Create a LocalProperties.xml File” on page 107.

*Note:* If you are using SAS Web Report Viewer to render reports, then configure SAS Web Report Viewer in a similar way. In the **LocalProperties.xml** file for SAS Web Report Viewer, set the **<wrs.activateReportOptimizer>** property to **false**.  $\Delta$

You might want to delete the library that is associated with the query cache if you are certain that you will not use the query cache in the future. If there's a chance that you will re-enable the query cache, then you should leave the library in place.

---

## Re-Create and Redeploy SAS Web Report Studio

After initial installation, if you make configuration changes, then you might be instructed to re-create and redeploy SAS Web Report Studio. All of the procedures in the documentation explicitly state when you must redeploy SAS Web Report Studio.

To re-create and redeploy SAS Web Report Studio, complete these steps :

- 1 Create a new **SASWebReportStudio.war** file by running **sas.wrs.config.sh** (UNIX) or **sas.wrs.config.bat** (Windows). These scripts are located in **SAS-install-dir\SASWebReportStudio\3.1\**.
- 2 If you are using WebLogic, then run **sas.wrs.weblogic.prepare.bat** to explode the new WAR file into a specified directory. For example, from a command prompt, change to the SAS Web Report Studio installation directory and enter this command:

```
sas.wrs.weblogic.prepare.bat C:\bea\webapps SASWebReportStudio  
SASWebReportStudio.war
```

**CAUTION:**

The contents of the target directory are deleted by this script, so do not reference a directory that includes files for other Web applications. For complete configuration instructions, see

*SAS-install-dir\SASWebReportStudio\3.1\deployment.html*.  $\Delta$

- 3 Deploy the new WAR file. For deployment instructions, see *SAS-install-dir\SASWebReportStudio\3.1\deployment.html*.

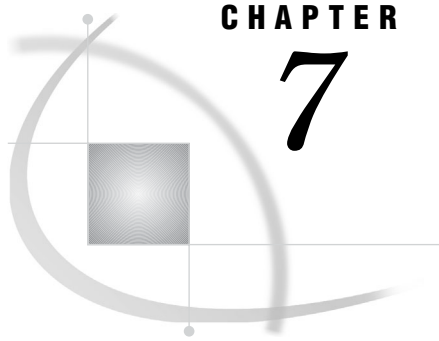
*Note:* If you are deploying into Tomcat, you will need to run a second script that disables Tomcat's serialization feature, that provides an explicit context for SAS Web Report Studio, and that explodes the WAR file. See the **deployment.html** file for details.  $\Delta$

- 4 Restart the servlet container or J2EE application server.

For some procedures, such as configuring trusted Web authentication, you might also be instructed to redeploy SAS Web Report Viewer. The deployment steps are similar. Create a new **SASWebReportViewer.war** file by running **sas.wrv.config.sh** or **sas.wrv.config.bat** located in *SAS-install-dir\SASWebReportViewer\3.1\*. Then deploy the new WAR file. For detailed instructions, see the **deployment.html** file.







## CHAPTER

## 7

# Managing SAS Web Report Studio Content and Users

<i>Setting Up Storage for Reporting</i>	<b>119</b>
<i>Overview: Setting Up Storage for Reporting</i>	<b>119</b>
<i>Standard Storage Containers for Reporting</i>	<b>120</b>
<i>Verifying Your Reporting Storage Structure</i>	<b>122</b>
<i>Adding Folders to Your Report Storage Structure</i>	<b>122</b>
<i>Adding Content for Use by Report Creators</i>	<b>123</b>
<i>Overview: Adding Content for Use by Report Creators</i>	<b>123</b>
<i>Making Data Sources Available to SAS Web Report Studio</i>	<b>123</b>
<i>Making Stored Processes Available to SAS Web Report Studio</i>	<b>124</b>
<i>Making Images Available to SAS Web Report Studio</i>	<b>125</b>
<i>Making Fonts Available to SAS Web Report Studio</i>	<b>127</b>
<i>Importing Reports that Conform to the SAS Report Model</i>	<b>128</b>
<i>Importing Legacy Reports</i>	<b>128</b>
<i>Setting up Users for SAS Web Report Studio</i>	<b>129</b>
<i>Overview: Setting up Users for SAS Web Report Studio</i>	<b>129</b>
<i>Designate a Surrogate Metadata Identity</i>	<b>129</b>
<i>Using SAS Web Report Studio Roles</i>	<b>130</b>
<i>Understanding SAS Web Report Studio Roles</i>	<b>130</b>
<i>Using the Roles: Example Scenario</i>	<b>132</b>
<i>Additional Authentication for SAS Web Report Studio Users</i>	<b>134</b>
<i>Managing Access to Reports</i>	<b>134</b>
<i>Overview: Managing Access to Reports</i>	<b>134</b>
<i>Changing Access to Reports</i>	<b>136</b>
<i>Security Considerations for Pre-generated Batch Reports</i>	<b>136</b>
<i>Considerations for Row-level Security</i>	<b>137</b>
<i>Protecting Report Content in the WebDAV Server</i>	<b>137</b>
<i>Protecting Data in the SAS Web Report Studio Temporary Files</i>	<b>138</b>

## Setting Up Storage for Reporting

### Overview: Setting Up Storage for Reporting

Proper storage of reports and report-related objects is important for these reasons:

- Storage of reports (and some report-related objects) must always be synchronized between your metadata repository and your external content server.
- Moving information maps or stored processes from one location to another breaks any reports that are based on those objects in their original locations.

- Information maps and stored processes must be registered in a particular location in the metadata repository in order to be available to users of SAS Web Report Studio.
- If you organize storage of reports appropriately for your environment, then controlling access to reports will be easier.

---

## Standard Storage Containers for Reporting

These are the standard storage containers for reports and report-related objects:

### **ReportStudio**

The top-level storage container for the reporting environment.

### **ReportStudio/BannerImages**

The location where SAS Web Report Studio looks for banner images when building a report.

### **ReportStudio/ConditionalHighlightingImages**

The location where SAS Web Report Studio looks for conditional highlighting images when building a report.

### **ReportStudio/Maps**

The top-level container for information maps. SAS Web Report Studio users can access only those information maps that are stored in the **Maps** folder or in a subfolder of the **Maps** folder.

### **ReportStudio/Shared**

The top-level container for report-related objects that will be accessed by multiple users. Although saving reports in this container is an easy way to make the reports widely available, this practice can create clutter that is difficult to secure and to navigate.

### **ReportStudio/Shared/Images**

The top-level container for images that can be included in reports (other than banner images and conditional highlighting images).

### **ReportStudio/Shared/Reports**

The top-level container for reports that will be accessed by multiple users. It is recommended that you store reports in subfolders beneath this container.

### **ReportStudio/Shared/Reports/StoredProcesses**

The initial location where SAS Web Report Studio looks for stored processes when you attempt to insert a stored process section into a report.

### **ReportStudio/Shared/ReportTemplates**

The container for the templates that are used when you create reports with custom layouts in SAS Web Report Studio. This folder is created automatically when templates are used.

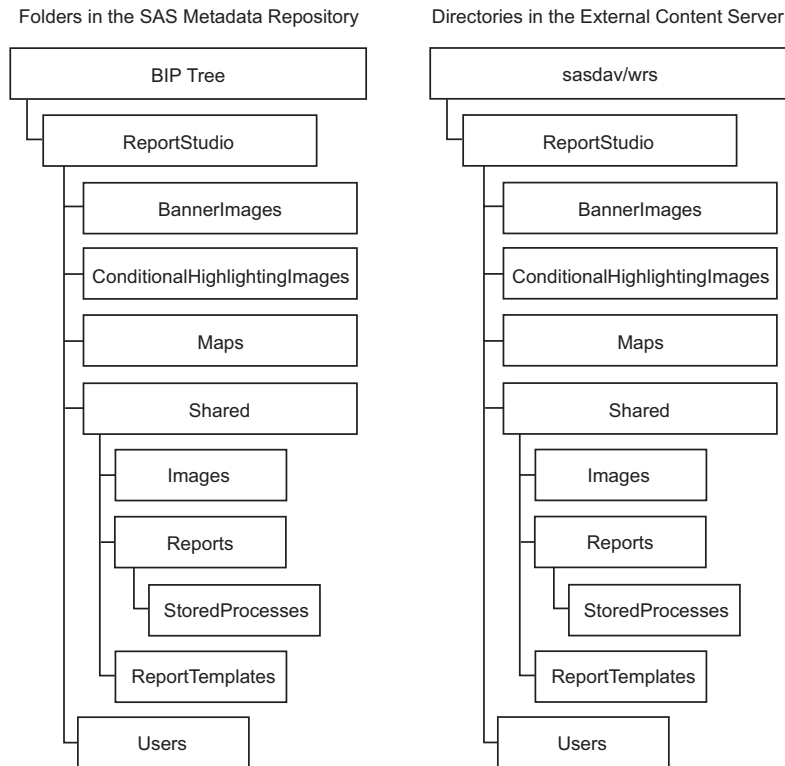
### **ReportStudio/Users**

The storage container for the personal folders in which each user can store private reports. A personal folder is created for each user when the user first logs on to SAS Web Report Studio. By default, only the user who is associated with a folder can read and write metadata in that folder. These default permissions are set only when a user folder is created automatically. For this reason, you should not manually create user folders.

*Note:* This **Users** folder is used only by SAS Web Report Studio. The SAS Information Delivery Portal has its own **Users** folder in a different location.  $\Delta$

This set of storage containers for SAS Web Report Studio must exist in both the foundation SAS Metadata Repository *and* the external content server (third-party WebDAV server or file system). The following figure depicts the parallel storage structures that are required by SAS Web Report Studio. The figure also depicts the relationship between the top-level report storage containers. The **BIP Tree** folder in the metadata repository corresponds to the **sasdav/wrs** directory in the content server. These top level containers should be created during the installation process.

**Figure 7.1** Parallel Structure for Report Storage Containers



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- 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.** Using SAS applications to manage your reports preserves the synchronization. Using the external content server or interacting directly with the report content can leave the corresponding report metadata objects in an inconsistent state. For example, using Xythos WebFile Server to delete or rename files or directories will break the synchronization. Similarly, using a text editor to make changes to a report definition XML file without also updating the corresponding report metadata object will result in a nonfunctional report.  $\Delta$

During the first use of SAS Web Report Studio, all of the standard storage containers are created for you in both the metadata repository and your external content server. These containers should be in place at the end of the installation process, because the installation includes a verification step in which someone logs on to SAS Web Report Studio. For each additional report creator who logs on to SAS Web Report Studio, one additional personal report storage container is added under the **Users** container.

---

## Verifying Your Reporting Storage Structure

To view the report storage containers in your metadata repository, complete these steps:

- 1 Log on to SAS Management Console and access the foundation repository. It is recommended that you log on as an unrestricted user such as the SAS Administrator (sasadm) for this task. This ensures that you will not be prevented from seeing a folder because of access controls in the metadata layer.
- 2 Navigate to **Environment Management**  $\blacktriangleright$  **BI Manager**  $\blacktriangleright$  **BIP Tree**  $\blacktriangleright$  **ReportStudio**.
- 3 Verify that the **BannerImages**, **Maps**, **Shared**, and **Users** folders exist beneath the **ReportStudio** folder. Also verify that the subfolders inside the **Shared** folder exist.

To verify that the same set of storage containers exists in your external content server, complete these steps:

- 1 In SAS Management Console, navigate to **Environment Management**  $\blacktriangleright$  **BI Manager**  $\blacktriangleright$  **BIP Tree**  $\blacktriangleright$  **Properties** to see the content server and content path that your deployment of SAS Web Report Studio has been configured to use.
- 2 In your external content server, navigate to the content path that is specified in metadata for the BIP Tree, and verify that the **BannerImages**, **Maps**, **Shared**, and **Users** directories exist under a **ReportStudio** directory.

For example, if you are using Xythos WebFile Server as your content server and you used the standard ports and names, then you would open a Web browser to `<machine-name>:8300/sasdav/wrs` and log on as the SAS Web Administrator (saswbadm). (You might first need to grant the SAS Web Administrator access to this folder. For instructions, see “Protecting Report Content in the WebDAV Server” on page 137.)

---

## Adding Folders to Your Report Storage Structure

In the metadata repository, each report, information map, and stored process inherits permissions from the folder in which it is stored. For example, the access controls that you set on a report folder are inherited by all of the reports within that folder. It is easier to manage permissions on folders than on individual resources, so you might want to define additional subfolders within the standard structure. For example, within the **ReportStudio/Shared/Reports** container you could create separate report

subfolders for Human Resources, Finance, and Sales. Then, within the **Sales** container, you could create an additional subfolder for each sales region.

Each new subfolder inherits the effective access controls of its parent folder. You can view the permission settings for any folder by navigating to the folder in SAS Management Console under **Environment Management ► Authorization Manager ► Resource Management ► By Application ► BIP Tree ► ReportStudio** and accessing the folder's **Authorization** tab. This structure enables you to store reports that everyone will access in the **ReportStudio/Shared/Reports** folder and to limit access to each of the subfolders so that only certain user groups can see or modify the folder and its contents.

You should use the BI Manager in SAS Management Console to add folders to the report storage structure. For each folder that you create using the BI Manager, a corresponding directory is also automatically created in your content server. In this way, the BI Manager preserves the necessary synchronization between the folders in the metadata repository and the content server.

Users can also add folders from within SAS Web Report Studio. However, users cannot set permissions on folders from within the application. You can prevent users from creating subfolders by denying WriteMetadata permission to the parent folder. However, this will prevent users from adding resources to the parent folder. The inherited denial of WriteMetadata permission from the parent folder will also prevent users from modifying or deleting the resources in that folder (unless you set a direct grant of WriteMetadata permission on each resource).

---

## Adding Content for Use by Report Creators

---

### Overview: Adding 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 foundation metadata repository. The following sections describe how to add this metadata to the repository.

*Note:* If you plan to use ESRI maps in reports, then see Appendix 4, “Configuring the ESRI Map Component,” on page 369. △

---

### Making Data Sources Available to SAS Web Report Studio

In SAS Web Report Studio, users do not interact directly with SAS data sets, relational database tables, or SAS OLAP cubes. Instead, users interact with information maps that provide a business view of the underlying data. In SAS Web Report Studio, the term "data source" refers to an information map. Each report that is created in SAS Web Report Studio can use no more than one information map.

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 exist in the same foundation repository that the user of SAS Web Report Studio is accessing.
- 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.

*Note:* Beginning with Service Pack 4, end 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.  $\Delta$

---

## Making 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. The output of the stored process is rendered in the user's Web browser.

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.

*Note:* Stored process reports that were created by using SAS Enterprise Guide do not support any layout design.  $\Delta$

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, complete these steps:

- 1 Insert a `*PROCESSBODY` statement.
- 2 Insert a `%STPBEGIN` statement prior to 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 cannot 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 7.1** 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.* <b>GOPTIONS DEVICE=ACTIVEX;</b>

\* 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 will be substituted automatically.

To make a stored process available to users of SAS Web Report Studio, complete these steps:

- 1 Register the stored process in the metadata by using either SAS Management Console or SAS Enterprise Guide. The stored process must be registered in the foundation repository that will contain the reports that use the stored process. Within that repository, the stored process must be registered in the **BIP Tree/ReportStudio/** folder structure.
- 2 In SAS Management Console, navigate to the stored process under **Environment Management ► Authorization Manager ► Resource Management ► By Application ► BIP Service ► BIP Tree ► ReportStudio**. On the **Authorization** tab for the stored process, verify that your SAS Web Report Studio users have `ReadMetadata` access to the stored process.

*Note:* To learn more about stored processes, see "SAS Stored Processes" in *SAS Integration Technologies: Developer's Guide* at [support.sas.com/rnd/itech/doc9/dev\\_guide/stprocess/](http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/). △

---

## Making 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 7.2** 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 <b>ReportStudio/BannerImages</b>.</p> <p>By default, the <b>BannerImages</b> folder is empty. You should use the BI Manager 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 will be displayed in every table cell where the condition is met. Conditional highlighting images are stored in <b>ReportStudio/ConditionalHighlightingImages</b>.</p> <p>You should use the BI Manager in SAS Management Console to manage your conditional highlighting images. (The initial set of images do not appear in the <b>ConditionalHighlightingImages</b> folder until after the Conditional Highlighting dialog box has been accessed from within SAS Web Report Studio.)</p>
Other images	<p>Any report can include additional images for decorative or other purposes. These images are stored under the <b>ReportStudio/Shared/Images</b> folder.</p> <p>By default, the <b>Images</b> folder is empty. You should use SAS Web Report Studio to add images to this folder. For instructions, see the Help for SAS Web Report Studio.</p>

To make a banner image or conditional highlighting image available to users of SAS Web Report Studio, complete these steps:

- 1 If you are using a WebDAV server, make sure that the server is running.
- 2 In SAS Management Console, navigate to the appropriate images folder beneath **Environment Management ► BI Manager ► BIP Tree ► ReportStudio**.
- 3 From the menu bar, select **Actions ► Add Content From External File**.
- 4 From the Specify a Source File dialog box, 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. However, SAS Web Report Studio does not detect banner images or conditional highlighting images that are stored in sub-folders of the **BannerImages** and **ConditionalHighlightingImages** folders.  $\triangle$

- 5 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.
- 6 Click **OK** to close the Enter Description text box.
- 7 The imported images are available in SAS Web Report Studio within 10 minutes. 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 previous 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, complete these steps:

- 1 In SAS Management Console, navigate under **Environment Management ► BI Manager ► BIP Tree ► ReportStudio** and select the image that you want to delete.
- 2 From the menu bar, select **Edit ► Delete**.

*Note:* The minimum screen resolution that is supported for clients (browsers) is 1024 x 768. △

---

## Making 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.

To supply additional fonts, 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 **customer** subdirectory of your SAS Web Report Studio installation. 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 on 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 previous 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 **<font>**, are case-sensitive, and should be specified exactly as shown. △

If you plan to use SAS Web Report Viewer to render reports, create and edit a **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 Web Report Viewer **customer** subdirectory, 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 “Re-Create and Redeploy SAS Web Report Studio” on page 116.

---

## Importing 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, complete 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 BI Manager and select the folder into which you will import the report.
- 3 From the menu bar, select **Actions ► Import**.
- 4 Select the report to import.
- 5 Select **Open**.

---

## Importing 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 IntraNet. 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 on the FILENAME statement. When you use the SASXPGRP access method on the FILENAME statement, a SAS Business Intelligence Protocol (SBIP)\* URL identifies the external file that you want to write to. 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"`

The userID to access the server.

---

\* SBIP is a proprietary protocol for specifying the location of resources in a SAS Metadata Repository. For example, this path **SBIP://Foundation/BIP Tree/ReportStudio/Shared/Reports/MyReport.srx** specifies the location of a report named **MyReport** within a repository named **Foundation**.

PASSWORD="*password*"

The password to access the server.

DOMAIN="*domain*"

The domain name for the server.

OMRHOST="*host*"

The network name of the machine hosting the metadata repository.

OMRPORT="*nnnn*"

The port number for accessing the repository.

OMRUSER="*user ID*"

The user ID to access the repository. This can be the same as the server user ID, or it can be different.

OMRPASSWORD="*password*"

The password to access the repository. This can be the same as the server password, or it can be different.

OMRREPOSNAME="*name*"

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/Bip Tree/ReportStudio/Users/xyz/Reports"
  userid="xyz" password="bip2004" domain="thisDomain"
  OMRHost="bipsvrxyz.na.sas.com" OMRPort="9999" OMRUser="xyz"
  OMRPassword="bip2004" OMRReposName="RepName"
  ;

option noovp;
ods sasreport file="myreport.xml" path=dest;
proc print data=sashelp.class;
run;
ods sasreport close;
```

---

## Setting up Users for SAS Web Report Studio

---

### Overview: Setting up Users for SAS Web Report Studio

Information about adding users to a deployment is described in “Planning User Accounts and Their Organization into Groups” on page 21.

These are the aspects of setting up users that are specific to SAS Web Report Studio:

- It is recommended that you designate a metadata identity to serve as a surrogate for SAS Web Report Studio users who do not have their own metadata identities. *This is a requirement if you are using Web authentication.*
- You can use roles to control access to SAS Web Report Studio functionality.

The following sections provide instructions and details for each of these topics.

---

### Designate a Surrogate Metadata Identity

By default, SAS Web Report Studio users who do not have their own individual metadata identities use the PUBLIC group’s metadata identity. It is recommended that

you instead use the SAS Guest User metadata identity as a surrogate for these public-only users. This is a requirement if you are using Web authentication.

To designate the SAS Guest User as a surrogate identity for public-only users of SAS Web Report Studio, complete these steps:

- 1 Open the **LocalProperties.xml** file. For instructions on creating a **LocalProperties.xml** file, see “Create a LocalProperties.xml File” on page 107.
- 2 In the **wrs.pfs** section, edit the properties to specify these values:
  - Set the **allowPublicUsers** property to **true**.
  - Set the **publicUserSurrogate.activate** property to **true**.
  - Set the **publicUserSurrogate.uid** property to **sasguest**.
  - Set the **publicUserSurrogate.pw** property to the password that your site is using for the **sasguest** account.
  - Set the **publicUserSurrogate.domain** property to **web** (if you are using Web authentication) or **DefaultAuth** (if you are not using Web authentication).

If your **LocalProperties.xml** file does not contain the **wrs.pfs** section, then you can add it to the file by copying the section block from the **WebReportStudioProperties.xml.orig** file.

- 3 Redeploy SAS Web Report Studio. For instructions, see “Re-Create and Redeploy SAS Web Report Studio” on page 116.

After you make these changes, the access controls and logins that are defined for the SAS Guest User are applied to every public-only user. For example, all of the resources that are available to the SAS Guest User are also available to all public-only users. The report folders that belong to public-only users function as shared folders; these folders are not protected.

*Note:* If you are not using Web authentication, you can choose to configure SAS Web Report Studio to require each user to have an individual metadata identity in order to log on. This is controlled by the **wrs.pfs.allowPublicUsers** property.  $\Delta$

## Using SAS Web Report Studio Roles

### Understanding SAS Web Report Studio Roles

By default, everyone who can log on to SAS Web Report Studio can view, edit, and create new reports.

To implement security, each user of SAS Web Report Studio can be assigned to one or more standard roles. A user’s role assignments determine which SAS Web Report Studio menu items are available to that user. These roles facilitate administration in a couple of ways. First, by default, all users have full permissions. When you are ready to lock down your environment, you need only limit roles to particular groups of users that you have defined in metadata. In addition, there is a small number of roles for you to manage. Each role can be assigned multiple groups of users.

The roles correspond to user groups in metadata. You can use the User Manager in SAS Management Console to add or remove users and other groups. For instructions on using the User Manager, select User Manager in the SAS Management Console navigation pane and then select Help from the menu bar.

The roles are created for you during installation. If you need to recreate a role, then you must specify the exact role name as specified in your application properties files. This documentation assumes that the default role names are used.

The next table describes a group of roles that have the following characteristics:

- By default, all SAS Web Report Studio users implicitly have the role. However, if you explicitly assign any members to the role, then only the explicitly-assigned members will have the role. This enables you to start using SAS Web Report Studio immediately after installation, yet still have the ability to restrict user access when locking down your deployment.
- Each role is a superset of the preceding role. For example, members of the "WRS Report Author" role have all the permissions that apply to the "WRS Report Consumer".
- Once you explicitly assign members to a role, you must explicitly assign members to each superset role. For example, if you assign members to the "WRS Report Author" role, then all of the subsequent superset roles (in this example, "WRS Advanced User") also become explicitly-assigned roles. The reason is that WRS Advanced User is a superset of WRS Report Author.
- Once you explicitly assign members to a role, then any user who is not assigned to a role, or who has no metadata identity, can only view reports and manipulate reports (for example, select new data items to view in report objects).

**Table 7.3** User Roles That Apply to All SAS Web Report Studio Users by Default

Role (Default Group Name in Metadata)	Capabilities
WRS Report Consumer	Users who have this role can view reports and manipulate report data in the View Report view. Users can copy, move, save, rename, or delete reports. Users cannot create new reports with the report builder or report wizard.
WRS Report Author *	In addition to the abilities assigned to WRS Report Consumers, users who have this role can create reports with the report builder or report wizard. Users can also schedule reports.
WRS Advanced User	In addition to the abilities assigned to WRS Report Authors, users who have this role can distribute reports. Users cannot create or delete recipient lists that are used for report distribution.

\* By default, WRS Report Authors can schedule reports, though you can change the default behavior and limit the scheduling feature to WRS Advanced Users. To do this, in your **LocalProperties.xml** file, specify true for the **schedulingRequiresAdvancedUserRole** property. For instructions on creating a **LocalProperties.xml** file, see “Create a LocalProperties.xml File” on page 107.

The following table describes user roles that do not have any members by default. You must explicitly add members to these roles.

**Table 7.4** User Roles That Have No Members by Default

<b>Role (Default Group Name in Metadata)</b>	<b>Capabilities</b>
WRS Administrator	<p>Users who have this role can perform all tasks that are associated with SAS Web Report Studio, including the ability to create and delete recipient lists that are used for report distribution.</p> <p>This role provides full permissions to SAS Web Report Studio and should be safeguarded accordingly. This role provides application level administrator functionality. However, this role has no effect on metadata access (authorization) rights to report data.</p>
WRS Prohibited	<p>Users who have this role cannot log on to SAS Web Report Studio. Regardless of the user's membership in any of the previous roles, if the user attempts to log on, the logon page displays the following error message: "This user is not allowed to access SAS Web Report Studio. Please contact your administrator."</p> <p>Some organizations might apply this role for users who are allowed to access some SAS applications but not SAS Web Report Studio. Alternatively, if an organization has multiple Web Report Studio installations, this role can be used to restrict some users from specific instances.</p> <p>The corresponding metadata group entity is not created during installation. You must manually create the group in metadata if you want to use this user role.</p>

## Using the Roles: Example Scenario

Suppose that your sales organization consists of the following three groups of people:

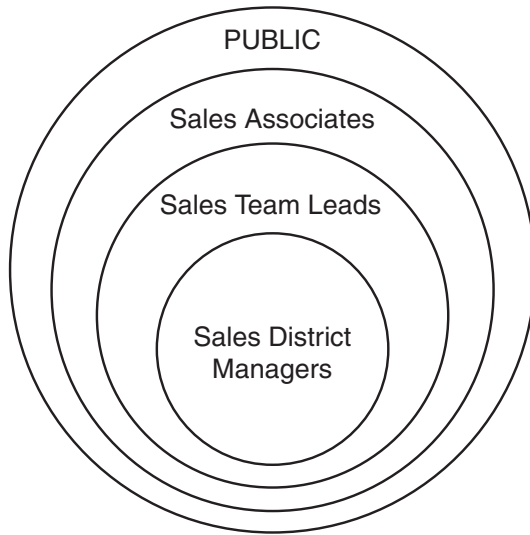
- Sales associates, who can create their own reports in order to evaluate trends, perform customer analysis, and measure their own sales performance.
- Sales team leads, who can create and distribute reports that are of interest to their team. For security purposes, you want to restrict report distribution to team leads.
- Sales district managers, who can create and distribute reports that are of interest to their district. In addition, they manage the recipient lists that are used for all report distribution. It is important that reports be distributed only to authorized people, so management of these lists is critical to your organization's security.

Everyone in the company can view sales reports that are made publicly available (that is, they are saved to the Shared folder in SAS Web Report Studio).

After you decide how these groups relate to each other, you can define the groups in metadata. In the metadata layer, one group can be a member of other groups. In this example, the Sales Associates group includes the Sales Team Leads and the Sales District Managers groups. The Sales Team Leads group includes the Sales District Managers group. The Sales District Managers group does not include any other group, although it might include the SAS administrator.

The following figure depicts this relationship:

**Figure 7.2** Example Group Relationships



The groups in this example can now be associated with the following user roles:

Sales Group	Associated User Role
All employees (PUBLIC group)	WRS Report Consumer
Sales Associates	WRS Report Author
Sales Team Leads	WRS Advanced User
Sales District Managers	WRS Administrator

Each user role has a corresponding group in metadata. The metadata groups were created when you installed SAS Web Report Studio. By default, all users belong to all the user roles, except the WRS Administrator. Your goal is to limit user roles in report creation and distribution for security purposes.

To associate the groups with the user roles, do the following:

- 1 In SAS Management Console, create your sales groups in metadata. For instructions on creating groups, see the User Manager Help in SAS Management Console.
- 2 Take no action on the WRS Report Consumer group. There is no need to limit the users for this user role because you have already decided that everyone in your organization can view reports.
- 3 Add the Sales Associates group as a member of the WRS Report Author group.
 

With this change, only the Sales Associates group (and any groups it contains) can create reports. In addition, the WRS Advanced User group now has no members because it is a superset of the WRS Report Author group. Once you explicitly assign members to a user role, you must explicitly assign members to any of its superset groups.
- 4 Add the Sales Team Leads group as a member of the WRS Advanced User group.
 

With this change, only the Sales Team Leads group (and any groups it contains) can distribute reports.

- 5 Add the Sales District Managers group as a member of the WRS Administrator group.

With this change, only the Sales District Managers group can manage recipient lists for report distribution.

After you have created groups and associated the groups with user roles, you can add users as members of the groups.

---

## Additional Authentication for SAS Web Report Studio Users

SAS Web Report Studio users might require additional credentials in order to access one or more of servers (such as a workspace server or a stored process server). For more information, see “Planning User Accounts and Their Organization into Groups” on page 21.

Changes that you make to the default SAS Web Report Studio configuration can affect your user’s ability to access servers. For example:

- If you modify your deployment to use pooled workspace servers, then it is no longer necessary for your users to have accounts or logins for the purpose of accessing the workspace server. Instead, each SAS Web Report Studio user must be a member of at least one group that is associated with a puddle within the pool. In the simplest case, the PUBLIC group is associated with the only puddle, so all users can access the pooled workspace servers. In a more restricted environment, the puddle might be associated with a user-defined group that includes only those users who should access the pooled workspace servers.
- If you modify your deployment to use Web authentication, then additional logins are required. For example, if you are using Web authentication with an LDAP authentication provider, you could choose to meet the additional authentication requirements as follows:
  - To enable users to access the SAS Workspace Server, use a pooled configuration (with a single puddle that is associated with the PUBLIC group).
  - To enable users to access the SAS OLAP Server, use the same LDAP authentication provider for the SAS OLAP Server as you are using for the Web application server.
  - To enable users to access the SAS Stored Process Server, give each user an individual or shared account in the host operating system and an additional individual or group login in the metadata. The additional login must include the credentials for the operating system account. The additional login must be associated with the stored process server’s authentication domain.

*Note:* In all of these scenarios, if you set up a surrogate public user, then the surrogate user’s logins are available to users who do not have their own metadata identities.  $\Delta$

---

## Managing Access to Reports

---

### Overview: Managing Access to Reports

The following table summarizes the basic security considerations for reports.



**Table 7.5** Report Security Considerations

In order to protect	You must secure
Report definitions	The metadata objects that are associated with the reports The physical storage location of the report definitions
Underlying report data	The metadata objects that are associated with the report data The physical storage location of the report data The information maps that reference the report data The stored processes that reference the report data The report definition (if the report includes embedded data) The generated report (if the report is a batch report)

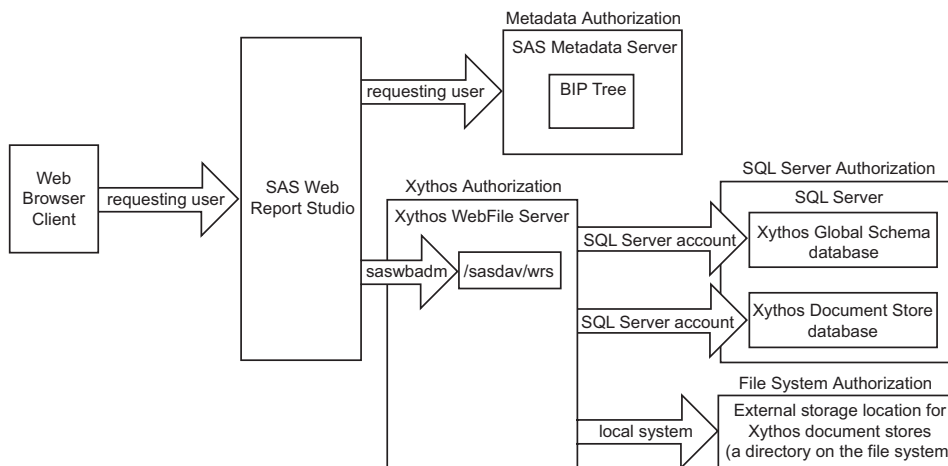
Different types of reports require different security measures. For example, if there is no embedded data of a sensitive nature in a report definition, then the report definition can be considered secure if the report’s underlying data, information maps, stored processes, and output are secure. However, batch 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 Xythos WFS content server with the Xythos document store located in a directory in the file system.

**Figure 7.3** Authorization Layers for SAS Web Report Studio



In the figure, the requesting user’s access to reports is subject to controls in the metadata, Xythos, SQL 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 **ReportStudio** folder structure includes appropriate metadata access controls for the **Shared** folder and the **Users** report folders. Additional steps should be taken in the metadata layer to protect your reports.
- The only relevant permissions for reports are ReadMetadata and WriteMetadata. The Read, Write, Create, Delete, CheckInMetadata, and Administer permissions have no effect on these objects. For more information, see “Which Actions are Controlled by Each Permission?” in the “Understanding Authorization” section in the *SAS Intelligence Platform: Security Administration Guide*.
- The effective permissions for a report folder are inherited by all of the reports within that folder. For more information, see “Inherited Access Controls” in the “Understanding the Metadata Authorization Layer” section in the *SAS Intelligence Platform: Security Administration Guide*.
- The ability to view or work with a report can be affected by access to each of the report’s underlying components. For information about the metadata layer requirements for working with reports and report folders, see “Access Requirements for Reports” in the “Access Guidelines and Requirements” section in the *SAS Intelligence Platform: Security Administration Guide*.
- If your organization uses publication channels to deliver reports, the reports can also be protected by controlling access to the publication channels. A user must have ReadMetadata access to a SAS Publication Channel in order to self-subscribe to that channel. In addition, you might need to edit the policy file to add or remove permissions for the folders that correspond to the channels. To learn how to set up a publication channel, see “Adding SAS Publication Channels” on page 292.

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 Xythos process is running (local system in this example) to access this physical file location.

Similarly, SAS Web Report Studio uses only one account (saswbadm) to communicate with the WebDAV content server, so you cannot make access distinctions between individual users by setting access controls in Xythos WFS. However, you should use this layer to protect your SAS report content, as described in “Protecting Report Content in the WebDAV Server” on page 137.

---

## Changing Access to Reports

Each new report subfolder that you create in metadata inherits the effective access controls of its parent folder. You can change the permission settings for any folder by navigating to the folder in SAS Management Console under **Environment Management ► Authorization Manager ► Resource Management ► By Application ► BIP Tree ► ReportStudio** and accessing the folder’s **Authorization** tab. This structure enables you to store reports that everyone will access in the **ReportStudio/Shared/Reports** folder and to limit access to each of the subfolders so that only certain user groups can see or modify the folder and its contents.

---

## Security Considerations for Pre-generated Batch Reports

When a batch report is generated, 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 batch report is configured to use an identity named BATCH, then the report can include anything that BATCH is able to access. Regardless of who actually views the report, the report content is always based on the access controls that apply to BATCH. This means that any user who has ReadMetadata permission for a batch 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 batch report uses for generation, and you must secure the batch reports that you create.

*Note:* Once a user refreshes the report data, that user sees only the content that he or she has permission to see. △

---

## Considerations for Row-level Security

If you implement 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. 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.

For more information about row-level security, see the *SAS Intelligence Platform: Security Administration Guide*.

---

## Protecting 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 recommended approach is to have SAS Web Report Studio use an area named **sasdav/wrs** on the content server, and to use the content server's access controls to give the SAS Web Administrator account (saswbadm) exclusive access to that area. The user ID and password of the SAS Web Administrator are available to SAS Intelligence applications through the SAS metadata repository. Applications that are not aware of SAS metadata do not have access to the saswbadm user ID and password, so these applications cannot access the **sasdav/wrs** area of the content server.

For example, if you are using Xythos WFS as your content server, you can verify that these protections are in place by completing these steps:

- 1 In SAS Management Console, navigate to **Environment Management ► BI Manager ► BIP Tree ► Properties** and verify that your deployment of SAS Web Report Studio has been configured to use the SAS Web Administrator (saswbadm) to access the **sasdav/wrs** area in your content server.

*Note:* If you have installed the SAS Foundation Services 1.2, then you should use the BI Manager plug-in instead of the Business Report Manager plug-in. Starting with that release, the BI Manager replaces the Business Report Manager, and is the recommended SAS Management Console plug-in for administering reports. △

- 2 Access the Xythos administration console by opening a Web browser to `<machine-name>:8300/xythosadmin` and logging on as the Xythos administrator.
- 3 Select **File System ► Directory & File Admin**.
- 4 Select the permissions icon for the **sasdav/wrs** directory.

- 5 On the access permissions page for the **sasdav/wrs** directory, verify that the SAS Web Administrator has exclusive, full access to this directory (and to this directory's subdirectories and files).
- 6 Verify that the SAS Web Administrator (saswbadm) can access the **sasdav/wrs** directory by completing these steps:
  - a Open a Web browser to **<machine-name>:8300/sasdav/wrs** and use the credentials of the SAS Web Administrator to log on.
  - b On the index page for **sasdav/wrs**, select Launch Web Folder.
  - c Drag and drop a local file onto the page to add it to the folder.
  - d Delete the file that you added to the folder.
- 7 Verify that other users cannot directly access the **sasdav/wrs** area by completing these steps:
  - a Delete the credentials that were cached for the SAS Web Administrator on the machine where you are working.
  - b Open a Web browser to **<machine-name>:8300/sasdav/wrs** and use the credentials of the SAS Demo User to log on.
  - c Instead of seeing the index page for the **sasdav/wrs** directory, you should see a "Page Not Found" message.

These are the only Xythos layer access controls that you should set in the **sasdav/wrs** content area, because these are the only Xythos layer access controls that are meaningful for SAS Web Report Studio. Administration of the SAS Information Delivery Portal requires you to use Xythos layer access controls to manage access for other content areas within the WebDAV server.

---

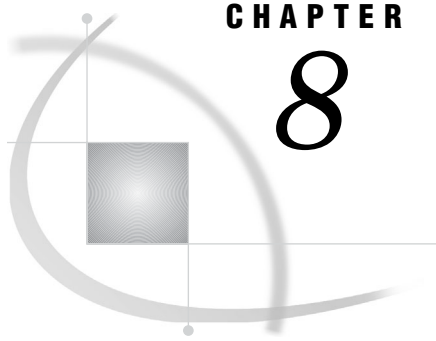
## Protecting Data in the SAS Web Report Studio Temporary Files

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 Tomcat, this location might be **C:\Tomcat4.1\work\Standalone\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, you should do these things:

- Place the computer on which SAS Web Report Studio is deployed in a physically secure location.
- 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.



## CHAPTER

## 8

## Customizing Reports

<i>Add Disclaimer Text to Graphs and Tables</i>	139
<i>Customizing Report Styles</i>	140
<i>Overview of Providing Custom Report Styles</i>	140
<i>Specify a Style in the Properties File</i>	140
<i>CSS Formats for Custom Report Styles</i>	141
<i>About CSS Formats</i>	141
<i>Tables</i>	142
<i>Graphs</i>	144
<i>Text</i>	147
<i>Synchronized Objects Container</i>	148
<i>Display Filters</i>	149
<i>Supported Properties</i>	150

### 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 Open the **LocalProperties.xml** file. For instructions on creating the **LocalProperties.xml** file, see “Create a LocalProperties.xml File” on page 107.
- 2 In **LocalProperties.xml**, add or edit the following element block:

```
<wrs.disclaimer.tableAndGraph>
MyDisclaimer
</wrs.disclaimer.tableAndGraph>
```

In the above block, replace *MyDisclaimer* with your own disclaimer text.

The text will wrap automatically; you cannot specify separate lines by inserting a carriage return or new line. Graphics are not supported in disclaimer text.

- 3 Save your changes.
- 4 Restart the servlet container.

The disclaimer text does not affect existing reports. The text will appear beneath the tables and graphs of all new reports.

---

# Customizing Report Styles

---

## Overview of Providing Custom Report Styles

Report styles affect the colors, fonts, and other elements that are used in tables and graphs. By default, report viewers can select one of the following styles for a report: Meadow, Seaside, or Festival. You can add your own custom styles to the list of available styles.

*Note:* The ability to apply custom styles is currently available only for applications that run with a U.S. locale.  $\Delta$

SAS Web Report Studio relies on cascading style sheets (CSS) to render styles. To supply a custom style, follow these steps:

- 1 Create a CSS file and define the formats that you want for the style. For details about the supported formats as well as a sample CSS file, see “CSS Formats for Custom Report Styles” on page 141.
- 2 In your properties file, provide information that SAS Web Report Studio needs in order to locate and render the style. For instructions, see “Specify a Style in the Properties File” on page 140.

---

## Specify a Style in the Properties File

In your **LocalProperties.xml** file, you must provide information that SAS Web Report Studio needs in order to locate and render the style that you want to use. To provide this information, complete these steps:

- 1 Open the **LocalProperties.xml** file in a text editor. If you don't have this file, then you can create it. See “Create a LocalProperties.xml File” on page 107.
- 2 In **LocalProperties.xml**, add the following element block if it's not already there (you can copy and paste from the **WebReportStudioProperties.xml.orig** file):

```
<sas.wrs.style>
<css></css>
<schemelist>
Seaside,Festival,Meadow
</schemelist>
<defaultscheme></defaultscheme>
</sas.wrs.style>
```
- 3 In the element block, modify the elements to specify your CSS file and style scheme. The following table describes the elements:

**Table 8.1** Report Style Elements in LocalProperties.xml

Element	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 file name 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 &lt;css&gt; list (but without the file path or CSS file name 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 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 Seaside.</p>

This example shows how these properties can be specified:

```
<sas.wrs.style>
<css>C:\styles\CustomScheme1.css,C:\styles\CustomScheme2.css</css>
<schemelist>
Seaside,Festival,Meadow,CustomScheme1,CustomScheme2
</schemelist>
<defaultscheme>CustomScheme1</defaultscheme>
</sas.wrs.style>
```

- 4 Save your changes.
- 5 If you intend to use SAS Web Report Viewer to render reports, then you must make similar changes for SAS Web Report Viewer. Open the **LocalProperties.xml** file for SAS Web Report Viewer, and repeat the previous steps.
- 6 You must restart the servlet container before your changes take effect.

## CSS Formats for Custom Report Styles

### About CSS Formats

In order to provide custom report styles, you create one or more CSS files. A CSS file enables specified formats (CSS rule sets) to be available for end users to modify in SAS Web Report Studio.

Here are the elements that 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 on making the CSS formats available to SAS Web Report Studio, see “Specify a Style in the Properties File” on page 140. For information about CSS files in general, consult the W3C organization’s Web site at <http://www.w3.org/TR/CSS21/>.

## Tables

The following figure shows a sample list table.

**Display 8.1** Sample List Table

Table Title				
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
<b>253</b>	<b>1184.4</b>	<b>Total</b>		<b>1900.5</b>

Here are the supported style formats for elements in the list table.

**Table 8.2** CSS Formats for List Tables

Callout Number	Selector	Supported Property Types
① (title)	Table Caption	text
② (headings)	Table Column Label	text cell border



Callout Number	Selector	Supported Property Types
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 vs. 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 details about the supported property types, see “Supported Properties” on page 150.

The following figure shows a sample crosstabulation table.

**Display 8.2** Sample Crosstabulation Table

usregion		Mid Atl	S Atl	Total			
year	product	Sum Of Pop 2000	Sum Of Year	Sum Of Pop 2000	Sum Of Year	Sum Of Pop 2000	Sum Of Year
2002	Bed	160975680	1209208	204788404	4708704	365764084	5917912
	Chair	160975680	1209208	204788404	4708704	365764084	5917912
	Desk	160975680	1209208	204788404	4708704	365764084	5917912
	Sofa	160975680	1209208	204788404	4708704	365764084	5917912
	Subtotal	643902720	4836832	819153616	18834816	1463056336	23671648
2003	Bed	160975680	1209812	204788404	4711056	365764084	5920868
	Chair	160975680	1209812	204788404	4711056	365764084	5920868
	Desk	160975680	1209812	204788404	4711056	365764084	5920868
	Sofa	160975680	1209812	204788404	4711056	365764084	5920868
	Subtotal	643902720	4839248	819153616	18844224	1463056336	23683472
Total	Subtotal	1287805440	9676080	1638307232	37679040	2926112672	47355120

Here are the supported style formats for elements in the crosstabulation table.

**Table 8.3** CSS Formats for Crosstabulation Tables

Callout Number	Selector(s)	Supported Property Types
① (title)	Table Caption	text
② (headings)	Table Rowgroup Label	text
	Table Rowgroup Row Label	cell
	Table Columngroup Label	border
	Table Columngroup Column Label	

Callout Number	Selector(s)	Supported Property Types
③ (border)	Table	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
⑥ (subtotals)	Table Rowgroup Rows Summary	text
	Table Columngroup Columns Summary	cell
		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 vs. 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.  $\Delta$

For more details about the supported property types, see “Supported Properties” on page 150.

## 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. Since subgrouping is data dependent (one subgroup might have three values, whereas the same subgroup on different data might have nine values), 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 : DIAMONDFILLED;
    marker-size : 10px;
    line-thickness : 2px;
}
```

```

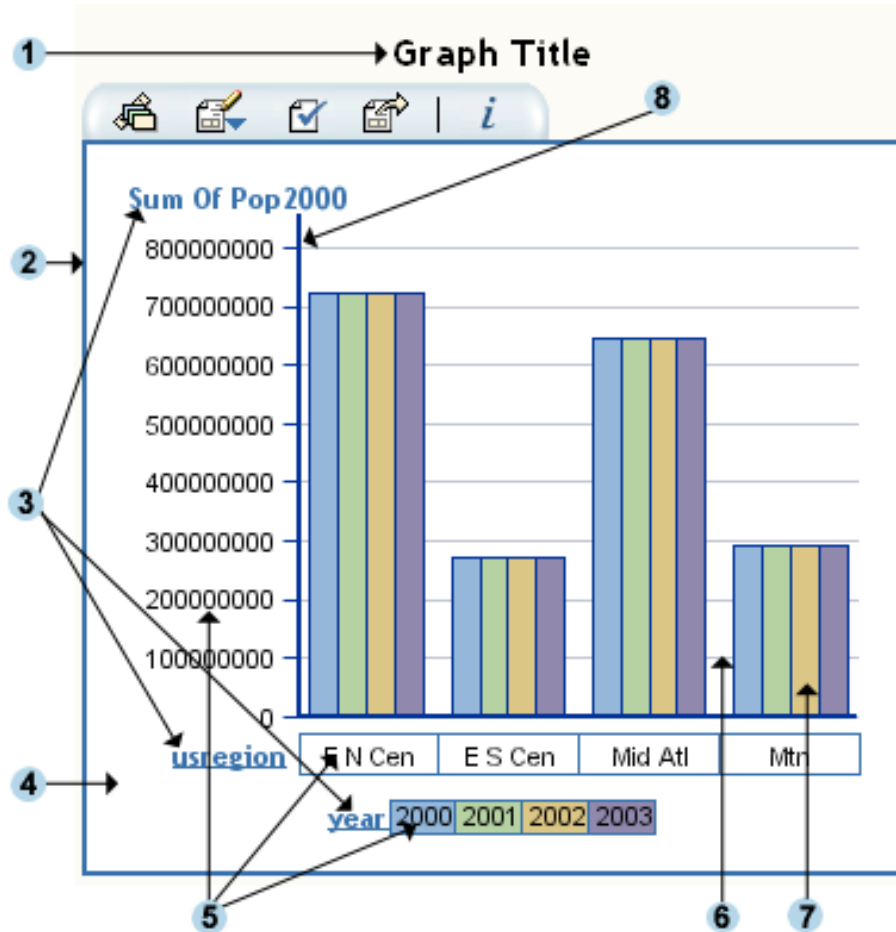
Graph GraphDataStyle3
{
    color : blue;
    marker-symbol : DIAMONDFILLED;
    marker-size : 10px;
    line-thickness : 2px;
}
    
```

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 chart 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.

**Display 8.3** Sample Graph



**Table 8.4** CSS Formats for Graphs

Callout Number	Selector	Supported Properties and Property Types
❶ (title)	Graph TitleText	text
❷ (border)	Graph BorderLines	line-color property
❸ (axis and legend labels)	Graph LabelText	minimal text
❹ (background)	Graph BackFill	fill-color property
❺ (axis and legend values)	Graph ValueText	minimal text
❻ (axis and legend values)	Graph LegendFill	fill-color property
❼ (grid lines)	Graph GridLines	line-color property
❼ (data)	Graph GraphDataStylen	graph data styles
❽ (horizontal and vertical axis)	Graph AxisLines	line-color property line-thickness property

For more details about the supported property types, see “Supported Properties” on page 150.

The geographical (ESRI) chart supports only the border style.

The progressive bar chart does not support the *GraphDataStylen* formats. Instead, the chart uses three different formats for its data styles. These formats are unique to the progressive bar chart.

Display 8.4 Sample Progressive Bar Chart

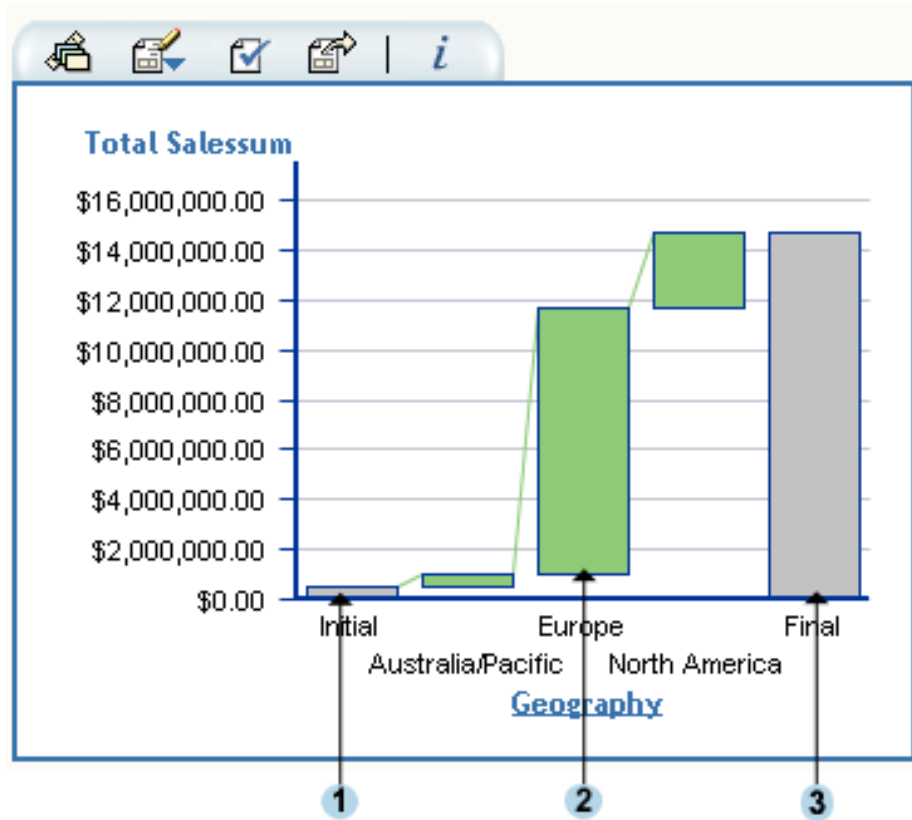


Table 8.5 CSS Formats for Progressive Bar Charts

Callout Number	Selector(s)	Supported Properties
① (initial bar)	Graph InitialDataStyle	fill-color
② (positive/negative bars)	Graph ThreeColorRamp Graph ThreeColorAltRamp	fill-gradient-start-color fill-gradient-end-color
③ (final bar)	Graph FinalDataStyle	fill-color

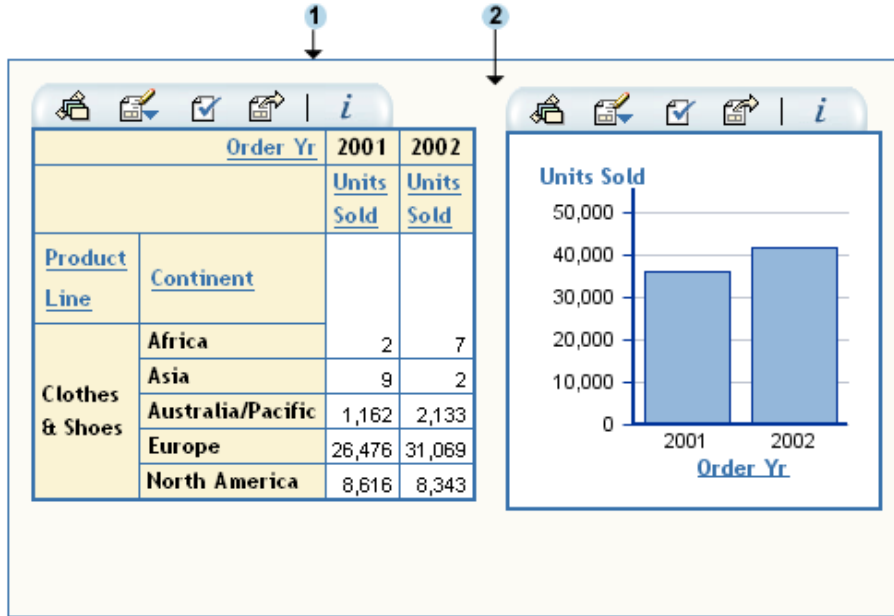
## Text

Text elements, including headers and footers, use the `text` property type, and support all text formats.

## Synchronized Objects Container

The following illustration shows a container for synchronized objects, followed by a list of the supported formats.

**Display 8.5** Sample Synchronized Objects Container



**Table 8.6** CSS Formats for a Synchronized Objects Container

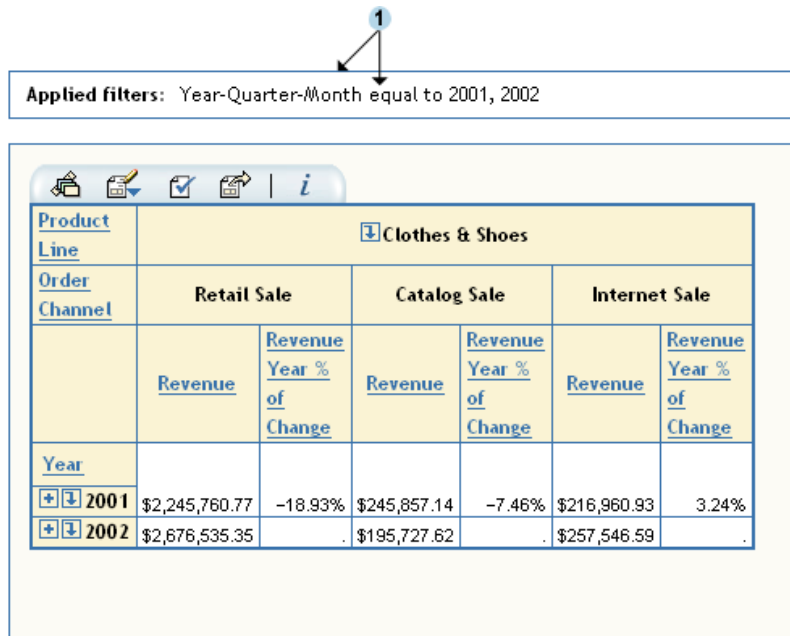
Callout Number	Selector	Supported Properties or Property Types
❶ (container border)	LinkedContainer	border
❷ (container)	LinkedContainer	background-color property padding property

For details about the supported property types, see “Supported Properties” on page 150.

## Display Filters

Display Filters must be specified individually for graphs, tables, and the containers for synchronized objects. The following figure shows display filters for a table. Display filters are similar for graphs and synchronized object containers.

**Display 8.6** Sample Display Filters



Here are the supported style formats for display filters. 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.

**Table 8.7** CSS Formats for Display Filters

Callout Number	Selector	Supported Property Types and Properties
① (filter)	Graph DisplayFilter	text border margin-bottom
① (filter)	Table DisplayFilter	text border margin-bottom
① (filter)	LinkedContainer DisplayFilter	text border margin-bottom

*Note:* For each of the formats in the table, 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. △

## Supported Properties

This table lists the properties that are supported for the property types that are mentioned in the previous sections.

**Table 8.8** Supported Properties for CSS Formats

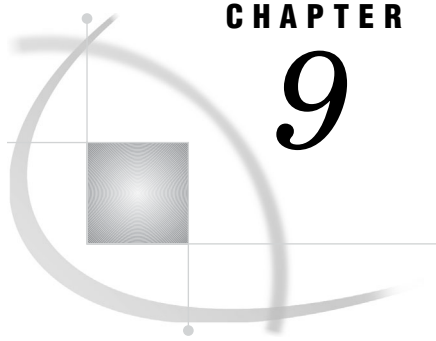
Property Type	Supported Properties
text	font-family font-weight color background-color text-align font-size text-decoration font-style
minimal text	font-family font-weight text-color font-size font-style
border	border border-color border-top-color border-bottom-color border-right-color border-left-color 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



<b>Property Type</b>	<b>Supported Properties</b>
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.





## CHAPTER

## 9

## Scheduling and Distributing Pre-generated Reports

<i>Overview: Scheduling and Distributing Pre-generated Reports</i>	153
<i>Understanding Pre-generated Reports</i>	153
<i>Required Permissions for Scheduling and Distributing Reports</i>	154
<i>Methods for Scheduling and Distributing Reports</i>	155
<i>How Report Scheduling Differs From Report Distribution</i>	155
<i>Main Administrative Tasks for Scheduling and Distributing Reports</i>	156
<i>Setting Up a Distribution Library and Recipient List</i>	156
<i>Overview: Setting Up a Distribution Library and Recipient List</i>	156
<i>Create a Library for Recipient Lists</i>	157
<i>Creating a Recipient List for Report Distribution</i>	158
<i>Understanding How Recipient Lists Enable Report Distribution</i>	158
<i>Create a Recipient List</i>	159
<i>Alternative Example: Create a Recipient List Using PROC SQL</i>	162
<i>Considerations for Creating Recipient Lists</i>	163

### Overview: Scheduling and Distributing Pre-generated Reports

#### Understanding Pre-generated Reports

SAS Web Report Studio enables you to run queries against reports and provide the results to users. One advantage of providing pre-generated results is improved performance. It takes less time to view the report 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, which are described in “Security Considerations for Pre-generated Batch Reports” on page 136.

Pre-generated reports can be provided to users in three ways:

- You can save any report as a static report in PDF format. The report is saved in your repository and is made available to authorized users. Users cannot interact with a report that has been saved in PDF format.
- You can create manually refreshed reports. Manually refreshed reports are saved reports that contain data from a pre-generated query. The report is stored in your repository, and the results can be viewed in SAS Web Report Studio or SAS Web Report Viewer. When users open the report, they can manually refresh the report data.

There are two ways to create a manually refreshed report:

- You can select **Data can be manually refreshed** from the Save As dialog box when you save the report. This type of report typically requires a manual refresh each time it is opened.
- You can schedule saved reports to be run at a specified time. This feature enables you to refresh report data at specified intervals or times. You can also specify an archive in order to maintain older versions of 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 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.
- The third way to provide a pre-generated report is to create a snapshot of report data, and then distribute the static results to recipients.

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.

The distributive nature of this feature presents some inherent risks for an organization. For more information about those risks, see “Security Considerations for Pre-generated Batch Reports” on page 136.

The remainder of this topic describes scheduling manually refreshed reports and distributing static snapshot reports.

---

## Required Permissions for Scheduling and Distributing Reports

Users must be assigned to the following roles in order to schedule or distribute reports:

**Table 9.1** User Roles for Scheduling and Distributing Tasks

Task	Role Requirement
Schedule a report *	WRS Report Author
Schedule a folder of reports	WRS Advanced User
Specify an archive for reports and control the size of the archive	WRS Advanced User

Task	Role Requirement
Distribute a report	WRS Advanced User
Create or delete a recipient list for distributed reports	WRS Administrator

\* You can specify that all users must have WRS Advanced User permissions in order to schedule a report by changing the value of `schedulingRequiresAdvancedUserRole` to `true` in the `LocalProperties.xml` file. If you don't have a `LocalProperties.xml` file, then you can create one. See "Create a LocalProperties.xml File" on page 107.

For more information about these user roles, see "Using SAS Web Report Studio Roles" on page 130.

*Note:* In addition to the above role requirements, you must have configured a *trusted user* (sastrust by default) in order to schedule or distribute reports. You configured the trusted user during installation. The SAS Trusted User is used to establish a trust relationship with the metadata server. The `OutputManagementConfigTemplate.xml` file contains the user ID and password for the SAS Trusted User. The `OutputManagementConfigTemplate.xml` file resides in the installation directory.  $\Delta$

## Methods for Scheduling and Distributing Reports

Once you have enabled scheduling (as described in "Main Administrative Tasks for Scheduling and Distributing Reports" on page 156), you can schedule or distribute reports using either of these methods:

- In SAS Web Report Studio, report creators can create reports and control when those reports are updated or distributed. For instructions, see the Help for SAS Web Report Studio.

Administrators can limit access to this scheduling functionality by assigning only selected users to the WRS Advanced User role. For details, see the discussion of user roles in "Setting up Users for SAS Web Report Studio" on page 129.

- In SAS Management Console, administrators can use BI Manager to create a job for scheduling and then use Schedule Manager to schedule the jobs. For instructions, see the Help for each of these plug-ins.

## 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 `outputgen.exe` executable to create output. (The `outputgen.exe` tool replaces the `batchgen.exe` tool that was included with previous releases.)

However, there are several differences between scheduling and distribution. The following table summarizes these differences:

**Table 9.2** 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.	Reports can be pushed to a publication channel. Publication channels are specified in the recipient list.

Report Scheduling	Report Distribution
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.	Reports cannot be archived.
Scheduling does not include the ability to preview a schedule.	Users can run a test to preview the distribution of a report. 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.	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. △

---

## Main Administrative Tasks for Scheduling and Distributing Reports

The scheduling and distribution features use trusted authentication and rely on the SAS Query and Reporting Services, which were configured during installation.

In addition to this initial configuration, you must satisfy the following requirements before users can schedule or distribute reports:

- Before users can schedule or distribute reports, you must have installed Platform LSF (Load Sharing Facility) and SAS scheduling software. In addition, you must complete the configuration steps that are documented in "Enabling the Scheduling of Reports" in the *SAS Intelligence Platform: System Administration Guide*. (For an overview of scheduling, see "SAS Scheduling Overview" in the *SAS Intelligence Platform: System Administration Guide*.)
- Before users can distribute reports, you must do the following:
  - 1 Create a library in the metadata repository to contain the tables for your distribution recipient lists.
  - 2 Create the recipient lists for the reports.
 For details and instructions, see "Setting Up a Distribution Library and Recipient List" on page 156.

---

## Setting Up a Distribution Library and Recipient List

---

### Overview: Setting Up a Distribution Library and Recipient List

In order to distribute reports to recipients, you must define those recipients in the metadata repository. First you create a library to contain the tables for your recipient

lists. Then you can create the recipient list tables for your reports. The sections that follow explain how to create the library and the recipient lists.

---

## Create a Library for Recipient Lists

SAS Web Report Studio stores information about recipients in data sets within a SAS library. Depending on how you installed SAS Web Report Studio, you might have been prompted to specify the name of that library. Regardless of whether you were prompted for the name, a library path was created on your host machine. You must define that library in the metadata repository before the library can become available for use by SAS Web Report Studio.

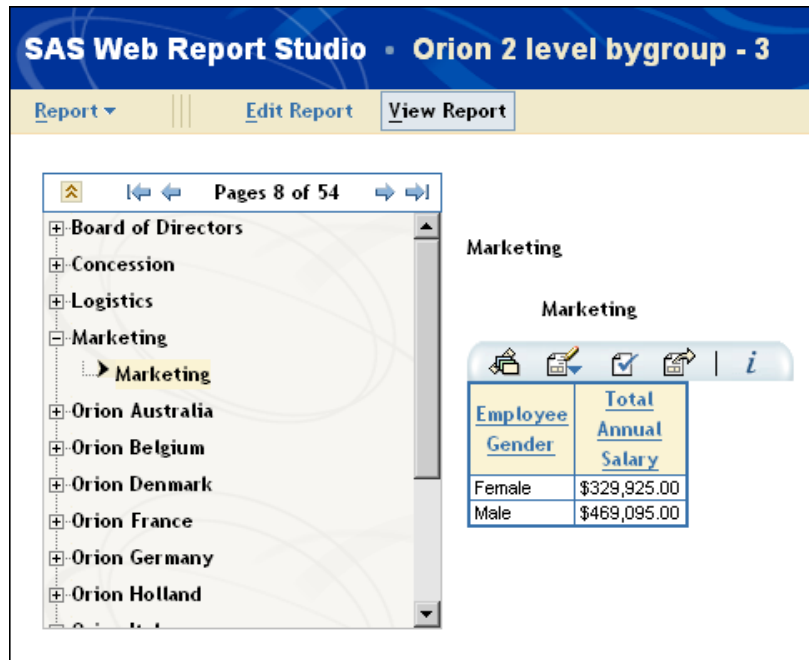
Perform the following steps in SAS Management Console:

- 1 Double-click the **Data Library Manager**. Right-click the **SAS Libraries** icon. Then, select the **New Library** option to access the first screen of the **New Library Wizard**.
- 2 Select **SAS Base Engine Library** from the list of **SAS Libraries**. Click **Next** to access the next screen of the wizard.
- 3 In the **Name** field, enter the name of the library that was specified during installation. This value is specified in the `<libname>` element in the **WebReportStudioProperties.xml** file, which is found in the WEB-INF directory of your deployment. Optionally, supply a description of the library, and then click **Next** to access the next screen of the wizard.
- 4 Provide a `libref` value for the library, and specify *Base* for the engine. Then provide a path for the library. The path should match the library path that is found in `SAS-config-dir\Lev1\SASMain\appserver_autoexec.sas`. The default path is `SAS-config-dir\Lev1\SASMain\Data\wrsdist`. Click **Next** to access the next screen of the wizard.
- 5 Select one or more SAS servers. The library is assigned to the server or servers that you select from this list. Click **Next** to access the next screen of the wizard.
- 6 Examine the finish screen of the wizard to ensure that the proper values have been entered. Click **Finish** to save the settings.

## Creating a Recipient List for Report Distribution

### Understanding How Recipient Lists Enable Report Distribution

Suppose that you want to distribute an employee salary graph to human resources (HR) representatives in different locations around the world. The following report summarizes salary information for men and women based on company location:



The screenshot shows the SAS Web Report Studio interface. The title bar reads "SAS Web Report Studio - Orion 2 level bygroup - 3". Below the title bar are buttons for "Report", "Edit Report", and "View Report". The main content area shows a tree view on the left with folders like "Board of Directors", "Concession", "Logistics", "Marketing", "Orion Australia", "Orion Belgium", "Orion Denmark", "Orion France", "Orion Germany", and "Orion Holland". The "Marketing" folder is selected. On the right, the report content is displayed for "Marketing". It includes a table with the following data:

Employee Gender	Total Annual Salary
Female	\$329,925.00
Male	\$469,095.00

This sample report was created with a group break on a variable named Company. The result is a separate report page for each value of Company (each main division or corporate office location).

In order to distribute the relevant report to each HR representative, you must create a recipient list that maps each Company value to one or more recipients. A recipient list 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.



The following SAS data set illustrates a sample recipient list for a report that has a group break on Company:

	Company	EMAIL	CHANNEL
1	Board of Directors		
2	Concession		
3	Logistics		
4	Marketing		
5	Orion Australia		
6	Orion Belgium		
7	Orion Denmark		
8	Orion France		
9	Orion Germany		
10	Orion Holland		
11	Orion Italy		
12	Orion Spain		
13	Orion UK		
14	Orion USA		
15	Purchasing		

As shown in the data set, the EMAIL and CHANNEL columns are empty. You must provide e-mail and channel information for recipients of the distributed report. For details, see “Create a Recipient List” on page 159.

*Note:* You can create recipient lists for reports that have more than one group break. For more information about nested group levels, see “Considerations for Creating Recipient Lists” on page 163.  $\Delta$

## Create a Recipient List

To create a recipient list, you first use SAS Web Report Studio to create an initial list that includes your group breaks. After you create the list, you use Base SAS to provide e-mail addresses and publication channels for that list.

To create a recipient list, complete these steps:

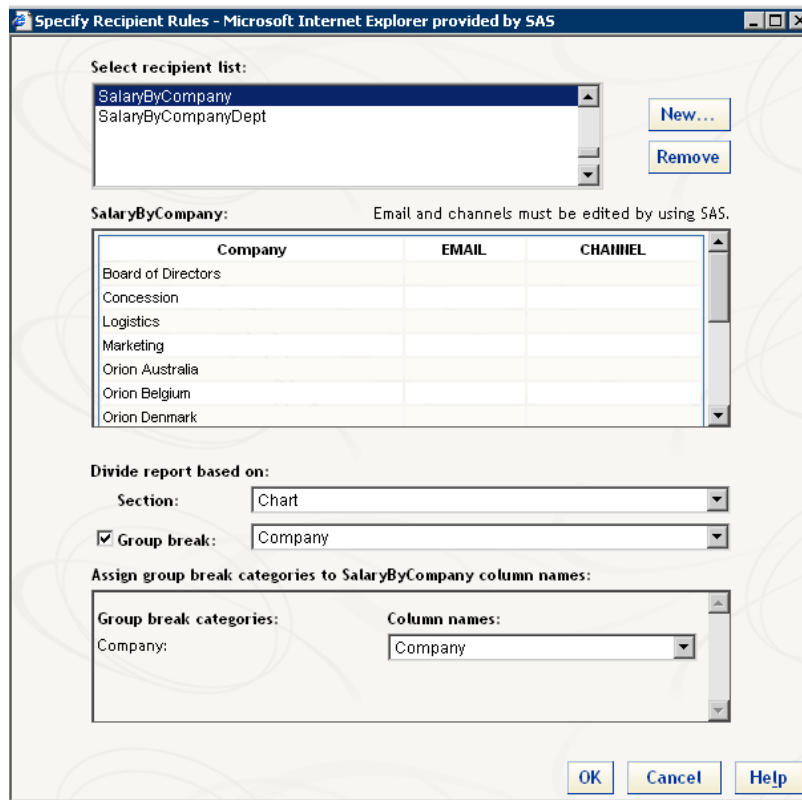
- 1 Log on to SAS Web Report Studio as a member of the WRS Administrator role. For information about this role, see “Using SAS Web Report Studio Roles” on page 130.
- 2 In SAS Web Report Studio, select the report that you want to distribute.
- 3 Start the Distribution wizard, and proceed as though you were creating a distribution.

Here is a summary of the steps that you take in the wizard:

- a In step 1 of the wizard, specify a date and time.
- b In step 2 of the wizard, click **Specify Recipient Rules**.

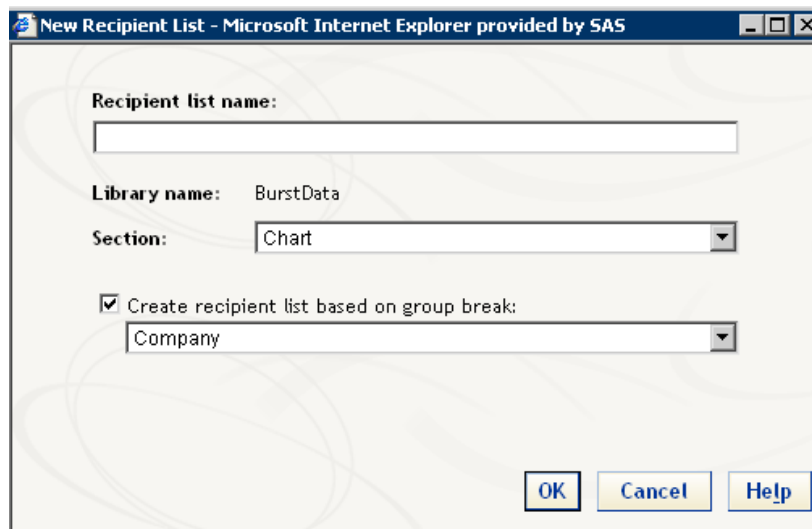
For full instructions, see the SAS Web Report Studio online Help.

- 4 In the Specify Recipient Rules dialog box, select **New**.



The **New** and **Remove** buttons are available only if you are logged on as a member of the WRS Administrator role. Be aware that if you remove a list, any distribution that references the list becomes nonfunctional.

- 5 In the New Recipient List dialog box, provide information about the recipient list that you want to create.



The following table explains the fields in this dialog box:

**Table 9.3** Fields in the New Recipient List dialog box

<b>Field</b>	<b>Description</b>
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 that you created in metadata for recipient lists. You will need this library name when you subsequently edit the list to add e-mail or channel recipients.
Section	The section that you want to use to subset the report by specifying a group break. This field is available only if the report contains more than one section.
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.

**6** Click **OK**.

**7** 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 library.

Next, you must specify the actual recipient e-mail addresses and/or publication channels.

**8** In Base SAS, open the data set that corresponds to the recipient list that you just defined. The data set will reside in the library that corresponds to the **Library name** value from the New Recipient List dialog box. You must run SAS from a system that can access the library location, and then you must assign the library by using the LIBNAME command.

*Note:* You should run SAS with the VALIDVARNAME= system option set to ANY. The SAS system option VALIDVARNAME controls the type of column names that can be used during a SAS session. For more information, see SAS Help and Documentation.

Note also that the column sizes are fixed. The group break columns are 256kb, and the EMAIL and CHANNEL columns have a fixed name and a fixed size of 1024kb. Do not change the name of the EMAIL and CHANNEL columns.  $\Delta$

**9** In the data set, provide the e-mail addresses and/or publication channels that are appropriate for the recipients.

Note the following about e-mail addresses and publication channels:

- The e-mail address must be in the form **John.Doe@abc.com**. Use this format also if you specify a distribution list (for example, **hr.mylist@abc.com** as opposed to **My List**).
- SAS Web Report Studio does not check the validity of the e-mail address that you provide. You are responsible for ensuring that every e-mail address is valid.
- To specify multiple e-mail addresses in a single row, delimit the e-mail addresses with a comma.
- To specify multiple publication channels, delimit the channel names with a comma.

If you specified a group break for the report, then the data set will be grouped according to the level that is associated with the group break. In the example that is

shown in “Understanding How Recipient Lists Enable Report Distribution” on page 158, you would specify the e-mail address and/or the publication channel for a recipient in each Company.

You can later return to the recipient data set and make additional changes. Note, however, that on UNIX, the data set assumes its permissions from the administrator who created the data set. Often, the administrator’s permission mask is set to restrict "group" and "other" from write permission. If someone other than the file’s creator wants to edit the data set, then that person might not have the required permissions. The person can either use the same account name as the data set creator, the creator can change the mask, or the creator can change the permissions that are assigned to the data set.

As an alternative to creating the recipient list in SAS Web Report Studio, you can create a recipient table in Base SAS, for example by using a 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. The next section illustrates a sample program.

### 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.

The example uses a library that is named OMDData, a SAS table named Burst, and a group break variable named Year. You will need to change these values as applicable for your environment. For the library name, use the value that is found in the output-generation configuration file (**OutputManagementConfigTemplate.xml**) that you created when you configured the scheduling of reports, as instructed in “Main Administrative Tasks for Scheduling and Distributing Reports” on page 156.

```
libname OMDData ' \\server\c$\DataSources\SAS\OMData ';
proc sql;
create table OMDData.Burst (Year num, EMAIL char(256), CHANNEL char(256));
insert into OMDData.Burst
values (2000, 'email1@abc.com', 'channelname')
values (2001, 'email2@abc.com', 'channelname')
values (2002, 'email3@abc.com', 'channelname')
;
quit;
```

After you have created the SAS table for your environment, you must import the table into metadata.

Here is a summary of the steps to import the table into metadata:

- 1 In SAS Management Console, expand the Data Library Manager until you reach the library that you specified in the LIBNAME instruction (in this example, OMDData).
- 2 Right-click the library name and choose **Import Tables**.
- 3 Follow the prompts to provide information about the table that you created (in this example, Burst).

For complete instructions on importing a table, see the Help for the Data Library Manager.

## 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 could have been created with multiple group break levels, as seen here:

**Select recipient list:**

SalaryByCompany  
SalaryByCompanyDept

**SalaryByCompanyDept:** Email and channels must be edited by using SAS.

Company	Department	EMAIL	CHANNEL
Board of Directors	Executives		
Board of Directors	Group Financials		
Board of Directors	Secretary of the Board		
Board of Directors	Strategy		
Concession	Concession Management		
Logistics	Logistics Management		
Logistics	Stock & Shipping		

**Divide report based on:**

Section: Chart

Group break: Department

**Assign group break categories to SalaryByCompanyDept column names:**

Group break categories:      Column names:

Company:                      Company

Department:                Company

OK    Cancel    Help

When you open the table in SAS, you see something like this:

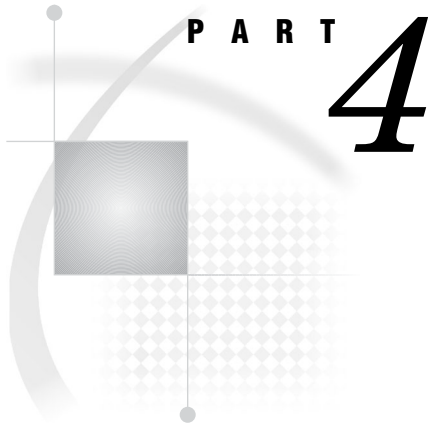
	Company	Department	EMAIL	CHANNEL
8	Marketing	Marketing		
9	Orion Australia	Administration		
10	Orion Australia	Engineering		
11	Orion Australia	Sales		
12	Orion Australia	Sales Management		
13	Orion Belgium	Administration		
14	Orion Belgium	Engineering		
15	Orion Belgium	Sales		
16	Orion Belgium	Sales Management		
17	Orion Denmark	Administration		
18	Orion Denmark	Engineering		
19	Orion Denmark	Sales		
20	Orion Denmark	Sales Management		
21	Orion France	Administration		
22	Orion France	Engineering		

In this example, the recipient list (SalaryByCompanyDept) can still be used for the report that is grouped by Company. In addition, the list can be used for any report that is grouped by Company and then by Department. In this example, you would specify the e-mail address and/or the publication channel for a recipient in each department.

- 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 will 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 doesn't validate the relationship between group break columns in a recipient list and group breaks in the corresponding report.
- 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.

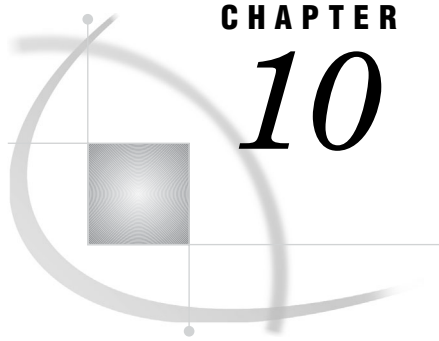


## **SAS Web OLAP Viewer Administration**

- Chapter 10* . . . . . **Introduction to SAS Web OLAP Viewer for Java Administration** 167
- Chapter 11* . . . . . **Configuring SAS Web OLAP Viewer for Java** 169
- Chapter 12* . . . . . **Customizing SAS Web OLAP Viewer for Java** 173







## CHAPTER

## 10

## Introduction to SAS Web OLAP Viewer for Java Administration

<i>Introduction to SAS Web OLAP Viewer for Java</i>	167
<i>Prerequisites for Administering SAS Web OLAP Viewer for Java</i>	167
<i>Main Tasks for Administering SAS Web OLAP Viewer for Java</i>	168
<i>Additional Documentation for SAS Web OLAP Viewer for Java</i>	168

### 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. You can configure SAS Web OLAP Viewer for Java to support single sign-on with the portal.

### Prerequisites for Administering SAS Web OLAP Viewer for Java

This documentation assumes that you have successfully installed and configured SAS Web OLAP Viewer for Java. There are two methods by which you might have performed your installation:

- A planned installation (personal or advanced) uses information from a planning document as input to the SAS Software Navigator and the SAS Configuration Wizard. For this type 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 Configuration Wizard. In addition, you can verify that SAS Web OLAP Viewer for Java has been deployed properly by reading the deployment instructions that are provided in the **config.pdf** file. The **config.pdf** file resides in your installation directory.
- An installation using the Software Index is performed without the use of plans and SAS project directories. For this type of installation, you should follow all of the post-installation configuration and deployment steps that are provided in the **config.pdf** file.

If you have OLAP information maps that were created with SAS Web OLAP Viewer for Java 1.2 or SAS Information Map Studio 1.0.1, then those information maps must be updated to the new SAS Information Map Studio 3.1 format. For instructions, see

“Upgrade Information Maps to the SAS Information Map Studio 3.1 Format” on page 169.

For a comprehensive overview of installation, see the *SAS Intelligence Platform: Installation and Configuration Guide*. For instructions on using the SAS Web OLAP Viewer for Java interface, see the product’s online Help.

---

## 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 169.
- Upgrade information maps, if necessary.
 

If you have OLAP information maps that were created with SAS Web OLAP Viewer for Java 1.2 or SAS Information Map Studio 1.0.1, then those information maps must be updated to the new SAS Information Map Studio 3.1 format. For details, see “Upgrade Information Maps to the SAS Information Map Studio 3.1 Format” on page 169.
- 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 170.
- 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 4, “Configuring the ESRI Map Component,” on page 369.
- Improve the performance of SAS Web OLAP Viewer for Java.
 

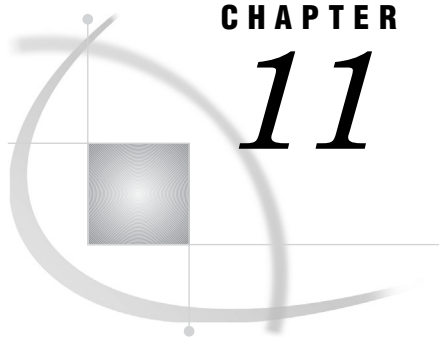
Manage static content and take advantage of clustering capabilities. For details, see “Improving the Performance of SAS Web OLAP Viewer for Java” on page 171.
- Customize the display.
 

You can customize the default display for the viewer, specify a default data source, and perform other customization tasks. For details, see Chapter 12, “Customizing SAS Web OLAP Viewer for Java,” on page 173.

---

## 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.
- The **config.pdf** file, located in the SAS Web OLAP Viewer for Java installation directory, contains configuration and deployment information.
- Chapter 4, “Best Practices for Configuring Your Middle Tier,” on page 57 contains information that is associated with middle-tier administration.
- Chapter 3, “Setting Up and Managing Middle-Tier Security,” on page 19 contains information about authentication, single sign-on, Secure Sockets Layer, and other security related administration.



## CHAPTER

## 11

## Configuring SAS Web OLAP Viewer for Java

<i>Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java</i>	169
<i>Upgrade Information Maps to the SAS Information Map Studio 3.1 Format</i>	169
<i>Configure Logging for SAS Web OLAP Viewer for Java</i>	170
<i>Improving the Performance of SAS Web OLAP Viewer for Java</i>	171
<i>Re-Create and Redeploy SAS Web OLAP Viewer for Java</i>	171

### 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.
- The user of SAS Web OLAP Viewer for Java must have ReadMetadata and Read permission for the information map.

*Note:* Beginning with Service Pack 4, end 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. △

- If you upgraded from a version earlier than SAS Web OLAP Viewer for Java 3.1, then you might need to upgrade your existing information maps. For details, see “Upgrade Information Maps to the SAS Information Map Studio 3.1 Format” on page 169.

### Upgrade Information Maps to the SAS Information Map Studio 3.1 Format

If you have OLAP information maps that were created with SAS Web OLAP Viewer for Java 1.2 or SAS Information Map Studio 1.0.1, then those information maps must be updated to the new SAS Information Map Studio 3.1 format.

Updates can occur in two ways:

- If the source cube is more recent than the existing information map, then SAS Web OLAP Viewer for Java automatically generates an updated map from the source cube.
- If the source cube is as old as the existing information map, then you must manually update the information map.

To update one or more information maps manually, follow these steps:

- 1 Open SAS Information Map Studio 3.1 and connect to the server and repository that contains the information maps that you want to modify.
- 2 From the menu bar, select **Tools ► Administrative Tools**.
- 3 Select the **Metadata Format** tab, and browse to the metadata repository folder that contains the information maps that require updates.
- 4 Click **Run**. The tool will be run on all of the information maps that you have permission to access in the specified folder and its subfolders.

---

## Configure Logging for SAS Web OLAP Viewer for Java

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.

To configure logging, perform these steps:

- 1 In SAS Management Console, navigate to **Foundation Services Manager ► SAS Web OLAP Viewer Local Services ► BIP Core Services**.
- 2 Select **Platform Logging Service**.
- 3 From the menu bar, select **File ► Properties**.
- 4 In the **Service Configuration** tab, click **Edit Configuration**. The Logging Service Configuration dialog box is displayed. There are three possible predefined logging contexts:
  - The **RootLoggingContext** context processes all logging messages for a selected priority.
  - The **com.sas.services** context processes service-related logging messages.
  - The **com.sas.webapp.dataexplorer** context processes all SAS Web OLAP Viewer for Java specific messages.

The Outputs list boxes show the available and the selected outputs for the logging contexts. By default, the output for all contexts is to the console. A log file context is also available, but is not selected by default.

- 5 To edit a context, select the context, click **Edit**, and make your changes. For example, you can do the following:
  - Select a different priority for the context
  - Enable log file output for the context by moving the appropriate output from the **Available** to the **Selected** column.
- 6 To change an output, in the **Outputs** tab select the output that you want to change, and then click **Edit**. For file outputs, you can specify the path to the logging file. You can also control the format of your logging. For more information, click **Help**.

- 7 You can also modify the contexts by assigning new priority levels, adding new outputs, or changing other properties. For more information, click **Help** in the properties dialog box.

---

## Improving the Performance of SAS Web OLAP Viewer for Java

There are a few administration tasks that you can perform to improve overall performance. Most administration tasks are performed in conjunction with other SAS Web applications, such as SAS Information Delivery Portal.

To optimize the performance of SAS Web OLAP Viewer for Java, you can do the following:

- Configure the application's static content to be served from an HTTP server, such as Apache HTTP Server. This will reduce the number of requests made to the J2EE application server. For more information, see “Configuring an HTTP Server to Serve Static Content for SAS Web Applications” on page 86.
- Deploy the application as part of a cluster configuration. See “Configuring a Cluster of J2EE Application Servers” on page 84.

For more information about middle-tier deployment configurations and performance enhancements, see Chapter 4, “Best Practices for Configuring Your Middle Tier,” on page 57.

---

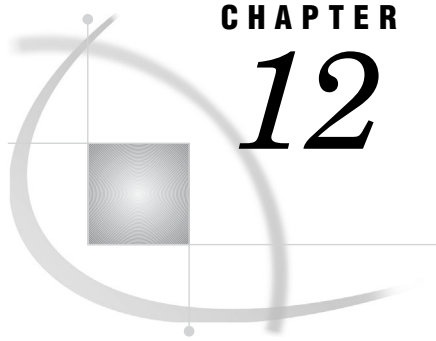
## Re-Create and Redeploy SAS Web OLAP Viewer for Java

After initial installation and configuration, if you make changes to your configuration, then you must re-create and redeploy SAS Web OLAP Viewer for Java. For example, changes that you make to `WebOLAPViewerConfig.xml` require that you redeploy SAS Web OLAP Viewer for Java.

To re-create and redeploy SAS Web OLAP Viewer for Java, follow these steps:

- 1 Create a new `SASWebOLAPViewer.war` file by running `configure.sh` (UNIX) or `configure.bat` (Windows). These scripts are located in the SAS Web OLAP Viewer for Java installation directory.
- 2 Deploy the updated WAR file into your J2EE application server or servlet container. Follow the deployment instructions for your platform that are described in the `config.pdf` file. This file is located in `SAS-install-dir\SAS\SASWebOlapViewerforJava\3.1`.





## CHAPTER

## 12

## Customizing SAS Web OLAP Viewer for Java

<i>Main Steps for Customizing SAS Web OLAP Viewer for Java</i>	173
<i>Changes That Can Be Made to WebOLAPViewerConfig.xml</i>	174
<i>Specify a Default Data Source</i>	174
<i>Specify an Information Map</i>	174
<i>Specify a Data Exploration</i>	174
<i>Specify Common Public Data Explorations</i>	175
<i>Customize the Default Display for Viewers</i>	175
<i>Specify the Column Layout</i>	176
<i>Specify the Default Panel</i>	177
<i>Customize the Header and Footer Styles</i>	177
<i>Customize the Main Header</i>	177
<i>Specify a Main Footer</i>	178
<i>Specify an Alternate Header and Footer for Exporting and for Printing</i>	178
<i>Change the Personal Data Explorations Folder</i>	179
<i>Customize the Open Dialog Box</i>	179
<i>Specify a Default Initial Path</i>	179
<i>Customize Available File Types</i>	180
<i>Enable the Theme That Is Used for the Current Portal User</i>	180
<i>Disable the Logoff Link</i>	180

### Main Steps for 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. To customize SAS Web OLAP Viewer for Java, complete the following steps:

- 1 Modify the **WebOLAPViewerConfig.xml** file, which is located in the **SAS-install-dir\SASWebOlapViewerforJava\3.1\SASWebOLAPViewer\WEB-INF** directory. Instructions for customizing that file are presented here, and can also be found as comments within the file.
- 2 After you modify the **WebOLAPViewerConfig.xml** file, your changes won't take effect until you re-create and redeploy SAS Web OLAP Viewer for Java. For instructions, see "Re-Create and Redeploy SAS Web OLAP Viewer for Java" on page 171.

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 will be 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://repository-name/path-to-map
```

You can optionally 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://Foundation/BIP Tree/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 will be displayed in the user interface. When you change the value to **true**, a default query will not be generated for the information map. End users will 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 will be 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://repository-name/path-to-exploration
```

You can optionally 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://Foundation/BIP Tree/Users/saswbadm/SampleDE"
  activeBookmarkName="SampleBookmark">
</DataExplorations>
```

## Specify Common Public Data Explorations

To specify data explorations that will be available to all users in the Bookmarks panel, modify the **<DataExploration>** element within the **<DataExplorations>** element.

Use the following format to specify a fully-qualified path to a data exploration:

```
SBIP://repository-name/path-to-exploration
```

For example, the following code specifies two data explorations:

```
<DataExplorations>
  <DataExploration pathURL=
    "SBIP://Foundation/BIP Tree/Users/saswbadm/SampleDE1">
  <DataExploration pathURL=
    "SBIP://Foundation/BIP Tree/Users/saswbadm/SampleDE2">
</DataExplorations>
```

In this example, all users will be able to access SampleDE1 and SampleDE2 from the Public Data Explorations group in the Bookmarks panel.

## Customize the Default Display for Viewers

The **<Viewers>** element determines which viewers will be 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 will 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:

**Table 12.1** Viewer Elements

Element	Viewer
<Table>	table viewer
<BarChart>	bar chart viewer
<ColorMappedTable>	color-mapped table viewer
<PieChart>	pie chart viewer
<ScatterPlot>	scatter plot viewer

Element	Viewer
<LineChart>	line chart viewer
<BarLineChart>	barline 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 12.2** 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 12.3** `<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 will result 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 12.4** `<Header>` Elements

Element	Description
<code>&lt;BackgroundImage&gt;</code>	The background image for the header. The default image is a blue SAS background.
<code>&lt;Image&gt;</code>	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.
<code>&lt;Text&gt;</code>	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.”

Element	Description
<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 <b>useDynamicText</b> attribute specifies whether the value of <b>SecondaryText</b> is generated from the name of the data source that you are viewing.
<TemplateFileName>	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 <code>VisualDataExplorerHeader.html</code> .

*Note:* Images are stored in the **images** subdirectory under the SAS Web OLAP Viewer for Java deployment in the servlet container.  $\Delta$

## 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 12.5** <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.
<SecondaryText>	Secondary text that follows the value of <Text>. The <b>useDynamicText</b> 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 templates 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 12.6** 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.

Element	Description
<Text>	The main text for the alternate header or footer.
<SecondaryText>	Secondary text for the alternate header or footer. The <b>useDynamicText</b> attribute is only available in the <b>&lt;AlternateHeader&gt;</b> element, and it specifies whether the value of <b>&lt;SecondaryText&gt;</b> is generated from the name of the data source you are viewing. The <b>newLine</b> attribute specifies whether the value of <b>&lt;SecondaryText&gt;</b> is displayed on a new line beneath the value of <b>&lt;Text&gt;</b> .

For the previous elements, you can specify the following attributes:

**Table 12.7** Attributes for Alternate Header and Footer Elements

Attribute	Description
hAlignment	Specifies the horizontal alignment for the item. Specify either left, center, or right.
vAlignment	Specifies the vertical alignment for the item. Specify either top, middle, or bottom.
column	Specifies which column the item is placed in.
row	Specifies which row the item is placed in.

## Change the Personal Data Explorations Folder

To change the root folder in metadata for personal data explorations, edit the **<PersonalDataExplorationFolder>** element in the **WebOLAPViewerConfig.xml** file. The root folder that you specify will contain subdirectories for each user, in the following format: /user-name/Data Explorations.

Here is the default value:

```
SBIP://repository-name/BIP Tree/Users
```

## 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://repository-name/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://Foundation/BIP Tree/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.  $\Delta$

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>

```

---

## Enable the Theme That Is Used for the Current Portal User

A theme is a collection of specifications (for example, colors, fonts, and font styles) and graphics that control the appearance of an application. By default, SAS Web OLAP Viewer for Java uses the default theme that is used by SAS Information Delivery Portal. You can configure SAS Web OLAP Viewer for Java to use theme resources that are in effect for the particular user who is logged on.

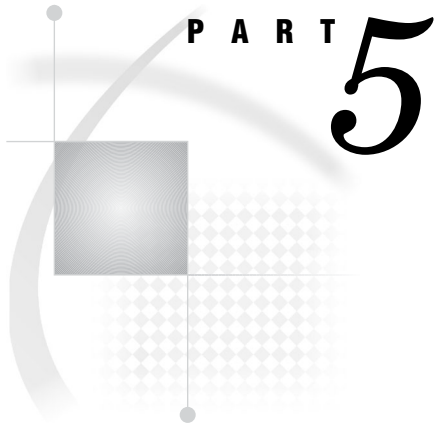
To enable the theme for the current user's profile, in the **WebOLAPViewerConfig.xml** file, change the **<SASThemes enabled="false"/>** element to **<SASThemes enabled="true"/>**.

---

## 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 could 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"/>**.

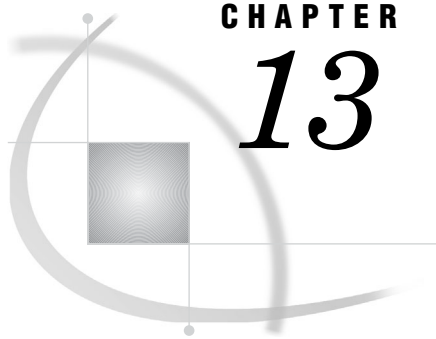


## **Portal Web Application Administration**

- Chapter 13*..... **Overview of the Portal Web Application** 183
- Chapter 14*..... **Introduction to Portal Administration** 191
- Chapter 15*..... **Using the Portal Administration Tools** 209
- Chapter 16*..... **Administering Portal Authorization** 219
- Chapter 17*..... **Adding Content to the Portal** 237
- Chapter 18*..... **Administering SAS Business Intelligence Dashboard** 303
- Chapter 19*..... **Customizing the Portal's Display** 317
- Chapter 20*..... **Foundation Services and WebDAV Server Deployment** 335
- Chapter 21*..... **Redistributing Portal Web Applications and Servers** 347







## CHAPTER

## 13

## Overview of the Portal Web Application

<i>Introduction to the Portal Web Application</i>	183
<i>Understanding the SAS Web Infrastructure Kit and the SAS Information Delivery Portal</i>	184
<i>Comparison of the SAS Web Infrastructure Kit and the SAS Information Delivery Portal</i>	184
<i>Features of the SAS Web Infrastructure Kit</i>	185
<i>Features of the SAS Information Delivery Portal</i>	185
<i>Summary of Portal Features and Their Software Requirements</i>	186
<i>Understanding the Portal Components</i>	188

### Introduction to the Portal Web Application

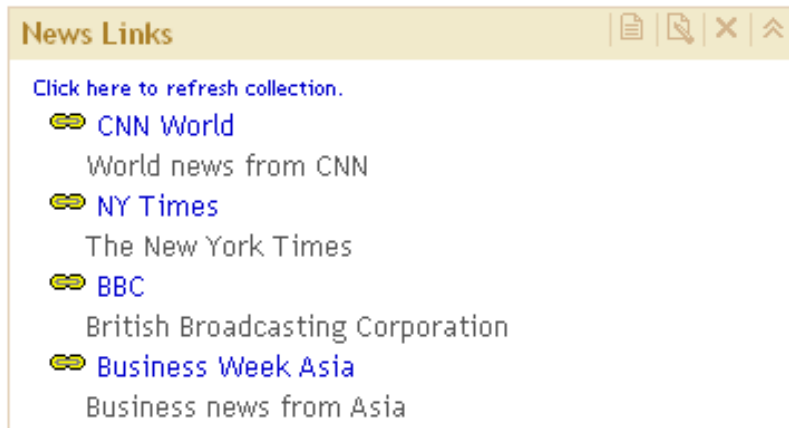
The SAS Web Infrastructure Kit, which is provided with SAS Integration Technologies, serves as the underlying infrastructure for the following:

- SAS Portal Web Application Shell: a Web application that is included in the SAS Web Infrastructure Kit and is used as a portal by other SAS Web applications
- SAS Information Delivery Portal: a separate SAS product that is installed with the SAS Web Infrastructure Kit in order to fully implement the capabilities of the SAS Portal Web Application Shell

*Note:* In this documentation, “portal Web application” is a generic term that refers to either of these portal entities. Even if you don’t have the SAS Information Delivery Portal installed, the same general concepts apply to both. The main distinction between the two is that the SAS Information Delivery Portal has more features. For details, see “Understanding the SAS Web Infrastructure Kit and the SAS Information Delivery Portal” on page 184. △

The portal Web application 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 Web application uses portlets to organize information on Web pages. Here is a sample portlet that contains links to Web sites that provide business or world news:



The portal Web application 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 Web Infrastructure Kit to create custom portlets. The SAS Web Infrastructure Kit also includes a framework that enables users to launch SAS Stored Processes and have the results displayed dynamically in the browser. The kit also provides tools to facilitate secure integration and information sharing with remote applications.

---

## Understanding the SAS Web Infrastructure Kit and the SAS Information Delivery Portal

---

### Comparison of the SAS Web Infrastructure Kit and the SAS Information Delivery Portal

The SAS Web Infrastructure Kit includes the following components:

- a Web application shell that displays content in portlets and pages and that provides log-on and log-off capabilities, metadata searching, bookmarking, and content administration features
- a SAS Stored Process application that enables stored processes to run from the Web
- predefined portlets for content viewing and navigation
- administrative tools for deploying services, portlets, and additional Web applications
- SAS Java components, Web infrastructure components, and a services infrastructure

If the SAS Information Delivery Portal is installed, then the following additional capabilities are provided:

- 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 WebDAV (Web-Based Distributed Authoring and Versioning)
- 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 Web Infrastructure Kit

The SAS Web Infrastructure Kit provides the following features and capabilities:

- Developers can create custom portlets using a framework that conforms to industry-standard Model-Viewer-Controller (Model 2) design patterns. Portlet deployment is simplified through the use of portlet deployment descriptors and portlet archive (PAR) files. These features enable new portlets to be deployed without the need to restart the Web server. A set of action and initializer classes is provided to reduce the need for developing custom programs. Access to SAS custom tags and to tags in the Struts development framework simplifies the development and localization of JavaServer Page (JSP) pages.
- Developers can integrate other applications with the portal Web application shell. An application can be integrated in either of the following ways:
  - as a remote portlet, which is a portlet that calls a remote Web application. The Web application returns an HTML fragment to be displayed within the portlet's borders.
  - as a stand-alone application that is invoked from the portal Web application shell but is executed remotely and displayed in a separate browser window.

The SAS Web Infrastructure Kit provides tools to facilitate secure information sharing between the portal Web application shell and the remote application. One type of information sharing is the single sign-on feature, which enables applications to be invoked from the portal without requiring the user to re-enter a user name and password. Other information related to a portal session can be shared as well.

For information about developing applications that share services, see the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/library/toc\\_portaldev.html](http://support.sas.com/rnd/itech/library/toc_portaldev.html).

- Users can launch SAS Stored Processes and have the results displayed dynamically in the browser. For information about how to execute stored processes, click the Help link in the banner of the portal's user interface.
- All viewing of content and launching of applications are subject to multilayered security features, which ensure that each user can access only the content that the user is authorized to access.
- Developers or administrators can create custom themes that specify the text attributes, backgrounds, logos, and other graphical elements to be incorporated into the user interface. For information about creating themes, see the *SAS Web Infrastructure Kit Developer's Guide*.
- The search tool enables users to perform keyword searches to locate content that they are authorized to access. Users can view the located content items and bookmark them to see later. For more information about the search tool, click the Help link in the banner of the portal's user interface.
- Users can generate e-mails that contain password-protected links to portal content, so that they can share information with other users in a secure environment. For more information about e-mailing, click the Help link in the banner of the portal's user interface.

---

## Features of the SAS Information Delivery Portal

In addition to the features listed previously, 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 one, two, or three columns on a page.
  - 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. If the Xythos WebFile Server (WFS) WebDAV server is installed, then 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.

- If Xythos WebFile Server (WFS) is installed, then 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. If Xythos WFS has been installed, then users can also publish portal content to a WebDAV repository and can 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 Web application provides support for the RSS (Rich Site Summary) 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 Viewer and SAS Information Map Studio, then users can access SAS information maps and SAS reports using the portal.

For information about using the report and information map features, click the Help link in the banner of the user interface.

---

## Summary of Portal Features and Their Software Requirements

You might have the SAS Information Delivery Portal installed along with the SAS Web Infrastructure Kit, or you might have only the SAS Web Infrastructure Kit. The main distinction between the two is that the SAS Information Delivery Portal has more features. Additionally, Xythos WFS provides its own set of features to your deployment.

Based on the software that your organization has installed, you can use the following table to determine the availability of features:

**Table 13.1** Features Based on the Software That Is Installed

Feature	SAS Web Infrastructure Kit		SAS Information Delivery Portal	
	With Xythos WFS WebDAV Server	Other WebDAV Server, or No WebDAV Server	With Xythos WFS WebDAV Server	Other WebDAV Server, or No WebDAV Server
Launch applications	Yes	Yes	Yes	Yes
View links	Yes	Yes	Yes	Yes
Use personalization features	Yes (content administrators only)	Yes (content administrators only)	Yes (all users)	Yes (all users)
Execute dynamic SAS Stored Processes	Yes	Yes	Yes	Yes
Execute SAS Stored Processes running in the background and view results stored in WebDAV	Yes	No	Yes	No
View files stored in WebDAV	Yes	No	Yes	No
Manage subscriptions to publication channels	No	No	Yes	Yes
Publish portal content to publication channels	No	No	Yes	Yes
Publish portal content to WebDAV	No	No	Yes	No
View published packages stored on a server	No	No	Yes	Yes
View published packages stored in WebDAV	No	No	Yes	No
View syndication channel content	No	No	Yes	Yes

Feature	SAS Web Infrastructure Kit		SAS Information Delivery Portal	
	With Xythos WFS WebDAV Server	Other WebDAV Server, or No WebDAV Server	With Xythos WFS WebDAV Server	Other WebDAV Server, or No WebDAV Server
View SAS reports	No	No	Yes	Yes
SAS Web Report Viewer must be installed.				
View SAS information maps	No	No	Yes	Yes
SAS Information Map Studio must be installed to create and edit information maps.				

## Understanding the Portal Components

The portal Web application runs in a servlet container or J2EE application server, and requires a Java 2 Software Development kit (SDK). The portal Web application also uses the SAS Foundation Services for both local and remote service functionality. Finally, the portal Web application 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 portal Web application:

- the Portal Web Application Shell, or the SAS Information Delivery Portal, which includes the following:
  - Portal Web Application Java Classes: The foundation of the portal Web application consists of Java classes contained in the Portal API. Refer to the Portal API class documentation for complete documentation of the Java classes included in these SDKs. If you want, you can use these classes to develop your own custom portlets and custom applications for deployment in the portal Web application. For details, see “Using the Portlet API” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/portlets/dg\\_api\\_portal.html](http://support.sas.com/rnd/itech/doc9/portal_dev/portlets/dg_api_portal.html).
  - Portal Web Application Java Servlets, JSPs, and JavaBeans: The portal Web application servlets, JSPs, and JavaBeans are the active components of the portal Web application. Using the portal Web application Java classes, these servlets, JSPs, and JavaBeans interact with the metadata server, SAS Stored Process Server, and the SAS Workspace Server to deliver portal Web application functionality and content to users.
  - Custom Themes: Themes control the appearance of the portal Web application and of SAS solutions that run in the Portal. A theme consists of cascading style sheets (CSSs) and graphical elements, including the portal Web application’s banner, background image, and logo. To create your own custom themes, see the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/themes/index.html](http://support.sas.com/rnd/itech/doc9/portal_dev/themes/index.html).
  - Property Files: The `install.properties` file contains parameters that control the operation of the portal Web application. The `install.properties` file

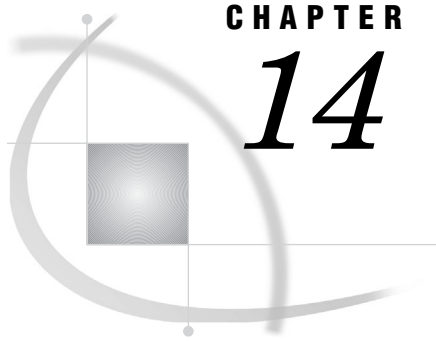
includes your installation directory locations, default locale setting, user ID and authentication domain information, and information about your SAS Metadata Server, Java RMI Server (for the SAS Services application), and WebDAV server.

- Package Viewer: The Package Viewer enables users to display packages in the portal Web application.
- Visual Data Explorer: The Visual Data Explorer, which is available if you have installed the SAS Information Delivery Portal, enables users to display SAS Information Maps in the portal Web application.
- SAS Stored Process Server Web application: The Stored Process Server Web application enables users to run stored processes. The Stored Process Web application can be run standalone or through the portal Web application. 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 Web application and other Web applications.
- SAS Preferences Web application: The SAS Preferences Web application manages user preferences for the portal Web application and for SAS solutions that are delivered through the portal Web application. The SAS Preferences Web application is run through the portal Web application.
- SAS Themes Web application: The SAS Themes Web application contains definitions for themes that are used by the portal Web application and by SAS solutions that are delivered through the portal Web application.
- 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 Report Viewer (optional): The SAS Web Report Viewer enables users to display SAS Reports in the portal Web application.
- SAS Web OLAP Viewer for Java (Optional): The SAS Web OLAP Viewer for Java enables users to display OLAP data in the portal Web application.
- Xythos WFS content services: The Web server also manages content that is accessible to HTTP clients. This content may be accessible through Uniform Resource Locators (URLs), or it may be accessible only through Web applications. Web Distributed Authoring And Versioning (DAV) 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 Web application requires Xythos WFS 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 Web application 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 Web application's content, metadata, and security services.

For details about developing portlets, see the *SAS Web Infrastructure Kit: Developer's Guide*.

- 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 Web application, the portal Web application passes the application the session and application context which can then be used to obtain the authenticated user (and allow single signon). For information about developing applications, see the *SAS Web Infrastructure Kit: Developer's Guide*.
- Other SAS Solutions Web Applications: The middle tier might also manage other Web applications, such as solutions or Web applications that are built using the servlet container services.





## CHAPTER

## 14

## Introduction to Portal Administration

<i>Prerequisites for Administering the Portal Web Application</i>	191
<i>What You Should Know</i>	191
<i>What You Should Do</i>	192
<i>Who Can Administer the Portal Web Application</i>	193
<i>Main Tasks for Administering the Portal Web Application</i>	196
<i>Provide Portal Content</i>	196
<i>Implement Security for the Portal</i>	196
<i>Set Up Portal Views</i>	197
<i>Set Up the Public Kiosk</i>	197
<i>Customize the Portal's Appearance</i>	197
<i>Perform Routine Maintenance</i>	198
<i>Reconfigure or Redistribute the Portal Web Application</i>	198
<i>Suggestions for Verifying Portal Operation</i>	198
<i>Important Portal Administrative Files</i>	199
<i>Loading Initial Metadata</i>	200
<i>Administering the Public Kiosk</i>	202
<i>Overview of the Public Kiosk</i>	202
<i>Create or Edit the Public Kiosk</i>	203
<i>Remove the Public Kiosk</i>	203
<i>Modifying the Logging Output Information and Location</i>	204
<i>Overview of Log Configuration Files</i>	204
<i>Change the Types of Messages That Are Stored in the Log</i>	205
<i>Change the Log Type, File Name, or Location</i>	205
<i>Change the Log Message Format</i>	206
<i>Example: Customize the Log File to Track User Logons</i>	206
<i>Example: Customize the Log File to Track Portal Content Usage</i>	207
<i>Get Additional Information</i>	207
<i>Additional Documentation for the Portal</i>	207

## Prerequisites for Administering the Portal Web Application

### What You Should Know

In order to administer the portal Web application, you should familiarize yourself with the following:

- The concepts that are listed in “Prerequisites for Administering the Web Applications” on page 4.
- Distinctions between the SAS Web Infrastructure Kit and the SAS Information Delivery Portal, and which features are available depending on the software that

you have installed. For full details, see “Understanding the SAS Web Infrastructure Kit and the SAS Information Delivery Portal” on page 184. For a table that summarizes the distinctions, see “Summary of Portal Features and Their Software Requirements” on page 186 .

- Which SAS users have permissions to administer the portal Web application, and what additional users you should define for administration purposes. See “Who Can Administer the Portal Web Application” on page 193.
- How to use the portal Web application, including information about the following:
  - How to start the servers and log on. See “Starting the Web Applications” on page 13.
  - What the portal Web application enables users to accomplish, and the main tasks that users can perform in the portal. See *Introduction to the SAS Information Delivery Portal 2.0* at <http://support.sas.com/rnd/web/portal/doc2/tour/index.html> for a general introduction and tour of the SAS Information Delivery Portal. (Even if you don’t have the SAS Information Delivery Portal installed, the same general concepts apply for the portal Web application that’s included in the SAS Web Infrastructure Kit.)
  - How to create a page in the portal Web application, 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, refer to the online Help that is provided with the portal Web application.
- The software components that are required for portal operation. For a description of these components, see “Understanding the Portal Components” on page 188.
- Location of important portal files. See “Important Portal Administrative Files” on page 199.
- The main tasks for administering the portal. See “Main Tasks for Administering the Portal Web Application” on page 196.

---

## What You Should Do

Before you can administer the portal Web application, the portal Web application must be functional. This means that you have done the following:

- 1 Installed and configured the server-side software that is required for portal operation. In summary, the portal Web application:
  - relies on the SAS metadata repository, and requires that the SAS System, SAS Management Console, and SAS Foundation Services be installed and configured
  - uses SAS application and data servers (SAS Workspace Server, SAS Stored Process Server, SAS OLAP Server) to render SAS data and run SAS programs
- 2 Installed and configured the portal Web application and the third-party software that it requires. (The portal Web Application Shell is included in the SAS Web Infrastructure Kit. You can also optionally install the SAS Information Delivery Portal.) In summary, the portal Web application:
  - runs in a servlet container or J2EE application server, and requires the Java 2 Software Development Kit (SDK)
  - (Optional) uses the Xythos WFS WebDAV server for read-write HTTP functionality

A planned installation uses information from a planning document as input to the SAS Software Navigator and the SAS Configuration Wizard. If you did not perform a planned installation (that is, you performed an Index installation), then you should follow all of the pre-installation, installation, and post-installation

steps that are provided in the `wik_readme.html` file that is included with the portal software.

- 3 Defined the required SAS users on the host and in the SAS Metadata Server. For a summary of these users, see Appendix 1, “Summary of the Required SAS Users and Groups,” on page 359. In addition, you might have created additional credentials to access SAS application and data servers (SAS Workspace Server, SAS Stored Process Server, SAS OLAP Server).
- 4 Optionally, you might have loaded initial portal metadata. If you haven’t already loaded the initial portal metadata, then you can do so before you start administering the portal. For more information, see “Loading Initial Metadata” on page 200.
- 5 Verified that your portal Web application operates correctly. See “Suggestions for Verifying Portal Operation” on page 198.

---

## Who Can Administer the Portal Web Application

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 would log on to the portal Web application 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 233. △

**Table 14.1** Users Who Administer the Portal Web Application

<b>User Name or Role</b>	<b>Default Metadata User ID</b>	<b>What the User Administers</b>	<b>Required Metadata Permissions</b>	<b>What the Permissions Allow</b>
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 Web application. This account should be used only for administering metadata in SAS Management Console.	All permissions	Unrestricted user
SAS Web Administrator	saswbadm any user who is a member of the Portal Admins group (members must be users, not groups)	Administers all aspects of the portal Web application.  In order to edit content that some other user created, this account might need to search for the content first in the portal Web application.	(All portal content) - ReadMetadata - WriteMetadata - Read - Delete	Create and share portal content  View, edit, share, unshare, and delete all content, including content that others create.
SAS Guest	sasguest	Administers the portal's Public Kiosk.  If you did not install the SAS Information Delivery Portal, then to enable the SAS Guest to administer the Public Kiosk, you must configure the SAS Guest as a group content administrator for the PUBLIC group.	(Public Kiosk content) - ReadMetadata - WriteMetadata - Read	Create, view, edit, and delete Public Kiosk content.

User Name or Role	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 224.	(Group and personal content) - ReadMetadata - WriteMetadata - Read	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 <b>Options</b> menu (the SAS Information Delivery Portal must be installed).  You must manually define portal users. See “Planning User Accounts and Their Organization into Groups” on page 21.	(Personal content only) - ReadMetadata - WriteMetadata - Read	View Public Kiosk content.  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 if the SAS Information Delivery Portal is installed.

By default, when you first install the portal Web application, all users who can access the metadata server are members of the PUBLIC group and have administrator permissions. The above 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.

For general information about permissions, see “Understanding Authorization” in the *SAS Intelligence Platform: Security Administration Guide*.

---

## Main Tasks for Administering the Portal Web Application

---

### Provide Portal Content

Determine the types of content that you want to provide, and then add content items to the portal Web application 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 17, “Adding Content to the Portal,” on page 237.

Developers in your organization will create many of these content items. In addition, your organization can develop custom portlets and themes for the portal Web application. For more information, see the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/library/toc\\_portaldev.html](http://support.sas.com/rnd/itech/library/toc_portaldev.html)

---

### Implement Security for the Portal

For general security tasks, see “Planning Your Middle-Tier Security Implementation” on page 20. The following security tasks apply specifically to the portal Web application:

- Set up users for the portal Web application.
  - Enable users to log on to the portal by creating metadata identities for the users, as explained in “Planning User Accounts and Their Organization into Groups” on page 21. See also “Planning for Portal Users and Groups” on page 220.
  - 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 224.
- 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 Web application 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 222.
    - Set up Java security for your Web applications. If you deploy Web applications that run from the portal, then you should configure permissions for those applications. For details, see “Adding Permissions to Policy Files” on page 45.

- Set up trusted Web authentication.

Optionally, you can configure the portal to use trusted Web authentication. For instructions, see “Changing to Trusted Web Authentication” on page 32.

## Set Up Portal Views

The portal Web application 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 Web application, 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 Web application, 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 could 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 224.

## Set Up the Public Kiosk

The Public Kiosk can be used to display pages and portlets that you want all users to be able to view, even those users who don't log on to the portal Web application. If you configure Web authentication for the portal, then users will directly access the portal Web application's logon page instead of the Public Kiosk.

The SAS Guest user is the administrator of the Public Kiosk. For more information, see “Administering the Public Kiosk” on page 202.

## Customize the Portal's Appearance

You can make some changes to the portal's appearance that affect all portal views:

- You can set up a default theme. When users log on to the portal Web application, they will see the theme that you specify as default. In addition, you can make new themes available to portal users.
- 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.

The above changes will be seen by all users who log on to the portal.

If the SAS Information Delivery Portal is installed, then 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.

For details about themes and preferences, see Chapter 19, “Customizing the Portal's Display,” on page 317.

---

## Perform Routine Maintenance

Here are some maintenance task that you might need to perform:

- Change passwords for users as needed. For more information, see “User ID and Password Management” in the *SAS Intelligence Platform: Security Administration Guide*.
- Add new users and groups.
- Add new custom-developed portlets, Web applications, themes, and other content.
- Update existing portal pages.
- Delete portal content items from the portal environment. Instructions can be found in the online Help that is provided with the portal Web application.
- Remove old publication channel files from the file system or from a Xythos WFS repository. (The expiration date for a package doesn’t delete files; it only removes them from the channel.)
- If necessary, remove and then reinstall portal metadata to its initial state (the state it was in after you installed the portal Web application). See “Using the SAS Portal Metadata Tool to Remove Portal Metadata” on page 215.

*Note:* You can use the Quiesce portlet to bring down the portal Web application gracefully. When you use the Quiesce portlet, you prepare the portal to be shut down by preventing new users from logging on. For details, see “Using the Quiesce Portlet to Bring Down the Portal” on page 213.  $\triangle$

---

## Reconfigure or Redistribute the Portal Web Application

To reconfigure specific features of your portal Web application, you can rerun the Web Infrastructure Kit installation (Index install) and change your initial parameters. For example, you can reconfigure the following parameters:

- the directory that is used for storing portal configuration files
- the Xythos WFS WebDAV server location or base path
- the locale that is used for the portal Web application

You might need to move portal components to different hosts. For example, if you initially installed the portal Web application 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 that are included in the SAS Web Infrastructure Kit are designed to operate in a tiered environment using various servers, each of which can run on a separate machine. When you are ready to deploy the portal Web application to your production environment, you might want to use a distributed configuration. For more information, see Chapter 21, “Redistributing Portal Web Applications and Servers,” on page 347.

For deployment scenarios that depict best practices, see “Sample Middle-Tier Deployment Scenarios” on page 70.

---

## Suggestions for Verifying Portal Operation

Instructions for installing the portal Web application are provided with the SAS Web Infrastructure Kit and SAS Information Delivery Portal installation wizards.

Here are some suggestions for verifying portal operation after you have completed the installation:



- 1 Start the portal Web application. For instructions, see “Starting the Web Applications” on page 13.
- 2 Log on to the portal Web application as the SAS Demo User (sasdemo) or the SAS Web Administrator (saswbadm). For help logging on, refer to the online Help that is included with the portal Web application. 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. See Appendix 3, “Logon Formats for the Web Applications,” on page 365.
- 3 In the portal, search for and execute a sample stored process (for example, the *Hello World* sample stored process). This verifies that the SAS Stored Process Server Web application functions correctly. The SAS Stored Process Server Web application is one of the Web applications that are included in the SAS Web Infrastructure Kit. You loaded the sample stored processes during installation and configuration. You should be able to see these sample stored processes in SAS Management Console. For more information, see “Adding and Administering SAS Stored Processes” on page 294. For help performing a search, refer to the online Help that is included with the portal Web application.
- 4 In the portal, view the online Help by clicking the **Help** link in the banner pane of the window. This verifies that the SAS Documentation Web application has been successfully installed.
- 5 In the portal, set some preferences (by using the portal’s **Options ► Preferences** menu). This verifies that the SAS Preferences Web application has been successfully installed.

*Note:* The SAS Information Delivery Portal must be installed in order for the SAS Demo User to perform this task. If you installed only the SAS Web Infrastructure Kit, then you must be logged on as the SAS Web Administrator to perform this task. △

- 6 If you are using Xythos WFS, then make sure that you can view the SAS Demo User profile in the Xythos Administrator. By default, the Xythos installation creates a top-level directory called `/sasdav/Users` on the WebDAV server. The SAS Demo User should appear under this top-level directory.
- 7 When you are done testing the portal Web application, log off the portal.

---

## Important Portal Administrative Files

Here are the locations of important portal files:

**Table 14.2** Portal Web Application Files

<b>Files</b>	<b>Location</b>
installation files	<code>SAS-install-dir\Web\Portal2.0.1</code>
configuration files	<code>SAS-config-dir\Lev1\web</code>
	For example, on Windows, these files might be found in <code>C:\SAS\EntBIServer\Lev1\web</code> .

Files	Location
initial configuration instructions	<p><i>SAS-config-dir</i>\<b>instructions.html</b> file (for a planned installation) or  <i>SAS-install-dir</i>\<b>Web</b>\<b>Portal2.0.1</b>\<b>wik_readme.html</b>(for an index installation)</p> <p>These configuration instructions were provided during installation. You might refer to these files to verify parts of your configuration.</p>
<b>configure_wik</b> (BAT or SH)	<p><i>SAS-install-dir</i>\<b>Web</b>\<b>Portal2.0.1</b></p> <p>This utility re-creates the portal Web applications. For more information about this utility , see “Re-Create and Redeploy the Portal Web Application” on page 211.</p>
<b>install.properties</b>	<p><i>SAS-install-dir</i>\<b>Web</b>\<b>Portal2.0.1</b>\<b>PortalConfigure</b></p> <p>This file contains application properties. Changes that you make to this file take effect after you re-create and redeploy the portal Web application.</p>
log file	<p><i>SAS-config-dir</i>\<b>Lev1</b>\<b>web</b>\<b>Deployments</b>\<b>Portal</b>\<b>log</b>\<b>portal.log</b></p> <p>You can change the location of this file. See “Modifying the Logging Output Information and Location” on page 204.</p>

## Loading Initial Metadata

When you installed and configured the portal Web application, you were given the option to load initial portal metadata. The programs that load this initial metadata are **LoadPortalStructure.sas** and **LoadPortalStructure\_utf8.sas** (uses UTF-8 character encoding). Both files are located in the *SAS-install-dir*\**Web**\**Portal2.0.1**\**OMR** directory.

If you did not load the initial metadata during installation, then you can do so afterward. To be successful, however, you must run the program under one of the following conditions:

- Load the metadata before you start the portal Web application for the first time. When you start the portal Web application, some metadata is created for the portal, and you can no longer run the \*.sas files (they will abort if you try to reload existing metadata).
- If you have already started the portal Web application, then you can remove any existing metadata, and then reload the metadata. For information about removing existing metadata, see “Using the SAS Portal Metadata Tool to Remove Portal Metadata” on page 215.

To load the metadata, run the appropriate **LoadPortalStructure\*.sas** file. For instructions, see “Install Metadata” in the **wik\_readme.html** file, which is located in the portal installation directory.

If you want to load the initial demonstration data, and if the SAS Metadata Server runs on a different machine from the one on which you installed the portal Web application, then before you run the \*.sas files, you must ensure that the encodings on the two machines are compatible. The \*.sas files contain localized metadata that is created in the encoding of the machine on which the portal Web application was installed:

- If the localized metadata cannot be represented in the default encoding of the SAS System on the SAS Metadata Server machine, then in most cases you should not transfer these files to that machine and submit them to the SAS System. However, you might be able to use the **-encoding** system option to change the encoding of the SAS Metadata Server machine's SAS System so that it successfully reads the **\*.sas** files.
- If the localized metadata was successfully created in the encoding of the portal Web application's machine, then you might be able to run the **\*.sas** files using the SAS System of the portal Web application's machine in order to load the metadata to the SAS Metadata Server's machine. Before you submit the **\*.sas** programs, use the SAS Program Editor to view the localized metadata and verify that it is correct.

The following table summarizes what happens when you load the initial metadata:

**Table 14.3** Differences Between Loading and Not Loading Initial Metadata

<b>Initial Metadata Is Loaded</b>	<b>Initial Metadata Is Not Loaded</b>
<p><b>LoadPortalStructure*.sas</b> creates metadata for the Public Kiosk. When users access the portal Web application, the first page that they see is the Public Kiosk.</p>	<p>No metadata is created for the Public Kiosk. When users access the portal Web application, the first page that they see is the logon page. You can log on as the SAS Guest user (sasguest) and create content metadata for the Public Kiosk. For more information, see "Administering the Public Kiosk" on page 202.</p>
<p><b>LoadPortalStructure*.sas</b> creates a Home page template. When users log on to the portal Web application, they see a Home page that contains a collection portlet and a Bookmarks portlet. This Home page is based on the template that is created by <b>LoadPortalStructure*.sas</b>.</p> <p>For information about pages, page templates, and the Home page template, see "Understanding Pages and Page Templates" on page 242.</p>	<p>No Home page template is created. When users log on to the portal Web application, they see a portal with no pages. You can create your own page template to use as the Home page.</p>

**Initial Metadata Is Loaded**

**LoadPortalStructure\*.sas** loads sample page templates, pages, portlets, and links. You can search for the sample content and add it to the portal. If the SAS Information Delivery Portal is installed, then users who can log on to the portal can add the sample content to their portal views.

**LoadPortalStructure\*.sas** creates the *Portal Application Tree* in SAS metadata. This tree is the highest level tree in which all the user and group permission trees are created for defining portal content access control.

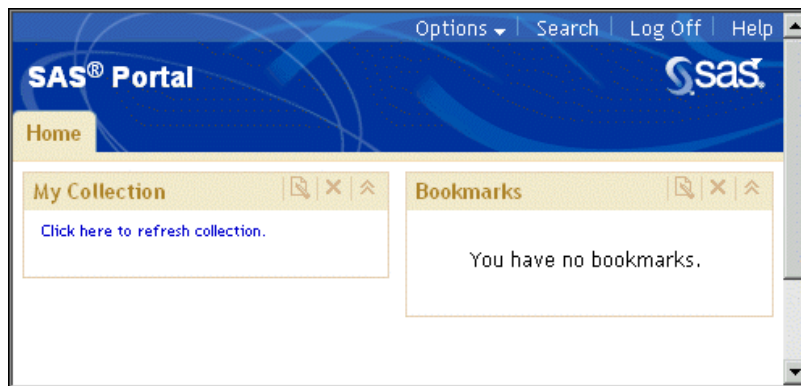
For information about the Portal Application Tree, see “Managing Portal Permission Trees in Metadata” on page 233.

**Initial Metadata Is Not Loaded**

No samples are loaded in the portal Web application.

The Portal Application Tree is not created until you start the servlet container.

This display shows a sample Home page that was created by **LoadPortalStructure\*.sas**:



## Administering the Public Kiosk

### Overview of the Public Kiosk

The Public Kiosk can be used to display pages and portlets that you want all users to be able to view as follows:

- If you are using the SAS Metadata Server’s authentication provider for user authentication, then when users access the portal Web application, you can display a Public Kiosk for users to view before they log on to the portal Web application. Anyone who can connect to the SAS Metadata Server (all PUBLIC users) can access the Public Kiosk.
- If you are using trusted Web authentication, then authenticated users will directly access the portal Web application instead of the Public Kiosk.

---

## Create or Edit the Public Kiosk

The Public Kiosk was created during installation when you loaded initial portal metadata (see “Loading Initial Metadata” on page 200 ). If you chose not to load the initial portal metadata, then you can create your own kiosk.

The SAS Guest user is the administrator of the Public Kiosk. When you log on to the portal Web application as the SAS Guest user, you can create, edit, and display content for the Public Kiosk. (The SAS Guest user’s personalized portal is the content that is displayed as the Public Kiosk).

*Note:* The default user account for SAS Guest is **sasguest**. Anything that SAS Guest creates is displayed in the Public Kiosk. Safeguard this account, and use it only to create content that is suitable for all portal viewers. △

To create or edit the Public Kiosk, follow these steps:

- 1 If you have installed only the SAS Web Infrastructure Kit, then configure the SAS Guest user as a group content administrator of the PUBLIC group so that SAS Guest can create the Public Kiosk. For instructions, see “Configure a Group Content Administrator” on page 224. If you installed the SAS Information Delivery Portal as well as the SAS Web Infrastructure Kit, then the SAS Guest User already has authorization to create the Public Kiosk.
- 2 Log on to the portal Web application as the SAS Guest user. A collection of publicly-available pages (with the default as a single Public Kiosk page) are defined for the SAS Guest user.
- 3 Add the desired Public Kiosk content to the SAS Guest user’s portal Web application. For details about adding content, see Chapter 17, “Adding Content to the Portal,” on page 237. Note the following about adding pages:
  - Any page that SAS Guest creates is automatically added to the Public Kiosk.
  - If a page has been shared with the PUBLIC group, then SAS Guest can search for and add the page to the Public Kiosk. (Even if the page is DEFAULT or STICKY, SAS Guest must add the page before it appears in the Public Kiosk.)
- 4 Specify the appropriate access controls for the content as necessary to enable the SAS Guest user to view the content. When users access the portal Web application’s Public Kiosk, they can only retrieve and view content that the SAS Guest user and the Public group are authorized to access. For more information, see Chapter 16, “Administering Portal Authorization,” on page 219 .

---

## Remove the Public Kiosk

In some cases, you might not want to display a Public Kiosk to users. Depending on whether you have installed the initial portal metadata, you can eliminate a Public Kiosk display as follows:

- If you have installed the initial portal metadata, then log on to the portal Web application as the SAS Guest user and delete all pages from the SAS Guest user’s portal Web application.
- If you have not installed the initial portal metadata, then do not define any pages for the SAS Guest user. No Public Kiosk will be available to users.

## Modifying the Logging Output Information and Location

### Overview of Log Configuration Files

You can modify the logging configurations for the portal Web applications and for the SAS Services Application by editing the logging configuration file that is associated with the applications. You can change the log file name and location, the types of messages that are stored in the log, and the log message format.

To edit the logging configuration file for an application, you must first locate the file using the following table:

**Table 14.4** Default Logging Files and Locations

Application	Default Logging Configuration File
SAS Services Application	<i>SAS-config-dir</i> \Lev1\web\Deployments\ RemoteServices\logging_config_svc.xml
Portal Web Application	<i>SAS-config-dir</i> \Lev1\web\Deployments\ Portal\logging_config_idp.xml
SAS Preferences Web Application	<i>SAS-config-dir</i> \Lev1\web\Deployments\ Portal\logging_config_prefs.xml
SAS Stored Processes Web Application	<i>SAS-config-dir</i> \Lev1\web\Deployments\ Portal\logging_config_stp.xml

*Note:* SAS Web Report Studio and SAS Web Report Viewer have similar log files. For details, see “Configuring the SAS Web Report Studio Logs” on page 109.  $\Delta$

When you open a configuration file, you will see several sections of lines such as the following (these sample lines were taken from the portal’s configuration file):

```
<RootLoggingContext
  priority="WARN">
  <OutputRef outputID="IDP_FILE_0"/>
  <OutputRef outputID="IDP_CONSOLE_0"/>
</RootLoggingContext>

<LoggingContext name="com.sas"
  priority="WARN"
  chained="false">
  <OutputRef outputID="IDP_FILE_1"/>
  <OutputRef outputID="IDP_CONSOLE_1"/>
</LoggingContext>

<LoggingContext name="com.sas.portal"
  priority="WARN"
  chained="false">
  <OutputRef outputID="IDP_FILE_2"/>
  <OutputRef outputID="IDP_CONSOLE_2"/>
</LoggingContext>
```

To customize logging, you edit the sections in the logging configuration file.

*Note:* Changes to the logging configuration files will be lost if you run the application's configuration script again. It is recommended that you make a copy of the configuration files.  $\Delta$

---

## Change the Types of Messages That Are Stored in the Log

To change the types of messages that are stored in the log, specify the priority level attribute for the appropriate logging context. Specify one of the following values:

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 abort.

For example, the following two examples show the priority attributes of WARN and INFO respectively:

```
<LoggingContext name="com.sas"
  priority="WARN"
  chained="false">
  <OutputRef outputID="IDP_FILE_1"/>
  <OutputRef outputID="IDP_CONSOLE_1"/>
</LoggingContext>

<LoggingContext name="com.sas.portal.container.deployment.PortletDeployer"
  priority="INFO"
  chained="false">
  <OutputRef outputID="CONSOLE_PortletDeployer"/>
</LoggingContext>
```

You must restart the servlet container before logging changes take effect.

---

## Change the Log Type, File Name, or Location

By default, the configuration file contains three file output types and four console output types.

- IDP\_FILE\_0, IDP\_FILE\_1, and IDP\_FILE\_2 are all set to output to a file
- IDP\_CONSOLE\_0, IDP\_CONSOLE\_1, IDP\_CONSOLE\_2, and IDP\_CONSOLE\_PortletDeployer are all set to output to the console

To change the log file, modify the **value** parameter for the file in the **<Output>** tag(s). The following example section shows the log file name in highlighted text:

```
<Output id="IDP_FILE_0"
  type="File"
  layoutPattern = "%d [%p] %c - %m%n">
  <param name = "File"
    value = "C:/SAS/EntBIServer/Lev1/web/Deployments/Portal/log/portal.log">
```

```
</Output>
```

Make the same change for the `IDP_FILE_1` and `IDP_FILE_2` output elements if you want to maintain the default behavior of logging to a single log file for all contexts. Alternatively, you can specify three different log files by entering different file names for the `IDP_FILE_0`, `IDP_FILE_1`, and `IDP_FILE_2` outputs.

*Note:* On Windows systems, the path must use either forward slash characters (/) or escaped backslash characters (\\).  $\Delta$

You can also create your own custom outputs by adding additional `<Output>` tags (each ID value must be unique). For example, suppose that you want to create a log output that prints to the console. You might add a block of code that is similar to this:

```
<Output id="IDP_CONSOLE_MyConsoleOutput"
type="Console"
layoutPattern = "%m%n">
</Output>
```

To use this new log output, you simply reference it as an `outputID` in one of your log contexts. The following example code shows the output reference in highlighted text:

```
<LoggingContext name="com.sas.services.information"
priority="DEBUG"
chained="false">
<OutputRef outputID="IDP_CONSOLE_MyConsoleOutput"/>
</LoggingContext>
```

You must restart the servlet container before logging changes take effect.

## Change the Log Message Format

To change the log format, modify the `layoutPattern` attribute for the `<Output>` tag. The following example shows the pattern in highlighted text:

```
<Output id="FILE"
type="File"
layoutPattern = "%d [%p] %c - %m%n">
<param name = "File"
value = "C:/SAS/cfg/Levl/web/Deployments/Portal/log/portal.log">
</Output>
```

For more information about the pattern syntax, see “Pattern Layout for easy formatting of output” in the SAS Foundation Services class documentation at <http://support.sas.com/rnd/gendoc/bi/api/Foundation/com/sas/services/logging/package-summary.html>.

You must restart the servlet container before logging changes take effect.

## Example: Customize the Log File to Track User Logons

This example illustrates how to log activity related to portal logons in order to monitor usage patterns. This code logs logon successes and failures.

To monitor portal logon successes and failures, add the following code to the `logging_config_idp.xml` file:

```
<LoggingContext name= "ApplicationMonitor.UserService"
priority="DEBUG"
chained="false">
```



```

    <OutputRef outputID="IDP_FILE_3"/>
</LoggingContext>

<Output id="IDP_FILE_3"
  type="File"
  layoutPattern = "%d{ISO8601} [%-5p] %-20.20u %-40.40c - %m%n">
<param name = "File"
  value = "C:/SAS/cfg/Levl/web/Deployments/Portal/log/portal.log" />
</Output>

```

You must restart the servlet container before logging changes take effect.

---

## Example: Customize the Log File to Track Portal Content Usage

This example illustrates how to log activity related to portal content usage. For this type of logging, the output includes more than just portal content, so you must parse the log output for the content that you want.

To monitor portal content usage, add the following code to the `logging_config_idp.xml` file:

```

<!-- With DEBUG: Shows User accessing SAS Stored Processes and SAS Web Reports -->
<LoggingContext name= "com.sas.portal.util.PortalContentUtil"
  priority="DEBUG"
  chained="false">
  <OutputRef outputID="IDP_FILE_3"/>
  <OutputRef outputID="IDP_CONSOLE_1"/>
</LoggingContext>

<!-- With DEBUG: Shows User accessing a SAS Information Map -->
<LoggingContext name= "com.sas.iquery.metadata.IntelligentQueryMetadataService"
  priority="DEBUG"
  chained="false">
  <OutputRef outputID="IDP_FILE_3"/>
  <OutputRef outputID="IDP_CONSOLE_1"/>
</LoggingContext>

```

You must restart the servlet container before logging changes take effect.

---

## Get Additional Information

A logging context is usually the fully-qualified class name of the class where the logging message originated.

For additional details about the elements in the logging configuration files, see the SAS Foundation Services class documentation for the `com.sas.services.logging` component at <http://support.sas.com/rnd/gendoc/bi/api/Foundation/com/sas/services/logging/package-summary.html>.

---

## Additional Documentation for the Portal

Here is additional documentation that is available for the portal Web application:

- The *Introduction to the SAS Information Delivery Portal* provides a general introduction and tour of the SAS Information Delivery Portal. (Even if you don't

have the SAS Information Delivery Portal installed, the same general concepts apply for the SAS Portal Web Application Shell that's included in the SAS Web Infrastructure Kit.)

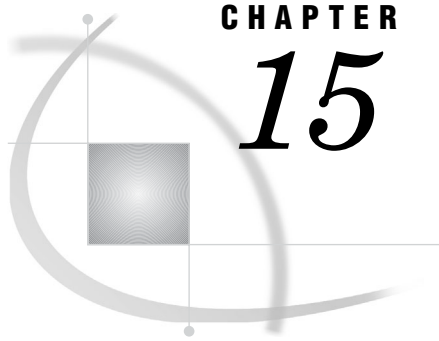
The *Introduction to the SAS Information Delivery Portal* is available at the following location:

**<http://support.sas.com/rnd/web/portal/doc2/tour/index.html>**

- Online Help in the portal's interface provides concepts and procedures that explain how to create pages in the portal Web application, 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 Web application.
- The **wik\_readme.html** file, located in the portal's installation directory, contains instructions for running the configuration script and for redeploying the portal.
- Chapter 4, "Best Practices for Configuring Your Middle Tier," on page 57 contains information that is associated with middle-tier administration.
- Chapter 3, "Setting Up and Managing Middle-Tier Security," on page 19 contains information about authentication, single sign-on, Secure Sockets Layer, and other security related administration.
- The *Web Infrastructure Kit: Developer's Guide* explains how to use the Web Infrastructure Kit to develop your own custom portlets and applications, or to customize and extend the features of the SAS Information Delivery Portal.

The *Web Infrastructure Kit Developer's Guide* is available at the following location:

**<http://support.sas.com/rnd/itech/library/library9.html>**



## CHAPTER

## 15

## Using the Portal Administration Tools

<i>Overview of the Portal's Administration Tools</i>	209
<i>Using the Portal Options Menu</i>	210
<i>Re-Create and Redeploy the Portal Web Application</i>	211
<i>Using <code>initPortalData</code> to Update Portal Permission Trees</i>	212
<i>What the <code>initPortalData</code> Utility Does</i>	212
<i>Run the <code>initPortalData</code> Utility</i>	213
<i>Using the Quiesce Portlet to Bring Down the Portal</i>	213
<i>Overview of the Quiesce Portlet</i>	213
<i>Add a Quiesce Portlet</i>	213
<i>Fields for the Quiesce Portlet</i>	214
<i>Using the SAS Portal Metadata Tool to Remove Portal Metadata</i>	215
<i>Overview of the SAS Portal Metadata Tool</i>	215
<i>What Data the Tool Does and Does Not Remove</i>	215
<i>Name and Location of the SAS Portal Metadata Tool Script</i>	216
<i>Run the SAS Portal Metadata Tool</i>	216
<i>Restore the Default Portal Metadata</i>	218

### Overview of the Portal's Administration Tools

The administrator can use the portal's administration tools for the following types of tasks:

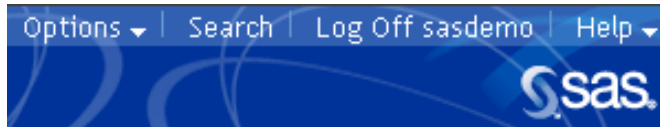
- Metadata and authorization administration: For content that is created in the portal Web application, such as links and pages, the portal Web application administers both content and authorization metadata. You can use the portal **Options** menu to create and share portal content.
- Personalization: Group content administrators, members of the Portal Admins group, and common users that have the SAS Information Delivery Portal installed can use the portal **Options** menu to set up and display the portal Web application's content.
- Prepare for shutdown: Members of the Portal Admins group can use the Quiesce portlet to prepare the portal Web application to be shut down.
- Re-create WAR files: You can use the `configure_wik` utility to re-create the WAR and configuration files for the portal Web application, the SAS Stored Process Web Application, and the SAS Services application (remote services). After you re-create the WAR files, you can redeploy them to your servlet container.
- Initialization of metadata for group permission trees: You can use the `initPortalData` utility to manually initialize metadata for group permission trees.

- Removal of portal-specific metadata: If necessary, you can use the SAS Portal Metadata Tool to remove the portal-specific metadata from the portal Web application's SAS Metadata Repository. The tool does not remove metadata that is managed by specific SAS Management Console plug-ins (such as Server Manager, BI Manager, or Publishing Framework).

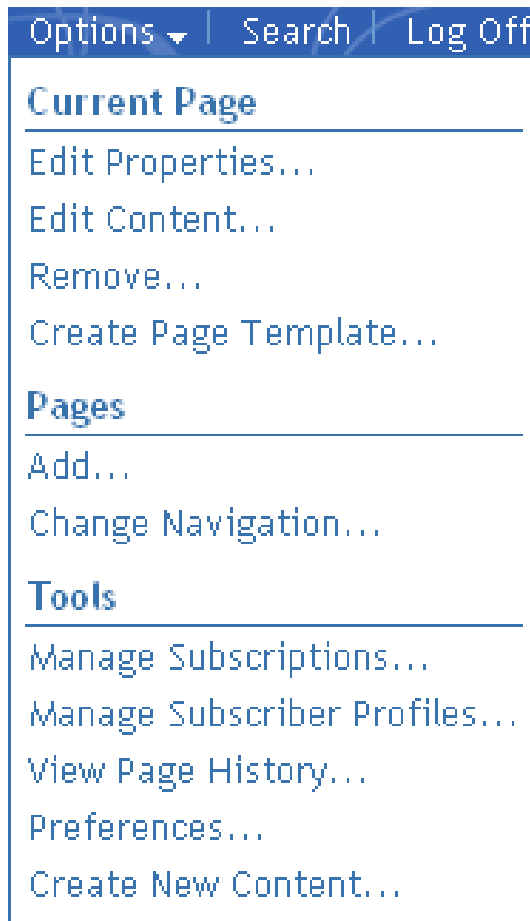
---

## Using the Portal Options Menu

After you log on to the portal Web application, the **Options** link appears in the banner, as shown here:



When you click **Options**, the portal Options menu appears:



The portal **Options** menu enables content administrators, members of the Portal Admins group, and common users that have the SAS Information Delivery Portal installed to perform several administration tasks:

- Add a page to your portal view by creating a new page or searching for and adding an existing page. When you add a page, you can share the page with a group that

has been defined in SAS metadata if you have administrator permissions for that group.

- After you have added a page, you can do the following:
  - Edit the properties of the page by changing the page's name, description, keywords, and rank. If the page has been shared and you have administrator permissions, you can unshare the page, share it with a different group, or change the page attribute, for example, from STICKY to DEFAULT.
  - Edit the contents of the page by adding, rearranging, and removing portlets.
  - Remove the page from your portal view, or delete it permanently from the portal Web application.
- Create a page template based on the current page (this option appears only if you are logged on as an administrator).
- Move the navigation bar to a different position (to the top or the side of the browser window).
- Manage publication channel subscriptions and subscriber profiles (if you have installed the SAS Information Delivery Portal).
- View and optionally clear user history in order to restore default pages that the user has removed.
- Change personal preferences for the country, language, and theme that is used in the Web application.
- Create content that is independent of any other portal component. For example, you can create a page without adding the page to your navigation bar, and you can create a portlet without adding the portlet to a page.

For complete information about using the **Options** menu or for instructions on performing any of the tasks that are listed here, refer to the online Help that is provided with the portal Web application.

---

## Re-Create and Redeploy the Portal Web Application

After initial installation and configuration, if you make changes to your portal configuration, then you must re-create and redeploy the portal Web application. For example, you change the portal configuration when you redistribute any of the portal's Web applications, when you change user IDs or passwords for particular SAS users (such as saswbadm or sastrust), when you change the method that is used for authentication, and any time that you edit the **install.properties** file.

To re-create and redeploy the portal Web application:

- 1 Complete the configuration changes that you want to make. Most often, these changes require you to edit the **install.properties** file.
- 2 Run the **configure\_wik** utility (**configure\_wik.bat** on Windows, **configure\_wik.sh** on UNIX). The **configure\_wik** utility is located in the **SAS-install-dir\Web\Portal2.0.1** directory. The **configure\_wik** utility creates the following files:
  - the portal Web application WAR file (**Portal.war**)
  - SAS Stored Process Web application WAR file (**SASStoredProcesses.war**)
  - SAS Preferences Web application WAR file (**SASPreferences.war**)
  - SAS Services application
  - other deployment and configuration files

The WAR files are created in the same directory where `configure_wik` resides. The `configure_wik` utility also copies the local and remote foundation services to the appropriate directory. For more information about these services, see Chapter 20, “Foundation Services and WebDAV Server Deployment,” on page 335.

- 3 Deploy the new **WAR** files into your servlet container. You might need to explode the WAR file before you deploy it. The “Deploy Web Application Files into the Servlet Container” section of the `wik_readme.html` file contains suggestions for deploying portal WAR files. (The `wik_readme.html` file is located in the `SAS-install-dir\Web\Portal2.0.1` directory.) For complete deployment instructions, consult the documentation that is provided for your servlet container. Unless otherwise instructed, deploy all of the **WAR** files that are listed in the previous step.
- 4 If you change particular values in the `install.properties` file, then you must delete and reimport the foundation services into the SAS Metadata Server. The foundation services are regenerated each time you run the `configure_wik` utility. If you have modified any server information related to the SAS Services Application, your WebDAV server, or the SAS Metadata Server in the `install.properties` file, then you must reimport the foundation services after you run `configure_wik`.

For details, see the following topics:

- “Changes That Require You to Reimport the Service Deployment Configurations” on page 339
  - “Reimport the Service Deployment Configurations” on page 340
- 5 If the SAS Services application is running, stop and restart it.
  - 6 If your servlet container or J2EE application server is running, stop and restart it.

*Note:* All of the procedures in this guide explicitly state whether you must redeploy the portal Web application after you perform the procedure. The procedures also specify which **WAR** files to redeploy, and whether you must delete and re-import the foundation services.  $\triangle$

---

## Using `initPortalData` to Update Portal Permission Trees

---

### What the `initPortalData` Utility Does

The `initPortalData` utility enables you to update permission metadata manually while you are logged out of the portal Web application. This capability is desirable when you have a large amount of metadata that requires updates.

The portal Web application will perform these same updates when you restart the servlet container or log on as the SAS Web administrator. However, if you have a lot of metadata changes, restarting the servlet container can take a long time. To avoid this time delay, you can instead create the folders before you start the servlet container by running the `initPortalData` utility.

Here is a high-level description of what `initPortalData` does:

- Creates the Portal Application Tree list in SAS metadata if the list has not already been created. For information about the Portal Application Tree list, see “Managing Portal Permission Trees in Metadata” on page 233.
- Creates new group permissions folders in the Portal Application Tree list if the folders have not already been created for those groups. This features enables you

to define new groups in metadata, and then initialize those groups before you log on to the portal.

- Removes unused permissions tree folders for users and groups. If you have deleted any user or group identities in metadata, then the utility removes their respective folders from the Portal Application Tree list. The utility also permanently deletes all data that is stored within the respective folders. In other words, when a permission tree folder is removed, all portal content (pages, template pages, portlets, links, and so on) that is owned by the user or group is also removed.
- Performs other metadata updates as needed. One example includes metadata that is changed or added between releases of the portal software.

## Run the `initPortalData` Utility

To run the `initPortalData` utility, complete these steps:

- 1 Start the SAS Metadata Server if it is not already running.
- 2 If you have not already defined the groups in metadata, then do so now. For instructions, see the SAS Management Console User Manager Help.
- 3 Run the `initPortalData` utility (`initPortalData.bat` on Windows, `initPortalData.sh` on UNIX). This utility is located in the `SAS-install-dir\SAS\Web\Portal2.0.1\Tools` directory.

If the `initPortalData` utility runs successfully, then a message like the following is displayed:

```
Done initializing metadata information
Transaction count: [0]
DONE
```

The transaction count in this message indicates the number of transactions that are still active when the utility exits. A value other than zero indicates an error.

## Using the Quiesce Portlet to Bring Down the Portal

### Overview of the Quiesce Portlet

The Quiesce portlet is a portlet template that enables the SAS Web Administrator to quiesce the portal Web application. *Quiescing* the portal Web application prepares it to be shut down by preventing new users from logging on.

New users who attempt to access a quiesced portal Web application receive the following message:

```
Portal authentication is currently disabled
```

*Note:* To re-enable new users to log on, you must restart the servlet container. △

### Add a Quiesce Portlet

To use the Quiesce portlet, you must first create an instance from the template and add it to a page:

- 1 Log on to the portal Web application as a SAS Web Administrator.

- 2 Edit the contents of a page in order to add the Quiesce portlet. (To add the portlet to a page, you will select **Quiesce Portlet** from the **Portlet Type** drop-down list.)

For complete instructions on adding a portlet, refer to the online Help that is provided with the portal Web application.

---

## Fields for the Quiesce Portlet

The screenshot shows a web-based configuration window for the Quiesce Portlet. The window has a title bar with the text 'Quiesce Portlet' and standard window controls (close, maximize). The main content area contains the following fields:

- Current State:** A text field displaying the value 'Running'.
- Quiesce SAS Services:** A label followed by an unchecked checkbox.
- Quiesce Wait:** A label followed by a text input field and the word 'seconds'.
- Quiesce:** A blue button with white text located at the bottom left of the window.

The Quiesce portlet contains the following fields:

**Current State** shows the operational status of the portal Web application. The values for this field are as follows:

<b>Running</b>	The portal Web application is running normally.
<b>Quiescing</b>	The Quiesce portlet is waiting for the interval specified in the Quiesce Wait field before quiescing the portal Web application.
<b>Quiesced</b>	The portal Web application is not accepting new users. As soon as the current users log off, the portal Web application can be safely shut down and restarted.

**Quiesce SAS Services** specifies whether Web applications that run within the portal Web application will also be quiesced.

*Note:* The SAS Services Application is not affected by the Quiesce portlet. △

**Quiesce Wait** specifies an interval (in seconds) to wait before quiescing the portal Web application.

**Quiesce** quiesces the portal Web application.

*Note:* When the portal Web application is quiescing or has already been quiesced, the Quiesce portlet only contains the **Current State** field. △



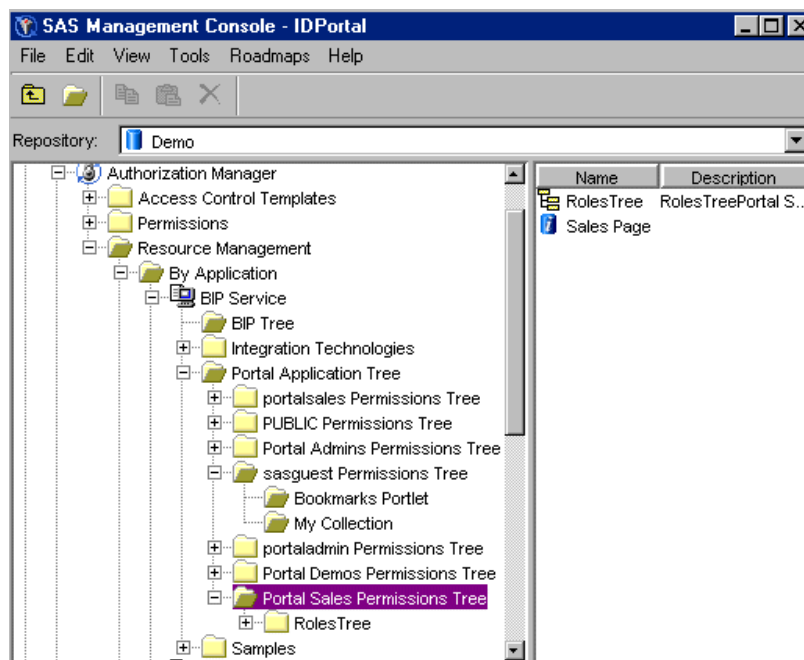
## Using the SAS Portal Metadata Tool to Remove Portal Metadata

### Overview of the SAS Portal Metadata Tool

The SAS Portal Metadata Tool removes portal-specific metadata from the SAS Metadata Repository. You might want to remove all of the portal-specific metadata that you have created for either of the following reasons:

- a requirement to start with a clean SAS Metadata Repository
- a requirement to create a new set of portal-specific metadata

The portal-specific metadata is defined in the Portal Application Tree of the SAS Management Console navigation tree, as shown here:



### What Data the Tool Does and Does Not Remove

The SAS Portal Metadata Tool removes all portal-specific metadata, which includes metadata for the following:

- links
- pages and page templates
- portlets
- syndication channels
- user and group permission trees
- Web applications

The tool does *not* remove the following metadata:

- metadata that is managed by specific SAS Management Console plug-ins (such as Server Manager, BI Manager, Publishing Framework, and User Manager), including metadata for the following:
  - SAS Stored Processes
  - published packages
  - users and groups

*Note:* BI Manager is available beginning with SAS Foundation Services 1.2. If you have not upgraded to this release, then you can use Stored Process Manager to register and manage stored processes. BI Manager replaces Stored Process Manager. For more information about using BI Manager or Stored Process Manager, see the Help in SAS Management Console.  $\Delta$

- metadata that is loaded by the **LoadDefaultProperties.sas**, **LoadPreferencesConnection.sas**, and **LoadThemeConnection.sas** programs. These programs load metadata that is not exclusive to the portal.

## Name and Location of the SAS Portal Metadata Tool Script

The SAS Portal Metadata Tool runs by invoking the following script: **RemovePortalMetadata** (**RemovePortalMetadata.bat** for Windows and **RemovePortalMetadata.sh** for UNIX).

**RemovePortalMetadata** is installed in the *SAS-install-dir\Web\Portal2.0.1\Tools* directory.

Note the following about the tool:

- Your installation might contain a second instance of the tool inside the *SAS-install-dir\Web\Portal2.0.1\Portal\Tools* directory. Do not run the tool from this directory. If you attempt to run the tool from this alternate location, the tool will not remove the metadata correctly.
- Do not move or rename the **portal-objects.xml** file that resides in the *SAS-install-dir\Web\Portal2.0.1\Tools* directory. This file contains the metadata information that the tool uses to remove the portal-specific metadata.

## Run the SAS Portal Metadata Tool

To run the SAS Portal Metadata Tool:

- 1 Back up the metadata server. For details, see “Using the %OMABAKUP Macro to Perform Backups and Restores” in the *SAS Intelligence Platform: System Administration Guide*.
- 2 Stop your servlet container and the SAS Services Application.
- 3 Use SAS Management Console to add the SAS Administrator user (for example, *<Windows domain or host name>\sasadm*) to the Portal Admins group. This is required in order to give the SAS Administrator permission to delete portal metadata from the Portal Application Tree. Follow these steps:
  - a In SAS Management Console, open the User Manager.
  - b Select the **Show Groups** check box. Then select the Portal Admins group, and select **Actions** ► **Properties**.
  - c In the Portal Admins Properties window, select the **Members** tab.
  - d Select **SAS Administrator**, select the right arrow, and then select **OK**.

- 4 Optionally, if you want to change the logging output for the SAS Portal Metadata Tool, then you can edit the **logger.config** file and specify the desired logging output configuration. This file is located in the *SAS-install-dir\Web\Portal2.0.1\Tools* directory.

The **logger.config** file follows the Log4j standard. For details about the Log4j standard, see the information for the product on the Apache Jakarta Web site <http://logging.apache.org/log4j/docs/index.html>. The **logger.config** file specifies which messages are output to the following log files:

- **error.log**
- **output.log**
- **xmlstatements.log**

- 5 Run the **RemovePortalMetadata** script. To locate the directory for running the script, see “Name and Location of the SAS Portal Metadata Tool Script” on page 216 . When you run **RemovePortalMetadata**, you can specify the following options on the command line:

**-help**  
displays a list of options.

**-nogui**  
does not display the GUI screen.

**-noprompt**  
does not display the warning prompt.

**logger=logfilename**  
outputs log4j messages to the specified file.

**-host=host name**  
specifies the SAS Metadata Server machine (see the `$_SERVICES_OMI_HOST$` property in the **install.properties** file).

**-port=port**  
specifies the SAS Metadata Server port (see the `$_SERVICES_OMI_PORT$` property in the **install.properties** file).

**repository=repository**  
specifies the SAS Metadata Server Repository (see the `$_SERVICES_OMI_REPOSITORY$` property in the **install.properties** file).

**user=user ID**  
specifies the fully qualified user ID for connection to the SAS Metadata Server. Use the SAS Administrator’s fully qualified user ID (`<Windows domain or host>\sasadm`). To use the SAS Portal Metadata Tool to delete the portal-specific metadata, the SAS Administrator must have “Delete” permissions for the metadata.

**pwd=password**  
specifies the password for connection to the SAS Metadata Server.

The SAS Portal Metadata Tool dialog box appears (unless you specify the **nogui** option):

- 6 In the SAS Portal Metadata Tool dialog box, enter the appropriate information for your SAS Metadata Server. For descriptions of the fields, see the options that are described in the previous step.
- 7 Select **Clean** to start the SAS Portal Metadata Tool. The tool runs and displays the logging messages to the screen, or to a file as specified by the **logging.config** file.

*Note:* The amount of metadata that is associated with the portal-specific metadata affects the time that it takes to remove that metadata. For example, if 100 users are associated with a particular page of the portal Web application, then the tool will take a longer time to remove that page from the metadata.  $\triangle$

- 8 After you successfully run the SAS Portal Metadata Tool, you might want to undo the metadata change that you made previously in step 3. That change was required in order to run the SAS Portal Metadata Tool. If you want to undo the change, then remove the SAS Administrator from the Portal Admins group.

If you plan to restore the default metadata after you run the SAS Portal Metadata Tool, then do not restart the servlet container until you have run the tool that restores the default metadata. For instructions, see “Restore the Default Portal Metadata” on page 218.

---

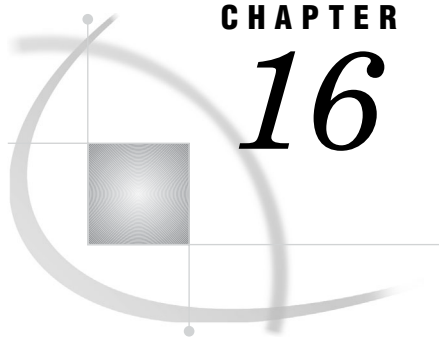
## Restore the Default Portal Metadata

After removing portal metadata, if you want to restore the default portal metadata, then do the following:

- 1 Do not restart the servlet container until you have completed the next step. When you start the servlet container, some metadata is created for the portal. If you then run the program that is described in the next step, the program will fail (it will abort if you try to reload existing metadata).
- 2 Run **LoadPortalStructure.sas**, which is located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory.

After you run **LoadPortalStructure.sas**, the next time you start your servlet container and log on to the portal Web application, you will see the default metadata. For more information about **LoadPortalStructure.sas**, or for a description of the default metadata, see “Loading Initial Metadata” on page 200.

*Note:* You should *not* run **loadWIKPrimer** script, which is run from **instructions.html** (the step that loads Web Infrastructure Kit "primer" metadata). The **loadWIKPrimer** script will try to re-run the **LoadDefaultPreferences.sas**, **LoadPreferencesConnection.sas**, **LoadThemeConnection.sas**, and **LoadPortalStructure.sas** programs. Re-running the **loadWIKPrimer** script will result in failures because metadata for preferences and themes already exists (the metadata was not removed by the **RemovePortalMetadata** tool).  $\triangle$



## CHAPTER

## 16

## Administering Portal Authorization

<i>Overview of Portal Authorization Tasks</i>	219
<i>Planning for Portal Users and Groups</i>	220
<i>Overview of Planning for Portal Users and Groups</i>	220
<i>Step 1: Analyze Content</i>	220
<i>Files</i>	221
<i>Packages Published on a Xythos WFS Server</i>	221
<i>SAS Publication Channels on a Xythos WFS Server</i>	221
<i>Other Types of Content</i>	221
<i>Step 2: Analyze and Group Users</i>	221
<i>Step 3: Assign Group Content Administrators</i>	222
<i>Understanding Portal Authorization</i>	222
<i>Overview of Authorization</i>	222
<i>Methods Used to Implement Authorization</i>	223
<i>Portal Items That Require Authorization, and Their Respective Authorization Methods</i>	224
<i>Configure a Group Content Administrator</i>	224
<i>Sharing Content in the Portal Web Application</i>	226
<i>Overview: Sharing Portal Content</i>	226
<i>Who Can Share Portal Content</i>	227
<i>Types of Changes That Can Be Made to Shared Content</i>	227
<i>About Shared Pages</i>	228
<i>Sharing Items That Contain Other Items</i>	228
<i>When Can You Share Content?</i>	229
<i>Suggestions for Sharing Content with Multiple Groups of Users</i>	229
<i>Setting Up Authorization for Stored Processes and Publication Channels</i>	229
<i>Implementing Authorization for the Xythos WebFile Server</i>	231
<i>Overview of Xythos WebFile Server Authorization</i>	231
<i>Example Scenario: Xythos WebFile Server Authorization</i>	231
<i>Managing Portal Permission Trees in Metadata</i>	233
<i>Overview of Permission Tree Folders</i>	233
<i>How Permission Tree Folders Are Created</i>	234
<i>How Permission Tree Folders Are Removed</i>	235
<i>Verify Permission Tree Folders and Permissions</i>	235

### Overview of Portal Authorization Tasks

In addition to the tasks that are outlined in “Planning Your Middle-Tier Security Implementation” on page 20, you will need to control access to the content that is added to the portal.

Here are the portal authorization tasks that you might need to perform:

- 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 220.
- Create group content administrators to help control the flow of information to particular groups of users. These administrators can assume responsibility for creating and sharing portal content with their respective groups. For details, see “Configure a Group Content Administrator” on page 224 and “Sharing Content in the Portal Web Application” on page 226.
- 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 222.
- Although permission trees are created for groups when you log on to the portal, you might want to create permission tree folders manually before logging on to the portal. This option is recommended when you have a large number of new groups that require permission tree folders. See “Using `initPortalData` to Update Portal Permission Trees” on page 212.

---

## Planning for Portal Users and Groups

---

### Overview of Planning for Portal Users and Groups

When you define users to access the portal Web application, 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 Web application security, and reduces the chance for errors. For example:

- As new content is added to the Web application, 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 group(s) for which they are a group content administrator.
- Any member of the Portal Admins group (for example, the SAS Web Administrator) can share any user’s content with any group.

For more information about planning portal users and groups, see “Planning User Accounts and Their Organization into Groups” on page 21. For instructions on adding users and groups, see “User and Group Management” in the *SAS Intelligence Platform: Security Administration Guide*.

The following steps outline basic tasks for planning your user groups.

---

### Step 1: Analyze Content

The first step in setting up groups is to analyze the content that is planned for the portal Web application. For each category of content, determine whether authorization

restrictions are needed. If restrictions are needed, then identify the types of users that should and should not be authorized to access the content.

For the portal Web application, in order to implement the appropriate security, you must define groups for the following content:

## Files

If you are storing file content on a Xythos WFS WebDAV repository, then you must set up groups for access to the appropriate group folders. (If you have installed the SAS Information Delivery Portal, then you might already have group folders set up for SAS reports that are stored on the Xythos WFS WebDAV server).

## Packages Published on a Xythos WFS Server

If you have installed the SAS Information Delivery Portal and you are publishing packages to a Xythos WebFile Server (WFS), then you must set up a group that contains all of the users who need the ability to publish to the Xythos WFS. For details about setting up users for publishing, see “Publishing to Secure Servers” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/publish/publish\\_security.html](http://support.sas.com/rnd/itech/doc9/admin_oma/publish/publish_security.html).

In addition, you should plan for the Xythos WFS personal and group folders in which you will publish and access the packages.

## SAS Publication Channels on a Xythos WFS Server

If you have installed the SAS Information Delivery Portal and are publishing packages to a SAS publication channel on Xythos WFS, then you must set up a group that contains all the users who need the ability to publish to the publication channel’s Xythos server. For details about setting up users for publishing, see “Publishing to Secure Servers” in the *SAS Integration Technologies: Administrator’s Guide*.

In addition, you should plan for the Xythos WFS WebDAV personal and group folders in which you will publish and access the packages.

## Other Types of Content

You might also set up groups based on the following content:

- Portal Web application content: You might define groups based on the portal Web application content that members of the group need to access. Portal Web application content includes Web applications, links, page templates, portlets, and syndication channels (SAS Information Delivery Portal only).
- Content on SAS application servers: You might define groups based on which users need access to data on particular servers (including SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers). In addition, you might set up a group definition for users to access a server in a different authentication domain than the SAS Metadata or Web server’s authentication domain.
- Groups that have already been created for SAS Reports or SAS Information Maps: Some of the groups that you need to define for portal Web application content might be the same groups that are already defined for SAS Reports, SAS Information Maps, or SAS Stored Processes.

---

## 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 could start by identifying large groups of users. You can then subdivide those large groups into smaller groups if necessary. For example, you could create an Accounting user group that needs access to financial files through the portal Web application. Within that group, you could identify a subgroup of users who need access to salary information files that should not be accessed by the rest of the group.

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

After you set up a group, you can configure a user to be a group content administrator. Members of the Portal Admins group, which includes the SAS Web Administrator, are automatically configured as group content administrators for all groups.

Group content administrators can create personal pages and share their personal pages with all members of their respective group. For general information about sharing portal content, see “Sharing Content in the Portal Web Application” on page 226.

For each group that you plan to define, determine if you need to assign a group content administrator for that group.

For instructions on configuring a group content administrator, see “Configure a Group Content Administrator” on page 224.

---

## Understanding Portal Authorization

---

### Overview of Authorization

The portal Web application uses the authorization (access control) metadata on the SAS Metadata Server to determine who can view content in the portal Web application.

By default, the SAS Guest and SAS Web Administrator users have ReadMetadata and WriteMetadata permission on the repository Access Control Template (ACT). Additionally, each portal user that you define must have ReadMetadata and WriteMetadata permission on the repository ACT in order to log on to the portal Web application. By default, all users who are defined in metadata are members of the PUBLIC group and have administrator permissions (including ReadMetadata and WriteMetadata). It is expected that you will add portal users to groups that you define in SAS metadata, grant the necessary permissions to those groups in the ACT, and then limit the permissions for the PUBLIC group.

See “Who Can Administer the Portal Web Application” on page 193 for a description of the permissions that are granted to the SAS Web Administrator and to portal users. For a better understanding of authorization and the ACT, see *SAS Intelligence Platform: Security Administration Guide*.

As part of your security implementation, you will also set up authorization for particular portal Web application content in order to allow or restrict user access to that content. For example, if the portal Web application displays SAS reports that contain employee salary information, you will want to 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 Web application 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. For portal-specific details about planning for and creating groups, see “Planning for Portal Users and Groups” on page 220.

After organizing your users into groups, the level of additional access control that you apply will depend on your user base and on the sensitivity of the content that you make available through the portal Web application. For a basic understanding of access controls, see “Understanding Authorization” in the *SAS Intelligence Platform: Security Administration Guide*.

---

## Methods Used to Implement Authorization

You can implement authorization in the following basic ways:

- Specify ownership (personal or shared) for content in the portal Web application.

By default, content that any user creates in the portal Web application is personal. *Personal content* is content that can be edited, viewed, and deleted only by the user who created it, or by a SAS Web Administrator. When you create content in the portal Web application, the content is added to the appropriate permission tree in SAS metadata. For example, if you log on to the portal Web application as the SAS Web Administrator and create a personal page, that page is added to the SAS Web Administrator’s permission tree. (To learn more about permission trees, see “Managing Portal Permission Trees in Metadata” on page 233.)

The portal Web application enables you to share content with a 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 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 SAS Web Administrator or a group content administrator for the respective group. For more information about sharing content, see “Sharing Content in the Portal Web Application” on page 226.

- Specify authorization in SAS metadata.

When you create content apart from the portal Web application, 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 stored processes and publication channels. This option provides flexibility in controlling access to portal Web application content. For more information, see “Setting Up Authorization for Stored Processes and Publication Channels” on page 229.
- Specify authorization for custom-developed portlets in the portlet’s descriptor file. The descriptor file can also contain an attribute that enables authorized portal users to share the portlet by using the portal Web application’s share feature.

For information about using the portlet deployment descriptor file to specify which users or groups are authorized to access the portlet, see “Creating a Deployment Descriptor” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/tasks/dg\\_portlet\\_descr.html](http://support.sas.com/rnd/itech/doc9/portal_dev/tasks/dg_portlet_descr.html).

- 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 Web application.) For details, see the applicable topic for adding page templates, Web applications, or syndication channels in Chapter 17, “Adding Content to the Portal,” on page 237.

- Specify authorization for Xythos WFS content using the Xythos administrator console. For more information, see “Implementing Authorization for the Xythos WebFile Server” on page 231.
- You can also set up Java security for your Web applications. If you deploy Web applications that run from the portal, then you should set up permissions for those applications. For details, see “Adding Permissions to Policy Files” on page 45.

---

## 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 240.

---

## Configure a Group Content Administrator

A group content administrator is a user who has **WriteMetadata** permission for the respective group. 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 Web administrator has **WriteMetadata** permission for all groups 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 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 “Managing Portal Permission Trees in Metadata” on page 233.

To configure a group content administrators, follow these steps:

- 1 Log on to SAS Management Console as the SAS Administrator.
- 2 Navigate to **Authorization Manager**  $\blacktriangleright$  **Resource Management**  $\blacktriangleright$  **By Application**  $\blacktriangleright$  **BIP Service**  $\blacktriangleright$  **Portal Application Tree**.
- 3 Select the group for which you wish to assign a group content administrator.
- 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 user who will be the content administrator. You must select an individual user, not a group.

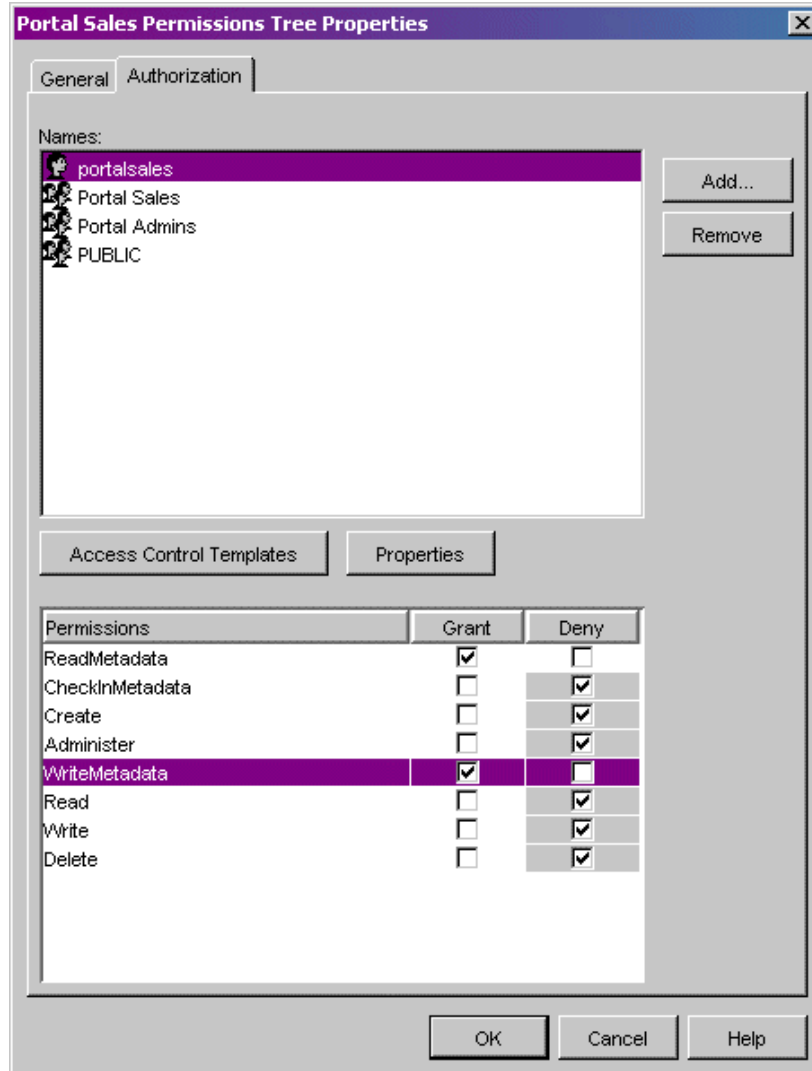
*Note:* If a particular user is not listed, click **Add** to add the user. When you return to the **Authorization** tab, make sure the appropriate user is selected in the **Names** list box.  $\triangle$

- 7 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 directly assigned, instead of inherited. 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.

The following display of the Authorization tab shows the users and groups who have permissions for the Portal Sales permission tree. The WriteMetadata permission is directly assigned to the portalsales user.



**8** 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 Web application and share personal content with that group.

---

## Sharing Content in the Portal Web Application

---

### Overview: Sharing Portal Content

After you have defined a group in SAS metadata and initialized its respective group permission tree, you can log on to the portal Web application and share portal content with that group. The portal's share feature provides an easy and efficient way to control access to particular types of portal content. (For information about permission trees, see "Managing Portal Permission Trees in Metadata" on page 233.)

The following content items can be shared from the portal Web application:

- pages
- portlets
- applications
- links
- syndication channels

When you create one of these items, if you have administrative permissions, then you 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 SAS Web Administrator or a group content administrator can edit the content.

*Note:* The portal Web application uses the authorization metadata on 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.  $\triangle$

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 229.

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 Web application. When you select a group in the **Location (group)** drop-down list, you share the page with that group:

## Who Can Share Portal Content

You must log on to the portal Web application with the appropriate permissions in order to share content. Here are the types of users that can share content:

**Table 16.1** Who Can Share Portal Content

User	Share Permissions
SAS Web Administrator	Can create and share portal content with any group that is defined in SAS metadata, including PUBLIC.
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 224.

For more information about the permissions that are granted to these users in SAS metadata, see “Who Can Administer the Portal Web Application” on page 193.

## Types of Changes That Can Be Made to Shared Content

After content has been shared with a group, any user who is authorized as an administrator for the group can do the following:

- 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 you unshare content, the content is removed from all of the portal views where that content is displayed. The content still exists in the portal environment. When you change the group to which the content is shared, the content is moved from the portal views for members of the original group to the portal views for members of the new group.
- 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 246. △

- 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 attribute (pages only). You can change the attribute of a shared page (STICKY, DEFAULT, AVAILABLE). For more information about these attributes, see “Page Attributes: AVAILABLE, DEFAULT, and STICKY” on page 245.

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 attribute (DEFAULT, AVAILABLE, or STICKY) 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 228.

For more information, see “Shared Pages” on page 246.

---

## Sharing Items That Contain Other Items

When you share portal content, you can specify whether you also want to share any items that are contained within the content that you share.

For example, if you share a page that contains portlets, then you can specify whether you also want to share those portlets. The portal Web application 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.

*Note:* When you share a page that contains an Alerts portlet, 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. △

Similarly, 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.

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 Xythos WFS 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 233.

In the portal Web application, 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, refer to the online Help that is provided with the portal Web application.

---

## Suggestions for Sharing Content with Multiple Groups of Users

The portal Web application 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 can be 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. You can use the portal’s access control features to block access to different portions of the content on a shared page. The effect is that different users see different content on the same shared page.

---

## Setting Up Authorization for Stored Processes and Publication Channels

For certain content, such as SAS Stored Processes and SAS publication channels, you might need to implement authorization by manually assigning access controls for the content metadata on the SAS Metadata Server. You can also implement authorization for SAS Stored Processes and SAS publication channels by manually assigning access controls for the server-level metadata on the SAS Metadata Server. By granting or denying permissions in the metadata for the portal Web application’s SAS Metadata Server, you can control security at virtually any level of granularity.

To specify access control permissions, you can log on to SAS Management Console as a SAS Administrator and use the Authorization Manager to specify authorization metadata. SAS Management Console’s Authorization Manager enables you to specify

access control for SAS publication channels, SAS Stored Processes, servers, and their resources as follows:

- Publishing Framework resources, including the following:
  - SAS publication channels: For a user to self-subscribe to SAS publication channels, the user must have **ReadMetadata** access on the SAS Publication Channel.
  - SAS subscribers: In the portal Web application, subscriber profiles designate a set of personal preferences for subscribing to SAS publication channels. The SAS Administrator can also specify authorization metadata for the subscriber profiles.

For more details, see the *SAS Integration Technologies: Administrator's Guide* at:

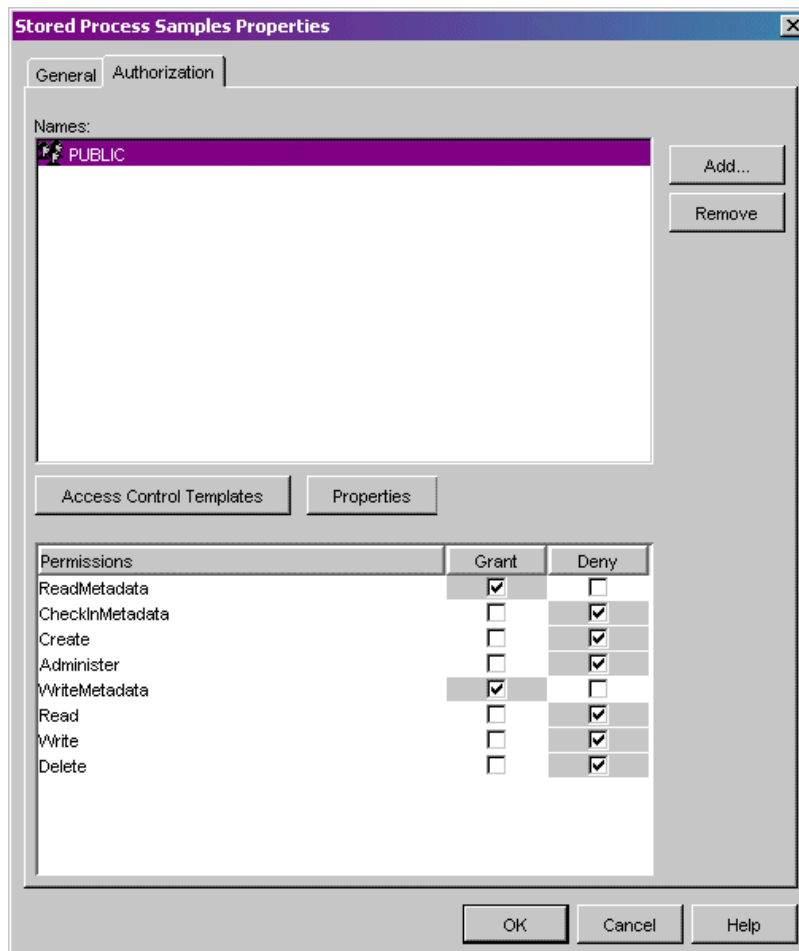
[http://support.sas.com/rnd/itech/library/toc\\_general.html](http://support.sas.com/rnd/itech/library/toc_general.html)

- BI Manager plug-in resources, including the following:
  - SAS Stored Processes
  - Paths in which stored processes reside

For more details, see the *SAS Integration Technologies: Developer's Guide* at:

[http://support.sas.com/rnd/itech/library/toc\\_devguide.html](http://support.sas.com/rnd/itech/library/toc_devguide.html)

After you create a resource, you can use SAS Management Console to associate access controls with the resource. You associate access controls with a resource by modifying the resource's properties. The **Authorization** tab of a resource's Properties window maintains the authorization information:





---

# Implementing Authorization for the Xythos WebFile Server

---

## Overview of Xythos WebFile Server Authorization

To authorize access to content on a Xythos WebFile Server (WFS), administrators can specify users and groups that are defined in a SAS Metadata Repository. The SAS User Management Customization that is provided with Xythos WFS enables you to specify which users or groups are authorized to access specific folders in the Xythos WFS repository, and what type of access permissions they have for the folders.

Use the Xythos WFS administration console to create folders and associate access controls with the folders.

*Note:* This topic does not describe authentication for Xythos WFS. By default, WebDAV users are authenticated against the SAS Metadata Server's authentication provider. You must define your WebDAV users on the appropriate authentication provider for the SAS Metadata Server. For a description of authentication, see the *SAS Intelligence Platform: Security Administration Guide*.

For full details about authentication and authorization, refer to the Xythos WFS product documentation.  $\Delta$

Before you can associate access controls with a folder, you must complete these tasks:

- 1 Use the WebDAV administration console to create the folder on the WebDAV server.
- 2 Ensure that the appropriate user, group, and login definitions exist on the SAS Metadata Server for the WebDAV users and groups for whom you wish to control access to the folder.

After you have created the WebDAV folders and have ensured that the appropriate user, group, and login definitions are created on the SAS Metadata Server, use the Xythos WFS administration console to associate access controls with the folder.

---

## Example Scenario: Xythos WebFile Server Authorization

Within your portal implementation, you might utilize the publish and subscribe capabilities to publish (write) and subscribe to (read) group folders on a DAV-based publication channel.

The following scenario shows a portal'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 Xythos WFS server. On the SAS Metadata Server, these teams are represented by two groups, Americas Sales and Sales Executives. In addition, the portal installation provides a group named Portal Admins, which has unrestricted access to the portal's metadata on the SAS Metadata Server.

This publish and subscribe scenario has a requirement for three different content areas, or group folders, on the WebDAV 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.

*Note:* The Portal Admins group can also read (subscribe to), write (publish), and delete information for all of the above directories.  $\Delta$

The following table summarizes this scenario's group-based folders on the WebDAV server, and the permissions for each user:

**Table 16.2** Summary of Folders on the WebDAV Server

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

To create this sample Xythos configuration, complete these steps:

- 1 In SAS Management Console, define the users, groups, and login credentials that will access the WebDAV server. When you define login credentials, you must specify the same authentication domain name that you specified for the Xythos WFS server during installation of the SAS User Management Customization tool. This authentication domain can be found in the **saswfs.properties** file.

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

Group Metadata Identities	User Metadata Identities	User ID	Authentication Domain
America Sales	salesusr1	salesusr1	DefaultAuth
Sales Executives	execusr1	execusr1	DefaultAuth
Portal Admins	saswbadm	saswbadm	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 Xythos WFS administration console, create your new directory under the sasdav directory. For this example, you would navigate to the **sasdav** directory, and then create these three subdirectories: **Catalog Sales**, **Field Sales**, and **Sales Execs**.

*Note:* Ignore any messages that state "The directory does not have an owner." Directory ownership is not a requirement for the SAS User Management Customization tool.  $\Delta$

- 3 In the Xythos WFS administration console, configure the access permissions for the folders that you created. For this example, you would set the access permissions for each subdirectory, using the following tables as guides:

**Table 16.3** 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
Portal Admins	Yes	Yes	Yes	Yes	Yes	Yes

**Table 16.4** 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
Portal Admins	Yes	Yes	Yes	Yes	Yes	Yes

**Table 16.5** 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
Portal Admins	Yes	Yes	Yes	Yes	Yes	Yes

---

## Managing Portal Permission Trees in Metadata

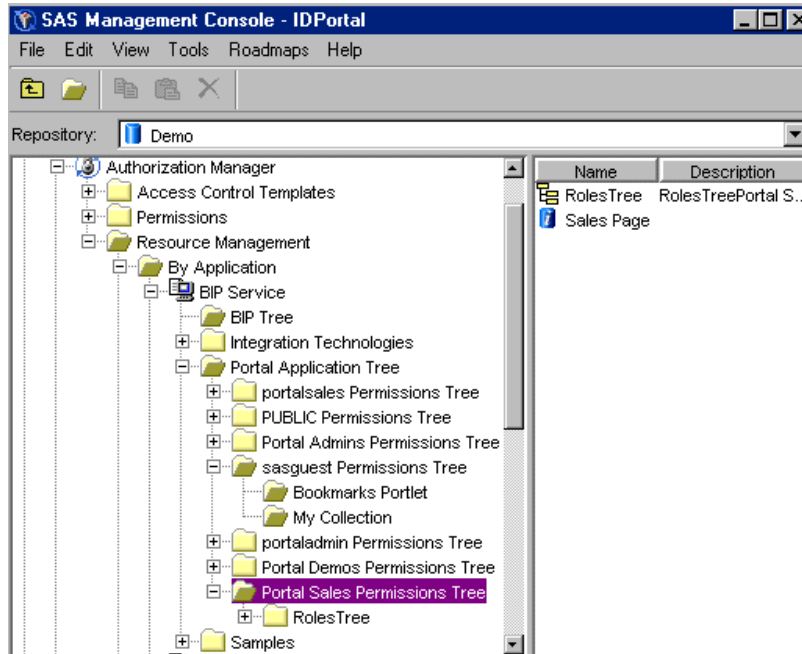
---

### 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 portal Web application 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*.

For example, suppose that you have created a Portal Sales group in metadata. In Authorization Manager, a folder named **Portal Sales Permission Tree** appears in the **Portal Application Tree** list. If you inspect the properties for the **Portal Sales Permission 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 234.)



When you add new users or groups to the metadata server, the portal Web application 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 Web application 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.

---

## 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 portal Web application uses to create permission tree folders depend on whether the metadata identity is a user or a group:

- User permission trees: The portal Web application creates a permission tree for a user entity that is defined in metadata when you log on to the portal Web application as that user.
- Group permission trees: The portal Web application creates a permission tree for one or more groups that are defined in metadata when you do any *one* of the following:
  - Restart the servlet container. The portal software creates permission tree folders for new groups each time the servlet container is started.
  - Log on to the portal Web application as the SAS Web administrator (**saswbadm**, or any member of the Portal Admins group).
  - Create permission tree folders manually by running the **initPortalData** utility. This option is recommended when you have a large number of new

groups that require permission tree folders. For instructions, see “Using `initPortalData` to Update Portal Permission Trees” on page 212.

*Note:* For instructions on defining users or groups in metadata, see the SAS Management Console User Manager Help. △

---

## How Permission Tree Folders Are Removed

After you delete a user or group identity from SAS metadata, the portal Web application removes the corresponding permission tree when you do any *one* of the following:

- Restart the servlet container.
- Log on to the portal Web application as the SAS Web administrator (`saswadm`, or any member of the Portal Admins group).
- Run the `initPortalData` utility. For instructions, see “Using `initPortalData` to Update Portal Permission Trees” on page 212.

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.

---

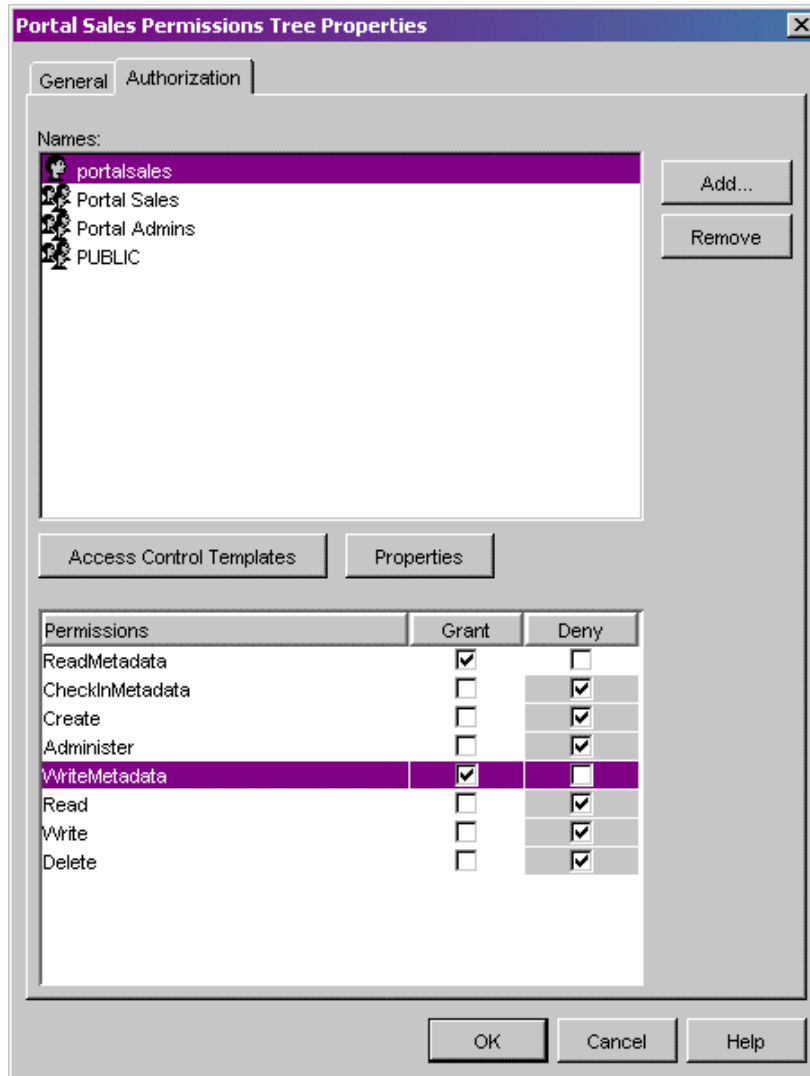
## Verify Permission Tree Folders and Permissions

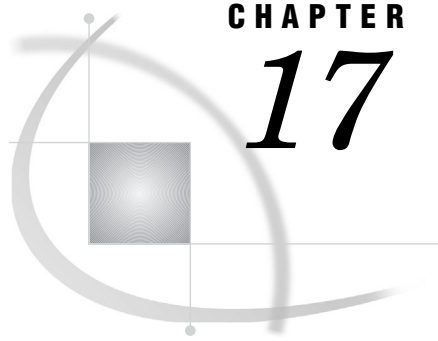
Follow these steps to verify that a permission tree folder has been created for a particular user or group. You can also verify the permissions that have been granted for the resources that are associated with the user or group:

- 1 To verify that a permission tree folder has been created:
  - a Log on to SAS Management Console as the SAS Administrator.
  - b Navigate to **Authorization Manage ► Resource Management ► BIP Service ► Portal Application Tree**.
  - c If the permission tree folder has been created, then the folder will appear in the **Portal Application Tree** folder list. (If the folder does not appear in the list, then you can create the folder by using one of the options described in “How Permission Tree Folders Are Created” on page 234.)
- 2 To verify the associated permissions for the permission tree, do the following:
  - a Select the permission tree folder in the **Portal Application Tree** folder list.
  - b From the main menu, select **File ► Properties**.
  - c In the Properties dialog box, select the **Authorization** tab.
  - d In the **Names** list box, select a user or group. The permissions for that user or 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.)

For information about the permissions, see “Who Can Administer the Portal Web Application” on page 193.

For example, the following display of the **Authorization** tab shows the users and groups that have permissions for the **Portal Sales** permission tree. The **WriteMetadata** permission is directly assigned to the **portalsales** user.





## CHAPTER

## 17

## Adding Content to the Portal

<i>Overview of Adding Content</i>	<b>239</b>
<i>Introduction to Adding Content</i>	<b>239</b>
<i>SAS Application Server Requirements</i>	<b>240</b>
<i>Metadata Requirements</i>	<b>240</b>
<i>Summary of Content That Can Be Added to the Portal</i>	<b>240</b>
<i>Understanding Pages and Page Templates</i>	<b>242</b>
<i>About Pages</i>	<b>242</b>
<i>Who Can Administer Pages</i>	<b>243</b>
<i>Understanding Customized Page Deployment</i>	<b>244</b>
<i>Page Attributes: AVAILABLE, DEFAULT, and STICKY</i>	<b>245</b>
<i>Personal Pages</i>	<b>245</b>
<i>Shared Pages</i>	<b>246</b>
<i>Types of Edits That Can Be Made to a Page</i>	<b>247</b>
<i>Page Templates</i>	<b>247</b>
<i>Overview of Page Templates</i>	<b>247</b>
<i>Main Features of Page Templates</i>	<b>247</b>
<i>The Home Page Template</i>	<b>248</b>
<i>Adding, Editing, and Removing Pages</i>	<b>249</b>
<i>Add and Share a Page</i>	<b>249</b>
<i>Edit a Page</i>	<b>250</b>
<i>Remove a Page from the Portal</i>	<b>250</b>
<i>Adding, Editing, and Removing Page Templates</i>	<b>251</b>
<i>Add a Page Template</i>	<b>251</b>
<i>Edit or Remove a Page That Was Created from a Page Template</i>	<b>256</b>
<i>Edit a Page Template</i>	<b>256</b>
<i>Delete a Page Template from the Portal</i>	<b>257</b>
<i>Understanding Portlets</i>	<b>258</b>
<i>Overview of Portlets</i>	<b>258</b>
<i>Custom-Developed Portlets</i>	<b>259</b>
<i>Portlet Templates (Editable Portlets)</i>	<b>259</b>
<i>Predefined Portlets That Are Provided with the Portal Web Application</i>	<b>261</b>
<i>Main Steps to Add a Portlet</i>	<b>262</b>
<i>Adding WebDAV Graph Portlets</i>	<b>264</b>
<i>Overview of Adding WebDAV Graph Portlets</i>	<b>264</b>
<i>Step 1: Prepare the Data Set That You Want to Graph</i>	<b>265</b>
<i>Step 2: Create an XML File and Add It to WebDAV</i>	<b>265</b>
<i>Step 3: Create and Share a WebDAV Graph Portlet</i>	<b>267</b>
<i>Adding Custom-Developed Portlets</i>	<b>268</b>
<i>Overview of Adding Custom-Developed Portlets</i>	<b>268</b>
<i>Step 1: Design and Code the Portlet</i>	<b>268</b>
<i>Step 2: Deploy the Portlet in the Portal Web Application</i>	<b>268</b>

Step 3: Ensure That the Appropriate Resource Metadata Is Added to the SAS Metadata Repository	269
Step 4: For Remote Portlets Only, Add the Permission Statements for the Portlet to the Required Policy Files	269
Step 5: Implement Authorization for the Portlet	269
Step 6: Add the Portlet to the Portal Web Application	270
Understanding Portlet Deployment	270
Overview of Portlet Deployment	270
Deploying Portlets	270
How Portlet Hot Deployment Works	271
How Local and Remote Portlets Execute	271
Hiding Portlets from Users	272
Overview of Hiding Portlets from Users	272
Associating the Portlet with a Group	272
Hide a Portlet	272
Adding Links	274
Adding Files	275
Overview of Adding Files	275
Step 1: Add the File to the Xythos WebFile Server (WFS)	275
Step 2: Implement Authorization (Access Control) for the File	276
Step 3: Make the File Available to Portal Users	276
Adding Web Applications	276
Overview of Adding Web Applications	276
Step 1: Design and Code the Web Application	277
Step 2: Deploy the Web Application's WAR File in the Servlet Container	278
Step 3: Ensure That the Appropriate User or Group Permission Tree Is Created in the SAS Metadata Repository	278
Step 4: Add the Web Application's Metadata to the SAS Metadata Repository	279
Step 5: Ensure That the Appropriate Resource Metadata Is Added to the SAS Metadata Repository	280
Step 6: Add the Permission Statements for the Web Application to the Required Policy Files	281
Step 7: Implement Authorization for the Web Application	281
Step 8: Make the Web Application Available in the Portal	281
Step 9: Optionally, Update or Remove the Web Application	282
Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java	282
Overview: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java	282
Step 1: Design and Code the Web Application	282
Step 2: Deploy the WAR Files in the Servlet Container	283
Step 3: Ensure that the Appropriate Group Metadata Exists in the SAS Metadata Repository	283
Step 4: Add the Application's Metadata to the SAS Metadata Repository	283
Step 5: Ensure That the Appropriate Resource Metadata Is Added to the SAS Metadata Repository	284
Step 6: Add the Permission Statements for the Web Application to the Required Policy Files	285
Step 7: Implement Authorization (Access Control) for the Web Application	285
Step 8: Make the Web Application Available in the Portal	286
Step 9: Optionally, Update or Remove the Web Application	286
Adding Syndication Channels	286
Overview of Adding Syndication Channels	286
Step 1: Add the Syndication Channel's Permission Statement to the Appropriate Policy File	287
Step 2: Ensure That the Appropriate User or Group Permission Tree Is Created in the SAS Metadata Repository	287
Step 3: Add the Syndication Channel's Metadata to the SAS Metadata Repository	288
Step 4: Implement Authorization for the Syndication Channel	289
Step 5: Make the Syndication Channel Available in the Portal	289



<i>Step 6: Optionally, Update or Remove the Syndication Channel</i>	290
<i>Adding SAS Packages</i>	290
<i>Overview of Adding SAS Packages</i>	290
<i>Step 1: If the Package Is Not Created, Create the Package</i>	291
<i>Step 2: Implement Authorization for the Package</i>	291
<i>Step 3: Make the Package Available in the Portal Web Application</i>	291
<i>Adding SAS Publication Channels</i>	292
<i>Overview of Adding SAS Publication Channels</i>	292
<i>About SAS Publication Channels</i>	292
<i>E-Mail Transport Restriction</i>	292
<i>WebDAV Publication Channel Considerations</i>	292
<i>Step 1 (Optional): If Publishing to an Archive, Add the SAS Publication Channel's Archive Permission Statement to the Appropriate Policy File</i>	293
<i>Step 2: Add the Publication Channel to the SAS Metadata Repository</i>	293
<i>Step 3: Implement Authorization (Access Control) for the SAS Publication Channel</i>	294
<i>Step 4: Make the SAS Publication Channel Available to Portal Users</i>	294
<i>Adding and Administering SAS Stored Processes</i>	294
<i>What Is a Stored Process?</i>	294
<i>How Stored Processes Are Executed from the Portal</i>	295
<i>Characteristics of Non-Streaming Stored Processes</i>	295
<i>Main Tasks for Administering Stored Processes</i>	296
<i>About Alerts</i>	298
<i>Adding SAS Information Maps</i>	299
<i>Overview of Adding SAS Information Maps</i>	299
<i>Step 1: Control Access the SAS Information Map</i>	299
<i>Step 2: Make the SAS Information Map Available in the Portal Web Application</i>	299
<i>Adding SAS Reports</i>	300
<i>Overview of Adding SAS Reports</i>	300
<i>Step 1: Control Access to the SAS Report</i>	300
<i>Step 2: Make the SAS Report Available to Portal Users</i>	301

---

## Overview of Adding Content

---

### Introduction to Adding Content

The portal Web application 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 “Summary of Portal Features and Their Software Requirements” on page 186.

The portal Web application 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

For more details, see “SAS Application Servers That Are Required for SAS Content” on page 363.

---

## Metadata Requirements

All content requires the addition of metadata to the SAS Metadata 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 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 Web application, such as links and pages, the portal Web application administers both content and authorization metadata.
- For WebDAV content, when you add the content to the WebDAV repository, the content metadata becomes available to the portal Web application. 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 240.

---

## 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 Web application. 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 Xythos WebFile Server repository. For more detail about an item, or for instructions on adding the item, see the applicable topic in this chapter (Chapter 17, “Adding Content to the Portal,” on page 237).

**Table 17.1** Summary of Portal Content and Metadata

<b>Content Type</b>	<b>Description</b>	<b>Where to Specify Metadata</b>	
		<b>Content Metadata</b>	<b>Authorization Metadata</b>
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 portal Web application users (who have access) to view a variety of document types in the portal Web application.	Xythos WebFile Server	Xythos WebFile 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 Web application that is a template page which contains portlets.	Portal (create feature), or a SAS program	Portal (share feature), or a SAS program
Page	A Web page in the portal Web application that contains portlets.	Portal (create feature)	Portal (share feature)
Custom-developed portlet	A rectangular display component of the portal Web application in which content and links to content are displayed. Administrators can add custom-developed local or remote portlets to the portal Web application.	Portlet Hot-Deploy Mechanism	Portal (share feature)*
Portlet	A rectangular display component of the portal Web application 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 Web application.	Portal (create feature)	Portal (share feature)
Syndication channel	(SAS Information Delivery Portal only) A channel that provides syndicated, continuously updated Web content.	Portal (create feature), or a SAS program	Portal (share feature), or a SAS program

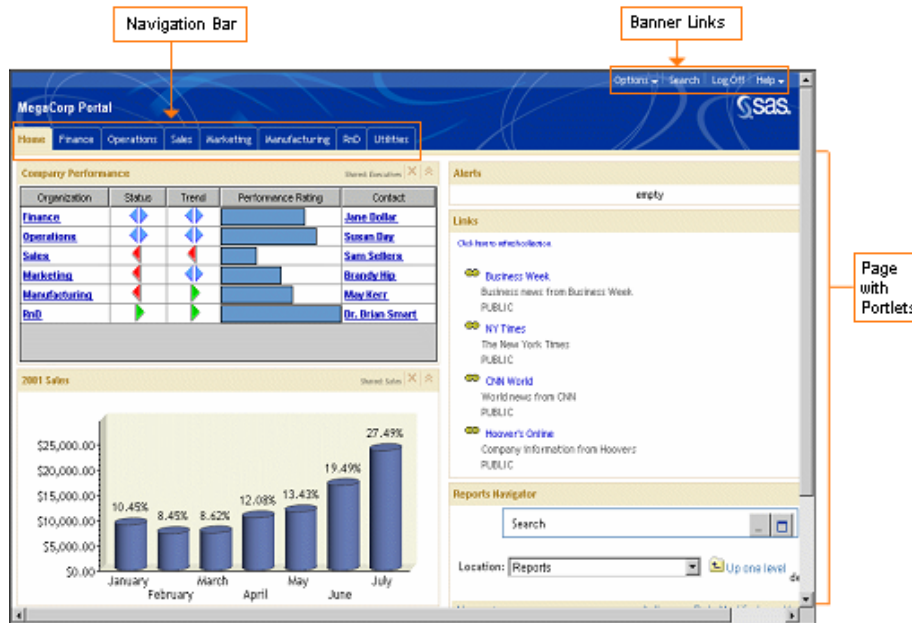
Content Type	Description	Where to Specify Metadata	
		Content Metadata	Authorization Metadata
SAS Publication Channel	(SAS Information Delivery Portal only) A channel created by the SAS Publishing Framework. Publication channels can be used to provide access to archived content published through the SAS Publication Framework.	SAS Management Console	SAS Management Console
Package	(SAS Information Delivery Portal only) A collection of structured and unstructured content that has been published to a publication channel or to Xythos WebFile Server.	The content metadata is part of the metadata for the SAS publication channel or Xythos repository	The authorization metadata is part of the metadata for the SAS publication channel or Xythos 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	(SAS Information Delivery Portal only) Business-oriented view of multidimensional and relational data, which can be used to develop reports. SAS Information Maps are available in the portal Web application if your organization has installed SAS Information Map Studio.	SAS Information Map Studio	SAS Information Map Studio
SAS Report	(SAS Information Delivery Portal only) 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 Web application if the portlet's descriptor file contains settings that enable sharing.

## Understanding Pages and Page Templates

### About Pages

The portal Web application 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 Web application. A page can contain any number of portlets.

A banner spans the top of the portal Web application, 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 Web application. A theme defines the portal's colors, fonts, banner, and graphical elements. (You deploy themes separately from pages. For details, see "Theme Deployment" on page 328.)

Each page in the portal Web application 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 Web application, you can choose to override page ranks by explicitly defining the order of pages.

You will determine how rank values will be 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 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 17.2** Page Administrators

<b>User Type</b>	<b>What the User Can Administer</b>
SAS Web administrator	Can share, unshare, edit, and delete any page in the portal Web application, including a page that someone else has created.
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. See “Configure a Group Content Administrator” on page 224.
SAS Guest User	Can edit and delete pages in the Public Kiosk.  Any page that SAS Guest creates is automatically added to the Public Kiosk. If a page is shared with the PUBLIC group, then SAS Guest can add the page to the Public Kiosk. For more information about the Public Kiosk, see “Administering the Public Kiosk” on page 202.
All portal users	Can create, edit, and delete only personal pages.  The SAS Information Delivery Portal must be installed in order for all portal users to have this capability. If the SAS Information Delivery Portal is not installed, then users can edit portlets on a page, but they cannot add or edit pages.  For more information about personal pages, see “Personal Pages” on page 245.

For more information about the permissions that are granted to these users in SAS metadata, see “Who Can Administer the Portal Web Application” on page 193.

## Understanding Customized Page Deployment

The portal Web application 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 Web application, 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.

Here is a typical scenario for deploying a page:

- 1 create the page
- 2 add portlets to the page
- 3 add content to the portlets
- 4 set up access control for the content
- 5 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. You would 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. Then, you would create pages and share them

with the appropriate group. The portal users who belong to the group would see only the pages that were 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 frequently-visited Internet sites, and change the navigation scheme. (For users to edit pages and personalize their portal views, the Information Delivery Portal must be deployed.)

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 then assume responsibility for creating and sharing pages with the respective group.

---

## Page Attributes: AVAILABLE, DEFAULT, and STICKY

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

There are three attributes that associate pages with portal views:

**Table 17.3** Page Attributes

Attribute	Description
AVAILABLE	The page is added manually to portal views. Users will 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 portal Web application. Users can later remove the page if they don't want it in their views.
STICKY	The page is added automatically to portal views. Users can <i>not</i> remove the page from their portal views.

If a DEFAULT or STICKY page is shared to a group that contains members of the Portal Admins group, then the page will not appear when those members log on to the portal. Each user who is a member of the Portal Admins group will need to add the page manually. The reason for this behavior is that a member of the Portal Admins group is a privileged user who has access to all user and group content. When users log on, the pages for every group they have access to are initialized. This could have a large performance impact when members of the Portal Admins group log on. In general, only a very limited number of users should be in the Portal Admins group.

As you might expect, DEFAULT is the default value for every page that you create in the portal. After you create a page, if you share that page with other users in a particular group, then you can apply any one of the three attributes to the shared page. For more details, see the section “Shared Pages” on page 246.

---

## 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. If the SAS Information Delivery

Portal is installed, then all users who can log on to the portal can create personal pages in their portal views. If the SAS Information Delivery Portal is not installed, then only a portal administrator or group content administrator can create pages.

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 page, except a SAS Web Administrator.
- If the owner of a page has administrator privileges, then the owner can share that page with others. Shared pages are described in the next section.

---

## Shared Pages

When you create a page, if you have administrative permissions, then you can share the page 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 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, then you can specify whether you also want to share the portlets and their contents.

*Note:* When you share a page that contains an Alerts portlet, 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. See “Page Templates” on page 247. △

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 STICKY) that you apply to the page determines how portal users access the page:

- Pages that have a share type of DEFAULT or STICKY are added automatically to portal views for the respective group members. (The page is not automatically added to the portal view for any user who is also a member of the Portal Admins group.)
- For pages that have a share type of AVAILABLE, group members must search for the page before they can see it. If the SAS Information Delivery Portal has been deployed, then 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 Web application. 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 both of the following are true:

- The SAS Information Delivery Portal has been deployed.
- The shared page has an attribute of DEFAULT or AVAILABLE. (If the shared page has an attribute of STICKY, then only the portal administrator or group content administrator can remove the page from a portal view.)

For general information about sharing portal content, see “Sharing Content in the Portal Web Application” on page 226.



---

## Types of Edits That Can Be Made to a Page

If you have the appropriate permissions, then 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 Web application

You edit pages by using the portal **Options** menu. For more information, refer to the online Help that is provided with the portal Web application. See also “Using the Portal Options Menu” on page 210 for an overview of the **Options** menu.

---

## 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 an explicitly-defined group). The page template must have an attribute of either DEFAULT or STICKY. You can create a page template from an existing page in the portal Web application, 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. (The page is not automatically added to the portal view for any user who is also a member of the Portal Admins group.)
- 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 theme templates. These are different entities that use the name “template.” For information about portlet templates, see “Understanding Portlets” on page 258. For information about theme templates, see “Developing Custom Themes” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/themes/index.html](http://support.sas.com/rnd/itech/doc9/portal_dev/themes/index.html). △

### 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:

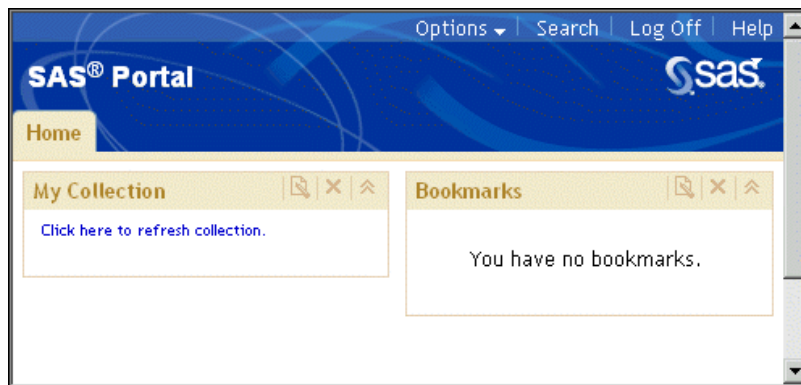
- Page templates enable you to deploy pages without logging on to the portal Web application. Unlike shared pages, the portal does not need to be running in order for you to create a page template and add it to metadata (although you can create a page template while you are logged on to the portal Web application).
- A separate page is created for each user. This means that, if the SAS Information Delivery Portal is installed, 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, regardless of whether the SAS Information Delivery Portal is installed. △
- Page templates can contain portlets that are not shareable. For example, a page template can contain 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 STICKY. 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, then users can remove the page from their portal views.
  - If the page is defined as STICKY, then 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 then create it again. For more information, see “Adding, Editing, and Removing Page Templates” on page 251.

## The Home Page Template

After you install and configure the portal Web application, when you first log on to the portal, you see the following Home page:



The Home page template is created when you install the demonstration metadata, as explained in the `wik_readme.html` file (located in the portal's installation directory). 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 STICKY, and is associated with the PUBLIC group. This means that all users (except a SAS Web Administrator) see the Home

page when they log on to the portal Web application. The SAS Web Administrator can search for and add the page to his or her portal view.

Users cannot remove the Home page from their portal views unless a SAS Web 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. If the SAS Information Delivery Portal has been deployed, then users can also edit the page. See “Types of Edits That Can Be Made to a Page” on page 247.

## Adding, Editing, and Removing Pages

### Add and Share a Page

One of your administrative tasks is to deploy custom views for particular groups of portal users. To accomplish this, you first create the pages, then 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 242.

Here is a summary of the steps required to add and share a page. For complete instructions, refer to the online Help that is provided with the portal Web application:

- 1 Verify that a permission 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 233.
- 2 Log on to the portal Web application as either a SAS Web Administrator or 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. See “Using the Portal Options Menu” on page 210 for an overview of the **Options** menu.
- 4 Add the portlets and the content that are appropriate for the group with which you intend to share the page. For instructions, refer to the online Help that is provided with the portal Web application.
- 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 222.
- 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 **STICKY**. For a description of these attributes, see “Page Attributes: **AVAILABLE**, **DEFAULT**, and **STICKY**” on page 245.

If the page contains portlets that you have permission to share, then 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 Web Application” on page 226.

*Note:* If you have installed the SAS Information Delivery Portal, then 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 245. △

---

## Edit a Page

You edit a page using the portal **Options** menu. For instructions, refer to the online Help that is provided with the portal Web application. See also “Using the Portal Options Menu” on page 210 for an overview of the **Options** menu.

Here are some main points related to editing pages:

- Log on to the portal Web application using a login that is appropriate for the type of page that you want to edit. For example, a group content administrator would log on to 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 243.
- After logging on, to edit a page that someone else created, you might first have to search for the page. Optionally, 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 247.

*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 251. △

---

## Remove a Page from the Portal

You remove a page using the portal **Options** menu. For instructions, refer to the online Help that is provided with the portal Web application.

Here are some of the main points related to removing pages:

- Log on to the portal Web application 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 243.
- You have the following options for removing a page:

**Table 17.4** Options for Removing a Page

Option	Description
Remove the page from your personal portal view.	The page will no longer appear in your personal portal view, but remains in the portal repository so that you can add it later. If the page is shared, this option has no affect on other portal views.
Remove and delete the page from the system permanently.	The page will be deleted from the SAS metadata, and therefore it will not be available to add later. If the page is shared, this option will remove the page from all portal views. If the page contains one or more portlets that you are authorized to delete, then the portal Web application gives you the option to delete those portlets and any of their contents that you are authorized to delete.

- If you have installed the SAS Information Delivery Portal, then 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, then all portal users who can access the page can remove the page from their portal views.
- If a page was shared with a **STICKY** attribute, then 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 251. △

---

## 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 will see the first time they log on to the portal Web application. 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 247.

Before you can create a page template and associate it with a group, you must create a permission tree in SAS metadata for the group. To verify that a permission tree exists, or to create one, see “Managing Portal Permission Trees in Metadata” on page 233.

There are two ways to create a page template and add it to the portal Web application:

- In the portal Web application, create a page template from an existing page.

When you create a page template in the portal Web application, 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, refer to the online Help that is provided with the portal Web application:

- 1 Log on to the portal Web application as either the SAS Web Administrator or the group content administrator for the group.
- 2 Create a page in the portal Web application, and add the contents that are appropriate for the page.
- 3 Use the **Options** menu to convert the page to a template. See “Using the Portal Options Menu” on page 210 for an overview of the **Options** menu.
- 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 Web Application” on page 226.

- Create a page template by running a SAS program.

You can 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 the *SAS-install-dir\Web\Portal2.0.1\OMR* 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 the next time that the user logs on to the portal Web application.

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 `$_SERVICES_OMI_HOST$` property in the `install.properties` file. 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 `$_SERVICES_OMI_PORT$` property in the `install.properties` file (located in the `PortalConfigure` subdirectory of the setup directory).

`metauser="user ID"`

Specify the user ID to use to connect to the SAS Metadata Server; this user ID is typically the SAS Administrator (default, `sasadm`). For Windows users, the user ID is domain or machine name qualified. For example:

```
machine\saswbadm
NTDOMAIN\saswbadm
```

In the above code, `machine` is the name of the local host, and `NTDOMAIN` is the Windows authentication domain.

`metapass="password"`

Specify the password for the `metauser`.

`metarepository="repository";`

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the `$_SERVICES_OMI_REPOSITORY$` property in the `install.properties` file (located in the `PortalConfigure` subdirectory of the setup directory).

`%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 permission 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=Sticky | Default;`

Specify whether the page is Sticky or Default, followed by a semicolon (;).

*Default*

Default user or group pages are automatically added to the portal Web application of the user, or of all users in a group. The users can later remove the page.

*Sticky* Sticky group pages are automatically added to the portal Web application of the user, 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 will have this page rank.

Pages are ordered in the portal Web application 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 (if the SAS Information Delivery Portal is installed).

`data pageTemplate;`

Specifies the SAS dataset 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 dataset. 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 CSV (comma separated values) 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 Web application 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 will identify the portlet on the page. If a portlet already exists with the same *portletName*, the existing portlet is used and a new portlet will not be 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
Welcome template

```

The template name for custom portlets will be "<portletName> template". This field cannot contain a comma.

data properties;

Specify the SAS dataset that defines any additional data properties that are needed by the portlets. This variable enables you to add default starting data to template portlets. Acceptable values for the data properties vary with the portlet that is being referenced. None of the pre-defined portlets use these properties, but some custom portlets might use them.

If there are no additional data properties for any portlets, then 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 dataset 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 ";;;" line describes each property (1 per line) that you want to define. The line is formatted as comma-separated values (CSV), and each column denotes a portion of the property definition. For example, the following lines create 4 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 dataset that defines any collection or bookmark links to add to collection or Bookmarks portlets.

If no links are required for any of the portlets, then 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 dataset 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 (1 per line) that you want to add to Collection or Bookmarks portlets. The line is formatted as comma-separated values (csv), and each column denotes a portion of the link definition. For example, the following lines create 3 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 that will 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
File * Document
Link Document
Package ** ArchiveFile
Page PSPortalPage
Portlet PSPortlet
Publication Channel ITChannel
Information Map Transformation
SAS Report Transformation
SAS Stored Process ClassifierMap
Syndication Channel Document
```

\* The file can not be stored in WebDAV.

\*\* A publication package must be stored in SAS metadata in order to be referenced in the page template. The package can not be stored in WebDAV.

**searchStr**

Specify an *XMLSelect* string that will uniquely locate 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 will continue executing, ignore this entry, and print 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 17.5** Rules for Editing or Removing Pages

Action	Instructions and Rules
Edit a page	<p>Edit a page by using the portal <i>Options</i> menu. For more information, refer to the online Help that is provided with the portal Web application. See also “Using the Portal Options Menu” on page 210 for an overview of the <i>Options</i> menu.</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>If the SAS Information Delivery Portal has been installed, then all portal users can edit a page that has been added to their portal views. If the SAS Information Delivery Portal is not installed, then users can edit only the portlets on the page.</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 247.</p>
Remove a page	<p>If the page has an attribute of <b>DEFAULT</b>, then you can remove the page from your portal view by using the portal <i>Options</i> menu. When you remove a page from your portal view, you do not affect the pages that have been added to other portal views. If the SAS Information Delivery Portal has been installed, then all portal users can remove a <b>DEFAULT</b> page that has been added to their portal views.</p> <p>If the page has an attribute of <b>STICKY</b>, then you cannot remove the page 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 257.</p>

---

## Edit a Page Template

After you have created a page template, you cannot edit it. However, you can delete the page template, and then create a new page template that contains the revised content.

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 Web application. After you create a page template, if you decide that it needs changes, then you can delete the page template along with all pages that might have been created in other users’ portal views. Then, 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 257.

---

## 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, then 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 Web application.

When you delete a page template in the portal Web application, 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, refer to the online Help that is provided with the portal Web application:

- 1 Log on to the portal Web application as either the SAS Web Administrator or the group content administrator for the group with which the page template is shared.
- 2 In the portal Web application, search for and then 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.

- Delete a page template by running a SAS program.

When you use this method to delete a page template, 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 then create a new template, the new template will 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 then review the changes that appear in the new page, remove the original page from their portal views, and re-personalize the new page.

Follow these steps:

- 1 Modify the SAS program **RemovePageTemplate.sas**, which is located in the *SAS-install-dir/Web/Portal2.0.1/OMR* 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 `$SERVICES_OMI_HOST$` property in the **install.properties** file. 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 `$SERVICES_OMI_PORT$` property in

the *install.properties* file (located in the *PortalConfigure* subdirectory of the setup directory).

`metauser="user ID"`

Specify the user ID to use to connect to the SAS Metadata Server; this user ID is typically the SAS Administrator (default, *sasadm*). For Windows users, the user ID should be qualified with either the domain name or the machine name. For example:

```
<machine>\saswbadm
<NTDOMAIN>\saswbadm
```

`metapass="password"`

Specify the password for the *metauser*.

`metarepository="repository";`

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the `$SERVICES_OMI_REPOSITORY$` property in the *install.properties* file (located in the *PortalConfigure* subdirectory of the setup 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 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.)

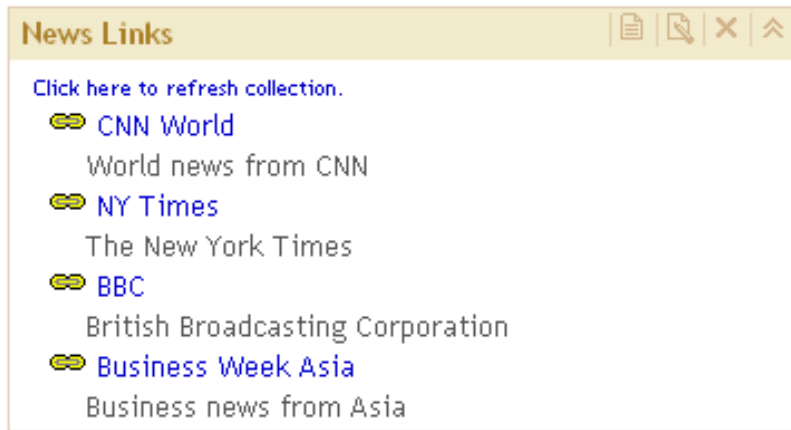
---

## Understanding Portlets

---

### Overview of Portlets

Portlets are the rectangular display components of the portal Web application, 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:* If you have installed the SAS Information Delivery Portal and the SAS Web Infrastructure Kit, then all users can add portlets to the portal Web application. If you have installed only the SAS Web Infrastructure Kit, then only a SAS Web Administrator or a group content administrator can add portlets to the portal Web application. △

For instructions on adding any of the portlets that are described here, refer to the online Help that is provided with the portal Web application (see the topic “About Portlets” in the Help). See “Main Steps to Add a Portlet” on page 262 for a process overview of creating and sharing portlets.

The portal Web application supports the following basic types of portlets:

- custom-developed portlets
- portlet templates (editable portlets)
- predefined portlets that are provided with the portal Web application

The following sections describe these different types of portlets.

---

## Custom-Developed Portlets

You can create custom portlets using the portlet development kit. For more information, see “Developing Custom Portlets” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/portlets/dg\\_portlets.html](http://support.sas.com/rnd/itech/doc9/portal_dev/portlets/dg_portlets.html).

---

## 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 portal Web application:

**Table 17.6** Portlet Templates

Template Name	Description
Collection Portlet	Contains links to content items (which can include Web links) in the portal Web application.
Information Map Viewer Portlet	<p>Displays information map views that users create by using bookmarks in the Visual Data Explorer component.</p> <p>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, refer to the online Help that is provided with the portal Web application (see the topics "About Data Explorations and Bookmarks" and "About Information Map Viewer Portlets" in the Help).</p>
Quiesce Portlet	Prepares the portal Web application to be shut down by preventing new users from logging on. This portlet type is only available to SAS Web administrators. For more information, see "Using the Quiesce Portlet to Bring Down the Portal" on page 213.
SAS Business Intelligence Dashboard Portlet	Displays one or more graphical indicators. If your installation includes SAS Business Intelligence Dashboard, 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. For administration of SAS Business Intelligence Dashboard, see Chapter 18, "Administering SAS Business Intelligence Dashboard," on page 303.
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 IFRAME option, which 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>

Template Name	Description
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"> <li>□ If the HTML fragment contains style definitions with class names that also occur in the portal theme, then the appearance of the portal could be affected when the portlet is displayed.</li> <li>□ If the HTML fragment contains JavaScript, use namespaces for the JavaScript functions to prevent conflicts with portal processing.</li> <li>□ 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.</li> </ul> <p>For more information, refer to 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 264.</p>
WebDAV Repository Navigator *	<p>Enables a user to explore the contents of a WebDAV repository.</p>

\*This portlet type is available only if you have configured a Xythos WFS WebDAV repository for use with the portal Web application. When a user adds one of these portlets, the user can view only Xythos WFS content that the user is authorized to access.

---

## Predefined Portlets That Are Provided with the Portal Web Application

A predefined portlet is automatically deployed when you install the portal Web application. 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 Web application.

Here is a list of predefined portlets that are deployed when you install the portal Web application.

**Table 17.7** Predefined Portlets

<b>Portlet Name</b>	<b>Description</b>
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 using the Search tool and 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 is subscribed to and enables the user 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 using the user's locale (language and country) preference. The Welcome Portlet is not interactive.

\*This portlet is available only if you have installed the SAS Information Delivery Portal.

\*\*This portlet is available only if you have configured a Xythos WFS WebDAV repository for use with the portal Web application.

---

## Main Steps to Add a Portlet

One of your administrative tasks 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 258.

Here is a summary of the steps that are required to add and share a portlet. For complete instructions, refer to the online Help that is provided with the portal Web application:

- 1 Verify that a permission tree folder exists for the group with which you want to share the portlet. If necessary, create a permission tree folder. See “Managing Portal Permission Trees in Metadata” on page 233.
- 2 Log on to the portal Web application as either a SAS Web Administrator or 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), then you must enable the portal Web application to access the URL. To enable this access, add permission statements for the portlet to the Java policy file for the portal Web application's servlet container.

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 permission statement to the policy file, depending on the URL type, do one of the following:

- For the file protocol, add a **java.io.FilePermission** that grants access to all of the files that make up the HTML fragment; these files include the HTML file and any resources it uses (such as images, CSS, 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** 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 “Adding Permissions to Policy Files” on page 45. For more information about URL display portlets, see “Understanding Portlets” on page 258. Refer also to the online Help that is provided with the portal Web application.

- 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 222.
- 7 Make the portlet available in the portal Web application.

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, then 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 (the SAS Information Delivery Portal must be installed in order for users to do this).

*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 display, regardless of the percentage that is associated with the portlet's column. In addition, a portlet's size will vary based on the content it contains. If a particular portlet cannot fit within a column, then the percentage that you specified for the column will be overridden by the width that is actually required in order to display the portlet.  $\Delta$

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 (the SAS Information Delivery Portal must be installed in order for users to add pages).

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, then the portlet is removed from all portal views.

For complete instructions about creating, sharing, editing, or deleting a portlet, refer to the online Help that is provided with the portal Web application. For information about sharing portal content in general, see “Sharing Content in the Portal Web Application” on page 226.

*Note:* If you have installed the SAS Web Infrastructure Kit and the SAS Information Delivery Portal, then all users can create portlets and add portlets to their pages by using the portal **Options** menu. If you installed only the SAS Web Infrastructure Kit, then only the SAS Web Administrator and group content administrators can use the portal **Options** menu to create personal content. In either case, only users who are authorized as an administrator for a group can share a portlet with the group, or can edit a shared portlet.  $\triangle$

---

## 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. If the SAS Information Delivery Portal has been installed, then all portal users can create their own WebDAV graph portlets.

Here is a sample WebDAV graph portlet:



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 done using the SAS programming language), the easier it is to display a graph in the portal Web application. For storage and performance reasons, it is preferable to presummarize a large dataset (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 portal Web application provides a macro that creates a properly formatted XML file from your SAS data set, and then 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.

Follow these steps to create and add the XML file:

- 1 Create a SAS program that invokes the **publishToWebDAV** macro. The macro file is located in the *SAS-install-dir\Web\Porta12.0.1\Samples\WebDAVGraph* 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.

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, then 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:

*sasdsn* Specify the full name of the SAS data set, in the format *libref.SAS-data-set*, and enclosed in single quotation marks. For example:

```
'sashelp.class'
```

*davloc* Specify the URL for the WebDAV location, enclosed in single quotation marks. For example:

```
'http://<WebDAVHost>:8300/sasdav/users/sasdemo/graph1'
```

In this example, the macro creates a **graph1** directory under **sasdav/users/sasdemo** on the WebDAV server, and then creates an XML file named **data.xml** in that directory.

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, then you will overwrite the existing XML file.

*userid* 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, then the user ID should be qualified with either the domain name or the machine name. For example:

```
<machine or Windows domain>\sasdemo
```

The user ID that you specify must be authorized to write to the WebDAV location that is specified for the *davloc* parameter. In the example, *sasdemo* has write permissions for the **sasdav/users/sasdemo** location on the WebDAV server.

*passwd* 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.

Use SAS proprietary 32-bit encryption to encrypt passwords. For example, to encrypt a password of *SASDemo1*, you would submit this code in the SAS Program Editor:

```
proc pwencode in='SASDemo1' method=sasenc;
run;
```

The encrypted password is written to your SAS log. When you use *method=sasenc*, the first part of the password is *{sasenc}* .

Here is a sample SAS program that includes and then invokes the **publishToWebDAV** macro:

```
%include 'publishToWebDAV.sas';
%publishToWebDAV('sashelp.class',
'http://localhost:8300/sasdav/users/sasdemo/graph1/',
'sasdemo', '{sasenc}E19F24391F7B576C45200E543F0B37B4');
```

---

## Step 3: Create and Share a WebDAV Graph Portlet

After you have successfully completed Step 2, then 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, refer to the online Help that is provided with the portal Web application:

- 1 Verify that a permission tree folder exists for the group with which you want to share the portlet. If necessary, create a permission tree folder. For more information, see “Managing Portal Permission Trees in Metadata” on page 233.
- 2 Log on to the portal Web application as either a SAS Web Administrator or 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 (optionally) 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.
  - Optionally, a statistic that is to be calculated based on the response variable (the variable that is used for the vertical axis). If you don't choose a statistic from the list that is provided, *Sum* is used as the default statistic.
  - Optionally, 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, then enter the path for the link. The target file can be stored in SAS metadata or in the portal's WebDAV repository.

Here are two examples of paths for a link. The first example illustrates a WebDAV location, and the second example illustrates a location in metadata:

```
sasdav/Sales/ToysDivisionMarketingPlan.doc  
Foundation/Samples/Stored Processes/Sample: Hello World
```

*Note:* To determine the path, you can search for the file in the portal Web application (to find a WebDAV file, search on the Files category). The path will be displayed in the search results.

If you share the WebDAV graph portlet with a group of users, then make sure that the group has permissions to access the target file. △

For complete descriptions of the properties listed above, refer to the online Help that is provided with the portal Web application (see the topic “Edit the Contents of a WebDAV Graph Portlet”).

- 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 installed both the SAS Web Infrastructure Kit and the SAS Information Delivery Portal, and you have added XML data files to the WebDAV server, then all portal users can create WebDAV graph portlets and add those portlets to their pages by using the portal **Options** menu. If you installed only the SAS Web Infrastructure Kit, then only the SAS Web Administrator and 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 can edit a shared portlet.

---

## Adding Custom-Developed Portlets

---

### Overview of Adding Custom-Developed Portlets

A portlet is a rectangular display component of the portal Web application in which content and links to content are displayed. You can develop a custom portlet and add it to the portal Web application. Your custom-developed portlet can be either a local portlet (contained in a PAR file) or a remote portlet (contained in a PAR file and a WAR file).

To design and develop a custom local or remote portlet for deployment in the Web application, 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. The SAS Web Infrastructure Kit: Developer’s Guide provides guidance for developing custom portlets.

The following sections describe the steps for developing a custom portlet and adding it to the Web application.

---

### Step 1: Design and Code the Portlet

For details about designing and coding the portlet, see “Developing Custom Portlets” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/portlets/dg\\_portlets.html](http://support.sas.com/rnd/itech/doc9/portal_dev/portlets/dg_portlets.html).

---

### Step 2: Deploy the Portlet in the Portal Web Application

To enable deployment of the portlet in the portal Web application, move or copy the portlet’s PAR file to the portlet deployment directory. The portal Web application automatically adds the portlet to the SAS Metadata Repository. For example, on a Windows system you might copy **portlet.par** to the **C:\Program Files\SAS\Web\Portal2.0.1\DeployedPortlets** directory. To verify the location of

your portlet deployment directory, see the \$PORTLET\_DEPLOY\_DIR\$ value in the `install.properties` file.

If the portlet is a remote portlet, you must also do the following:

- 1 Deploy the associated WAR file in the servlet container.
- 2 Add permissions to the SAS Services Application's policy files. For details, see "Access Permissions for Custom Portlets and Web Applications" on page 53.

*Related Topics:*

- For details about how the portal Web application deploys the portlets into the Web application and adds the portlet metadata to the Web application's metadata repository, see "Understanding Portlet Deployment" on page 270.
- For details about creating the PAR file, see "Creating a PAR File for Deployment in Your Application" in the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/tasks/dg\\_portlet\\_parfile.html](http://support.sas.com/rnd/itech/doc9/portal_dev/tasks/dg_portlet_parfile.html).
- For information about the WAR file, see "Creating a Remote Portlet" in the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/use\\_cases/dg\\_portlet\\_remote.html](http://support.sas.com/rnd/itech/doc9/portal_dev/use_cases/dg_portlet_remote.html).

---

### 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 portal Web applications content and security features, you must ensure that the appropriate metadata for each resource has been added to the portal Web application'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 the appropriate topic in this chapter (Chapter 17, "Adding Content to the Portal," on page 237).

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 servlet container'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 portal Web application and SAS Services Application codebases to the required policy file. For details, see "Access Permissions for Custom Portlets and Web Applications" on page 53.

---

### Step 5: Implement Authorization for the Portlet

When a portlet is developed, authorization metadata that specifies which users or groups can access the portlet can be included in the descriptor file for the portlet. The

descriptor file can also contain an attribute that enables authorized portal users to share the portlet by using the portal Web application's share feature.

For information about using the portlet deployment descriptor file to specify which users or groups are authorized to access the portlet, see “Creating a Deployment Descriptor” in the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/tasks/dg\\_portlet\\_descr.html](http://support.sas.com/rnd/itech/doc9/portal_dev/tasks/dg_portlet_descr.html).

*Note:* When you specify a user for portlet access, the user is granted ReadMetadata and WriteMetadata permissions to enable the user to view and edit the portlet. When you specify a group for portlet access the group is granted ReadMetadata permission to enable the group members to view the content. △

---

## Step 6: Add the Portlet to the Portal Web Application

To add your custom-developed portlet to the portal Web application, see “Main Steps to Add a Portlet” on page 262.

---

# Understanding Portlet Deployment

---

## Overview of Portlet Deployment

The portal Web application 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 Web application automatically deploys the portlets via a hot deployment mechanism that runs when the portal Web application's servlet container starts.
- state management of portlets. The portal Web application manages portlet state and keeps track of the portlet context.
- routing of user requests. The portal Web application 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 Web application, see “How Local and Remote Portlets Execute” on page 271.

For instructions on adding custom-developed portlets to the portal Web application, see “Adding Custom-Developed Portlets” on page 268.

---

## Deploying Portlets

To deploy portlets in the portal Web application, copy the PAR file to the portlet deployment directory. For instructions, see “Step 2: Deploy the Portlet in the Portal Web Application” on page 268. When the servlet container starts, the portal Web application deploys the portlets through the portal Web application's hot deployment mechanism. The portal Web application then handles portlets as follows:

- Deploying additional portlets: If you add a portlet and its resources to the servlet container while the portal Web application is running, the portal Web application automatically deploys the new portlet into the portal Web application.

*Note:* The portal Web application makes one attempt to deploy the PAR file. If the hot deployment is not successful, the portal Web application will not attempt to deploy the PAR file again. △



- Updating or removing portlets: If you update or remove a portlet and its resources in the servlet container, the portal Web application does not automatically update or remove the portlet from the portal Web application. To update or remove a portlet, you must stop and restart the servlet container. The portal Web application will then check the portlet deployment directory and update or remove the appropriate portlets from the portal Web application.

---

## How Portlet Hot Deployment Works

Although the portal Web application automatically deploys portlets when the servlet container starts, it is helpful to understand how this deployment works. Portlet deployment works as follows:

- 1 The portal Web application starts a **PortletDeployer** thread.
- 2 For each PAR file in the hot deployment directory, the **PortletDeployer** thread does the following:
  - a opens the PAR file, and locates the deployment descriptor file called **portlet.xml**.
  - b parses the **portlet.xml** file and determines the name of the portlet resource directory. The portlet resource directory is the portlet path followed by the portlet name.
  - c creates metadata entries if the portlet is being deployed for the first time. If localized values for title and description are provided, then the localized values are extracted from the appropriate **portletDisplayResources.properties** file.
  - d copies the portlet resources into the Web context under the **/portlet** folder. The portlet resources include JSPs, images, and other non-Java class files.
  - e registers the portlet actions with the PortletRegistry.

---

## 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 the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/index.html](http://support.sas.com/rnd/itech/doc9/portal_dev/index.html).

A *local portlet* is deployed inside the portal Web application and executes inside the portal's servlet container. Because a local portlet executes in the portal Web application's servlet container, it consumes the computing resources (for example, CPU, memory, and disk storage) of the server machine on which the portal Web application's servlet container 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 Web application.

A *remote portlet* might not execute within the same servlet container and Web application as the portal Web application. 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 Web application's servlet container 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 the *SAS Web Infrastructure Kit: Developer's Guide*.

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 would make a portlet unavailable in order to ensure that users don't 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, 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 258.

---

### 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 the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/tasks/dg\\_portlet\\_descr.html](http://support.sas.com/rnd/itech/doc9/portal_dev/tasks/dg_portlet_descr.html).

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 270.

- 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 224.

---

### Hide a Portlet

To deny ReadMetadata permissions for a portlet template, follow these steps:

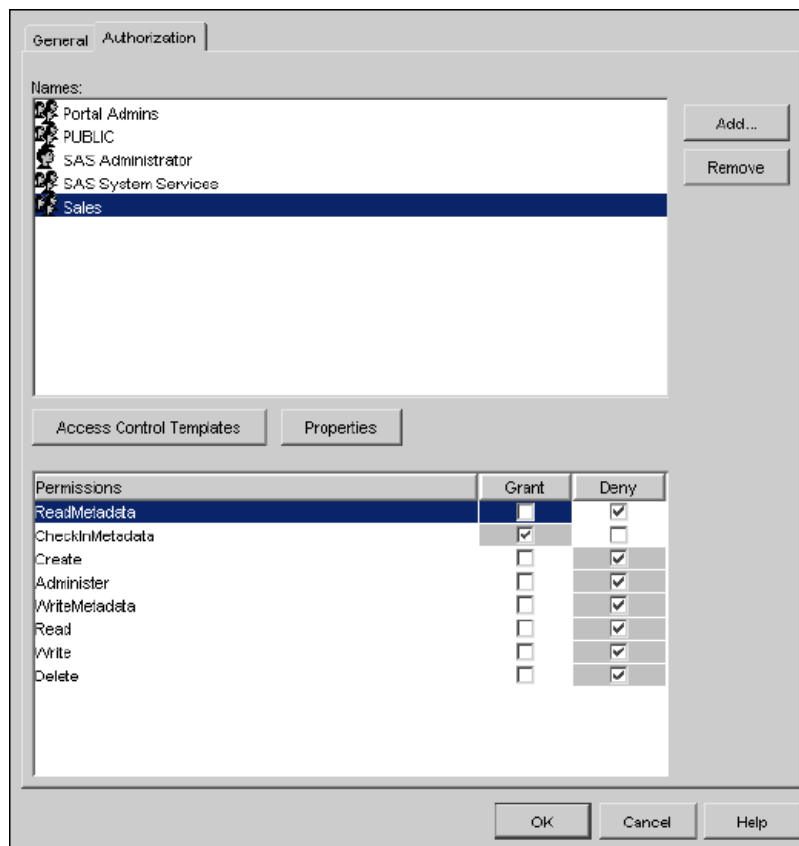
- 1 Log on to SAS Management Console as the SAS Administrator.
- 2 Navigate to the portlet template in the following location in metadata:  
**Environment Management ► Authorization Manager ► Resource Management ► By Type ► Prototype**
- 3 Select the template for the portlet that you want to hide.
- 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 use the Add Users and/or Groups dialog box to add the user or group. When you return to the **Authorization** tab, make sure that the appropriate user or group is selected in the **Names** list box.
- 7 To modify the permissions for the selected user, in the permissions list row for the ReadMetadata permission, select the **Deny** check box.

*Note:* Ensure that the permission is directly assigned, instead of inherited. 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. △

- 8 When you are done, click **OK**.

The following display of the *Authorization* tab shows the users and groups who have permissions for a portlet template. The ReadMetadata permission is directly denied to a group named Sales.



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 done with development, repeat these steps, but grant ReadMetadata permission to the group or users.

## 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 then share the links with groups of users. Users who can access the links can then include the links in collection portlets and display them on pages in the portal Web application. You can also add links to portlets that you share with users.

To create and share a link:

- 1 Verify that a permission tree folder exists for the group with which you want to share the link. If necessary, create a permission tree folder. See “Managing Portal Permission Trees in Metadata” on page 233.
- 2 Log on to the portal Web application as either a SAS Web Administrator or 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, then you can edit the link in order to share it.

*Note:* For instructions on creating and sharing a link, refer to the online Help that is provided with the portal Web application (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, then you might want to implement authorization on the report. For general information about authorization, see “Understanding Portal Authorization” on page 222.

*Note:* A link’s URL might target an HTTP document that is independent of the portal Web application or outside of the portal environment. The portal Web application does not secure the physical document for this type of link. However, you can secure the document through Web server security. Consult the documentation for your servlet container or Web server. △

- 5 Make the link available in the portal Web application by sharing the link. For general information about sharing content, see “Sharing Content in the Portal Web Application” on page 226.

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 Web application, including these options:

- You can edit a collection portlet in order to add the link to the portlet. You can then share the portlet with a group, including the PUBLIC group. If the SAS Information Delivery Portal is installed, then 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 that they log on, or group members can search for and add the page to their portal views (the SAS Information Delivery Portal must be installed in order for users to add pages).

*Note:* If you logged on as a SAS Web Administrator, then you can edit any portlet or page in the portal Web application. If you logged on as a group content administrator, then 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 then 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 on creating, sharing, editing, or deleting a link, refer to the online Help that is provided with the portal Web application. For general information about sharing portal content, see “Sharing Content in the Portal Web Application” on page 226.

*Note:* All portal users can create and add links to their collection portlets. Only users who are authorized as an administrator for a group can share a link with the group, or can edit a shared link. △

---

## Adding Files

---

### Overview of Adding Files

If you have installed the SAS Information Delivery Portal and if your organization’s portal installation includes Xythos WebFile Server (WFS) support, then users can add files to the portal Web application.

A file can be a file of any type. You must add files to the Xythos WFS repository in order to make them available in the portal Web application. Files enable portal users (who have access) to view a variety of document types in the portal Web application. 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 Xythos WebFile Server (WFS)

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 Xythos WFS repository for a user’s personal access, complete these steps:
  - 1 Log on to the portal Web application as the user in order to create the user’s personal repository directory on the Xythos WFS server.
  - 2 Use the administration tool provided with Xythos WFS to locate the appropriate user folder for the content.
  - 3 Use an administration tool, such as Microsoft Web Folders, to add content to the appropriate folder.
- To add the file to the Xythos WFS WebDAV repository for group access, complete these steps:
  - 1 Determine which SAS group(s) will access the content.
  - 2 Use the administration tool that is provided with Xythos WFS to create the appropriate group folder(s) for the content.

- 3 Use an administration tool, such as Microsoft Web Folders, to add content to the appropriate folder(s).

*Note:* If you have installed the SAS Information Delivery Portal, then you might already have group folders set up for SAS Reports that are stored on the Xythos WFS server.  $\Delta$

---

## Step 2: Implement Authorization (Access Control) for the File

Take any necessary steps to control access to the file by using the access control tools that are provided with Xythos WFS. See “Implementing Authorization for the Xythos WebFile Server” on page 231.

---

## 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 then 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.

Users can also publish the file to a SAS Publication Channel and then either add the SAS Publication Channel or SAS Package to a collection portlet, or add the Publication Channel Subscriptions Portlet to their portal Web application.

For more information, refer to the portal’s online Help.

---

# Adding Web Applications

---

## Overview of Adding Web Applications

To design and develop a custom Web application for access from the portal, you should have a working knowledge of JavaServer Pages (JSPs), Java servlets, and the Java programming language. A Web application can be accessed both from within the portal Web application and from outside the portal Web application. To implement a Web application in the portal Web application, do one of the following:

- Implement a remote portlet and a corresponding Web application: A remote portlet looks like any other portlet, but it calls a remote Web application. The remote Web application returns an HTML fragment to the portal Web application to be displayed within the portlet’s borders. This approach is useful when you want to incorporate a portion of the output from your application into the portal. To add a remote portlet and corresponding Web application to the portal Web application, see “Adding Custom-Developed Portlets” on page 268 .
- Implement a stand-alone application that is invoked from the portal Web application but executed remotely: The Web application returns a complete HTML page, which is displayed in a separate browser window. This approach is useful when you want to enable portal Web application users to invoke your application from the portal, but the application output needs to appear separately. To add a stand-alone Web application, follow the instructions in this topic.

For most Web applications, you will typically implement single sign-on with the portal Web application. *Single sign-on* is an authentication model that enables users to access a variety of Web applications without being repeatedly prompted for their user

IDs and passwords. For a general overview of single sign-on, see “Understanding Single Sign-On” on page 24.

*Note:* This documentation assumes that you will implement single sign-on and includes the required steps in all instructions. △

For examples that show how to add SAS Web applications to the portal Web application, see “Examples: Adding SAS Web Report Studio and SAS Web OLAP Viewer for Java” on page 282.

In order for Web applications to participate in a single sign-on configuration, the applications must share a common authentication provider and user context. The user context varies between SAS Metadata Server authentication and trusted Web authentication. Therefore, prior to adding Web applications to the portal environment, you should decide which authentication provider to use for your portal deployment. For more information about authentication providers, see “Choosing an Authentication Provider” on page 20.

If you use the servlet container to authenticate users, then you will configure a Web user role for the portal and related Web applications. The Web user role that you configure establishes the authority constraint and signifies that the servlet container is required to authenticate and authorize users. This authorization is for the application only. SAS content items (information maps, reports, OLAP cubes, and so on) within the application are still authorized through the SAS Metadata Server.

Each Web application that you create and add to the portal must participate in the Web realm that is associated with the user role that you configured. You must configure the Web application’s configuration XML file for the Web realm. This configuration tells the Web server to authenticate users who try to connect to a Web application within that realm. The realm can be defined in a database, LDAP directory, flat file, or some other format that the servlet container supports. If the servlet container is using LDAP for authentication, then the realm would be defined in an LDAP directory.

If you use an HTTP server for authentication, then you will not configure an XML file. The HTTP server authenticates the user and then passes the authenticated user name and password to the servlet container.

For more information about configuring trusted Web authentication, see “Changing to Trusted Web Authentication” on page 32.

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 portal users. For information about developing and integrating Web applications with the portal Web application, see “Integrating Other Web Applications With the Portal” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/webapps/dg\\_webapps.html](http://support.sas.com/rnd/itech/doc9/portal_dev/webapps/dg_webapps.html).

Single sign-on implementation varies with the type of authentication that you are using:

- SAS Metadata Server Authentication

When you use SAS Metadata Server authentication, your custom applications must use the SAS Foundation Services API to manage user and session information. The remote services are shared among all Web applications that participate in the single sign-on configuration. Both the remote and local SAS Foundation Services are deployed by the SAS Services Application. You will use the local services API to create a session, authenticate users, and log messages. You will use the remote services to obtain the user context when your application is launched from the SAS Information Delivery Portal. For a general overview of the authentication process, see “Implementation That Uses Metadata Server

Authentication” on page 27. For an example of a Web application that uses the SAS Foundation Services, see “Sample: Web Application (HelloUserWikExample)” in the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/samples/webapp/dg\\_sample\\_webapp.html](http://support.sas.com/rnd/itech/doc9/portal_dev/samples/webapp/dg_sample_webapp.html).

*Note:* If you do not intend to implement single sign-on with a Web application, then the Web application is not required to use the local or remote foundation services. △

□ Trusted Web Authentication

When you use trusted Web authentication, the J2EE application server, servlet container, or Web server is responsible for authenticating users. Authentication can occur using one of the standard Web authentication mechanisms (basic, digest, and so on). The credentials are submitted in the form of header name-value pairs on each HTTP request. Your application must be able to access the remote user identity in the HTTP request. Typically, the application uses the container's API call `request.getRemoteUser()` to obtain a user identity. There are multiple ways to implement single sign-on, and you should carefully choose the implementation that is best for your environment. For complete details, refer to the documentation that is provided by your servlet container or Web server.

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. To see which SAS applications support single sign-on, see “Which SAS Web Applications Support Single Sign-On?” on page 25.

## Step 2: Deploy the Web Application's WAR File in the Servlet Container

After you code the Web application, you will then deploy your Web application into the servlet container.

The "Deploy Web Application Files into the Servlet Container" section of the `wik_readme.html` file contains suggestions for deploying WAR files. For complete instructions, consult the documentation that is provided for your servlet container.

## Step 3: Ensure That the Appropriate User or Group Permission Tree Is Created in the SAS Metadata Repository

Before you can define a Web application and share it with a group, you must create a permission tree in SAS metadata for the group. To verify that a permission tree exists, or to create one, see “Managing Portal Permission Trees in Metadata” on page 233.

After you add the Web application 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 metadata by running a SAS program, then you can associate the Web application with a user instead of a group. The user must have a permission tree in metadata. To create a permission tree for a user, log on to the portal Web application 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 Web application as the SAS Web Administrator in order to share the Web application with a group (you might need to search for the Web



application first). If the user is a group content administrator, then the user can share the Web application with the respective group.

---

## Step 4: Add the Web 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 portal Web application. When you create a Web application in the portal Web application, the portal adds the Web application's metadata to the metadata repository. Here are the high-level steps:
  - 1 Log on to the portal Web application as either the SAS Web Administrator or 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 Web Application" on page 226.

*Note:* For complete instructions on creating and sharing Web applications, refer to the online Help that is provided with the portal Web application (see the "Applications" section of the Help). △
- Create the Web application by running a SAS program. You can edit and run a SAS program that creates a Web application and adds the application's metadata to the SAS metadata repository. Follow these steps:
  - 1 Modify the SAS program **LoadWebApplicationExample.sas**, which is located in the *SAS-install-dir/Web/Portal2.0.1/OMR* 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 the \$SERVICES\_OMI\_HOST\$ property in the **install.properties** file (located in the **PortalConfigure** subdirectory of the portal installation 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 \$SERVICES\_OMI\_PORT\$ property in the **install.properties** file.

**metauser="user ID"**

Specify the user ID to use to connect to the SAS Metadata Server; this user ID is typically the SAS Administrator (default, sasadm). For Windows users, the user ID is domain or machine name qualified. For example:

```
<machine>\saswbadm
<NTDOMAIN>\saswbadm
```

**metapass**="password"

Specify the password for the metauser.

**metarepository**="repository";

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the \$SERVICES\_OMI\_REPOSITORY\$ property in the **install.properties** file.

**%let groupOrUserName**=SAS User or Group;

Specify the SAS group or user that you want to add the data to, followed by a semicolon (;). The name you specify must be defined in metadata and have an associated permission tree. Specify the user or group name as it appears in metadata.

**%let identityType**=IdentityGroup | Person;

Specify the type of the identity, followed by a semicolon (;). The identity type indicates whether the value that you provided for the groupOrUserName variable refers to a group or to a user.

**%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://host/SASWebReportStudio/logonFromPortal.do;
```

If your Web application has request parameters, then you must encode the parameters as HTML markup, add the parameters to the webappURI value, and enclose the entire string in single quotes ('). For example, a URL that takes the form:

```
http://host/webapp?param1=value1&param2=value2
```

would be entered in **LoadWebApplicationExample.sas** as:

```
%let webappURI='http://host/webapp?param1=value1&param2=value2';
```

Note that **&** is replaced with *&amp;*, and the entire URL string is enclosed in single quotes.

---

## Step 5: 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, then you can skip this step.

To enable the Web application to leverage SAS content and security features, you must ensure that the appropriate metadata for each resource has been added to the 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 information about content metadata, see the metadata addition steps in the appropriate content section of this chapter (Chapter 17, “Adding Content to the Portal,” on page 237).

In addition, when you add SAS publication channels, syndication channels, and servers, you must enable your Web application to access the content by specifying the appropriate permissions in your servlet container’s policy file.

*Note:* Although the metadata for Web application sources must be added to the SAS Metadata Repository, it is not necessary for these data sources to be surfaced in Web applications. △

## Step 6: 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 Portal, and SAS Services codebases to the required policy file. For details, see “Access Permissions for Custom Portlets and Web Applications” on page 53.

If the Web application uses the SAS Foundation Services API, then you must add permissions to the SAS Services Application’s policy file. For information about permission statements that are required for the SAS Services Application’s policy file, see “Modifying the Java Policy File” on page 47.

## Step 7: Implement Authorization for the Web Application

If you create the Web application by running a SAS program, then access is limited to the user or group that you specify in `LoadWebApplicationExample.sas`.

If you create the Web application in the portal, then you can share the application with a group of users. Only the users in that group can access the application.

In addition, if you configured trusted Web authentication, then 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 only controlled 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 8: 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 portal Web application:

- You can edit a collection portlet in order to add the Web application to the portlet. You can then share the portlet with a group, including the PUBLIC group. If the SAS Information Delivery Portal is installed, then 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 that they log on, or group members can search for and add the page (the SAS Information Delivery Portal must be installed in order for users to add pages).

If you logged on as a SAS Web Administrator, then you can edit any portlet or page in the portal Web application. If you logged on as a group content

administrator, then 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 an administrator for a group can share a Web application with the group, or can edit a shared Web application. △

---

## Step 9: Optionally, Update or Remove the Web Application

After you have created a Web application, you can then edit it, remove it from a portlet, and delete it permanently from metadata. From the portal Web application, 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, then the Web application is removed from all portal views.

For instructions on editing, removing, or permanently deleting a Web application, refer to the online Help that is provided with the portal Web application.

---

## 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 Web Applications” on page 276 in order to illustrate how you might add SAS Web Report Studio and/or SAS Web OLAP Viewer for Java to the portal Web application. 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:

If your deployment meets the requirements for single sign-on, then 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 “Understanding Single Sign-On” on page 24.

- are invoked from the portal Web application, but executed remotely
- support single sign-on
- use Metadata Server authentication
- use the SAS Foundation Services API

The following sections describe these steps to add SAS Web Report Studio and SAS Web OLAP Viewer for Java to the portal Web application. If you want to add one or the other application, but not both, then 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. This step is included here only to maintain a parallel structure with the steps that are in “Adding Web Applications” on page 276 .

---

## Step 2: Deploy the WAR Files in the Servlet Container

If you have already installed and configured SAS Web Report Studio and SAS Web OLAP Viewer for Java, then their WAR files have already been deployed into the servlet container.

Here are the WAR files that should be deployed:

- **SASWebReportStudio.war**, which is located in the *SAS-install-dir\SASWebReportStudio\3.1* directory. If you need to deploy this file, refer to the instructions that are in the application's **deployment.html** file.
- **SASWebOLAPViewer.war**, which is located in the *SAS-install-dir\SASWebOlapViewerforJava\3.1* directory. If you need to redeploy this file, refer to the deployment instructions that are in the application's **config.pdf** file.

---

## Step 3: Ensure that the Appropriate Group Metadata Exists in the SAS Metadata Repository

Before you add a Web application and share it with a group, that group must be defined in SAS metadata.

- 1 In SAS Management Console, create two groups that you can easily identify for the two applications. For example, create the following two groups:
  - Portal Web Report Studio Users
  - Portal Web OLAP Viewer Users

For details about defining groups, see the SAS Management Console User Manager Help.

- 2 Add at least one user to each group. For example, add the SAS Demo User (sasdemo) as a member of each group.
- 3 Optionally, you can add additional groups as members of each group. For example, if you have defined a group named Financial Analysts, then you can add that group as a member of Portal Web Report Studio Users and Portal Web OLAP Viewer Users. Users who are members of Financial Analysts will then have access to SAS Web Report Studio and SAS Web OLAP Viewer for Java.
- 4 Create a permission tree in SAS metadata for each group that you created. One way to create the permission trees is to restart the portal Web application. For more information, see “How Permission Tree Folders Are Created” on page 234.

---

## 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 portal
- Create the Web application by running a SAS program

Following are instructions for running the SAS program. Run the program separately for SAS Web Report Studio and SAS Web OLAP Viewer for Java.

- 1 Make a backup copy of the SAS program **LoadWebApplicationExample.sas**, which is located in the *SAS-install-dir\Web\Portal2.0.1\OMR* 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=Portal Web Report Studio Users;
%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=Portal Web OLAP Viewer Users;
%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 servlet container. For example, if SAS Web Report Studio is installed on Tomcat, then the line **%let webappURI=/SASWebReportStudio/logonFromPortal.do;** would be resolved to:

```
http://<PortalTomcatHost>:8080/SASWebReportStudio/
logonFromPortal.do
```

If SAS Web Report Studio were installed on a remote system instead, then you would specify the location as a fully qualified URL that includes the remote host and port, such as:

```
%let webappURI=http://<remotePortalTomcatHost>:8090/
SASWebReportStudio/logonFromPortal.do
```

- For complete descriptions of the variables in **LoadWebApplicationExample.sas**, see “Adding Web Applications” on page 276.

- 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, then you might modify and run it for SAS Web OLAP Viewer for Java.

---

## Step 5: Ensure That the Appropriate Resource Metadata Is Added to the SAS Metadata Repository

If you have correctly installed and configured the portal Web application and the application that you want to add (SAS Web Report Studio or SAS Web OLAP Viewer for Java), then you can skip this step.

You can verify this metadata in Foundation Services Manager in SAS Management Console. You should see the following:

- ID Portal Local Services (for the portal Web application)
- Remote Services (for single sign-on support)

- Query and Reporting Services (local services for SAS Web Report Studio)
- SAS Web OLAP Viewer Local Services (for SAS Web OLAP Viewer for Java)

For more information about redeploying the local and remote services, see “Service Deployment Configurations” on page 336.

---

## Step 6: Add the Permission Statements for the Web Application to the Required Policy Files

Add the codebase and permissions for the Web application and for the foundation services to the required policy file. For details, see “Adding Permissions to Policy Files” on page 45.

The policy file specified for your servlet container must contain a permission statement that allows the portal Web application to access the host and port where SAS Web Report Studio or SAS Web OLAP Viewer for Java is deployed.

Here’s a Tomcat example for SAS Web Report Studio on a Windows system:

Policy file: **C:\Tomcat4\conf\catalina.policy**

```
grant codeBase "file:${catalina.home}/webapps/Portal/-" {
...other permission statements
    // Socket access to Web Report Studio  $\frac{1}{2}$  hostname,
    // port if different from Portal
    permission java.net.SocketPermission
        "webreportstudio.host.com:port",
        "listen, connect, accept, resolve";
    ...
}
```

Here’s a Tomcat example for SAS Web OLAP Viewer for Java on a Windows system:

Policy file: **C:\Tomcat4\conf\catalina.policy**

```
grant codeBase "file:${catalina.home}/webapps/Portal/-" {
...other permission statements
    // Socket access to Web Report Studio - hostname,
    // port if different from Portal
    permission java.net.SocketPermission
        "swovj.host.com:port",
        "listen, connect, accept, resolve";
    ...
}
```

---

## Step 7: 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), then no additional steps are required for access control.

*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.  $\Delta$

For general information about access control, see “Understanding Portal Authorization” on page 222.

---

## Step 8: 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 portal Web application:

- You can edit a collection portlet in order to add the Web application to the portlet. You can then share the portlet with a group, including the PUBLIC group. If the SAS Information Delivery Portal is installed, then 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.

If you logged on as a SAS Web Administrator, then you can edit any portlet or page in the portal Web application. If you logged on as a group content administrator, then 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 an administrator for a group can share a Web application with the group, or can edit a shared Web application. △

For instructions on adding or sharing portlets and pages, refer to the online Help that is provided with the portal Web application.

---

## Step 9: Optionally, Update or Remove the Web Application

After you have created a Web application, you can then 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, then the Web application is removed from all portal views.

For instructions on editing, removing, or permanently deleting a Web application, refer to the online Help that is provided with the portal Web application.

---

# Adding Syndication Channels

---

## Overview of Adding Syndication Channels

A syndication channel is a channel that provides syndicated, continuously updated Web content. The portal Web application provides support for the emerging RSS (Rich Site Summary) 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, CNET, CNN, Disney, Forbes, Motley Fool, Wired, Red Herring, Salon, Slashdot, and ZDNet 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 the portal Web application permission 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 portal Web application's machine (the machine that hosts the portal's servlet container) 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 are running an Apache Tomcat server and you want to add a syndication channel from *rssnews.acme.com*, add the following statements:

```
grant codeBase "file:${catalina.home}/webapps/Portal/-" {
    ...
    permission java.net.SocketPermission
        "rssnews.acme.com:80", "connect, resolve";
    ...
};
```

For more information about policy files, see "Adding Permissions to Policy Files" on page 45.

---

## Step 2: Ensure That the Appropriate User or Group Permission Tree Is Created in the SAS Metadata Repository

Before you can define a syndication channel and share it with a group, you must create a permission tree in SAS metadata for the group. To verify that a permission tree exists, or to create one, see "Managing Portal Permission Trees in Metadata" on page 233.

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, then you can associate the syndication channel with a user instead of a group. The user must have a permission tree in metadata. (To create a permission tree for a user, log on to the portal Web application 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 Web application as the SAS Web Administrator in order to share the syndication channel with a group (you might need to search for the syndication channel first). If the user is a group content administrator, then the user can share the syndication channel with the group.

---

## 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 portal Web application: When you create a syndication channel in the portal Web application, 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, refer to the online Help that is provided with the portal Web application:

- 1 Log on to the portal Web application as either the SAS Web Administrator or 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 Web Application” on page 226.

- Create the syndication channel by running a SAS program: You can edit and run a SAS program that creates a syndication channel and adds the channel's metadata to the SAS metadata repository. Complete these steps:

- 1 Modify the SAS program `LoadSyndicationChannelExample.sas`, which is located in the `SAS-install-dir\Web\Portal12.0.1\OMR` 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 `$SERVICES_OMI_HOST$` property in the `install.properties` file (located in the `SAS-install-dir\Web\Portal12.0.1\PortalConfigure` 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 `$SERVICES_OMI_PORT$` property in the `install.properties` file.

`metauser="user ID"`

Specify the user ID to use to connect to the SAS Metadata Server. This user ID is typically the SAS Administrator (sasadm). For Windows users, the user ID is

domain or machine name qualified. For example: *<domain or machine name>\saswbadm*

metapass=*“password”*

Specify the password for the metauser.

metarepository=*“repository”*;

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the \$SERVICES\_OMI\_REPOSITORY\$ property in the **install.properties** file.

%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://csociety.purdue.org/~jacoby/XML/CNN_US.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, then access is limited to the user or group that you specify in **LoadSyndicationChannelExample.sas**.
- If you create the syndication channel in the portal, then 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 then share the portlet with a group, including the PUBLIC group. If the SAS Information Delivery Portal is installed, then 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 that they log on, or group members can search for and add the page (the SAS Information Delivery Portal must be installed in order for users to add pages).

*Note:* If you logged on as a SAS Web Administrator, then you can edit any portlet or page in the portal Web application. If you logged on as a group content administrator, then 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: Optionally, Update or Remove the Syndication Channel

After you have created a syndication channel, you can then 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, then the syndication channel is removed from all portal views.

For instructions on editing, removing, or permanently deleting a syndication channel, refer to the online Help that is provided with the portal Web application.

---

## Adding SAS Packages

---

### Overview of Adding SAS Packages

A package is a collection of structured and unstructured content that has been published using the SAS Publishing Framework, or created or published by running a SAS Stored Process on a SAS Workspace Server.

If you have installed the SAS Information Delivery Portal, then users can view SAS packages from the portal Web application.

Packages are used to deliver the following:

- the content of publication channels, which publish information using the SAS Publishing Framework. If you publish a package from the SAS Information Delivery Portal, then the package might include any of the following archived content types:
  - files (if you have installed the SAS Information Delivery Portal with the Xythos WFS WebDAV server)
  - links
  - SAS Information Maps (which are published as a link (reference) that will display the SAS Information Map in the Visual Data Explorer of the portal)
  - SAS reports (which are published as a link (reference) that will display the SAS Report in the SAS Web Report Viewer of the portal)
- 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 Web application if the packages have been published to a SAS Publication Channel (SAS Information Delivery Portal only) or if the packages have been created or published to a Xythos WFS repository.

If a channel has a WebDAV persistent store that is not under the portal's default base path, then packages that are published to that channel will not (initially) be accessible using the portal's Search tool. Users can, however, access the channel in order to view the packages. A connection to the channel's base path is established when the channel is accessed. The portal's Search tool will then be able to find the packages for the remainder of the portal session.

The following sections describe how to add a package to the portal Web application.

---

## Step 1: If the Package Is Not Created, Create the Package

If the package is not already created, create and publish the package to one of the following locations:

- SAS publication channel (if you have installed the SAS Information Delivery Portal)
- Xythos WFS WebDAV repository

There are several ways that a package might be created and published:

- You can develop a SAS Stored Process that runs on a SAS Workspace Server and produces packages. These packages can be stored on a Xythos WFS server or published to a SAS publication channel. For details, see “SAS Stored Processes” in the *SAS Integration Technologies: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/dev\\_guide/stprocess/index.html](http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/index.html).
- If you have installed the SAS Information Delivery Portal, then users can use the portal **Options** menu to publish a package and add the package (content) metadata to the SAS Metadata Repository. For details about using the portal Web application to publish a package, refer to the portal’s online Help. When users publish the package to a SAS publication channel or to a Xythos WFS repository, the Web application adds the metadata for the package to the SAS Metadata or WebDAV Repository.
- You can use SAS Publishing Framework to publish a package. For details, see “Publishing Framework” in the *SAS Integration Technologies: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/dev\\_guide/publish/index.html](http://support.sas.com/rnd/itech/doc9/dev_guide/publish/index.html).
- You can publish a package in SAS Enterprise Guide.

---

## Step 2: Implement Authorization for the Package

The authorization metadata for a package is part of the metadata for the SAS publication channel or Xythos WFS repository to which the package is published.

Take any necessary steps to control access to files, reports, or other items that have been added to the package.

---

## Step 3: Make the Package Available in the Portal Web Application

You can make the package appear on the portal Web application in any of the following ways:

- Add a SAS publication channel and then add the package to the channel (if you have installed the SAS Information Delivery Portal). Authorized users can view the package from the SAS publication channel.
- Add a WebDAV navigator portlet and view the package from a WebDAV navigator portlet (if you have installed the Xythos WFS WebDAV server).
- Edit a collection portlet and add the package to the collection portlet. You can then share the content using the portal **Options** menu. For general information about sharing content, see “Sharing Content in the Portal Web Application” on page 226.

## Adding SAS Publication Channels

### Overview of Adding SAS Publication Channels

#### About SAS Publication Channels

If you have installed the SAS Information Delivery Portal, then users can access SAS publication channels from the portal Web application.

A SAS publication channel is a channel created by SAS Publishing Framework. 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 “Publishing Framework” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/publish/index.html](http://support.sas.com/rnd/itech/doc9/admin_oma/publish/index.html).
- For information about implementing the Publishing Framework capabilities in your applications, see “Publishing Framework” in the *SAS Integration Technologies: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/dev\\_guide/publish/index.html](http://support.sas.com/rnd/itech/doc9/dev_guide/publish/index.html).

In addition, you can publish files, links, SAS Information Maps, and SAS Reports to a publication channel. However, if you do not have the Xythos WebFile Server (WFS) installed, then you cannot publish to a channel that is defined as a WebDAV persistent store.

From the portal Web application, users can publish a package that might include any of the following archived content types:

- files (if you have installed the SAS Information Delivery Portal and Xythos WFS)
- 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)
- SAS Reports (which are published as a link that will display the SAS Report in the SAS Web Report Viewer of the portal)

The portal provides an interface through which users can subscribe to SAS publication channels. After subscribing to a channel, users can use the portal to view archived content that is published through the channel. When a user subscribes to a channel, a subscriber profile is used. This profile contains information on how the information that is published to the channels is to be delivered.

#### E-Mail Transport Restriction

When a user publishes a package to a channel from within the 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, the SAS Publisher user interface (which is part of Base SAS), or CALL routines within a SAS program or SAS stored process. For more information, see “Publishing Framework” in the *SAS Integration Technologies: Developer’s Guide*.

#### WebDAV Publication Channel Considerations

If a channel’s base path is not under the portal’s default base path, then packages that are published to the channel will not initially be accessible using the portal’s

Search tool. Users can, however, access the channel in order to view the packages. When a user accesses a channel that has a WebDAV persistent store, a connection to the channel's base path is established. The user's Search tool will then be able to find the package for the remainder of the portal session.

If you are setting up a WebDAV-based publication channel, then you must set up the appropriate SAS users and groups to enable users to publish to the Xythos WFS WebDAV server. For details, see "Planning for Portal Users and Groups" on page 220.

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): If Publishing to an Archive, Add the SAS Publication Channel's Archive Permission Statement to the Appropriate Policy File**

To enable the portal Web application 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 servlet container'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 an Apache Tomcat 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:${catalina.home}/webapps/Portal/-" {
    ...
    permission java.io.FilePermission
        "/sas/PubSub/-", "read,write";
    ...
};
```

For more information about policy files, see "Adding Permissions to Policy Files" on page 45.

## **Step 2: Add the Publication Channel to the SAS Metadata Repository**

Before you add the publication channel to the SAS Metadata Repository, ensure that you have the appropriate servers defined in the SAS Metadata Repository. Depending on the delivery method for your publication channel, you must have certain servers defined in your SAS Metadata Repository:

- If you are publishing to an archive on a SAS Workspace Server, then you must define a SAS Workspace Server and Spawner.
- If you are publishing to an archive on a Xythos WFS WebDAV server, then you must define a Xythos WFS WebDAV server.

- If you are publishing to an archive in the file system, then no server definition is needed for the publication channel.

For details about verifying which server definitions are required for your content, see Appendix 2, “SAS Application Servers That Are Required for SAS Content,” on page 363. To add a publication channel to the SAS Metadata Repository, complete these steps:

- 1 Log on to SAS Management Console as the SAS Administrator and use the Publishing Framework to define the channel in the metadata repository. For detailed instructions about defining channels, refer to the Publishing Framework Help. See also “Managing Channels” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/publish/publish\\_channels.html](http://support.sas.com/rnd/itech/doc9/admin_oma/publish/publish_channels.html).
- 2 Use either the Publishing Framework or the portal **Options** menu to define one or more subscribers for the channel in the metadata repository. For detailed instructions on defining subscribers, refer to the Publishing Framework Help and “Managing Subscribers” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/publish/publish\\_subscribers.html](http://support.sas.com/rnd/itech/doc9/admin_oma/publish/publish_subscribers.html).

---

### Step 3: Implement Authorization (Access Control) for the SAS Publication Channel

Take any necessary steps to control access to the Publication Channel. For details, see “Setting Up Authorization for Stored Processes and Publication Channels” on page 229.

---

### Step 4: Make the SAS Publication Channel Available to Portal Users

Depending on who has access permission to the publication channel, users can use one of several methods to make it appear in the portal Web application. Those methods include the following:

- The SAS Web Administrator or a group content administrator can edit a collection portlet in order to add the publication channel to the portlet. The SAS Web Administrator can then share the portlet with a group, including the PUBLIC group. A group content administrator can share the portlet with the group for which he is an administrator. If the SAS Information Delivery Portal is installed, then group members can search for and add the portlet to their pages.
- All users can use the Publication Channel Subscriptions portlet to display the SAS publications channels that they are subscribed to. This provides a convenient way to view content published to the channels. Users can add this portlet to multiple pages.

---

## Adding and Administering SAS Stored Processes

---

### 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 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 “SAS Stored Processes” in the *SAS Integration Technologies: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/dev\\_guide/stprocess/index.html](http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/index.html).

---

## How Stored Processes Are Executed from the Portal

The SAS Stored Process Web application is required in order to run stored processes from the portal Web application. The SAS Stored Process Web application was deployed when you installed and configured the portal Web application (part of the Web Infrastructure Kit installation).

All portal users who have the appropriate permissions can run a stored process by clicking its icon or link in the portal Web application. 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, and users can add a stored process to a collection portlet. If users add a Stored Process Navigator portlet to their portal views, then users can explore the stored processes that have been defined in metadata (provided they have the appropriate permissions). After running the stored process, users can e-mail or bookmark the results to other users.

When a portal 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 in the portal Web application:

- Streaming output: If a stored process was defined to stream output to the viewer, then 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, then 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 “Result Types” in the *SAS Integration Technologies: Developer’s Guide*.

The portal Web application 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 Web application 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 Results Navigator portlet for later viewing 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 then 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 Alerts portlet. From the Alerts portlet, users can track, view, and remove the stored process. For more information, see “About Alerts” on page 298.
- Depending on how the stored process was created, permanent package output can be published to a Xythos WebFile Server WebDAV repository.
- The Xythos WebFile Server must be installed in order to run the stored process if either of the following is true:
  - the stored process creates a permanent package result in either WebDAV or in the personal repository
  - the stored process runs in the background
- If the stored process package is a permanent package that is associated with a SAS publication channel, then 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.

---

## 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.

*Note:* If the stored process runs in the background or creates a permanent package, then make sure that the Xythos WebFile Server is installed and configured for your environment. If the stored process creates a package that is published to a SAS publication channel, then the SAS Information Delivery Portal must be installed before portal users can view the package or access the channel. △

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 portal Web application, add the stored process to a collection portlet, and then share the page that contains the portlet with portal users. You might need to search for the stored process before it becomes available to you.

For instructions on sharing a page or adding items to a portlet, refer to the online Help that is provided with the portal Web application.

*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, and if you have installed the SAS Information Delivery Portal, then you can add the SAS publication channel to the portal (if it

hasn't already been added). You can also define subscriber profiles and set up subscriptions for portal users.

After the publication channel is created, then 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. Group content administrators can add the channel to a page, and share that page with their respective groups.

- Ensure access to the Xythos WebFile Server WebDAV repository: If the stored process publishes a package to a Xythos WebFile Server WebDAV repository, then 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 220.
- Provide an Alerts portlet to portal users: You might want to provide an Alerts portlet to some or all portal users. The Alerts portlet displays an alert for any background stored process when the stored process is executed. To provide an Alerts portlet to a group of users, create a page template and define an Alerts portlet in the page template. For details about creating a page template, see “Adding, Editing, and Removing Page Templates” on page 251. You can also add stored processes to this same page template by defining a collection portlet in the page template, and then 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 Alerts, collection, and Stored Process Navigator portlets to their portal views if you have installed the SAS Information Delivery Portal.  $\Delta$

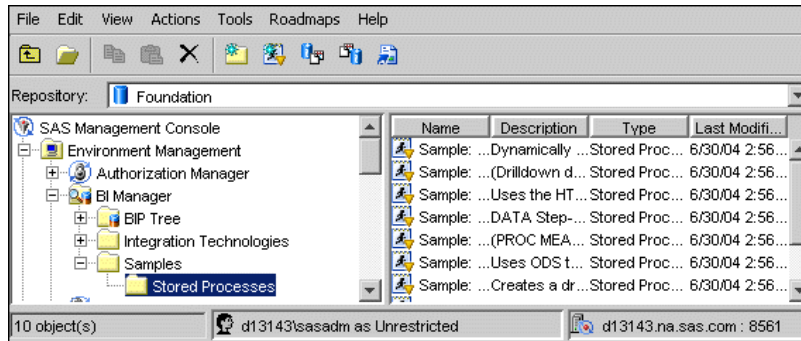
- Share stored process results: You can log on to the portal, run a stored process, and share the results with others by publishing or e-mailing the resultant package. (All portal users who have access permission to the stored process can perform this task.)
- Know how to register metadata for stored processes: Although developers register the metadata for a stored process when they create the stored process, you might occasionally need to confirm or change metadata. For example, you might need to change access permissions for a stored process. In SAS Management Console, use BI Manager to set access permissions for stored processes in metadata. See “Setting Up Authorization for Stored Processes and Publication Channels” on page 229.

*Note:* Custom input JSP files that your organization develops should be stored in the *SAS-install-dir\Web\Portal2.0.1\SASStoredProcess\input* directory rather than in the servlet container's deployment directory. If you deploy the files to your servlet container, then if you re-create and redeploy the portal files, your files will be overwritten.  $\Delta$

Example stored processes are installed automatically if you install the initial demo data for the portal Web application. You can log on to the portal Web application and search for SAS stored processes.

You can find the stored process sample code in the stored process sample path of the SAS installation. To check this value, see the `$STP_SAMPLE_PATH$` parameter of the `install.properties` file.

To view the metadata for the stored process, in SAS Management Console, expand the BI Manager, and then expand the **Samples** folder to locate the **Stored Process** samples directory, as shown in this display:



*Note:* BI Manager is available beginning with SAS Foundation Services 1.2. If you have not upgraded to this release, then you can use Stored Process Manager to register and manage stored processes. BI Manager replaces Stored Process Manager. For more information about using BI Manager or Stored Process Manager, see the Help in SAS Management Console. △

For more information about stored process metadata, see the BI Manager product Help in SAS Management Console. See also “SAS Stored Processes” in the *SAS Integration Technologies: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/dev\\_guide/stprocess/index.html](http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/index.html).

## About Alerts

An alert is an automatic notification of an electronic event that is of interest to you. Alerts are displayed in the Alerts portlet on a portal page. If the SAS Information Delivery Portal is installed, then all users in your organization with access to the portal can add an Alerts portlet to their personal portal views. You can also provide an Alerts portlet to a group of users by adding the portlet to a page template that you share with the respective group.

Here is a sample Alerts portlet:

<input type="checkbox"/>	Type	Name	Date/Time ▼	Description
<input type="checkbox"/>	STP	Daily Sales Analysis	June 3, 2005 1:02:46 PM EDT	Batch sales analysis
<input type="checkbox"/>	STP	Daily Sales Analysis	April 21, 2005 9:57:26 PM EDT	Batch sales analysis

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 Alerts portlet to see the results of the stored process. (If the stored process was defined with no output, then the alert is not linked.) From the Alerts portlet, you can also remove the results of the stored process by deleting the respective alert.

Stored processes that generate alerts require WebDAV or personal repositories be available. Therefore, the Xythos WebFile Server must be installed in order to run these stored processes.

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 an Alerts portlet or managing alerts, refer to the online Help that is provided with the portal Web application.

---

## 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.

*Note:* Beginning with Service Pack 4, Read permission for an information map is required in order to access data through that information map. This requirement is explained in your **instructions.html** file. △

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 portal Web application. 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 222.

---

### Step 2: Make the SAS Information Map Available in the Portal Web Application

Here are some ways to make a SAS Information Map appear in the portal Web application:

- The SAS Web Administrator or a group content administrator can edit a collection portlet in order to add an information map to the portlet. The SAS Web Administrator can then share the portlet with a group, including the PUBLIC group. A group content administrator can share the portlet with the group for which he is an administrator. If the SAS Information Delivery Portal is installed, then group members can search for and add the portlet to their pages.

- After adding the information map to a portlet, the SAS Web 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 (the SAS Information Delivery Portal must be installed in order for users to add pages).
- Users can use the portal Web application or SAS Publishing Framework to publish an information map to a SAS publication channel. Authorized users can then subscribe to the SAS publication channel and then add the Publication Channel Subscriptions portlet to their portal Web application. 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, then 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 the SAS Web Report Viewer to display the report. The SAS Web Report Viewer must be installed separately from the portal.

*Note:* Alternatively, you can configure the portal to display reports in SAS Web Report Studio. Then authorized users can edit reports without logging on to SAS Web Report Studio separately. For details, see “Using SAS Web Report Studio as the Default Report Viewer” on page 353. △

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 Web Report Studio then 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 portal Web application. Take any necessary steps to implement additional authorization (access control) for the page (that will contain the portlet with the SAS Report). For general information about access control, see “Understanding Portal Authorization” on page 222.

---

## Step 2: Make the SAS Report Available to Portal Users

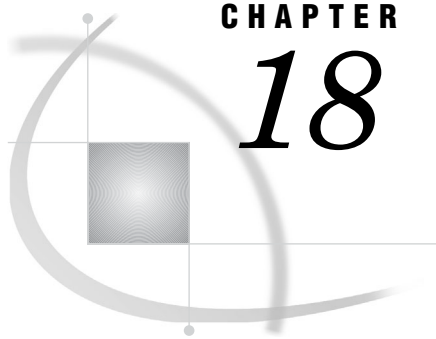
You can use one of several methods to make it appear in the portal Web application. These methods include the following:

- The SAS Web Administrator or a group content administrator can edit a collection portlet in order to add a report to the portlet. The SAS Web Administrator can then share the portlet with a group, including the PUBLIC group. A group content administrator can share the portlet with the group for which he is an administrator. If the SAS Information Delivery Portal is installed, then group members can search for and add the portlet to their pages.
- After adding the report to a portlet, the SAS Web 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 (the SAS Information Delivery Portal must be installed in order for users to add pages).
- Users can use the portal Web application or SAS Publishing Framework to publish a report to a SAS publication channel. Authorized users can then subscribe to the SAS publication channel and then add the Publication Channel Subscriptions portlet to their portal Web application. Users can also add the publication channel to a collection portlet.

For more information, refer to the portal's online Help.







## CHAPTER

## 18

## Administering SAS Business Intelligence Dashboard

<i>Overview of SAS Business Intelligence Dashboard</i>	<b>303</b>
<i>Main Tasks for Administering SAS Business Intelligence Dashboard</i>	<b>304</b>
<i>Understanding the Data Source XML (DSX) Files</i>	<b>304</b>
<i>Specify a JDBC Data Source for SAS Business Intelligence Dashboard</i>	<b>305</b>
<i>Improving the Performance of SAS Business Intelligence Dashboard</i>	<b>306</b>
<i>Overview of Improving Dashboard Performance</i>	<b>306</b>
<i>Configure a Data Cache</i>	<b>307</b>
<i>Configure Pooling of Dashboard JDBC Connections</i>	<b>308</b>
<i>Managing User Security for SAS Business Intelligence Dashboard</i>	<b>309</b>
<i>Overview of Dashboard Users and Security</i>	<b>310</b>
<i>Main Tasks for Implementing Dashboard Security</i>	<b>310</b>
<i>Enable Dashboard Security</i>	<b>310</b>
<i>Manage Users in Dashboard Groups</i>	<b>311</b>
<i>Verify or Set Permissions</i>	<b>312</b>
<i>Configuration for Dashboard Portlets That Are Shared</i>	<b>313</b>
<i>About Shared Dashboard Portlets</i>	<b>313</b>
<i>Enforce Portlet Security</i>	<b>313</b>
<i>(Optional) Create Additional Dashboard Groups</i>	<b>314</b>
<i>Overview of Creating Additional Dashboard Groups</i>	<b>314</b>
<i>Example Group: Dashboard Modelers</i>	<b>314</b>
<i>Example Group: Dashboard Analysts</i>	<b>315</b>
<i>Example Group: Finance Users</i>	<b>315</b>

### Overview of SAS Business Intelligence 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 two indicators. The indicator on the left contains a single gauge (map of the United States), whereas the indicator on the right contains several arrow gauges.

Display 18.1 Example Dashboard Portlet



SAS Business Intelligence 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.

*Note:* For instructions on using SAS Business Intelligence 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 only if you are logged on as a dashboard administrator. For more information about user permissions, see "Managing User Security for SAS Business Intelligence Dashboard" on page 309. △

SAS Business Intelligence Dashboard uses the portal's remote portlet architecture. This means that the portal Web application 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 Custom Portlets" in the *SAS Web Infrastructure Kit: Developer's Guide* at [http://support.sas.com/rnd/itech/library/toc\\_portaldev.html](http://support.sas.com/rnd/itech/library/toc_portaldev.html).

---

## Main Tasks for Administering SAS Business Intelligence Dashboard

The following list summarizes the administrative tasks that are specific to SAS Business Intelligence Dashboard:

- Configure data source XML files in order to specify the data sources that are created by data designers in your organization. See "Specify a JDBC Data Source for SAS Business Intelligence Dashboard" on page 305. (This administrator's guide does not describe how to create data sources.)
- (Optional) Improve the performance of SAS Business Intelligence Dashboard. See "Improving the Performance of SAS Business Intelligence Dashboard" on page 306.
- (Optional) Implement security in order to manage user access to dashboards objects. See "Managing User Security for SAS Business Intelligence Dashboard" on page 309.

---

## Understanding the Data Source XML (DSX) Files

During initial configuration, the SAS Business Intelligence 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-config-dir\Lev1\SASBIDashboard\dataSourceDefs* on the host machine. The following table summarizes the DSX files.

**Table 18.1** Summary of the DSX Files

Filename	Description
dboard_sas.dsx	<p>Registers a library in metadata and enables the SAS Business Intelligence Dashboard to connect with that library. Data modelers can issue SQL queries and read data using JDBC.</p> <p>By default, there are no data sets in the library. You can modify this file as follows:</p> <ul style="list-style-type: none"> <li>□ Specify a data source for the library. You must specify a data source before data designers at your site can create dashboards.</li> <li>□ Configure data caching and pooling of JDBC connections in order to improve dashboard performance.</li> </ul> <p>You can create additional DSX files for your JDBC connections.</p>
infomap.dsx	<p>Enables the SAS Business Intelligence Dashboard to use SAS Information Maps. Data modelers can specify information maps in order to read data.</p> <p>The only change you would normally make to this file is to configure performance settings (caching).</p>

*Note:* The file directory also includes an **spm.dsx** file that is used if SAS Strategic Performance Management has been installed at your site. △

*Related Tasks:*

- To specify a dashboard data source, see “Specify a JDBC Data Source for SAS Business Intelligence Dashboard” on page 305.
- To improve dashboard performance, see “Improving the Performance of SAS Business Intelligence Dashboard” on page 306.

## Specify a JDBC Data Source for SAS Business Intelligence Dashboard

Before the data designers at your site can create custom dashboard objects that use your organization’s data, you must specify the data sources so that SAS Business Intelligence Dashboard can find the data.

You use one or more DSX files to specify and register SAS libraries that your organization’s data designers create. The libraries referenced in DSX files use the SAS Workspace Server for processing.

For a description of the DSX files, including their location, see “Understanding the Data Source XML (DSX) Files” on page 304.

To make custom dashboard data available, 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-config-dir\Lev1\SASBIDashboard\sas-datasets* on the host machine.

- 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>DBOARD 'c:/myDashboard/sas-datasets';</LibRefs>
```

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, complete 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 would contain the following element:

```
<DataSourceDef id="myData"
  providerClass="com.sas.bi.dashboard.provider.JdbcProvider">
```

- 2 Change 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 you specify here will appear 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 servlet container or J2EE application server before your changes take effect.

---

## Improving the Performance of SAS Business Intelligence Dashboard

---

### Overview of Improving Dashboard Performance

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.

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.

## Configure a Data Cache

By default, dashboards obtain their data at the time the dashboard is requested. This default configuration can cause scalability problems and isn't necessary if the underlying data changes infrequently. For greater performance and scalability, Business Intelligence Dashboard employs an in-memory Least Recently Used (LRU) cache.

SAS Business Intelligence Dashboard serves a data model from cache if the underlying data is not stale. If the underlying data is stale, then SAS Business Intelligence 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 isn't available, then the least recently used data models will be dropped from the cache when the cache reaches its memory limit.

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 304.)

To enable caching for a data source, complete 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 you would remove the comment delimiters from `<DefaultTimingCacheDirective>` in the `infomap.dsx` file.
- 2 After you make changes to the DSX files, you must restart the servlet container or J2EE application server before your changes take effect.

Here are descriptions of the `<DefaultTimingCacheDirective>` attributes:

**Table 18.2** Attributes in the `<DefaultTimingCacheDirective>` Element

Attribute	Description
<code>cacheDisplayValueForRefresh</code>	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 <code>cacheDisplayMultiplierForRefresh</code> attribute.) Enter the number in quotation marks.  You can enter <code>'0'</code> 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.
<code>cacheDisplayMultiplierForRefresh</code>	Specifies the unit of measure for the value that is entered in <code>cacheDisplayValueForRefresh</code> . Valid values are "SECONDS," "MINUTES," and "HOURS." Enter the value in quotation marks.

Attribute	Description
cacheDisplayValueForStale	<p>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.) Enter the number in quotation marks.</p> <p>The value provided here should be greater than the value that you provide for cacheDisplayValueForRefresh. Otherwise, the data becomes stale before it can be refreshed, and the cacheDisplayValueForRefresh value has no effect.</p>
cacheDisplayMultiplierForStale	<p>Specifies the unit of measure for the value that is entered in cacheDisplayValueForStale. Valid values are "SECONDS," "MINUTES," and "HOURS." Enter the value in quotation marks.</p>
maxDataModelCacheSize	<p>Specifies the approximate upper limit of the data model cache size. When additional data models are cached after this size is met, the least recently used data models are dropped from the cache. The value is in bytes.</p>

Here is example code with sample values for the attributes:

```
<DefaultTimingCacheDirective
  cacheDisplayValueForRefresh="5.0"
  cacheDisplayValueForStale="15.0"
  cacheDisplayMultiplierForRefresh="MINUTES"
  cacheDisplayMultiplierForStale="MINUTES"
  maxDataModelCacheSize=2097152/>
```

In this example, the data will never be more than 15 minutes old. 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.

---

## Configure Pooling of Dashboard JDBC Connections

Caching is a preferable optimization mechanism for scalability, but caching might require more memory than desired or it might not meet your data freshness requirements.

In addition to caching, SAS Business Intelligence Dashboard enables you to 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 several different SQL queries. Regardless of whether you use data caching, you can configure pooled JDBC connections in order to improve performance.

By default, SAS Business Intelligence Dashboard uses JDBC connection pooling. You would perform the following procedure only if you want to change the values that are used for pooling. You don't need to do anything in order to use pooling with the default values.

To configure pooling for a data source, complete 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 304.

- 2 After you make changes to the DSX files, you must restart the servlet container or J2EE application server before your changes take effect.

Here are descriptions of the pooling attributes that you add to the `<DataSourceDef>` element:

**Table 18.3** Attributes in the `<DataSourceDef>` Element

Attribute	Description
<code>maxPoolSize</code>	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.
<code>maxWaitForPooledConnection</code>	Specifies the number of milliseconds to wait for a pooled connection before returning an error that the connection failed. The default value is 5000.
<code>lingerTime</code>	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 don't have to be re-established within a single HTTP request. The default value is 300000.
<code>alwaysConnectAsAdminUser</code>	Specifies that clients always connect as the administrator user that is specified in the <code>BIDashboard.config</code> file ( <code>saswbadm</code> is the default). 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 code snippet that shows pooling attributes added to the `<DataSourceDef>` element:

```
<DataSourceDef id="dboard_sas"
providerClass="com.sas.bi.dashboard.provider.JdbcProvider"
maxPoolSize="5"
maxWaitForPooledConnection="60000"
lingerTime="60000"
alwaysConnectAsAdminUser="true">
. . .
</DataSourceDef>
```

## Managing User Security for SAS Business Intelligence Dashboard

You can manage access to dashboard objects, such as dashboards, indicators and models, by adding users or groups to the appropriate dashboard group in metadata. This topic describes the dashboard groups and lists the permissions that the groups require in metadata. The topic also describes what to configure for dashboard portlets that are shared in the portal.

---

## Overview of Dashboard Users and Security

Information about adding users or groups to a deployment is described in “Planning User Accounts and Their Organization into Groups” on page 21.

These are the aspects of setting up users that are specific to SAS Business Intelligence Dashboard:

- The portal Web application is responsible for authentication.
- The ability to create and manage dashboard objects is controlled by permissions that are set on the BIDashboard folder in SAS Management Console and by the permissions that are granted in the Default ACT.
- SAS Business Intelligence Dashboard uses a metadata account to connect to SAS servers. By default, this account is the SAS Web Administrator (saswbadm).

The SAS Web Administrator must be included in another group or must appear explicitly in the Default ACT with ReadMetadata and WriteMetadata permissions. These permissions are established during initial configuration.

- If a user doesn't have permission to view the underlying data for a dashboard, then the user will see the dashboard but not the data. For example:
  - If a user doesn't have read permissions on an underlying information map, cube, or data set, then the query will fail and an error message will be returned.
  - If an information map employs row-level permissions, then only the data that is readable by a particular user will appear in a dashboard indicator when that user is logged on to the portal.
  - If an OLAP cube employs granular permissions, then only the data that is readable by a particular user will appear in a dashboard indicator.

---

## Main Tasks for Implementing Dashboard Security

- You must enable SAS Business Intelligence Dashboard security before your security implementation can take effect. See “Enable Dashboard Security” on page 310.
- Manage access to dashboard objects by doing the following:
  - Create groups in metadata and add users to those groups. See “Manage Users in Dashboard Groups” on page 311.
  - Assign permissions to the groups on the BIDashboard folder. See “Verify or Set Permissions” on page 312.
- Configure settings that enforce read-only access for dashboard portlets that are shared in the portal. Without this configuration, some users might be able to change the dashboard that is displayed in a shared portlet. This change would be visible to all users who can access the shared portlet. See “Configuration for Dashboard Portlets That Are Shared” on page 313.
- (Optional) Create additional groups in metadata in order to achieve more granular access controls. See “(Optional) Create Additional Dashboard Groups” on page 314.

---

## Enable Dashboard Security

You must enable SAS Business Intelligence Dashboard security before your security configuration can take effect. Until you enable dashboard security, all users who can access the portal Web application have read and write access to dashboard objects.



To enable dashboard security, in **BIDashboard.config**, remove the comment delimiters from the following properties:

- jaasConfig
- metadataRootFolder
- adminGroup
- userGroup

Here is an example of a block with the comment delimiters removed:

```
jaasConfig=C:\SAS\EntBIServer\Lev1\web\Deployments\Portal\login.config
metadataRootFolder=BIP Tree/BIDashboard
adminGroup=Dashboard Admins
userGroup=Dashboard Users
```

The values for these properties in your **BIDashboard.config** file might differ from the values that are shown here. The value of the `jaasConfig` property should be set to the location of the **login.config** file that is installed with the SAS Information Delivery Portal. This is typically located in *SAS-config-dir\Lev1\web\Deployments*. You will specify the values for the `adminGroup` and `userGroup` properties after you create their respective groups in metadata. For instructions, see “Manage Users in Dashboard Groups” on page 311.

After you make changes to the **BIDashboard.config** file, you must restart the servlet container or J2EE application server before your changes take effect.

If you later want to disable dashboard security, simply comment all of the properties.

---

## Manage Users in Dashboard Groups

Use dashboard groups to manage access to dashboard objects. Typically, you will grant access to data designers so that they can create the dashboards, indicators, and data models using the graphical interface. You will typically limit access for other users who need only to see dashboards on the portal page. You manage user access by adding users to the appropriate group and then by assigning permissions to the groups on the BIDashboard folder.

These groups determine which dashboard objects users can access and manipulate as follows:

**Table 18.4** User Groups and Their Access to Dashboard Objects

Group*	Type of Access
Dashboard Users	<p>Members of this group can view dashboards in portlets and change the dashboard layout. Members cannot create, edit, or delete dashboard objects. This group derives access from the PUBLIC group in the Default ACT.</p> <p>The SAS Guest user account (sasguest) should be added to this group in order to enable the addition of dashboard portlets to the Public Kiosk.</p>
Dashboard Admins	<p>Members of this group can view dashboards in portlets and change the dashboard layout. Members also have access to a <b>Manage Dashboards</b> link in the portlet. After they click this link, members can create, edit, and delete dashboard objects.</p> <p>The SAS Web Administrator account (saswbadm) should be added to this group.</p>

---

\* The names listed here are only suggestions. You can use different names in your configuration.

To manage dashboard groups, complete these steps:

- 1 Define the groups in SAS Management Console and add users to the groups.

*Note:* While it is possible to use the PUBLIC group for the Dashboard Users group, this is not recommended because users can possibly gain full access to dashboard objects. If you use the PUBLIC group, then be sure to deny the group WriteMetadata permission on the BIDashboard folder. See “Verify or Set Permissions” on page 312. △

- 2 Assign permissions to these groups on the BIDashboard folder in metadata. For details, see “Verify or Set Permissions” on page 312.
- 3 In the **BIDashboard.config** file, specify the names that you gave to these groups. Enter the names as values for the adminGroup and userGroup properties.

Here is an example of these properties:

```
adminGroup=Dashboard Admins
userGroup=Dashboard Users
```

The default location for the **BIDashboard.config** file is *SAS-config-dir\Lev1\SASBIDashboard*.

- 4 After you make changes to the **BIDashboard.config** file, you must restart the servlet container or J2EE application server before your changes take effect.

*Related Notes:*

- For instructions on creating groups or adding users to groups, see the User Manager Help in SAS Management Console.
- For a list of all the users and groups that should be defined in metadata, see “Users and Groups That Are Defined in Metadata” on page 360.

---

## Verify or Set Permissions

Permissions must be set on the BIDashboard folder 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 will need to set permissions manually.

You specified the location of the BIDashboard folder during initial configuration. To obtain the location for your deployment, look at the value specified for the metadataRootFolder property in the **BIDashboard.config** file.

To verify or set permissions on the BIDashboard folder, complete these steps:

- 1 In the SAS Management Console Authorization Manager, navigate to the BIDashboard folder.

For example, you might navigate to the following path: **Authorization Manager ► Resource Management ► By Application ► BIP Service ► BIP Tree ► BIDashboard**.

- 2 From the main menu, select **File ► Properties**.
- 3 In the Properties dialog box, select the **Authorization** tab.
- 4 On the **Authorization** tab, select a dashboard group and confirm or set the permissions for that group. The permissions are listed here:

**Table 18.5** Group and User Permissions

Group or User	Permissions
Dashboard Admins group	Explicitly grant Read, Write, Create, Delete, ReadMetadata, and WriteMetadata permissions.
Dashboard Users group	Explicitly grant Read and ReadMetadata permissions. Explicitly deny all other permissions.
PUBLIC and SASUSERS groups	Explicitly deny all permissions.
SAS Web Administrator	Explicitly grant Read, Write, Create, Delete, ReadMetadata, and WriteMetadata permissions. Alternatively, you can add this user to the Dashboard Admins group, which has those permissions.
SAS Guest user	Explicitly grant Read and ReadMetadata permissions. Alternatively, you can add this user to the Dashboard Users group. This user can add dashboard portlets to the Public Kiosk.
SAS Administrator and SAS System Services	(Optional) As good practice, change all permissions with a gray background to explicit grants or denials.

*Note:* In SAS Management Console, permissions with a gray background color obtain their settings from the repository ACT. To make a grayed permission setting explicit, select the check box. The gray background is removed and the check box is still selected. △

When you apply these permissions on the BIDashboard folder, the permissions automatically apply to the subfolders beneath the BIDashboard folder.

---

## 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 won't 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 SAS Web Administrator or a group content administrator for the respective group. For more information about sharing portlets, see "Sharing Content in the Portal Web Application" on page 226.

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 an `enforcePortletSecurity` property to true. The following section describes this setting.

### Enforce Portlet Security

The `enforcePortletSecurity` property in the `BIDashboard.config` file determines whether users can edit a portlet that is shared. The `enforcePortletSecurity` property can be set to true or false, as described in the following table.

**Table 18.6** Values for the enforcePortletSecurity Property

Property Value	Description
true	This 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 (default)	If enforcePortletSecurity is set to false or is absent from the file, 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.)

After you make changes to the **BIDashboard.config** file, you must restart the servlet container or J2EE application server before your changes take effect.

## (Optional) Create Additional Dashboard Groups

### Overview of Creating Additional Dashboard Groups

The topic “Manage Users in Dashboard Groups” on page 311 describes a Dashboard Admins group that has full access to dashboard objects and a Dashboard Users group that has read-only access. This configuration is adequate for sites with a limited number of users who create and administer dashboards. However some sites might want to implement more granular access controls. For example, a site might want to separate the modeling responsibilities from the administrative functions.

To achieve more granular access controls, you can do the following:

- 1 Define additional dashboard groups in metadata and add users to those groups. For instructions on creating groups and adding users, see the User Manager Help in SAS Management Console.
- 2 Set permissions in order to enable or disable a group's access to various dashboard objects. You can accomplish this by setting permissions on the object's respective subfolder under the BIDashboard folder. For instructions on setting permissions, see “Verify or Set Permissions” on page 312.

For example, you might create new groups and apply permissions as described in the following sections. These examples assume that dashboard security has been enabled (see “Enable Dashboard Security” on page 310).

### Example Group: Dashboard Modelers

Members of this group can create new dashboards, indicators, models, and ranges.

To configure this group, complete these steps:

- 1 Define the Dashboard Modelers group in SAS Management Console and add users to the group.
- 2 Add the Dashboard Modelers group as a member of the Dashboard Users group. (Users should not be explicit members of both Dashboard Modelers and Dashboard Users groups because this will result in conflicting permissions.)
- 3 Explicitly grant the Dashboard Modelers group ReadMetadata, WriteMetadata, Read, Write, and Create permissions on the following subfolders under the BIDashboard folder:
  - Dashboards
  - Indicator Definitions
  - DataPoint Models

Though members of this group have full access to objects that they create, members have read-only access to any data that already exists in metadata.

### **Example Group: Dashboard Analysts**

Members of this group can create new dashboards. Members have read-only access to any data that already exists in metadata.

To configure this group, complete these steps:

- 1 Define the Dashboard Analysts group in SAS Management Console and add users to the group.
- 2 Add the Dashboard Analysts group as a member of the Dashboard Users group. (Users should not be explicit members of both Dashboard Analysts and Dashboard Users groups because this can result in conflicting permissions.)
- 3 Explicitly grant this group ReadMetadata, WriteMetadata, Read, Write, Create permissions on the Dashboards subfolder under the BIDashboard folder.
- 4 Explicitly grant this group Read and ReadMetadata permissions on the Indicator Definitions and DataPoint Models subfolders.

### **Example Group: Finance Users**

Members of the Finance Users group can view a particular dashboard. Other dashboard users and the general public are restricted from seeing the dashboard.

This example assumes that data modelers have created a Finance Dashboard along with two indicators named Finance Indicator 1 and Finance Indicator 2, and that both indicators use the same model named Finance Model. You want to ensure that only members of the Finance Users group can view the dashboard in the portal.

To configure this group, complete these steps:

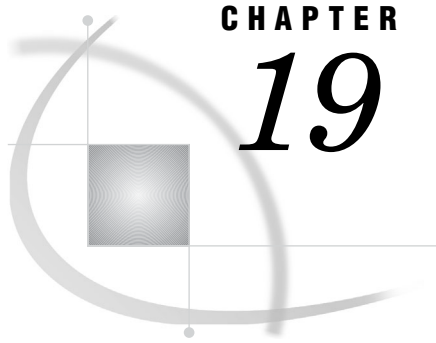
- 1 Define the Finance Users group in SAS Management Console and add users to the group.
- 2 Add the Finance Users group as a member of the Dashboard Users group. (Users should not be explicit members of both Finance Users and Dashboard Users groups because this can result in conflicting permissions.)
- 3 Set permissions directly on the following dashboard objects:

**Table 18.7** Finance Dashboard Objects That Require Permissions

<b>Subfolder Under BIDashboard</b>	<b>Example Dashboard Objects</b>
Dashboards	Finance Dashboard
Indicator Definitions	Finance Indicator 1 Finance Indicator 2
Data Point Models	Finance Model

Set permissions on these dashboard objects as follows:

- Explicitly grant the Finance Users group Read and ReadMetadata permissions on the Finance Dashboard objects. Explicitly deny this group all other permissions.
- Explicitly deny the Dashboard Users group all permissions on the Finance Dashboard objects.
- Explicitly deny the PUBLIC and SAS Users groups all permissions on the Finance Dashboard objects.



# CHAPTER 19

## Customizing the Portal's Display

<i>Overview of Portal Customization</i>	<b>317</b>
<i>Changing the Default Preferences</i>	<b>318</b>
<i>Overview of the Default Preferences</i>	<b>318</b>
<i>Options for Changing the Default Preferences</i>	<b>318</b>
<i>Changing the Default Preferences for your Entire Implementation</i>	<b>319</b>
<i>Descriptions of the Preferences</i>	<b>320</b>
<i>SoftwareComponent Element</i>	<b>323</b>
<i>Upgrading 9.1.2 Preferences to the 9.1.3 Preferences Format</i>	<b>327</b>
<i>Theme Deployment</i>	<b>328</b>
<i>Overview of Theme Deployment</i>	<b>328</b>
<i>Rules for Compressing Theme Files</i>	<b>328</b>
<i>Deploy a New Theme to the Servlet Container</i>	<b>329</b>
<i>Deploy a New Theme to an HTTP Server</i>	<b>330</b>
<i>Migrate Existing Themes to an HTTP Server</i>	<b>331</b>
<i>Changing the Default Theme</i>	<b>332</b>
<i>Deleting Custom-Developed Themes</i>	<b>333</b>

### Overview of Portal Customization

SAS provides tools for customizing the portal's appearance so that it is suitable for your organization. When you customize the portal's appearance, the changes that you make are seen by all portal users.

You can customize all portal views in the following ways:

- Change the default preferences, which determine the date format, locale, navigation, and other aspects of the portal. (Individual users who log on to the portal can override the default preferences by using the SAS Preferences Web application.) For information about changing the default preferences, see “Changing the Default Preferences” on page 318.
- Add company logos, images, and other content to the Home page template. An initial Home page template was created during installation. For details about the Home page template, see “Loading Initial Metadata” on page 200 and “Page Templates” on page 247.
- Add company logos, images, and other content to the Public Kiosk. For more information about the Public Kiosk, see “Administering the Public Kiosk” on page 202.
- Create and deploy your own custom theme. Here are the main tasks involved with deploying custom themes:
  - 1 Your developers create a custom theme and specify the logos, colors, and fonts that best suit your organization. Your developers can also specify the name of

the application that appears in the banner of the portal. For details, see “Developing Custom Themes” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/themes/index.html](http://support.sas.com/rnd/itech/doc9/portal_dev/themes/index.html).

- 2 You deploy the custom theme. For instructions, see “Theme Deployment” on page 328.
- 3 Optionally, specify your custom theme as the default theme. All portal users will then see the theme when they log on to the portal. Users can later change the default theme that is used for their personal portal views. To change the default theme, see “Changing the Default Theme” on page 332.

In addition to customizing the portal display, individual users can personalize their own portal views. For example, users can create links to frequently-visited Internet sites, change the order in which pages appear in the portal, and move the navigation bar from the top of the page to the side of the page. Information about personalizing portal views is available in the portal’s online Help.

---

## Changing the Default Preferences

---

### Overview of the Default Preferences

When you installed and configured the portal Web application, you set default values for particular portal preferences. These default preferences apply globally for any user in your organization who logs on to the portal Web application.

The SAS Preferences Web application manages user preferences for the portal Web application and for SAS solutions that are delivered through the portal Web application. If the SAS Information Delivery Portal has been installed, then portal users can change the preferences in their portal views by selecting a new preference in the SAS Preferences Web application. The SAS Preferences Web application enables users to specify preferences for the following features:

- date format
- locale
- themes
- e-mail
- navigation
- notification
- time format

User-specified preferences apply only for the user who is logged on to the portal Web application. User-specified preferences will override the default preferences that have been set. For details about the SAS Preferences Web application, refer to the online Help.

After installation, you can change the default preferences. The changes that you make apply to all portal users all SAS solutions that your organization has installed to run in the portal.

---

### Options for Changing the Default Preferences

You can change the default preferences in the following ways:



- Change a single default preference.

Modify and run the **UpdatePreferenceDefault.sas** file to set an individual default preference. You can use **UpdatePreferenceDefault.sas** to change only one preference at a time, but **UpdatePreferenceDefault.sas** can be run any number of times. The **UpdatePreferenceDefault.sas** file is located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory.

- Change several or all default preferences at once.

If you need to change several preferences, then you can remove and then reset all of your preference definitions. For instructions, see “Changing the Default Preferences for your Entire Implementation” on page 319.

- Update only the default theme.

If you want to change only the default theme, then you can run the **UpdateDefaultTheme.sas** program instead of **UpdatePreferenceDefault.sas**. With **UpdateDefaultTheme.sas**, you don't need to modify the file before you run it. For details, see “Changing the Default Theme” on page 332.

- Change the list order for packages that are published to a channel.

By default, packages that are published to a channel are listed with the oldest package at the top and the newest package at the bottom. Starting with hot fix 913WEBINFRAKIT02, you can change the order so that the newest packages are listed at the top, and the oldest packages are at the bottom.

To change the list order for packages, run **LoadPackageOrderPreference.sas** or **LoadPackageOrderPreference\_utf8.sas**. Both files are located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory.

After you successfully run **LoadPackageOrderPreference\*.sas**, the preference for package list order becomes available in the portal, and the newest packages in a channel will be listed at the top. As with other preferences, portal users can override this preference in their portal views via the SAS Preferences Web application. If you later want to revert back to the original list order, you can edit and run **UpdatePreferenceDefault.sas**

If you need to upgrade your preferences to the 9.1.3 format, see “Upgrading 9.1.2 Preferences to the 9.1.3 Preferences Format” on page 327.

---

## Changing the Default Preferences for your Entire Implementation

To change the default preferences for your entire implementation, complete these steps:

- 1 Run **Remove913PreferenceDefs.sas** or **Remove913PreferenceDefs\_utf8** (uses UTF-8 character encoding) to remove all of your preference definitions. The **Remove913PreferenceDefs\*.sas** files are located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory.
- 2 Edit the **LoadDefaultPreferences.sas** or **LoadDefaultPreferences\_utf8.sas** file as follows:

To change all preferences except the navigation bar position, locate the **SoftwareComponent** element for the SAS Application Infrastructure. For the structure of the **SoftwareComponent** element, see “SoftwareComponent Element” on page 323.

- a Using the table in “Descriptions of the Preferences” on page 320 for guidance, locate the **Property** elements that you want to update and determine the valid default values for the properties.

*Note:* For the **Property** element named `Locale.DefaultLocale`, the portal Web application uses the default language specified by your browser instead of the **DefaultValue** attribute specified in the **SoftwareComponent** element of the `LoadDefaultPreferences.sas` file. Do not change the **DefaultValue** attribute for `Locale.DefaultLocale`.  $\triangle$

(The `LoadDefaultPreferences.sas` contains a **Prototype** element for each preference; the **Prototype** element specifies the valid default values for that preference. When you update the **DefaultValue** attribute for a **Property** element (in the **SoftwareComponent** element), you must specify one of the valid values listed in the corresponding **Prototype** element. These valid values are listed in the following table.)

- b In the **Property** elements, update the **DefaultValue** attribute for any default preferences that you want to update.

To change the navigation bar position, locate the **PropertyGroup** element named `Application Configuration`. In the **Property** element named `Portal.PreferredNavigation`, update the **DefaultValue** attribute with the new default navigation bar position. (See the table in “Descriptions of the Preferences” on page 320 for valid values).

- 3 Run the `LoadDefaultPreferences.sas` file to update the default values for the preferences.

---

## Descriptions of the Preferences

The following table lists the valid default values for each of the preferences in the `LoadDefaultPreferences.sas` file (described in “Changing the Default Preferences for your Entire Implementation” on page 319 ). The table also specifies whether the default values can be updated by portal users via the SAS Preferences Web application.

**Table 19.1** Valid Values for Preferences in `LoadDefaultPreferences.sas`

Property Name of the Preference	Valid Values	Description	Updatable in SAS Preferences Web Application
<code>Email.SMTPHost</code>	<code>mailhost.company.com**</code> <a valid SMTP host>	host name of mail server (SMTP)	no
<code>Email.DefaultFrom</code>	<code>Company**</code> <a valid name for the From field>	default value for the FROM field in e-mail sent by the portal Web application	no
<code>Email.Admin</code> <code>EmailAddress</code>	<code>admin@company.com**</code> <a valid e-mail address for the system administrator>	default e-mail address for the system administrator	no
<code>Email.ErrorNotification</code> <code>List</code>	<code>admin@company.com**</code> <a valid list of e-mails>	one or more e-mail addresses to which notifications of system errors or other messages will be sent	no

Property Name of the Preference	Valid Values	Description	Updatable in SAS Preferences Web Application
Email.HTMLFormatFlag	true false	indicates whether to use the text MIME type or the html MIME type for e-mail	no
Email.Charset	UTF8** <a valid e-mail character set>	character set that is used to encode e-mail	no
Email.SMTPPort	25** <a valid SMTP port>	default port for e-mail	no
CharacterEncoding.Input	ISO8859_1** <a valid character encoding input>	character encoding that is used for Web input	no
CharacterEncoding.Output	UTF8** <a valid character encoding output>	character encoding that is used for Web output	no
Default.Theme	default** <a theme name>	default display theme for Web applications	yes
Notifications.EmailType	text (Plain-text e-mail) HTML** (HTML-formatted e-mail) digested (Digested e-mail)	the format in which you want to receive E-mail Notifications that are generated by SAS solutions that run inside the portal.	yes
Notifications.AlertsType	email (Via e-mail) portal** (My alerts portlet) emailandportal (Both e-mail and alerts portlet)	the format in which you want to receive Alerts Notifications that are generated by SAS solutions that run inside the portal.	yes
Format.TimeDate	standard** (dd-MMM-yyyy HH:mm:ss) yearmonthdatetime (yyyy-MMM-d HH:mm:ss aa)	format settings for dates and times for the SAS solutions that run inside the portal.	yes
Format.LongDate	eeemmmddyyyy** (EEE MMM dd yyyy) mmmmddyyyy (MMMM dd, yyyy) ddmmmyyyy (dd MMMM yyyy)	format settings for long dates for the SAS solutions that run inside the portal.	yes

Property Name of the Preference	Valid Values	Description	Updatable in SAS Preferences Web Application
Format.ShortDate	ddmmmyyyy (dd MMMM yyyy) ddmmyyyy (dd.MM.yyyy) ddmmmyyyy (dd.MMM.yyyy) ddmmyyyy (dd-MM-yyyy) ddmmmyyyy (dd-MMM-yyyy) ddmmyyyy (dd/MM/yyyy) ddmmmyyyy (dd/MMM/yyyy) mmmddyyyy (MMM dd yyyy) mmdyyyyy.dot (MM.dd.yyyy) mmdyyyyy.dash (MM-dd-yyyy) mmdyyyyy.slash** (MM/dd/yyyy) yyyymmdd.dot (yyyy.MM.dd) yyyymmdd.dash (yyyy-MM-dd) yyyymmdd.slash (yyyy/MM/dd) yyyymmdd.dot (yyy.MMM.dd) yyyymmdd.dash (yyyy-MMM-dd) yyyymmdd.slash (yyyy/MMM/dd)	format settings for short dates for the SAS solutions that run inside the portal.	yes
Format.Currency	parenthesis** ((\$1,234.56)) negsign (-\$1,234.56)	format settings for currency numbers for the SAS solutions that run inside the portal.	yes
Format.CurrencyDisplay	symbol** (Symbol (US\$)) isocode (ISO Code (USD))	format settings for currency display for the SAS solutions that run inside the portal.	yes

Property Name of the Preference	Valid Values	Description	Updatable in SAS Preferences Web Application
Version.Number	9.1.3** <a version number>	version number or class to use	no
Portal.Preferred Navigation	Horizontal** (Top) Vertical (Side)	the position of the portal navigation bar	yes

\*\*This is the default preference in the version of the **LoadDefaultPreferences.sas** file that is shipped with the portal Web application.

## SoftwareComponent Element

Here are the contents of the **SoftwareComponent** element of the **LoadDefaultPreferences.sas** file, which you should use to specify default preferences for your entire company. For a list of valid default values (**DefaultValue** attribute), see “Descriptions of the Preferences” on page 320.

*Note:* For the **Property** element named **Locale.DefaultLocale**, the portal Web application uses the default language that is specified by your browser instead of the **DefaultValue** attribute that is specified in the **SoftwareComponent** element of the **LoadDefaultPreferences.sas** file. Do not change the **DefaultValue** attribute for **Locale.DefaultLocale**.

△

```
<SoftwareComponent Name="SAS Application Infrastructure"
  Desc="SAS Application Infrastructure"
  ProductIdentifier="30"
  Id="$NewGlobalSWComponent">

  <PropertyGroups>
  <PropertyGroup Name="Application Configuration"
    Desc="Common properties for SAS">
    <GroupedProperties>

    <Property Name="Locale.DefaultLocale"
      PropertyName="Locale.DefaultLocale"
      Desc="Default to use App Server's Locale Locale"
      DefaultValue="en_US" >
    <OwningType>
    <PropertyType ObjRef="$Stringtypeforpropertyobjects" />
    </OwningType>
    <UsingPrototype>
    <Prototype ObjRef="$LocaleDefaultLocaleGUIDEF" />
    </UsingPrototype>
    </Property>

    <Property Name="Email.SMTPHost"
      PropertyName ="Host name of mail server (SMTP)"
      Desc="The SMTP mail host for your organization"
      DefaultValue="mailhost.fyi.sas.com">
```

```

<OwningType>
<PropertyType ObjRef="Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Email.DefaultFrom"
PropertyName="Value of FROM field"
Desc="The default value for the FROM
field in email sent by the app"
DefaultValue="SAS">
<OwningType>
<PropertyType ObjRef="Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Email.AdminEmailAddress" \
PropertyName="Administrator email address"
Desc="The default email address for the
system administrator"
DefaultValue="sasadmin@sas.com">
<OwningType>
<PropertyType ObjRef="Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Email.ErrorNotificationList"
PropertyName="Recipients of error notifications"
Desc="One or more email addresses
to which notifications of system
errors or other messages are sent"
DefaultValue="sasadmin@sas.com" >
<OwningType>
<PropertyType ObjRef="Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Email.HTMLFormatFlag"
PropertyName="Email.HTMLFormatFlag"
Desc="Flag indicating whether text/html MIME
type is used for email"
DefaultValue="true" >
<OwningType>
<PropertyType ObjRef="Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Email.Charset"
PropertyName="Email.Charset"
Desc="The default value for the charset used
to encode email"
DefaultValue="UTF8" >
<OwningType>
<PropertyType ObjRef="Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Email.SMTPPort"
PropertyName="Email.SMTPPort"
Desc="The default port for email"
DefaultValue="25" >

```

```

<OwningType>
<PropertyType ObjRef="$$Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="CharacterEncoding.Input"
PropertyName="CharacterEncoding.Input"
Desc="Character encoding used for Web input"
DefaultValue="ISO8859_1">
<OwningType>
<PropertyType ObjRef="$$Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="CharacterEncoding.Output"
PropertyName="CharacterEncoding.Output"
Desc="Character encoding used for Web output"
DefaultValue="UTF8">
<OwningType>
<PropertyType ObjRef="$$Stringtypeforpropertyobjects" />
</OwningType>
</Property>
<Property Name="Default.Theme"
PropertyName="Default Theme"
Desc="Default display theme for Web applications"
DefaultValue="default" >
<OwningType>
<PropertyType ObjRef="$$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$$DefaultThemeGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Notifications.EmailType"
PropertyName="Notifications.EmailType"
Desc="Default type of email"
DefaultValue="HTML">
<OwningType>
<PropertyType ObjRef="$$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$$NotificationsEmailTypeGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Notifications.AlertsType"
PropertyName="Notifications.AlertsType"
Desc="Default source for alert notifications"
DefaultValue="Portal" >
<OwningType>
<PropertyType ObjRef="$$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$$NotificationsAlertsTypeGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Format.TimeDate"

```

```

Property Name="Format.TimeDate"
PropertyDesc="Default time/date format"
PropertyDefaultValue="standard">
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$FormatTimeDateGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Format.LongDate"
Property Name="Format.LongDate"
PropertyDesc="Default long date format"
PropertyDefaultValue="eeemmmddyyyy">
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$FormatLongDateGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Format.ShortDate"
Property Name="Format.ShortDate"
PropertyDesc="Default short date format"
PropertyDefaultValue="mmddyyyy.slash">
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$FormatShortDateGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Format.Time"
Property Name="Format.Time"
PropertyDesc="Default time format"
PropertyDefaultValue="HHmmssaa">
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$FormatTimeGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Format.Currency"
Property Name="Format.Currency"
PropertyDesc="Default currency number format"
PropertyDefaultValue="parenthesis">
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$FormatCurrencyGUIDEF" />
</UsingPrototype>
</Property>

```



```

<Property Name="Format.CurrencyDisplay"
PropertyName="Format.CurrencyDisplay"
Desc="Default currency display format"
DefaultValue="symbol">
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
<UsingPrototype>
<Prototype ObjRef="$FormatCurrencyDisplayGUIDEF" />
</UsingPrototype>
</Property>
<Property Name="Version.Number"
PropertyName="Version.Number"
Desc="Version number or class to use"
DefaultValue="9.1.2" >
<OwningType>
<PropertyType ObjRef="$Stringtypeforpropertyobjects" />
</OwningType>
</Property>
</GroupedProperties>
</PropertyGroup>
</PropertyGroups>
</SoftwareComponent>

```

---

## Upgrading 9.1.2 Preferences to the 9.1.3 Preferences Format

For the 9.1.3 version of the portal Web application, the format for the preferences metadata has changed from the 9.1.2 preferences format. Therefore, if you have installed the 9.1.2 portal Web application, you must upgrade your preferences metadata by deleting the previously loaded default preferences, and loading 9.1.3 default preferences. Upgrading your 9.1.2 preferences affects your 9.1.2 default preferences as follows:

- Default preferences that were loaded using the SAS program **LoadDefaultPreferences.sas**: If you manually updated any of the default values for 9.1.2 preferences, then you must specify the updated default values in the 9.1.3 version of the **LoadDefaultPreferences.sas** file, and then run the program again.
- Default preferences that were selected by the individual user with the SAS Preferences Web application: If individual users have used the SAS Preferences Web application to customize their preferences, then these default preferences will not be changed when you replace the 9.1.2 preferences with 9.1.3 preferences.

To upgrade 9.1.2 preferences to the preference format for 9.1.3:

- 1 Run the **Remove912PreferenceDefs.sas** SAS file, which is located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory.
- 2 If you need to change any of the global default preferences, edit the **LoadDefaultPreferences.sas** file (in the **OMR** directory) and change the appropriate values. For details about changing the default preferences, see “Changing the Default Preferences” on page 318.
- 3 Run the SAS program **LoadDefaultPreferences.sas**.

After you run the SAS program **LoadDefaultPreferences.sas**, your default preferences are upgraded to the 9.1.3 preferences metadata format.

---

## Theme Deployment

---

### Overview of Theme Deployment

The installation of the portal Web application includes a default theme that specifies colors, fonts, and graphics for the user interface of the portal Web application and the SAS solutions that run in the portal. You can use the default theme as a basis for creating as many additional themes as you want. For details about creating new themes, see “Developing Custom Themes” in the *SAS Web Infrastructure: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/themes/index.html](http://support.sas.com/rnd/itech/doc9/portal_dev/themes/index.html).

After you create a new theme, you must deploy the theme so that it will be available to the portal Web application and to other SAS applications. You have two options for deploying the theme:

- Deploy the new custom theme to your servlet container. This is where SAS deploys the default theme when you install and configure the portal application.
- Deploy the new custom theme to a separate HTTP server, such as Apache HTTP Server. Because theme files are static, they do not require the services of a J2EE server application. Allowing an HTTP server to handle as much static content as possible is good practice because it frees the J2EE application server to handle Web applications that are resource intensive. You can use any HTTP 1.1 compliant server for this purpose.

If you choose to deploy custom themes to an HTTP server, you might want to migrate the default theme to that HTTP server as well.

After deploying a new theme, you can specify that theme as the default theme in the portal. For details, see “Changing the Default Theme” on page 332.

---

### Rules for Compressing Theme Files

When following the procedure to deploy themes, you might be instructed to compress all of the theme’s files into a single package. Here are some general rules for compressing theme files:

- Use the Java **jar** command to compress the files rather than an alternative zip program. Using programs such as WinZip can cause problems.
- Compress the theme’s directory, files, and sub-directories into a jar that has an extension of **.war**.
- Include the **WEB-INF** directory and the **web.xml** file that it contains. This file is required in order for the theme to be displayed.
- Make sure that the WAR file name does not contain spaces and does not begin with the characters **SASTheme**. (**SASTheme** is reserved for the default theme.)
- When compressing the files, run the jar command from the deployment directory and maintain the theme’s directory structure. For example, if you deploy to Tomcat, you would use a directory structure similar to this:

```
\Tomcat4.1\webapps\MyTheme
  \themes
  \WEB_INF
```

To run the `jar` command for this example, navigate to `\Tomcat4.1\webapps\MyTheme` and issue a command similar to this:

```
jar -cvf MyTheme.war .
```

---

## Deploy a New Theme to the Servlet Container

Follow these instructions to deploy a new theme to your servlet container:

- 1 Locate the directory containing all the files required for the theme. This directory should have the same name as the theme.
- 2 Compress the theme's directory, files, and sub-directories into a WAR file. When creating the WAR file, follow the recommendations outlined in "Rules for Compressing Theme Files" on page 328.

*Note:* Instead of creating a WAR file, you can place the directory that contains the new theme directly into the servlet container.  $\Delta$

- 3 Deploy the WAR file by using the appropriate procedures for the J2EE servlet container or J2EE application server.
- 4 Stop the servlet container or J2EE application server.
- 5 In SAS, open the program `LoadThemeConnection.sas`, which is located in the `SAS-install-dir\Web\Portal2.0.1\OMR` directory.
- 6 In the `LoadThemeConnection.sas` file, modify the following fields :

`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 `$SERVICES_OMI_PORT$` property in the `install.properties` file (located in the `PortalConfigure` subdirectory of the setup directory).

`metauser="user ID"`

Specify the user ID to use to connect to the SAS Metadata Server. This user ID is typically the SAS Administrator (sasadm). For Windows users, the user ID is qualified by the domain or the machine name, for example:

```
<machine or Windows domain>\sasadm
```

`metapass="password"`

Specify the password for the metauser.

`metarepository="repository";`

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the `$SERVICES_OMI_REPOSITORY$` property in the `install.properties` file (located in the `PortalConfigure` subdirectory of the portal's installation directory).

`%let SWCName=SASTheme_default;`

Replace `SASTheme_default` with the name of your WAR file, without the characters `.WAR`. Follow the name with a semicolon (;).

*Note:* If you did not create a WAR file, then change `SASTheme_default` to the name of the directory that you placed in the servlet container, and follow the name with a semicolon (;).  $\Delta$

`%let protocol=http;`

If you are using Secure Sockets Layer, then replace `http` with `https`.

`%let desc=The URL to the root context of the theme WAR;`

This is a textual description of the theme connection, and the text does not require any changes. If you choose to replace the text, then make sure the new text ends with a semicolon (;).

```
%let hostName=Host Name;
```

Specify the host name of the machine on which the theme is deployed, followed by a semicolon (;).

```
%let port=Port;
```

Specify the port number of the Web server on which the theme is deployed, followed by a semicolon (;).

```
%let service=SASTheme_default;
```

Replace *SASTheme\_default* with the name of your WAR file, but omit the characters *.WAR*. Follow the name with a semicolon (;).

*Note:* If you did not create a WAR file, then change *SASTheme\_default* to the name of the directory that you placed in the servlet container, and follow the name with a semicolon (;).  $\Delta$

- 7 When you have completed your edits, save **LoadThemeConnection.sas** with your changes.
- 8 Run the **LoadThemeConnection.sas** program. This program updates your metadata repository with information about the new theme.
- 9 Stop and restart the SAS Services application.
- 10 Start your servlet container or J2EE application server.

After performing the above steps, users will see the new theme as a selection on the Preferences page in the portal Web application.

If you later want to replace this theme with an updated version, repeat the previous steps 1-3, and overwrite the existing theme files with the updated files. If you make any changes to the HTML templates in the theme, you should stop and restart the servlet container or J2EE application server. You don't need to stop and restart the SAS Services application unless you make configuration changes that affect these services.

---

## Deploy a New Theme to an HTTP Server

Follow these instructions to deploy a new theme to an HTTP server. You can use any HTTP 1.1 compliant server:

- 1 Locate the directory containing all the files required for the theme. This directory should have the same name as the theme.
- 2 In the HTTP server's document area, create a directory with the same name as the theme.
- 3 Copy the theme's directory, files, and sub-directories into the directory that you created.
- 4 Stop and restart the HTTP server.
- 5 Stop the servlet container or J2EE application server where the portal Web applications are running. This step is required in order for these applications to recognize the new theme.
- 6 In SAS, open the program **LoadThemeConnection.sas** and edit that file to provide metadata information about the new theme. For descriptions of the fields in this file, see the previous procedure ("Deploy a New Theme to the Servlet Container" on page 329).
- 7 When you have completed your edits, save **LoadThemeConnection.sas** with your changes.

- 8 Run the **LoadThemeConnection.sas** program. This program updates your metadata repository with information about the new theme.
- 9 Stop and restart the SAS Services application.
- 10 Start your servlet container or J2EE application server.

After performing the above steps, authorized users will see the new theme as a selection on the Preferences page in the portal Web application.

If you later want to replace this theme with an updated version, just repeat the previous steps 1-3, and overwrite the existing theme files with the updated files. If you make any changes to the HTML templates in the theme, you should stop and restart the servlet container or J2EE application server. You don't need to stop and restart the SAS Services application unless you make configuration changes that affect these services.

---

## Migrate Existing Themes to an HTTP Server

The portal Web application contains two theme Web applications: **SASTheme\_default.war** and, starting with Service Pack 2, **SASTheme\_winter.war**. You initially deployed these themes to your servlet container or J2EE application server when you ran the portal's installation and configuration programs. If you later deploy your own custom themes to an HTTP server, you should migrate these existing themes to that HTTP server as well.

*Note:* To understand important concepts about using an HTTP server for theme content, or for example of setting up communication between the HTTP server and the servlet container, see "Configuring an HTTP Server to Serve Static Content for SAS Web Applications" on page 86. △

Follow these instructions to migrate the existing themes from the servlet container or J2EE application server to an HTTP server:

- 1 In the HTTP server's document area, create a directory named **SASTheme\_default**.
- 2 Locate the **SASTheme\_default.war** file in the **Portal12.0.1** directory of your portal installation.
- 3 Extract the contents of the **SASTheme\_default.war** file to the directory that you created (**SASTheme\_default**).
- 4 Deploy the **SASTheme\_winter.war** files in a similar way. Create a **SASTheme\_winter** directory in the HTTP server's document area, and then extract the contents of the **SASTheme\_winter.war** file to that directory.
- 5 Stop and restart the HTTP server.
- 6 Update the themes connection URL that is stored in metadata so that the URL targets the HTTP server. To update this metadata, you modify and run the **UpdateThemeConnection.sas** program.

*Note:* You must modify and run **UpdateThemeConnection.sas** separately for **SASTheme\_default** and for **SASTheme\_winter**. For instructions, see "Redistributing the SAS Themes Web Application" on page 351. △

- 7 Stop and restart the SAS Services application.
- 8 Optionally, remove the themes from the servlet container. To remove the themes, manually delete the theme files (Tomcat) or use the administrator console (WebLogic and WebSphere) to remove the themes. One reason to remove the themes is to avoid possible confusion if you upgrade to a new release or service pack. If the themes reside on both the servlet container and the HTTP server, then it's possible that you might accidentally update the themes in the wrong location.
- 9 If the servlet container or J2EE application server is running, stop and restart it.

After performing the above steps, authorized users will see the new theme as a selection on the Preferences page in the portal Web application.

If you later need to replace either theme with an updated version, just repeat the previous steps 1-3 to overwrite the existing theme files with the updated files, and then stop and restart the HTTP server. If you make any changes to the HTML templates in the theme, you should stop and restart the SAS Services application.

---

## Changing the Default Theme

After you have deployed a new theme, you can use the program **UpdateDefaultTheme.sas** to specify the new theme as the default theme. The new theme will then be in effect for users who have not selected a different theme on the Preferences page (the SAS Preferences Web application).

To specify a new theme as the default theme, complete these steps :

- 1 Modify the following lines in the SAS program **UpdateDefaultTheme.sas** (located in the **OMR** directory of the portal installation):

`options metaserver="host"`

Specify the host name of your SAS Metadata Server (for example, localhost or *<machine.mycompany.com>*). Use the value of the `$SERVICES_OMI_HOST$` property in the **install.properties** file (located in the **PortalConfigure** subdirectory of the portal's installation directory).

`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 `$SERVICES_OMI_PORT$` property in the **install.properties** file.

`metauser="user ID"`

Specify the user ID to use to connect to the SAS Metadata Server. This user ID is typically the SAS Administrator (sasadm). For Windows users, the user ID is qualified by the domain or the machine name, for example:

*<machine or Windows domain>\sasadm*

`metapass="password"`

Specify the password for the metauser.

`metarepository="repository";`

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 `$SERVICES_OMI_REPOSITORY$` property in the **install.properties** file.

`%let defaultTheme=theme name;`

Specify the name of the theme that is to be the default for the Portal Web application, followed by a semicolon (;). The theme name and case must exactly match the name and case that's in the theme's descriptor (XML) file. For example:

`%let defaultTheme=winter;`

- 2 Run the SAS program **UpdateDefaultTheme.sas**.

## Deleting Custom-Developed Themes

If you need to delete a custom-developed theme from the deployment for the portal Web application, you can use the SAS program **DeleteThemeConnection** to delete the theme's metadata from the SAS Metadata Repository. If users have selected this theme (using the SAS Preferences Web application), then the selected theme will be changed to the default theme.

*Note:* It is recommended that you test your custom-developed themes in a non-production environment so that you do not need to delete the theme connection metadata from a production SAS Metadata Server. △

Use these steps to delete a custom-developed theme from the portal Web application's deployment:

- 1 Modify the following lines in the **DeleteThemeConnection.sas** file (located in the **OMR** directory of your installation):

```
options metaserver="host"
```

Specify the host name of your SAS Metadata Server (for example, localhost or *<machine.mycompany.com>*). Use the value of the `$SERVICES_OMI_HOST$` property in the **install.properties** file (located in the **PortalConfigure** subdirectory of the portal's installation directory).

```
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 `$SERVICES_OMI_PORT$` property in the **install.properties** file.

```
metauser="user ID"
```

Specify the user ID to use to connect to the SAS Metadata Server. This user ID is typically the SAS Administrator (sasadm). For Windows users, the user ID is qualified by the domain or the machine name, for example:

```
<machine or Windows domain>\sasadm
```

```
metapass="password"
```

Specify the password for the metauser.

```
metarepository="repository";
```

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the `$SERVICES_OMI_REPOSITORY$` property in the **install.properties** file.

```
%let themeName=theme name;
```

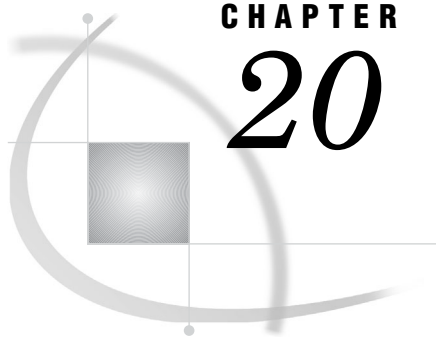
Specify the name of the theme to delete, followed by a semicolon (;).

- 2 Save **DeleteThemeConnection.sas** with your changes, and then run it.

After you run the **DeleteThemeConnection** SAS program, the theme will no longer be available as a theme selection in the SAS Preferences Web application.







## CHAPTER

## 20

## Foundation Services and WebDAV Server Deployment

<i>Overview of the SAS Foundation Services That Are Used by the Portal</i>	<b>335</b>
<i>Introduction to the SAS Foundation Services</i>	<b>335</b>
<i>Summary of the Foundation Services That Are Used by the Portal</i>	<b>336</b>
<i>Service Deployment Configurations</i>	<b>336</b>
<i>Understanding Service Deployment Configurations</i>	<b>336</b>
<i>Ensuring JRE Communication Between the Portal and the SAS Services Application</i>	<b>338</b>
<i>Changes That Require You to Reimport the Service Deployment Configurations</i>	<b>339</b>
<i>Reimport the Service Deployment Configurations</i>	<b>340</b>
<i>SAS Foundation Service Deployment and Use</i>	<b>341</b>
<i>Overview of SAS Foundation Service Deployment and Use</i>	<b>341</b>
<i>How the Portal Web Application Deploys SAS Foundation Services</i>	<b>341</b>
<i>Accessing Service Deployments from the SAS Metadata Server</i>	<b>342</b>
<i>Accessing Service Deployments from XML Files</i>	<b>343</b>
<i>How the Portal Web Application Components Are Distributed</i>	<b>344</b>
<i>How Applications Locate the SAS Foundation Services</i>	<b>344</b>
<i>How the Portal Web Application Shares SAS Foundation Services</i>	<b>344</b>
<i>Run Remotely Deployed Services as a Windows Service</i>	<b>345</b>
<i>WebDAV Server Metadata</i>	<b>345</b>

### Overview of the SAS Foundation Services That Are Used by the Portal

#### Introduction to the SAS Foundation Services

SAS Foundation Services is a set of infrastructure and extension services that support the development of integrated, scalable, and secure applications based on Java. The design model for SAS Foundation Services supports both local and remote resource deployment, and promotes resource sharing among applications.

The portal Web application uses both local and remote deployments of the foundation services. These are described in “Service Deployment Configurations” on page 336. When you installed the portal software, these deployments were configured for your environment. You typically do not need to perform any configuration after installation in order to use the portal.

Understanding the service configurations, and how the portal Web application deploys and accesses the services, is helpful for these reasons:

- In order for the portal Web application to access the SAS Foundation Services, the services must be deployed. The portal Web application deploys the local foundation services. An application called SAS Services Application is used to remotely deploy the foundation services. As part of your administrative tasks, you might

occasionally need to restart the SAS Services Application, and then to restart the servlet container.

- For some configuration changes, such as setting up Web authentication, you might need to re-import the foundation services into the metadata repository. It's useful to understand what you are importing.
- You can use the SAS Foundation Services to develop custom applications and portlets that integrate with the portal Web application.

For details, see “Using SAS Foundation Services With the Portal” in the *SAS Web Infrastructure Kit: Developer's Guide*, at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/webapps/dg\\_found.html](http://support.sas.com/rnd/itech/doc9/portal_dev/webapps/dg_found.html). See also the SAS Foundation Services class documentation at <http://support.sas.com/rnd/gendoc/bi/api/Foundation/overview-summary.html>.

- Understanding how the portal Web application accesses the foundation services can be helpful if you plan to redistribute your portal Web application implementation.

## Summary of the Foundation Services That Are Used by the Portal

The portal Web application uses the following core foundation services for infrastructure:

- Connection Service, for IOM connection management
- Discovery Service, for locating and binding to deployed services
- Event Services, for event notification and information delivery
- Information Service, for repository federation, searching repositories, a common entity interface, and creating personal repositories
- Logging Service, for run-time execution tracing and error tracking
- Publish Service, for access to the Publishing Framework
- Security Service, for user authentication, content authorization, action authorization, and task authorization
- Session Service, for context management, resource management, and context passing
- Stored Process Service, for access to stored process execution and package navigation
- User and Authentication Service, for access to authenticated user context, access to global, solution-wide, and application-specific profiles, and access to personal objects such as ad hoc results, bookmarks, documents, and alerts

## Service Deployment Configurations

### Understanding Service Deployment Configurations

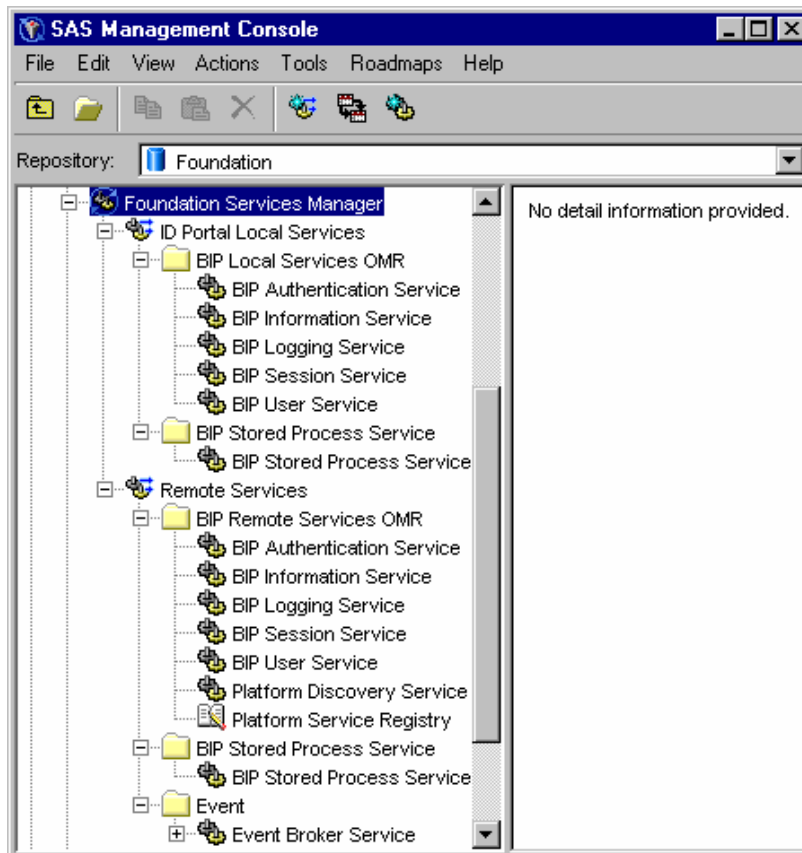
A service deployment configuration is a collection of foundation services that specifies the data necessary to deploy the services. A service deployment can be a local (accessible within a single Java Runtime Environment (JRE)) or remote service deployment (accessible within a single JRE, but available to other JRE processes).

The portal Web application, the SAS Services Application, and other foundation service-enabled applications use the service deployment configurations to deploy and access the foundation services. The portal Web application deploys local foundation services; you must start the SAS Services Application in order to deploy the remote foundation services. For details about service deployments, see “Understanding Service

Deployments” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/platserv/ps\\_servdep.html](http://support.sas.com/rnd/itech/doc9/admin_oma/platserv/ps_servdep.html).

Depending on how you installed the portal Web application, you might have been prompted to choose whether to store the metadata for local and remote service deployment configuration on the SAS Metadata Server or in an XML file:

- SAS Metadata Server: When you import the local and remote service deployment configurations into the SAS Metadata Server, the BIP Local Services and BIP Remote Services groups are displayed in the Foundation Services Manager in SAS Management Console:



- XML Files: The XML files for the foundation services deployment configuration for the portal Web application include both a local services deployment (`sas_services_idp_local_omr.xml`) and a remote services deployment (`sas_services_idp_remote_omr.xml`).

Regardless of whether the service deployments are accessed from the SAS Metadata Server or the XML files, the SAS Metadata Server or XML files contain the following service deployments and service deployment groups:

- ID Portal Local Services (`sas_services_idp_local_omr.xml`)
  - BIP Local Services: The portal Web application deploys the Authentication Service, Information Service, Logging Services, Session Service, and User Service as a local services deployment. The portal Web application has exclusive access to the locally deployed services.
  - BIP Stored Process Service: The SAS Stored Process Web application deploys the Stored Process Service as a local service deployment. The SAS Stored

Process Web application has exclusive access to the locally deployed Stored Process Service.

- Remote Services (`sas_services_idp_remote_omr.xml`)
  - BIP Remote Services: The SAS Services Application deploys the Authentication Service, Discovery Service, Information Service, Logging Service, Session Service, and User Service as a remote service deployment and shares the services using a Java Remote Method Invocation (RMI) server. The remote service deployment enables other applications to access the services. To enable remote access to services, the remote services deployment registers the remote services with the Java Remote Method Invocation (RMI) service registry. The portal Web application, and other applications and portlets can then share session and user information by locating and accessing the remote services. The remote-accessible Session Service enables single sign-on and communication between Web applications by enabling other Web applications, such as the SAS Stored Process Web application, to access the remote Session Service.
  - BIP Stored Process Service: The Stored Process Service is available as a remote service deployment, but it is not deployed by the SAS Services Application by default. When deployed, the remote-accessible Stored Process Service enables other applications to access the Stored Process Service.
  - Event: The Event Broker Service is available as a remote service deployment, but it is not deployed by the SAS Services Application by default. When deployed, the remote-accessible Event Broker Service enables other applications to access the Event Broker Service.

For more information about how foundation services are deployed and located by the portal Web application, and how they are accessed and used by applications and portlets, see “SAS Foundation Service Deployment and Use” on page 341.

---

## Ensuring JRE Communication Between the Portal and the SAS Services Application

When the portal Web application and the SAS Services application run on the same machine, ideally they should use the same JRE. However, there are situations in which the portal Web application and the SAS Services application run on different machines or use different JREs. For those situations, you must modify the permissions for the SAS Services application in order to enable communication.

If you use restricted permissions with the SAS Services application, then the two applicable grant principal statements in the policy file should reflect the JRE that the SAS Services application is using. The statements should specify "PFSIBMPrincipal" for an IBM JRE, or "PFSSunPrincipal" for a SUN JRE.

For example, suppose that you are running the portal Web application on a Windows machine within the IBM WebSphere Application Server, which comes with its own JRE. If the SAS Services application was set up to use the Sun JRE, then communication won't work correctly even though the SAS Services application is on the same machine as the portal and WebSphere. As a result, you would not be able to invoke other Web applications in order to access preferences, stored processes, or other portal functionality.

To enable communication in this example, the applicable grant principal statements might look like this:

```
// =====
// User and/or Group specific permissions
// =====
```

```
grant principal com.sas.services.security.login.PFSSunPrincipal
  "Portal Admins@DefaultAuth" {
  permission com.sas.services.user.UserContextPermission "*", "read";
  permission com.sas.services.session.SessionPermission "*", "quiesce";
  permission com.sas.services.session.SessionContextPermission "read";
};

grant principal com.sas.services.security.login.PFSSunPrincipal
  "SAS System Services" {
  permission com.sas.services.user.UserContextPermission "*", "read";
  permission com.sas.services.session.SessionContextPermission "read";
};
```

This discussion applies only if you have restricted permissions for the remote services. If you are running the SAS Services application with all permissions, then no grant principal is required. For more information about restricting permission, see “Adding Permissions to Policy Files” on page 45.

The default policy file for the SAS Services Application is `SAS-config-dir\Lev1\web\Deployments\RemoteServices\sas.wik.allpermissions.sasservices.policy`. You can find the location of the policy file that is being used on your system by looking in one of the following locations:

- If the SAS Services Application is running as a Windows service, then the policy file is specified in the file:

```
SAS-config-dir\Lev1\web\Deployments\RemoteServices\WEB-INF\conf\wrapper.conf
```

The following line specifies the policy file:

```
wrapper.java.additional.2=-Djava.security.policy= "C:\SAS\EntBIServer\Lev1\
web\Deployments\RemoteServices\sas.wik.sasservices.policy"
```

- If the SAS Services Application is started by a script, then the policy file is specified in the `StartRemoteServices` start-up script (`StartRemoteServices.bat` on Windows and `StartRemoteServices.sh` on UNIX), which is located in the `SAS-install-dir\Web\Portal2.0.1\SASServices\WEB-INF` directory. The start-up script contains the following two lines that specify the policy file that is used by the SAS Services application:

This line specifies the directory:

```
set SERVICES_DIR=C:\SAS\EntBIServer\Lev1\web\
Deployments\RemoteServices
```

This line specifies the file name:

```
set SERVICES_OPTS=%SERVICES_OPTS%
-Djava.security.policy="%SERVICES_DIR%\
sas.wik.allpermissions.sasservices.policy"
```

---

## Changes That Require You to Reimport the Service Deployment Configurations

If you reconfigure your SAS Metadata Server, Java Remote Method Invocation (RMI) server, or WebDAV server, then you must run the `configure_wik` utility to update your service deployment XML files. If you access the service deployment from the SAS Metadata Server, then you must then import the updated service deployment XML files into the SAS Metadata Repository.

The following list shows the values in the `install.properties` file that correspond to each server:

- Java RMI server:

```
$SERVICES_RMI_HOST$
$SERVICES_RMI_PORT$
```

- SAS Metadata Server:

```
$SERVICES_OMI_REPOSITORY$
$SERVICES_OMI_HOST$
$SERVICES_OMI_PORT$
$SERVICES_OMI_DOMAIN$
$SERVICES_OMI_USER_ID$
$SERVICES_OMI_USER_PASSWORD$
```

- WebDAV server:

```
$DAV_AUTOCONNECT$
$DAV_BASE$
$DAV_DOMAIN$
$DAV_HOST$
$DAV_REPOSITORY$
$DAV_PORT$
```

---

## Reimport the Service Deployment Configurations

If you make any changes that are specified in “Changes That Require You to Reimport the Service Deployment Configurations” on page 339, then the service deployment XML files will be reconfigured when you run the **configure\_wik** utility. You must reimport the foundation services after you run **configure\_wik**.

To reimport the services, complete these steps:

- 1 Start SAS Management Console and log on to the foundation metadata repository as the SAS Administrator.
- 2 In the Foundation Services Manager, remove the **ID Portal Local Services** node.
- 3 Right-click and select **Import Service Deployment**.
- 4 Choose the following file: *SAS-config-dir\Lev1\web\Deployments\Portal\sas\_services\_idp\_local\_omr.xml*.
- 5 In the Foundation Services Manager, remove the **Remote Services** node.
- 6 Right-click and select **Import Service Deployment**.
- 7 Choose the following file: *SAS-config-dir\Lev1\web\Deployments\Portal\sas\_services\_idp\_remote\_omr.xml*.

For more information, see the Foundation Services Manager Help.

Any changes that had been made to the previous service deployments on the SAS Metadata Server will be lost.

---

## SAS Foundation Service Deployment and Use

---

### Overview of SAS Foundation Service Deployment and Use

Understanding how the portal Web application deploys and accesses the SAS Foundation Services can help you determine how to redistribute your portal Web application implementation. Understanding how other Web applications and portlets use the foundation services can help you understand how to integrate applications and portlets with the portal Web application. The following sections explain how the portal Web application deploys, distributes, locates and shares foundation services.

For information about developing foundation service-enabled applications and portlets that are integrated with the portal Web application, see “Integrating Web Applications With the Portal” in the *SAS Web Infrastructure Kit: Developer’s Guide* at [http://support.sas.com/rnd/itech/doc9/portal\\_dev/webapps/dg\\_webapps.html](http://support.sas.com/rnd/itech/doc9/portal_dev/webapps/dg_webapps.html).

---

### How the Portal Web Application Deploys SAS Foundation Services

SAS Foundation Services are deployed as follows:

- For local service deployment, the portal Web application uses the `sas_metadata_source_client.properties` properties file to locate the portal Web application local service deployment configuration from one of the following:
  - the SAS Metadata Server
  - the XML file for the local service deployment configuration, `sas_services_idp_local_omr.xml`

The portal Web application uses the local service deployment configuration to deploy the local services. The portal Web application then has exclusive access to the locally deployed services. The Stored Process Web application accesses the Stored Process Service local service deployment configuration from one of the following:

- the SAS Metadata Server
- the XML file for the local service deployment configuration, `sas_services_idp_local_omr.xml`

The Stored Process Web application then uses the local service deployment to deploy the Stored Process Service as a local service. (The SAS Preferences Web application also accesses its local service deployment from the SAS Metadata Server or from the XML file for the local service deployment configuration, `sas_services_idp_local_omr.xml`. It then deploys the same local services as the portal Web application for its own exclusive local service deployment.)

- For remote service deployment, use the `StartRemoteServices.bat` or

```
StartRemoteServices.sh
```

utility to start the SAS Services application. You must start the remote services before you start the servlet container. (With SAS 9.1.3 and higher, the SAS Services Application can be run as a Windows service; for details, see “Run Remotely Deployed Services as a Windows Service” on page 345.) The SAS Services application uses the `sas_metadata_source_server.properties` properties file to locate the remote service deployment configuration from one of the following:

- the SAS Metadata Server

- the XML remote service configuration file, `sas_services_idp_remote_omr.xml`) and deploy the remote services

All of the remote services are registered with a remote Discovery Service. The remote services registration enables the portal Web application, the SAS Stored Process Web application, the SAS Preferences Web application, and other applications and portlets to use the remote Discovery Service to locate and use the remotely deployed services.

In addition, other applications and portlets might have their own local service deployment configurations to locally deploy particular SAS Foundation Services. The applications and portlets can access their local and remote service deployment configurations from the SAS Metadata Repository or from an XML configuration file.

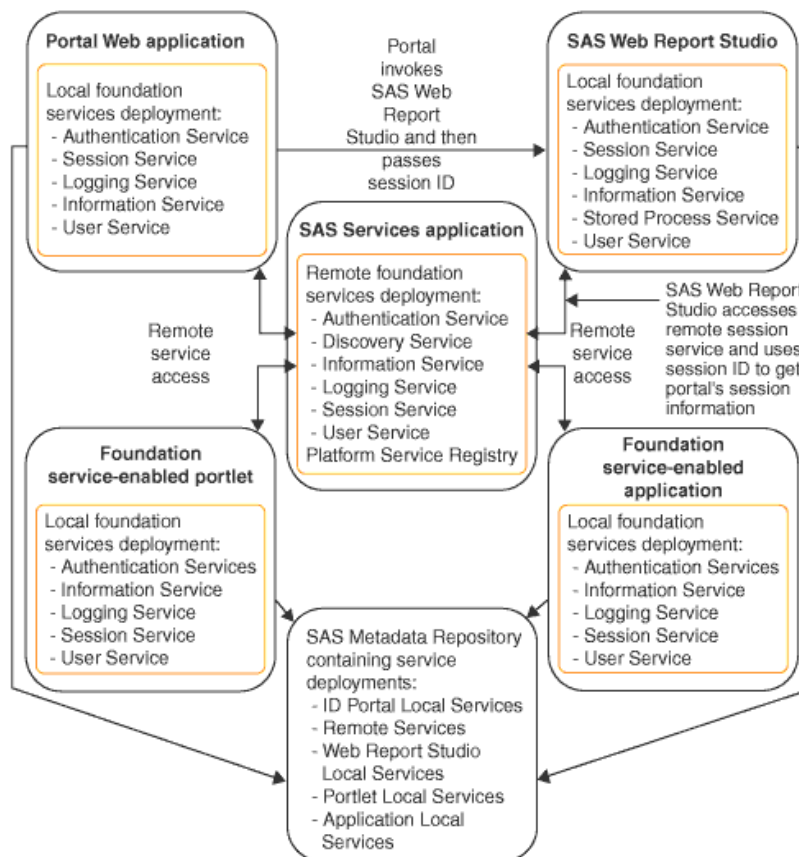
*Note:* The remote service deployment configurations, whether they reside in the metadata repository or in an XML file, must all contain the same configuration information for the remote service deployment.  $\Delta$

---

## Accessing Service Deployments from the SAS Metadata Server

The following diagram shows how components of the portal Web application access service deployments from the SAS Metadata Server.

**Display 20.1** How Components Access Service Deployments from the SAS Metadata Server





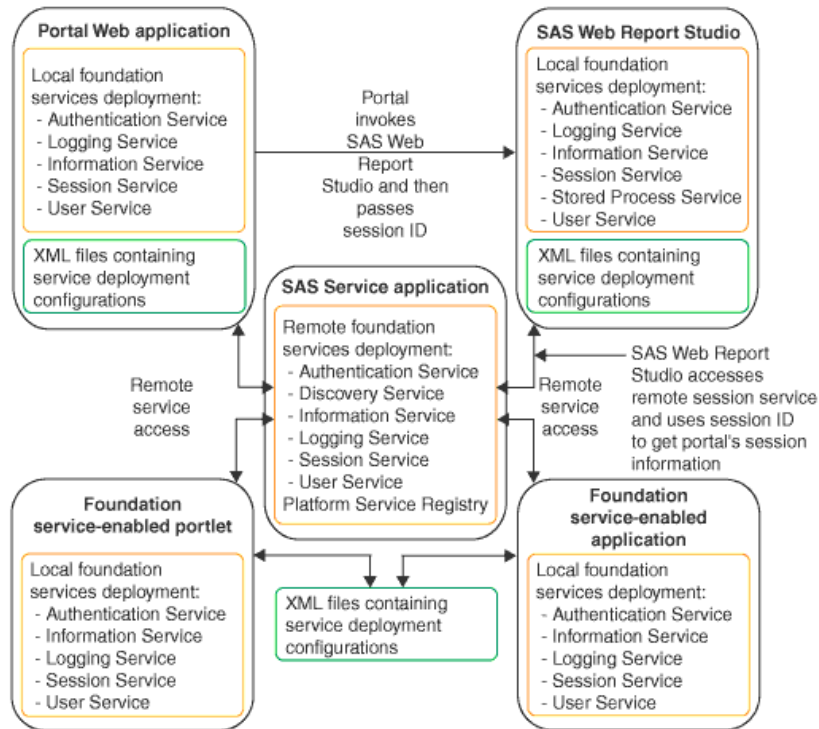
In the previous diagram, your portal Web application, SAS Services application, SAS Web Report Studio application, and foundation service-enabled portlet and application all access their local and remote service deployment configurations from the SAS Metadata Server. All of the applications share the same remote service deployment. In addition, each application has a local service deployment for its own exclusive access.

The SAS Stored Process Web application and the SAS Preferences Web application (not shown) also access their local and remote service deployment configurations from the SAS Metadata Server, deploy their own set of local services, and share the same remote service deployment as the other applications.

## Accessing Service Deployments from XML Files

The following diagram shows how components of the portal Web application access service deployments from XML files.

**Display 20.2** How Components Access Service Deployments from XML Files



In the previous diagram, your portal Web application, SAS Services application, SAS Web Report Studio application, and foundation service-enabled portlet and application all access their local and remote service deployment configurations from an XML service deployment configuration file. The remote service deployment configurations are all the same, and all of the applications share the same remote service deployment. In addition, each application has a local service deployment for its own exclusive access.

The SAS Stored Process Web application and SAS Preferences Web application (not shown) also access their local and remote service deployment configurations from XML files, deploy their own set of local services, and share the same remote service deployment as the other applications.

---

## How the Portal Web Application Components Are Distributed

The default installation deploys the SAS Stored Process Web application and the SAS Preferences Web application on the same machine as the portal Web application (the portal Web application installation machine). The default portal Web application installation also deploys both the local and remote services deployment (SAS Services application) on that same machine. You can move the SAS Stored Process Web application, the SAS Preferences Web application, and the SAS Services application (remote services) to separate machines.

For example, the different components in the previous diagram might exist on the same Web server or on different Web servers. In the diagram, the SAS Stored Process Web application or SAS Preferences Web application might exist on the same machine as the portal Web application (default installation) or on a machine that can access the remotely deployed services (SAS Services application). In addition, the remotely deployed services (SAS Services application) might exist on the same machine as the portal Web application (default installation) or on a separate machine that is accessible to the applications and portlets that need to use the services.

To deploy the SAS Stored Process Web application, the SAS Preferences Web application, or the SAS Services application (remote services) to a servlet container on a separate machine, see Chapter 21, “Redistributing Portal Web Applications and Servers,” on page 347.

---

## How Applications Locate the SAS Foundation Services

For information about how applications locate and bind to specific services, see “Understanding How Applications Locate Foundation Services” in the *SAS Integration Technologies: Administrator’s Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/platserv/ps\\_howservacc.html](http://support.sas.com/rnd/itech/doc9/admin_oma/platserv/ps_howservacc.html).

---

## How the Portal Web Application Shares SAS Foundation Services

An application or portlet can use the foundation services to access the portal Web application’s session context. To bind to the portal Web application’s remote session service, the application must provide the user ID of a privileged user. This user must be a member of the SAS System Service group, and it must be specified in the user service of the application’s local service deployment. For more details, see the following topics in the *SAS Web Infrastructure Kit: Developer’s Guide*:

- “Sample: Web Application (HelloUserWikExample)”
- “Using SAS Foundation Services With the Portal”

As shown in the previous diagram, the SAS Web Report Studio and portal Web applications use the same remotely deployed session service. When the portal Web application launches the SAS Web Report Studio application, it passes the portal Web application’s session ID to the SAS Web Report Studio application. The SAS Web Report Studio application can then bind to the remote session service and obtain and use the portal Web application’s session, user, and context information. This allows the user to seamlessly pass through to the SAS Web Report Studio application without the requirement for a separate logon.

In order to seamlessly integrate with the portal Web application, SAS Web Report Studio must be able to access the remote service deployment on startup. Therefore, you must start the remote services (by starting the SAS Services application) before starting SAS Web Report Studio. If SAS Web Report Studio cannot access the remote

services upon startup, then when you start the portal Web application and try to view a report with SAS Web Report Studio, you will not be able to seamlessly access SAS Web Report Studio from the portal Web application. Instead, you will need to log on to SAS Web Report Studio.

---

## Run Remotely Deployed Services as a Windows Service

SAS 9.1.3 and higher provides the capability to run the remote SAS Services application as a Windows service. This capability is enabled through the use of the Java Service Wrapper from Tanuki Software, which is provided with SAS Foundation Services.

To install the remote SAS Services application as a Windows service, follow these steps:

- 1 Go to the directory that contains the deployment information for the remote SAS Services application. If you performed a planned (Advanced or Personal) installation and accepted the default path, then this information is located in *SAS-config-dir*\Lev1\web\Deployments\RemoteServices. To determine the correct directory if you are not using the default path, check the value of the `$SERVICES_REMOTE_DIR$` property in the `install.properties` file, which is located in the *SAS-install-dir*\Web\Portal2.0.1\PortalConfigure directory.
- 2 In the **WEB-INF** subdirectory, run the `InstallRemoteServices.bat` script. The script installs a Windows service with the name **SAS Remote Services**.
- 3 You can now start the **SAS Remote Services** service from the Windows Services console.

To uninstall the service, use the script `UninstallRemoteServices.bat`, which is also located in the **WEB-INF** subdirectory.

Console output for the **SAS Remote Services** Windows service is stored in the file `wrapper.log`, which is located in the following path:

*SAS-config-dir*\Lev1\web\Deployments\RemoteServices\log\

For details about the Java Service Wrapper, see the Tanuki Software Web site at

<http://wrapper.tanukisoftware.org>.

For details about how to use the Java Service Wrapper to install and run remote SAS Foundation Services as Windows services for use with any foundation services-enabled application, see *Installing and Running Foundation Services as a Windows Service* in the *SAS Integration Technologies: Administrator's Guide* at [http://support.sas.com/rnd/itech/doc9/admin\\_oma/platserv/ps\\_windows\\_serv.html](http://support.sas.com/rnd/itech/doc9/admin_oma/platserv/ps_windows_serv.html).

---

## WebDAV Server Metadata

With the exception of reports, which can be stored on any type of WebDAV server, the portal Web application supports only Xythos WebFile Server (WFS) content (for SAS publication channels, files, and SAS Stored Process package output).

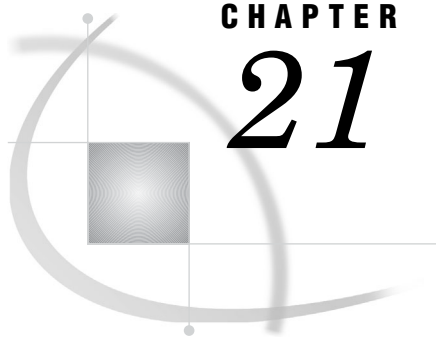
A WebDAV server definition on the SAS Metadata Server is required for the following:

- run SAS Stored Processes that publish to a Xythos WFS server
- configure WebDAV-based SAS publication channels or WebDAV package subscribers
- run other applications (such as SAS Web Report Studio) that require a WebDAV server definition

When you installed the Xythos WFS WebDAV Server, you specified an authentication domain for the WebDAV server. To verify this value, look at the `$DAV_DOMAIN$` value in the `install.properties` file. (The authentication domain is also listed in the Xythos `saswfs.properties` file.) When you defined the WebDAV server definition on the metadata server, you used this same authentication domain.

The SAS User Management Customization (provided with the Xythos WFS installation) enables the WebDAV server to use authentication and authorization metadata on the SAS Metadata Server. For information about setting up security for the Xythos WFS server, see “Implementing Authorization for the Xythos WebFile Server” on page 231.

*Note:* By default, Xythos is configured to return only a limited number of results (the default number is 50) in response to a search request. To obtain accurate results when searching for WebDAV content (for example, files), portal users should enter key words instead of using the wild card (\*). You can also change the default setting to a larger number in Xythos. Consult the Xythos documentation for instructions on changing this setting.  $\triangle$



## CHAPTER

## 21

## Redistributing Portal Web Applications and Servers

<i>Overview of Redistributing Applications and Servers</i>	347
<i>Redistributing the SAS Services Application (and Java RMI Server)</i>	348
<i>Redistributing the SAS Stored Process Web Application</i>	349
<i>Redistributing the SAS Preferences Web Application</i>	350
<i>Redistributing the SAS Themes Web Application</i>	351
<i>Portal Configuration After Redistributing SAS Web Report Viewer</i>	352
<i>Portal Configuration After Redistributing SAS Web Report Studio</i>	353
<i>Using SAS Web Report Studio as the Default Report Viewer</i>	353
<i>Portal Configuration After Redistributing the SAS Metadata Server</i>	354

### Overview of Redistributing Applications and Servers

Depending on the performance considerations of your implementation, you might want to redistribute the application pieces of the portal Web application, or the servers upon which they rely.

When you redistribute portal software, you can set up an environment that provides enhanced performance, security, availability, and scalability. For example, you might use a cluster of J2EE applications servers that are protected by a firewall. You can also modify Java Virtual Machine parameters for performance enhancement. For more information about enhancements and possible configurations, see “Sample Middle-Tier Deployment Scenarios” on page 70.

You can redistribute applications and servers as follows:

- Applications: The default installation deploys the SAS Stored Process Web application, the SAS Services application, the SAS Preferences Web application, and the SAS Themes Web application on the same machine as the portal Web application (the portal Web application’s installation machine). You can move these applications to separate machines.

*Note:* It is recommended that you leave the SAS Services Application (remote services and Java RMI server) on the same machine as the portal Web application.

△

For details about redistributing Web and other applications, see the following topics:

- “Redistributing the SAS Services Application (and Java RMI Server)” on page 348
- “Redistributing the SAS Stored Process Web Application” on page 349
- “Redistributing the SAS Preferences Web Application” on page 350
- “Redistributing the SAS Themes Web Application” on page 351

In addition, you might have installed the SAS Web Report Studio Web application and the SAS Web Report Viewer application. You can redistribute those applications as well. You can also configure the portal to display reports in SAS Web Report Studio instead of SAS Web Report Viewer. See the following topics:

- “Portal Configuration After Redistributing SAS Web Report Viewer” on page 352
- “Portal Configuration After Redistributing SAS Web Report Studio” on page 353
- “Using SAS Web Report Studio as the Default Report Viewer” on page 353
- Servers: When you installed the portal Web application and specified the configuration information, you specified the machine and port for the servers. However, after your initial installation, you might be required to move one or more of these servers to a different machine. Before you move servers to machines with different operating systems, be sure that you understand and have planned for your authentication domain(s), which are described in detail in *SAS Intelligence Platform: Security Administration Guide*. In addition, when you move servers to a new machine, you must update any permission statements for the servers. For details, see “Adding Permissions to Policy Files” on page 45.

For performance reasons, it is recommended that you install the SAS Metadata Server on a separate machine. If you move the metadata server, then you must reconfigure and redeploy the portal. For details, see “Portal Configuration After Redistributing the SAS Metadata Server” on page 354.

For information about moving other SAS servers, see *SAS Intelligence Platform: Application Server Administration Guide*.

---

## Redistributing the SAS Services Application (and Java RMI Server)

The SAS Services Application deploys the remote services for access by the portal Web application, the SAS Stored Process application, the SAS Preferences Web application, the SAS Themes Web application, remote portlets, and other applications such as SAS Web Report Studio. The SAS Services application uses a Java RMI server on its host machine to share the remote services with other applications. The remote services deployment configuration of the portal Web application gives you the flexibility to distribute the remote services as required. The recommended configuration (and the default installation) call for the local and the remote foundation services to run on the same machine.

However, for some implementations, you might want to have the remote services (SAS Services Application) deployed on a separate machine that can be accessed by the portal Web application, the SAS Stored Process Web application, the SAS Themes Web application, the SAS Preferences Web application, and other portlets and applications such as SAS Web Report Studio.

To redistribute the SAS Services application to a new machine:

- 1 On the new machine, install the SAS Web Infrastructure Kit (Index installation). When the installation program prompts you for the RMI server host, specify the new remote SAS Foundation Services host machine.
- 2 Perform the following post-installation instructions in the `wik_readme.html` file:
  - a Step 6: Run configuration scripts.
  - b Step 10: Prepare the servlet container environment.
  - c Step 13: Set up the SAS Services application.
  - d Step 15: Tune Web applications.
- 3 On the machine on which the portal Web application and the SAS Stored Process Web application are installed, follow these steps:
  - a Shut down the existing SAS Services application.

- b Shut down the servlet container in which all of the SAS Web applications are running.
  - c Remove cached JSP and servlet files, remove any expanded files that are produced from the WAR file, and remove any deployed WAR files. For details, see Step 11 in the `wik_readme.html` file.
- 4 Use a text editor to edit the `install.properties` file (located in the `PortalConfigure` subdirectory of the installation directory). Locate the following lines:

```
# RMI
$SERVICES_RMI_PORT$=5099
$SERVICES_RMI_HOST$=localhost
```

The remote services deployment uses a Java RMI registry to register remote services. These lines specify the machine and port on which your Java RMI service registry runs. Replace the `$SERVICES_RMI_HOST$` entry value with the name of the new machine on which the remote foundation services will be deployed (by the SAS Services Application). For example:

```
$SERVICES_RMI_HOST$=a1234.us.abc.com.
```

- 5 Re-create and redeploy the portal Web application as follows:

- a Run the `configure_wik.bator`

```
configure_wik.sh
```

- utility to create new service deployment configurations and new WAR files.
- b Deploy the WAR files to your servlet container.
- c Delete and reimport the Remote Services.

For complete instructions, see “Re-Create and Redeploy the Portal Web Application” on page 211.

- 6 Ensure that other applications and portlets have the appropriate permissions in their policy files to enable access to the new machine for the SAS Services Application. For details, see “Adding Permissions to Policy Files” on page 45.
- 7 On the SAS Service Application’s machine, run the `StartRemoteServices.bator`

```
StartRemoteServices.sh
```

utility to start the remote foundation services.

- 8 On the portal Web application’s machine and the SAS Stored Process Web application’s machine, restart the servlet container.

---

## Redistributing the SAS Stored Process Web Application

To redistribute the SAS Stored Process Web application to a new machine:

- 1 Uninstall the SAS Stored Process Web application from the servlet container in which the portal Web application is installed.
- 2 On the new machine, install the SAS Web Infrastructure Kit. When you run the installation program, be sure to specify the appropriate RMI Server host and port on which your SAS Services application will deploy the remote services. After the installation is complete, perform *only* the following post-installation instructions:
  - a Step 6: Run configuration scripts.
  - b Step 10: Prepare the servlet container environment.
  - c Step 11: Deploy the Web application files into the servlet container.

*Note:* Deploy *only* the **SASStoredProcess.war** file into the servlet container.  $\triangle$

d Step 15: Tune Web applications.

3 On your portal Web application machine, follow these steps:

- a Edit the **InfrastructureContent.xml** file that is located in the *SAS-install-dir\Web\Portal2.0.1\Portal\WEB-INF\content* directory.
- b Edit the **viewer** property for the stored process content so that it specifies the URL for the host name (and if required, the port number) of the new machine for the SAS Stored Process Web application, as shown in the following example:

```
<Content
  interface=
    "com.sas.services.storedprocess.metadata.StoredProcessInterface"
  category="storedprocess"
  icon="StoredProcess.image"
  viewer= http://host name:portnumber/SASStoredProcess/
    do?_action=form,properties
  isViewerExternal="true"
  appendSessionInfo="true"
  passObjectInSession="false"
  searchFilter=
    com.sas.services.storedprocess.metadata.StoredProcessFilter,
    com.sas.portal.filters.StoredProcessAttributeFilter
  searchRepositories="OMR"
  searchFoundationOnly="false"
  version="2.0">
</Content>
```

- 4 Run the **configure\_wik.bat** utility and redeploy the new **Portal.war** file. See “Re-Create and Redeploy the Portal Web Application” on page 211.

---

## Redistributing the SAS Preferences Web Application

To redistribute the SAS Preferences Web application to a new machine or servlet container, you must first redeploy the SAS Preferences Web application. Then you must update the application’s connection metadata on the SAS Metadata Server. You can update the metadata for the SAS Preferences Web application by running the SAS program **UpdatePreferencesConnection.sas**.

To redistribute the SAS Preferences Web application to a new machine or servlet container:

- 1 Deploy the **SASPreferences.war** file (from the servlet container on the portal Web application’s machine) to the new servlet container.
- 2 Modify the following fields in the **UpdatePreferencesConnection.sas** file, which is located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory:

```
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  $\$SERVICES\_OMI\_PORT\$\mathit{}$



property in the *install.properties* file (located in the *PortalConfigure* subdirectory of the setup directory).

metauser=*user ID*

Specify the user ID to use to connect to the SAS Metadata Server. This user is typically the SAS Administrator, whose default user ID is sasadm. For Windows users, the user ID is domain- or machine-name qualified. For example: “<domain or machine name>\saswbadm”

metapass=*password*

Specify the password for the metauser.

metarepository=*repository*;

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the *\$SERVICES\_OMI\_REPOSITORY\$* property in the *install.properties* file (located in the *PortalConfigure* subdirectory of the setup directory).

%let hostName=*Host Name*;

Specify the host name of the machine on which the SAS Preferences Web application is deployed, followed by a semicolon (;).

%let port=*Port*;

Specify the port number of the Web server on which the SAS Preferences Web application is deployed, followed by a semicolon (;).

%let URLPath=*base URL*;

Specify the URL path in the servlet container in which the SAS Preferences Web application is deployed, followed by a semicolon (;). This value might change if the WAR file is renamed before it is deployed.

%let protocol=*http*;

If you are using Secure Sockets Layer, then replace *http* with *https*.

- 3 Run the SAS program **UpdatePreferencesConnection.sas**.

---

## Redistributing the SAS Themes Web Application

The portal Web application contains two theme Web applications: **SASTheme\_default.war** and, starting with Service Pack 2, **SASTheme\_winter.war**. You initially deployed these themes to your servlet container or J2EE application server when you ran the portal’s installation and configuration programs. If you later want to deploy the themes to a different machine or servlet container, then you must update the application’s connection metadata on the SAS Metadata Server.

(To learn how to deploy your own custom themes, or to deploy themes to an HTTP server instead of to a servlet container, see “Theme Deployment” on page 328.)

To redistribute the SAS Themes Web applications to a new servlet container:

- 1 Locate the existing **SASTheme\_default.war** file that is deployed in your servlet container, and deploy the WAR file to the new servlet container using the appropriate procedures for the J2EE servlet container.
- 2 Modify the following fields in the **UpdateThemeConnection.sas** file, which is located in the *SAS-install-dir\Web\Portal2.0.1\OMR* directory:

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 *\$SERVICES\_OMI\_PORT\$* property in the *install.properties* file (located in the *PortalConfigure* subdirectory of your installation).

metauser=*“user ID”*

Specify the user ID to use to connect to the SAS Metadata Server. This user is typically the SAS Web Administrator, whose default user ID is saswbadm. For Windows users, the user ID is domain- or machine name- qualified. For example: “<domain or machine name>\saswbadm”

metapass=*“password”*

Specify the password for the metauser.

metarepository=*“repository”*;

Specify the name of the SAS Metadata Repository in which your portal Web application metadata is stored, followed by a semicolon (;). Use the value of the `$SERVICES_OMI_REPOSITORY$` property in the `install.properties` file (located in the `PortalConfigure` subdirectory of your installation).

%let themeName=*Theme Name*;

Specify the name of the theme to update, followed by a semicolon (;).

%let hostName=*Host Name*;

Specify the host name of the machine on which the theme is deployed, followed by a semicolon (;).

%let port=*Port*;

Specify the port number of the new servlet container or J2EE application server, followed by a semicolon (;).

%let URLPath=*base URL*;

Specify the URL path of the new servlet container or J2EE application server, followed by a semicolon (;).

*Note:* If you are using Secure Sockets Layer (SSL), specify `https` instead of `http` as the protocol for the URL.  $\Delta$

%let protocol=*http*;

If you are using Secure Sockets Layer, then replace `http` with `https`.

- 3 Run the SAS program `UpdateThemeConnection.sas`.
- 4 Repeat the previous steps 1-3 for `SASTheme_winter.war`.
- 5 Stop and restart the SAS Services Application.
- 6 Update the policy file for the appropriate applications in order to specify the new location for the SAS Themes Web applications. You can update the policy files in either of the following ways:
  - Manually edit the policy files to specify the new `$CONTAINER_HOST$` and `$CONTAINER_PORT$` for the SAS Themes application. For details, see “Adding Permissions to Policy Files” on page 45.
  - In the `install.properties` file, update the `$CONTAINER_HOST$` and `$CONTAINER_PORT$` parameters to specify the new SAS Themes location. Then, run the `configure_wik` utility and re-apply the policy files for the portal Web application, SAS Preferences Web application, and any remote portlets and applications that access the SAS Themes Web application.

---

## Portal Configuration After Redistributing SAS Web Report Viewer

After you redistribute SAS Web Report Viewer to a new machine or servlet container, perform the following steps to ensure access from the portal Web application:

- 1 If you changed the SAS Web Report Viewer `wrv.config` file, then re-create and redeploy SAS Web Report Viewer. Instructions are similar to those for SAS Web Report Studio. See “Re-Create and Redeploy SAS Web Report Studio” on page 116.

- 2 Deploy the **SASWebReportViewer.war** file (from the servlet container on the portal Web application's machine) to the new servlet container.
- 3 On your portal Web application machine, follow these steps:
  - a Edit the **PortalContent.xml** file that is located in the *SAS-install-dir\Web\Portal2.0.1\Portal\WEB-INF\content* directory.
  - b Edit the **viewer** property for the report content so that it specifies the URL for the new machine on which SAS Web Report Viewer is running. Include the host name, the port number, and the HTTP protocol as shown in the following example:

```
<Content interface="com.sas.report.repository.ReportEntryInterface"
  category="report"
  icon="Report.image"
  isViewerExternal="true"
  viewer="http://host name:port number/SASWebReportViewer/
  logonFromPortal.do"
  appendSessionInfo="true"
  newViewerWindow="true"
  passObjectInSession="false"
  searchFilter=com.sas.portal.filters.PortalReportFilter,
  com.sas.portal.filters.PortalReportAttributeFilter
  searchRepositories="OMR"
  searchFoundationOnly="false"
  version="2.0">
</Content>
```

- 4 Run the **configure\_wik.bat** utility and redeploy the new **Portal.war** file. See “Re-Create and Redeploy the Portal Web Application” on page 211.

---

## Portal Configuration After Redistributing SAS Web Report Studio

After you redistribute SAS Web Report Studio, perform the following steps to ensure access from the portal Web application:

- 1 Verify access to the redeployed SAS Web Report Studio application. See “Re-Create and Redeploy SAS Web Report Studio” on page 116 for details on redeployment.
- 2 Within the portal Web application, modify any link or application content type to point to the new host and port number of the redistributed SAS Web Report Studio application

---

## Using SAS Web Report Studio as the Default Report Viewer

By default, when a user selects a link for a SAS Report, the portal displays that report in SAS Web Report Viewer. You can change the default behavior so that the portal displays reports in SAS Web Report Studio. This way, authorized users can edit reports without logging on to SAS Web Report Studio separately.

To make SAS Web Report Studio the default report viewer, complete these steps:

- 1 Edit the **PortalContent.xml** file that is located in the *SAS-install-dir\Web\Portal2.0.1\Portal\WEB-INF\content* directory.

- Where the file specifies **SASWebReportViewer**, change the reference so that it specifies **SASWebReportStudio** instead, as shown in the following example:

```
<Content interface="com.sas.report.repository.ReportEntryInterface"
category="report"
icon="Report.image"
isViewerExternal="true"
viewer="/SASWebReportStudio/logonFromPortal.do"
appendSessionInfo="true"
newViewerWindow="true"
passObjectInSession="false"
searchFilter=com.sas.portal.filters.PortalReportFilter,
com.sas.portal.filters.PortalReportAttributeFilter
searchRepositories="OMR"
searchFoundationOnly="false"
version="2.0">
</Content>
```

- Run the **configure\_wik.bat** utility and redeploy the new **Portal.war** file. See “Re-Create and Redeploy the Portal Web Application” on page 211.

After you make this change, the portal uses SAS Web Report Studio to display all reports. If users have been added to the SAS Web Report Studio user roles, then those user roles are applied to the reports. Users can edit reports only if they have permissions based on their associated user role.

---

## Portal Configuration After Redistributing the SAS Metadata Server

If you move the SAS Metadata Server to a different machine, then you must update the portal’s settings.

- Shut down the servlet container for the portal Web application and the SAS Stored Process Web application.
- Open the **install.properties** file in a text editor, and update it as follows:

- Locate the following lines:

```
$SERVICES_OMI_HOST$=localhost
$SERVICES_OMI_PORT$=8561
```

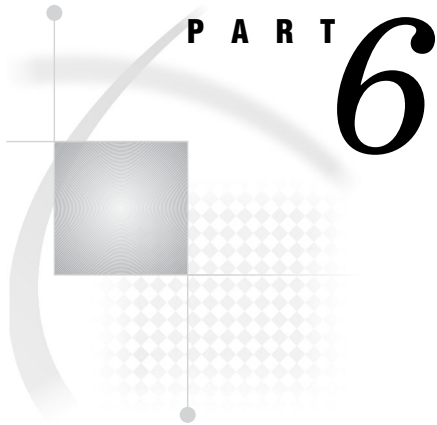
These lines specify the machine and port of your SAS Metadata Server.

- Replace the value of the **\$SERVICES\_OMI\_HOST\$** parameter with the new machine name of your SAS Metadata Server. For example, **\$SERVICES\_OMI\_HOST\$=a1234.us.abc.com**.
- If you are changing the default port of the SAS Metadata Server, then replace the value of the **\$SERVICES\_OMI\_PORT\$** parameter with the new port of your SAS Metadata Server.
- Locate the following lines:

```
$SERVICES_OMI_USER_ID$=sasadm
$SERVICES_OMI_USER_PASSWORD$={sas001}QWRtaW4xMjM=
$SERVICES_OMI_USER_NAME$=sasadm
$PORTAL_GUEST_ID$=sasguest
$PORTAL_GUEST_PASSWORD$={sas001}R3Vlc3QxMjM=
$PORTAL_GUEST_NAME$=sasguest
$PORTAL_ADMIN_ID$=saswbadm
```







## Appendixes

- Appendix 1* . . . . . **Summary of the Required SAS Users and Groups** 359
- Appendix 2* . . . . . **SAS Application Servers That Are Required for SAS Content** 363
- Appendix 3* . . . . . **Logon Formats for the Web Applications** 365
- Appendix 4* . . . . . **Configuring the ESRI Map Component** 369
- Appendix 5* . . . . . **Recommended Reading** 377





## APPENDIX

## 1

## Summary of the Required SAS Users and Groups

---

<i>Overview of the Required SAS Users and Groups</i>	359
<i>Users That Are Configured on the System</i>	359
<i>Users and Groups That Are Defined in Metadata</i>	360

---

### Overview of the Required SAS Users and Groups

When you installed the middle-tier software, you created particular SAS users and groups on the host and in SAS metadata. The user accounts on the host are required for authentication; the user and group definitions in SAS metadata are required for authorization (access control). These accounts are described in the *SAS Intelligence Platform: Pre-installation Checklists*.

If you have set up a distributed environment, or if various SAS servers (for example, stored process or workspace servers) were configured with different authentication domains, then the SAS users and groups must have additional logins in order to authenticate against those servers.

Global notes about the SAS users:

- When you installed the Web applications, you might have specified different metadata identity names and user IDs for these users. The identity and user ID names that are used here represent the default values that are provided by the Configuration Wizard. This administrator's guide always refers to these identities using the default values.
- Due to the permissions granted to the users, it is recommended that you safeguard the accounts for these users, and exercise caution if you share the accounts with others.

The following sections summarize the required users and groups that are required for operation:

- "Users That Are Configured on the System" on page 359
- "Users and Groups That Are Defined in Metadata" on page 360

---

### Users That Are Configured on the System

The following table lists the default user names for the SAS users that require an account on the SAS Metadata Server host system.

By default, user authentication occurs on the SAS Metadata Server host machine. If you later decide to set up an alternative form of authentication (for example, Web or LDAP authentication), then you would add these accounts to the chosen authentication provider.

**Table A1.1** Host System Users

Type of User	Account Name (Default)
SAS Administrator	sasadm
SAS Trusted User	sastrust
SAS Guest User	sasguest
SAS Demo User (optional)	sasdemo
SAS General Server	sassrv
SAS Web Administrator	saswbadm
SAS Installer (UNIX, z/OS only)	sas
LSF Administrator	lsfadmin
LSF User	lsfuser

*Note:* The accounts for your deployment might vary, depending on the software that you have installed. Some accounts are required in all cases, while others are required only if your deployment plan contains specific products. For details about which accounts you need to create at your site, see the *SAS Intelligence Platform: Installation and Configuration Guide*.  $\Delta$

The SAS Demo user is not actually required for operation, but is very helpful in developing and testing the middle-tier applications. This administrator's guide uses the SAS Demo User as the model for setting up typical users on the operating system and in metadata.

## Users and Groups That Are Defined in Metadata

At the end of the installation process, certain metadata objects must exist in your metadata repository. This section lists the User and Group objects that must be defined in the metadata in order for your servers and applications to work correctly. You can use the User Manager in SAS Management Console to verify that these objects have been created.

**Table A1.2** Summary of Metadata Identities

Metadata Identity	User ID*	Password**	Authentication Domain	Group Membership Information
User: SAS Administrator	sasadm			
User: SAS Trusted User	sastrust			member of: SAS System Services group, SAS General Servers group
User: SAS Guest	sasguest	*****	DefaultAuth	
User: SAS Demo User	sasdemo	*****	DefaultAuth	member of: Portal Demos

Metadata Identity	User ID*	Password**	Authentication Domain	Group Membership Information
User: SAS Web Administrator	saswbadm	*****	DefaultAuth	member of: Portal Admins group, SAS System Services group
Group: SAS System Services				members: SAS Trusted User, SAS Web Administrator
Group: SAS General Servers	sassrv	*****	DefaultAuth	members: SAS Trusted User
Group: Portal Admins***				members: SAS Web Administrator
Group: Portal Demos***				members: SAS Demo User
Group: WRS Administrator****				
Group: WRS Report Author****				
Group: WRS Advanced User****				

\* These are the recommended IDs. They should correspond to accounts in your authentication provider. On Windows, the user ID in the login should be fully qualified with a host or domain name, for example, *host-name\sasadm*.

\*\* If you are logged on to SAS Management Console as an unrestricted user, you will always see \*\*\*\*\* in the password column, even if no password was specified.

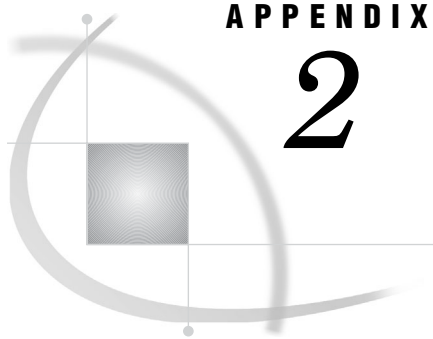
\*\*\*You need this metadata identity only if you are running SAS Information Delivery Portal.

\*\*\*\*You need this metadata identity only if you are running SAS Web Report Studio.

The users and groups that you have defined might vary, depending on the software that you have installed. Some users and groups are required in all cases, while others are required only if your deployment plan contains specific products. For details about which users and groups you need to create at your site, see the *SAS Intelligence Platform: Installation and Configuration Guide*.

For more information about these users and groups, or for instructions on managing the users and groups, see the *SAS Intelligence Platform: Security Administration Guide*.





## APPENDIX

## 2

## SAS Application Servers That Are Required for SAS Content

*SAS Application Servers That Are Required for SAS Content* 363

### SAS Application Servers That Are Required for SAS Content

The SAS Web applications enable you to exploit the analytical and reporting powers of SAS by delivering SAS data to the desktops of users. From client machines that have only a Web browser installed, authorized users can run SAS Reports and SAS Stored Processes, view information maps, and perform other data-specific tasks.

Before you can work with SAS content, the appropriate servers and spawners must be deployed. In addition, the servers must be started in the appropriate order. (For details, see “Starting the Web Applications” on page 13). The following table shows the metadata definitions and server deployment that is required for each type of content.

**Table A2.1** Metadata on the SAS Metadata Server

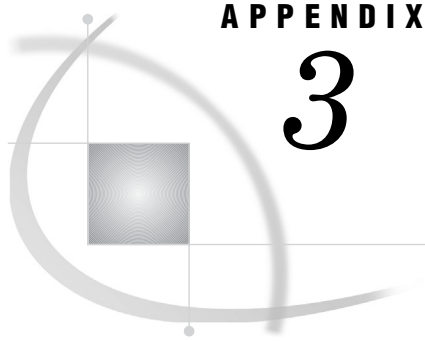
<i>Content</i>	<i>Required Server Definition</i>
SAS Information Maps	<input type="checkbox"/> for relational data, a SAS Workspace Server and Spawner <input type="checkbox"/> for multidimensional data, a SAS OLAP Server
Packages	<input type="checkbox"/> a SAS Workspace Server and spawner, if publishing to an archive on a SAS Workspace Server <input type="checkbox"/> a WebDAV server, if publishing to a WebDAV server <input type="checkbox"/> if publishing to a file, then no server definition is needed for the package
Publication Channels	<input type="checkbox"/> a SAS Workspace Server and spawner, if publishing to an archive on a SAS Workspace Server <input type="checkbox"/> a WebDAV server, if publishing to an archive on a WebDAV server <input type="checkbox"/> if publishing to an archive in the file system, then no server definition is needed for the publication channel
SAS Reports	<input type="checkbox"/> for relational data, a SAS Workspace Server and spawner <input type="checkbox"/> for multidimensional data, a SAS OLAP Server <input type="checkbox"/> for storing reports in WebDAV, a WebDAV server

---

<b><i>Content</i></b>	<b><i>Required Server Definition</i></b>
SAS Stored Processes - Package Results	<input type="checkbox"/> a SAS Workspace Server and spawner <input type="checkbox"/> a WebDAV server, if outputting a package to a WebDAV server
SAS Stored Processes - Streaming Results	a SAS Stored Process Server

---

*Note:* If you are publishing from the portal Web application to a WebDAV persistent store, then you must have the Xythos WebFile Server installed.  $\Delta$



## APPENDIX

## 3

## Logon Formats for the Web Applications

---

<i>Overview of Logon Formats</i>	365
<i>Logon Formats for SAS Metadata Server Authentication</i>	365
<i>Host Authentication</i>	365
<i>LDAP Authentication</i>	366
<i>Microsoft Active Directory Authentication</i>	366
<i>Logon Format for Web (Trusted) Authentication</i>	366

---

### Overview of Logon Formats

When a user logs on to the portal Web application, the user must provide valid credentials in the logon window. The exact format used to provide these credentials varies with the type of authentication that is configured for your portal environment. You have the option of using the following for authentication:

- the authentication provider that the metadata server uses. This can be the host operating system (default), an LDAP directory server, or a Microsoft Active Directory server. For the respective logon formats, see “Logon Formats for SAS Metadata Server Authentication” on page 365.

*Note:* For instructions on configuring the metadata server to use LDAP or Active Directory authentication, see the *SAS Intelligence Platform: Security Administration Guide*. △

- a Web server, servlet container, or J2EE application server (for trusted authentication). For logon information, see “Logon Format for Web (Trusted) Authentication” on page 366.

*Note:* For instructions on configuring trusted Web authentication, see “Changing to Trusted Web Authentication” on page 32. △

For instructions on logging on and logging off a Web application, refer to the online Help that is provided with the application.

---

### Logon Formats for SAS Metadata Server Authentication

---

#### Host Authentication

If you are using host authentication (the default authentication provider), then when a user logs on to the Web application, the user provides the user name and password

that were configured for this user on the metadata server's host operating system. The user name and password must also match the user ID and password that are defined for this user on the SAS Metadata Server.

For example:

```
User Name: User1
Password: User123
```

Here's an example that uses a Windows domain:

```
User Name: Domain\User1
Password: User123
```

---

## LDAP Authentication

If you have configured LDAP authentication, then when a user logs on to the Web application, the user must specify the LDAP domain that was configured in the SAS Metadata Server startup command and in the user definition on the SAS Metadata Server.

For example:

```
User Name: User1@LDAPAuthProv
Password: User123
```

---

## Microsoft Active Directory Authentication

If you have configured Microsoft Active Directory authentication, then when a user logs on to the Web application, the user must specify the authentication provider domain that was configured in the SAS Metadata Server startup command and in the user definition on the SAS Metadata Server.

The user provides a user name in *either* of the following formats:

```
userID@Windows_network_domain
```

```
Windows_network_domain\userID
```

For example:

```
User Name: User1@Sales
Password: User123
```

---

## Logon Format for Web (Trusted) Authentication

If you have configured trusted Web authentication, then the Web application does not prompt users for credentials. Instead, the Web component (Web server or servlet container) will prompt the user for a user name and password, and then authenticate that user before passing the credentials to the Web application.



The following display shows a sample logon to a Web server that has been configured for authentication:

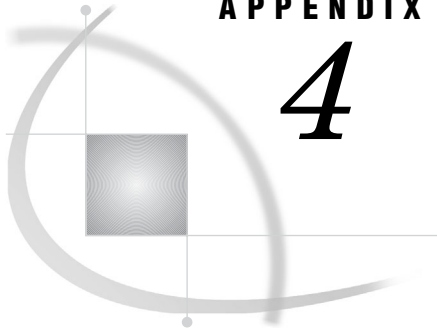


If the user credentials that the Web server passes contain a Windows domain, then specify a domain (for example, **WINNT\user1**).



## APPENDIX

## 4



## Configuring the ESRI Map Component

---

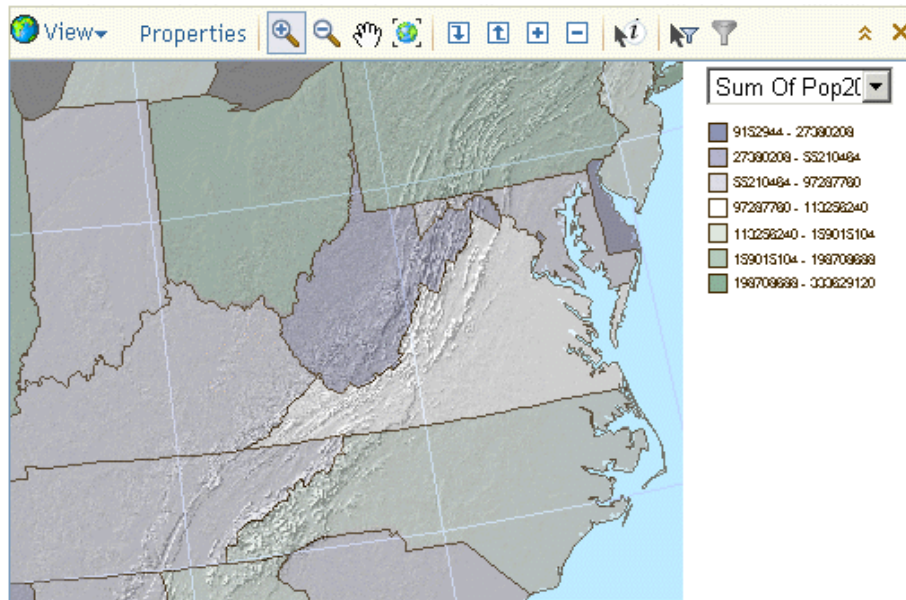
<i>Overview of the ESRI Map Component</i>	<b>369</b>
<i>Software Requirements</i>	<b>370</b>
<i>Define an ESRI Server</i>	<b>371</b>
<i>Configure Security for the ESRI Server</i>	<b>371</b>
<i>Define a Map Service</i>	<b>372</b>
<i>Overview of Defining a Map Service</i>	<b>372</b>
<i>Define a Map Service by Using the New Map Service Wizard</i>	<b>372</b>
<i>Define a Map Service by Creating an XML File (SAS Enterprise Guide Only)</i>	<b>372</b>
<i>Overview of Defining a Map Service by Creating an XML File</i>	<b>372</b>
<i>Create a Map Service XML File</i>	<b>373</b>
<i>Import the Map Service Metadata</i>	<b>373</b>
<i>Configure Your OLAP Cubes for ESRI Integration</i>	<b>374</b>
<i>Overview of Integrating Your OLAP Data with ESRI</i>	<b>374</b>
<i>Add ESRI Information to an Existing OLAP Cube</i>	<b>374</b>
<i>Specify ESRI Information in a New OLAP Cube</i>	<b>375</b>

---

### Overview of 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.

The following image shows a geographic map within SAS Web OLAP Viewer for Java:



Depending on the SAS product that you use, you can interact with your geographic maps in several ways. For example, you can zoom, scroll the map, drill up or down, and expand or collapse regions.

The following SAS products can use the ESRI map component to display geographic maps:

- SAS Web Report Studio
- SAS Information Delivery Portal
- SAS Web OLAP Viewer for Java
- SAS Enterprise Guide

## Software Requirements

The requirements for each ESRI-enabled SAS product are listed in the following table:

**Table A4.1** Requirements for ESRI Integration

SAS Product	Requirements
SAS Information Delivery Portal SAS Web OLAP Viewer SAS Web Report Studio	These products require access to ESRI ArcGIS Server 9.0 with Service Pack 3 or later.  <i>Note:</i> ArcGIS Server does not need to run on the same machine as your SAS software. $\Delta$
SAS Enterprise Guide	The ArcGIS engine runtime must be installed on the machine(s) where SAS Enterprise Guide is running.  Each ESRI map document must be specified in ArcGIS as a network location that SAS Enterprise Guide can access.

For more information about ESRI ArcGIS software, see the ESRI Web site at <http://www.esri.com>.

---

## Define an ESRI Server

In order to use the ESRI map component, you must create a server definition in metadata that identifies the host system for the ArcGIS Server.

To create an ESRI server definition:

- 1 In SAS Management Console, select **Server Manager** and then select **Actions ► New Server** from the main menu bar.
- 2 In the New Server Wizard, select **ESRI Map Server**. Click **Next**.
- 3 Enter a name for the server in the **Name** field, and an optional description for the server definition in the **Description** field. Click **Next**.
- 4 Enter the appropriate information for the ESRI ArcGIS server in the **Software Version** and **Vendor** fields. Click **Next**.
- 5 Select a domain from the **Authentication Domain** drop-down list, or click **New** to create a new authentication domain. Enter the host name for the machine where the ESRI server is running in the **Host Name** field. Click **Next**.

*Note:* It is recommended that you create a new authentication domain for the ESRI server. △

- 6 Confirm the information for your server definition, and then click **Finish**.

---

## Configure Security for the ESRI Server

Each user who uses the ESRI map component must be able to access a login for the machine where ArcGIS Server is running. This login must be a member of the agsusers group, which is created automatically when you install ArcGIS Server.

For example, you might want to configure the security for the ESRI map component as follows:

- 1 On the machine where ArcGIS Server is running, create a new user named esriuser. Make this user a member of the agsusers group.
- 2 Define metadata for a special access group:
  - a In SAS Management Console, select **User Manager**. Select **New ► Group** to create a new group.
  - b On the **General** tab of the New Group Properties dialog box, enter the name **ESRI Users**.
  - c On the **Members** tab, move all of the groups that need to access the ESRI server to the **Current Members** pane.
  - d On the **Logins** tab, click **New** to create a new login.
  - e Specify the credentials for the user that you created in Step 1. Prefix your user name with your machine name (for example, **machine\esriuser**). From the **Authentication Domain** drop-down list, select the domain that you specified in the definition for the ESRI server.

In the preceding example, all of the members of the ESRI Users group can read the login metadata for esriuser and use that login to access the ESRI server.

*Note:* To grant ESRI access to every user who has a metadata identity, you can add your group login to the SASUSERS group. △

---

## Define a Map Service

---

### Overview of Defining a Map Service

The ESRI map component uses a metadata object called a map service to determine how ESRI data corresponds to your SAS OLAP data. You must create a map service for each cube that you want to associate with ESRI data.

If you have access to ArcGIS Server (required for SAS Information Delivery Portal, SAS Web OLAP Viewer, and SAS Web Report Studio), define a map service by using the New Map Service wizard. See “Define a Map Service by Using the New Map Service Wizard” on page 372.

If you do not have access to ArcGIS Server (SAS Enterprise Guide only), define a map service by creating an XML file. See “Define a Map Service by Creating an XML File (SAS Enterprise Guide Only)” on page 372.

---

### Define a Map Service by Using the New Map Service Wizard

To create a map service definition by using the New Map Service wizard:

- 1 In SAS Management Console, select **Map Service Manager**, and then select **Actions**  $\blacktriangleright$  **New Map Service**.
- 2 In the New Map Service wizard, enter the name for the map service in the **Name** field, and then select a map server from the **Map Server** drop-down list. Click **Next**.

*Note:* When you click **Next**, SAS Management Console attempts to connect to the ESRI server. If the connection fails, a warning dialog box appears. Ensure that your user ID can access a login definition for the ESRI server, and that you are not logged on to SAS Management Console as an *unrestricted user*.  $\triangle$

- 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 that you selected, select the fields that you want to associate with OLAP data. Select one or more fields from the **Columns** selection box. Click **Next**.
- 6 Review the information that you specified, and then click **Finish**.

---

### Define a Map Service by Creating an XML File (SAS Enterprise Guide Only)

#### Overview of Defining a Map Service by Creating an XML File

*Note:* If you have access to ArcGIS Server, you can also create a map service by using the New Map Service wizard. See “Define a Map Service by Using the New Map Service Wizard” on page 372.  $\triangle$

If you will use the ESRI map component only with SAS Enterprise Guide and you do not have access to ArcGIS Server, then you can create a map service definition by using an XML file.

To create a map service definition by using an XML file, you create the XML file and then import the map service metadata by using the Import Map Service wizard in SAS Management Console.

## 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 the all of your 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=** optionally specifies an alias for the layer.
- 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 Metadata

To import map service metadata from an XML file:

- 1 In SAS Management Console, select the Map Service Manager.
- 2 Select **Actions** ► **Import Map Service**. The Import Map Service wizard appears.
- 3 Specify a name for the map service, and then click **Browse** to locate and select the XML file that you want to import.
- 4 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.
- 5 In the **Layers** field, select the layers that you want to use. By default, all of the layers are selected. Click **Next**.
- 6 Verify the information from your XML file, and then click **Finish** to create your new map service.

## Configure Your OLAP Cubes for ESRI Integration

### Overview of Integrating Your OLAP Data with ESRI

To use the ESRI map component with an OLAP cube, the cube must contain information about how its columns correspond to fields in the ESRI data. You can either add this information to an existing cube, or create a new cube.

### Add ESRI Information to an Existing OLAP Cube

To add ESRI information to an existing OLAP cube:

- 1 In SAS OLAP Cube Studio, select the cube that you want to modify, and then select **Actions**  $\blacktriangleright$  **Edit Cube Structure**.
- 2 In the Cube Designer Wizard, click **Next** until you reach the Cube Designer – Dimensions page.
- 3 Select the dimension that contains geographic data, and then click **Modify**. In the Dimension Designer wizard, select **GEO** from the **Type** drop-down list.
- 4 Click **Next** until you reach the Dimension Designer – Hierarchy page, and then click **Finish**.
- 5 On the Cube Designer – Dimensions page, select the dimension that you edited in the preceding step, and then click **Specify Map**.
- 6 In the Specify Map for Dimension dialog box, select the resources that contain the ESRI data that you want to use from the **Map Server** and **Map Service** drop-down lists.

For each dimension level in the **Levels** selection box, select the level and fill in the following information:

**Map Layer**

specifies the ESRI map layer that corresponds to the selected OLAP dimension level.



**Map Field ID**

specifies an ESRI map field that identifies the regions of the selected map layer.

**Field ID Column**

specifies an OLAP data column that uniquely identifies the regions of the selected dimension level.

*Note:* For each level, the values of the OLAP data column must be identical (case sensitive) to the values of the ESRI map field. The names of the column and map field do not need to be identical.  $\Delta$

When you have filled out the information for each layer, click **OK** to return to the Cube Designer – Dimensions page.

- 7 Click **Next** until you reach the Cube Designer – Finish page. Select whether to re-create the cube immediately, and then click **Finish**.

*Note:* You must create your cube using SAS OLAP Cube Studio. The OLAP procedure does not support cubes with GEO dimensions.  $\Delta$

---

## Specify ESRI Information in a New OLAP Cube

To specify ESRI information in a new OLAP cube:

- 1 In SAS OLAP Cube Studio, select **File**  $\blacktriangleright$  **New Cube**.
- 2 In the Cube Designer wizard, fill out the General, Input, and Drill-Through pages as usual. See the SAS OLAP Cube Studio Help for more information about these pages.
- 3 On the Cube Designer – Dimensions page, create your dimensions and hierarchies as usual. For the dimension that contains geographic data, select **GEO** from the **Type** drop-down list.

When you have finished creating your dimensions, select the dimension that contains geographic data, and then click **Specify Map**.

- 4 In the Specify Map for Dimension dialog box, select the resources that contain the ESRI data that you want to use from the **Map Server** and **Map Service** drop-down lists.

**Specify map for dimension Geography**

Map Server: ESRI Server

Map Service: ContinentalUS\_Projected

Levels:

- county\_name
- statecode
- usregion

From the Map Service:

Map Layer: states

Map Field Id: STATE\_ABBR

From the Cube Input Table:

Field Id Column: statecode

OK Cancel Help

For each dimension level in the **Levels** selection box, select the level and fill in the following information:

**Map Layer**

specifies the ESRI map layer that corresponds to the select OLAP dimension level.

**Map Field ID**

specifies an ESRI map field that identifies the regions of the selected map layer.

**Field ID Column**

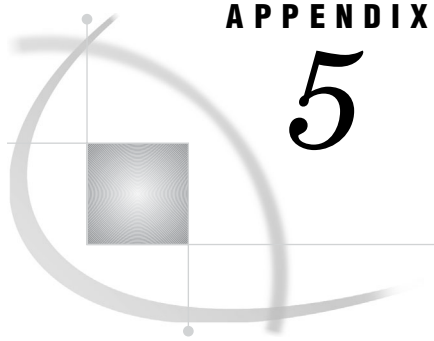
specifies an OLAP data column that uniquely identifies the regions of the selected dimension level.

*Note:* For each level, the values of the OLAP data column must be identical (case sensitive) to the values of the ESRI map field. The names of the column and map field do not need to be identical.  $\Delta$

When you have filled out the information for each layer, click **OK** to return to the Cube Designer – Dimensions page.

- 5 Finish defining your cube as usual, until you reach the Cube Designer – Finish page. Select whether to create your cube immediately, and then click **Finish**.

*Note:* You must create your cube using SAS OLAP Cube Studio. The OLAP procedure does not support cubes with GEO dimensions.  $\Delta$



## APPENDIX

## 5

## Recommended Reading

---

*Recommended Reading* 377

---

### Recommended Reading

Here is the recommended reading list for this title:

- Introduction to the SAS Information Delivery Portal*
- SAS Integration Technologies: Administrator's Guide*
- SAS Integration Technologies: Developer's Guide*
- SAS Intelligence Platform: Overview*
- SAS Intelligence Platform: Security Administration Guide*
- SAS Management Console: User's Guide*
- SAS Web Infrastructure Kit: Developer's Guide*
- SAS Web Report Studio: User's Guide*

For a complete list of administration documents for the SAS Intelligence Platform, see <http://support.sas.com/913administration>.

For a list of SAS documentation, see <http://support.sas.com/documentation/onlinedoc/sas9doc.html>.

For a complete list of SAS publications, see the current *SAS Publishing Catalog*. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at

SAS Publishing Sales  
 SAS Campus Drive  
 Cary, NC 27513  
 Telephone: (800) 727-3228\*  
 Fax: (919) 677-8166  
 E-mail: [sasbook@sas.com](mailto:sasbook@sas.com)

Web address: [support.sas.com/pubs](http://support.sas.com/pubs)

\* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.



# Glossary

---

**administrative user**

a special user of a metadata server who can create and delete user definitions and logins. An administrative user can also perform administrative tasks such as starting, stopping, pausing, and refreshing the metadata server. Unlike an unrestricted user, an administrative user does not have unrestricted access to the metadata. You are an administrative user if your user ID is listed in the adminUsers.txt file or if you connect to the metadata server using the same user ID that was used to start the metadata server.

**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 set of computing resources that use the same authentication process. An individual uses the same user ID and password for all of the resources in a particular authentication domain. Authentication domains provide logical groupings for resources and logins in a metadata repository. For example, when an application needs to locate credentials that enable a particular user to access a particular server, the application searches the metadata for logins that are associated with the authentication domain in which the target server is registered.

**authentication provider**

a software component that is used for identifying and authenticating users. For example, Windows NT and LDAP both 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.

**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](#), [subscribe](#).

**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.

**collection portlet**

a portlet that contains a list of portal content items. The items can be of any content type or combination of types.

**content administrator**

See [group content administrator](#).

**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 a particular user account that has been established either in the operating system or with an alternative authentication provider such as Microsoft Active Directory or Lightweight Directory Access Protocol.

**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 page that is automatically added to each user's personal portal. You can remove a default page from your personal portal if you do not need it.

**delivery method**

another term for delivery transport. See delivery transport.

**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.

**development environment**

a computing environment in which application developers use software tools to write, compile, and debug programs. See also testing environment, production environment.

**encryption**

the act or process of converting data to a form that only the intended recipient can read or use.

**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. In a SAS metadata environment, a user group is represented by an IdentityGroup object.

**group content**

content that a group of portal users can access. Users who are designated as group content administrators can use the SAS Information Delivery portal's Share option to 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.

**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 (IFRAME).

**information map**

a collection of data items and filters that describes and presents data in a form that is relevant and meaningful to a business user. A user of a query and reporting application such as SAS Web Report Studio can easily build a business report by using the parts of an information map as the building blocks for queries.

**inline frame (IFRAME)**

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. See also URL display portlet.

**IOM (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.

**IOM server**

a SAS object server that is launched in order to fulfill client requests for IOM services. See also IOM (Integrated Object Model).

**Java Development Kit**

See JDK (Java Development Kit).

**Java RMI**

See RMI (remote method invocation).

**Java Virtual Machine**

See JVM (Java Virtual Machine).

**JavaServer page**

See JSP (JavaServer page).

**JDK (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.

**JSP (JavaServer page)**

a type of servlet that enables users to create Java classes through HTML.

**JVM (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.

**LDAP (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.

**Lightweight Directory Access Protocol**

See LDAP (Lightweight Directory Access Protocol).

**link**

(1) a portal content item that can be accessed using a URL; (2) a character string in a portal that you can click in order 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 combination of a user ID, a password, and an authentication domain. Each login provides access to a particular set of computing resources. In a SAS metadata environment, each login can belong to only one individual or group. However, each individual or group can own multiple logins.

**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 tier in which J2EE Web applications and J2EE enterprise applications execute.

**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 (online analytical processing)**

a software technology that enables users to dynamically analyze data that is stored in cubes.

**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 (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. See also portlet.

**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 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 Web 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 pool consists of one or more puddles. See also puddle.

**pooling**

the act or process of creating a pool. See pool.

**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 which 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 (PAR) file**

See PAR (portlet archive) file.

**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 PAR (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, testing environment.

**Public Kiosk**

a public page that is displayed when a user starts the SAS Information Delivery Portal but has not yet logged on.

**publication channel**

an information repository that has been established using the SAS Publishing Framework and which 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 pool.

**remote method invocation**

See RMI (remote method invocation).

**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, 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, 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 (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. See also JVM (Java Virtual Machine).

**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 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 whose contents are 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 application server that provides access to multidimensional data. The data is queried using the multidimensional expressions (MDX) language.

**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 SAS data set.

**SAS Workspace Server**

a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces.

**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.

**servlet container**

an execution environment for Java servlets that contains a Java Virtual Machine. The servlet container also provides other services for servlets and for the Web applications that those servlets are part of. For example, the servlet container converts HTTP requests that are sent by clients to Java objects that servlets can

work with, and it converts the output of servlets to HTTP responses. An example of a popular servlet container is the Apache Tomcat server.

**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, 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.

**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, 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 relationship**

a logical association through which one component of an application accepts verification that has already been performed by another component. For example, the establishment of a trust relationship enables users to log on to the application once, and then to access all associated resources without the need for re-authentication. See also trusted user.

**trusted user**

a special user of a metadata server who can acquire credentials on behalf of other users in a multi-tier server environment.

**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 (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.

**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), 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 (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.

**Web Distributed Authoring and Versioning**

See WebDAV (Web Distributed Authoring and Versioning).

**Web Infrastructure Kit**

a set of infrastructure components that can be used to develop new portlets for the SAS Information Delivery Portal, to customize the SAS Information Delivery Portal, or to build new Web applications using portal technology. The kit includes common Java components as well as SAS Foundation Services. It is included with SAS Integration Technologies.



**Web server**

a server machine and software that enable organizations to share information through intranets and through the Internet.

**WebDAV (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. See also delivery transport.

**WebDAV repository**

a collection of files that are stored on a Web server so that authorized users can read and edit them. See also WebDAV (Web Distributed Authoring and Versioning).

**XML (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.



# Index

---

## A

- accessibility features 3
- ACTIVE X device driver 124
- adding page templates to the portal 251
- adding pages to the portal 249
- adding portlets to the portal 262
- additional authentication
  - SAS Web Report Studio 22, 134
- Alerts portlet 298
- Apache HTTP Server
  - configuring cache control for static content 96
  - configuring to serve static content 86
  - proxy plug-ins and J2EE application server 90
- application servers, required 363
- applications
  - redistributing 347
- ArcGIS Server 371
- architecture
  - middle tier 9
  - SAS Intelligence Platform 8
- authentication
  - choosing a provider 20
  - for single sign-on (metadata server) 27
  - logon formats for Web applications 365
  - middle tier 83
  - trusted Web 29
- authorization
  - layers for SAS Web Report Studio 135
  - portal Web application 219
  - Xythos WebFile Server (WFS) 231
- Authorization Manager 229

## B

- banner images 125
- batch reports
  - pre-generated 136
- BI Manager plug-in 107

## C

- cache control
  - configuring for static content 96
- caching 307
- cascading style sheets (CSS) 140
  - CSS formats 141
- clustering 80
  - J2EE application servers 84

- columns
  - layout for SAS Web OLAP Viewer for Java 176
- conditional highlighting images 125
- configuration
  - Apache cache control for static content 96
  - ESRI map component 369
  - group content administrator 224
  - logging for SAS Web OLAP Viewer for Java 170
  - logs for SAS Web Report Studio 109
  - of portal after moving SAS Metadata Server 354
  - of portal after moving SAS Web Report Studio 353
- cross-tabulation tables
  - CSS formats for 143
- CSS formats 141
  - for display filters 149
  - for graphs 144
  - for tables 142
  - for text 147
  - supported properties for 150
- cubes
  - viewing 169
- custom portlets
  - adding to portal 268
  - permissions for 53
- custom report styles 140
  - elements in LocalProperties.xml 140
  - specifying style in properties file 140
- custom themes 188
  - deleting 333
  - deployment 328
- customized page deployment 244
- customizing the portal display 317

## D

- dashboard groups 311, 314
- dashboards 303
  - administrative tasks 304
  - data cache configuration 307
  - data source DSX files 304
  - enabling security for 310
  - JDBC data source for 305
  - performance improvement 306
  - permissions 312
  - pooling JDBC connections 308
  - security for 309
  - shared portlets configuration 313
- data cache
  - configuring 307

- data explorations 174
  - folder for 179
  - public 175
- data sources
  - DSX files 304
  - JDBC 305
  - SAS Web Report Studio 123
- debug logging 110
- default preferences 320
- deleting page templates from the portal 257
- demilitarized zone (DMZ) 80
- disclaimer text
  - adding to graphs and tables 139
- display filters
  - CSS formats for 149
- distributed certificates 43
- distributing reports
  - See* report scheduling and distribution
- DMZ (demilitarized zone) 80
- DSX files 304

## E

- editing page templates 256
- editing pages in the portal 247, 250
- ESRI map component 369
  - configuring OLAP cubes 374
  - configuring security 371
  - defining a map service 372
  - defining an ESRI server 371
  - software requirements 370

## F

- files
  - adding to portal Web application 275
- filters
  - display filters 149
  - report filters 108
- folders
  - adding to report storage structure 122
  - for personal data explorations 179
- fonts
  - for SAS Web Report Studio 127
- footers
  - SAS Web OLAP Viewer for Java 177

## G

- garbage collector 63
  - configuring 64
- graph data styles 144
- graphs
  - adding disclaimer text to 139
  - CSS formats for 144
  - in stored process output 124
- group content administrator 224
- groups
  - associating portlets with 272
  - dashboard groups 311, 314
  - defined in metadata 360
  - for organizing user accounts 21
  - planning for portal 220
  - required 359

## H

- headers
  - SAS Web OLAP Viewer for Java 177
- HTTP servers
  - configuring to serve static content 86
  - deploying themes to 330
  - load-balancing software and hardware 82
  - migrating themes to 331
  - static content deployed in 77
- HTTP session timeout interval 14

## I

- images
  - for SAS Web Report Studio 125
- importing
  - legacy reports 128
  - reports 128
- information maps
  - adding to portal Web application 299
  - upgrading 169
- initPortalData utility 212
- installation
  - SAS Web OLAP Viewer for Java 167

## J

- J2EE application servers
  - configuring a cluster of 84
  - proxy plug-ins and HTTP Server 90
  - tuning 64
  - Web applications deployed across cluster of 79
  - Web applications in single server 75
- Java
  - See also* SAS Web OLAP Viewer for Java
  - SAS BI Web Services for Java 14
- Java classes 188
- Java policy files
  - See* policy files
- Java RMI server, redistributing 348
- Java Virtual Machine
  - garbage collector 63
  - Just-in-Time compiler 62
  - JVM arguments 66
  - JVM options 59
  - memory options 62
  - quick start settings 60
  - tuning 58
- JavaBeans 188
- JavaServer pages (JSP)
  - detecting changes in 64
- JDBC
  - data sources 305
  - pooling connections 308
- Just-in-Time compiler 62
- JVM arguments 66
- JVM options 59

## K

- Key User Action Log 111
  - output 111
  - reporting events in 112

**L**

- LDAP logon format 366
- legacy reports
  - importing 128
- libraries
  - for recipient lists 157
- links
  - adding to portal Web application 274
- list tables
  - CSS formats for 142
- load balancing
  - software and hardware 82
- loading initial metadata for portal Web application 200
- Local Services 337
- LocalProperties.xml file 107, 140
- log files
  - changing logging level 110
  - configuring debug logging dynamically 110
  - Key User Action Log 111
  - SAS Web Report Studio 109
- Log Off link 180
- logging
  - configuring for SAS Web OLAP Viewer for Java 170
  - configuring portal Web application for 204
  - logon formats for Web applications 365
- logs
  - changing message formats 206
  - changing types, filenames, or locations 205
  - changing types of messages 205
  - configuring for SAS Web Report Studio 109
  - Key User Action Log 111

**M**

- manually refreshed reports 153
- map service for ESRI map component 372
- metadata
  - adding for publication channels 293
  - adding for syndication channels 288
  - adding for Web applications 279
    - for portal content 240
    - for WebDAV server 345
  - initial loading for portal Web application 200
  - removing with SAS Portal Metadata Tool 215
  - restoring default portal metadata 218
  - storage for service deployment configurations 337
  - updating permissions for portal 212
- metadata identities
  - surrogate 129
- metadata objects
  - for reports 121
  - synchronizing report files with 121
- metadata server
  - See* SAS Metadata Server
- Microsoft Active Directory logon format 366
- middle tier 7
  - Apache cache control for static content 96
  - architecture 9
  - authentication 83
  - availability criteria 73
  - choosing configuration 71
  - cluster of J2EE application servers 84
  - configuration 58
  - deployment scenarios 70
  - HTTP server configuration 86
  - maintainability criteria 75
  - performance and scalability criteria 74
  - proxy plug-ins 90
  - redeploying applications 14
  - security criteria 72
  - security implementation 20
  - tuning J2EE application server 64
  - tuning Java Virtual Machine 58
  - tuning servlet container 64
  - tuning WebSphere 66

**O**

- ODS
  - importing legacy reports with 128
- OLAP
  - See* SAS Web OLAP Viewer for Java
- OLAP cubes
  - configuring for ESRI map component 374
- Open dialog box 179
- output
  - Key User Action Log 111
  - style for stored processes 124

**P**

- page templates 247
  - adding to the portal 251
  - deleting from the portal 257
- pages 242
  - adding to the portal 249
  - administration 243
  - attributes 245
  - customized deployment 244
  - editing in the portal 247, 250
  - personal 245
  - ranks 243
  - removing from the portal 250
  - shared 246
  - templates 247
- performance
  - dashboards 306
  - middle tier 74
  - SAS Web OLAP Viewer for Java 171
  - SAS Web Report Studio 113
- permission tree folders 233
  - creating 234
  - removing 235
  - verifying 235
- permission trees
  - creating for syndication channels 287
  - creating for Web applications 278
- permissions
  - adding to policy files 45
  - custom portlets and Web applications 53
  - dashboards 312
  - for report scheduling and distribution 154
  - planning for users and groups 23
  - portal components 50
  - portal content 49
  - provided by SAS 46
  - SAS Services Application 49
  - servers 48
- personal pages 245
- policy files 45
  - adding permissions to 45

- modifying 47
    - permissions for portal content 49
    - permissions for SAS Services Application 49
    - permissions for servers 48
    - security restrictions for 46
  - pooling JDBC connections 308
  - portal administration tools 209
    - for redeployment 211
    - initPortalData utility 212
    - portal Options menu 210
    - Quiesce portlet 213
    - SAS Portal Metadata Tool 215
  - portal authorization 222
    - configuring a group content administrator 224
    - implementing 223
    - permission trees 233
    - planning for users and groups 220
    - publication channels 229
    - stored processes 229
    - tasks 219
    - Xythos WebFile Server (WFS) 231
  - portal components 11
  - portal Options menu 210
  - portal views 197
  - portal Web application 183
    - adding applications 276
    - adding content 239
    - adding custom portlets 268
    - adding files 275
    - adding information maps 299
    - adding links 274
    - adding portlets 262
    - adding publication channels 292
    - adding reports 300
    - adding SAS packages 290
    - adding SAS Web OLAP Viewer for Java (example) 282
    - adding SAS Web Report Studio (example) 282
    - adding syndication channels 286
    - adding WebDAV graph portlets 264
    - administration tools 209
    - administrative tasks 196
    - changing default preferences 318
    - components 188
    - configuring for logging 204
    - custom themes 188
    - customizing the portal display 317
    - deploying custom themes 328
    - deploying portlets 270
    - deployment of SAS Foundation Services 341
    - execution of stored processes 295
    - hiding portlets from users 272
    - loading initial metadata 200
    - locations of administrative files 199
    - pages 242
    - permissions for components 50
    - portletlets 258
    - prerequisites for administration 191
    - quiescing 213
    - redeploying 211
    - sharing content 226
    - stored process administration 296
    - stored processes 294
    - upgrading preferences to 9.1.3 format 327
    - verifying operation 198
  - portletlets 258
    - adding to the portal 262
    - associating with groups 272
    - custom 268
    - deployment 270
    - execution of local and remote 271
    - hiding from users 272
    - hot-deploy 271
    - predefined 261
    - shared dashboard portlets 313
    - templates 259
    - WebDAV graph portlets 264
  - pre-generated reports 153
    - See also* report scheduling and distribution
    - batch 136
  - preferences, default 320
  - progressive bar charts
    - CSS formats for 146
  - proxy plug-ins 90
  - Public Kiosk
    - administration 202
    - removing 203
  - publication channels
    - adding to portal Web application 292
    - authorization 229
- ## Q
- query cache 114
    - disabling 116
    - library location 115
  - Quiesce portlet 213
- ## R
- recipient lists
    - considerations for creating 163
    - creating 159
    - creating with SQL procedure 162
    - for report distribution 158
    - library for 157
  - redemption
    - middle-tier applications 14
    - portal Web application 211
    - SAS Web OLAP Viewer for Java 171
    - SAS Web Report Studio 116
  - redistributing applications and servers 347
  - Remote Services 12, 338
  - removing pages from the portal 250
  - report content files
    - synchronizing with metadata objects 121
  - report definitions file 121
  - report distribution
    - See* report scheduling and distribution
  - report filters 108
  - report scheduling and distribution 153
    - library for recipient lists 157
    - methods for 155
    - permissions for 154
    - prerequisites 156
    - recipient lists 158
    - scheduling compared with distribution 155
  - Report Studio Configuration plug-in 108
  - report styles
    - customizing 140
    - elements in LocalProperties.xml 140
    - specifying style in properties file 140

- reporting
    - adding folders to storage structure 122
    - components 10
    - standard storage containers for 120
    - storage for 119
    - verifying storage structure 122
  - reports
    - See also* pre-generated reports
    - access to 134
    - adding to portal Web application 300
    - importing 128
    - importing legacy reports 128
    - manually refreshed 153
    - metadata objects for 121
    - protecting data in temporary files 138
    - protecting WebDAV server content 137
    - row-level security 137
    - SAS report model 128
    - SAS Web Report Studio as default viewer 353
    - static snapshot reports 154
  - restoring default portal metadata 218
  - roles
    - SAS Web Report Studio 130
  - row-level security 137
- S**
- SAS BI Web Services for Java
    - redeploying 14
  - SAS Business Intelligence Dashboard
    - See* dashboards
  - SAS Documentation Web application 189
  - SAS Foundation Services 9, 335
    - deployment in portal Web application 341
    - SAS Services Application 335
    - service deployment configurations 336
  - SAS Information Delivery Portal 11
    - See also* portal Web application
    - available worker threads 65
    - configuring for SSL 44
    - configuring HTTP Server for static content 86, 88
    - features 185
    - redeploying 14
    - viewing reports 300
  - SAS Information Map Studio
    - upgrading information maps 169
  - SAS Intelligence Platform
    - architecture 8
  - SAS Management Console
    - Authorization Manager 229
    - BI Manager plug-in 107
    - Report Studio Configuration plug-in 108
  - SAS Metadata Server 27
    - accessing service deployments from 342
    - authentication for single sign-on 27
    - logon formats 365
    - portal configuration after redistributing 354
  - SAS packages
    - adding to portal Web application 290
    - creating 291
  - SAS Portal Metadata Tool 215
  - SAS Preferences Web application 189
    - changing default preferences 318
    - redistributing 350
  - SAS programs
    - converting to stored processes 124
  - SAS Query and Reporting Services 11
  - SAS Services Application 12, 335
    - heap size 83
    - JRE communication with the portal 338
    - permissions for 49
    - redistributing 348
    - Remote Services 338
    - running as a Windows service 345
  - SAS Stored Process Web application 189
    - redistributing 349
  - SAS Themes Web application 189
    - redistributing 351
  - SAS Web Infrastructure Kit
    - components 11
    - features 185
  - SAS Web OLAP Viewer for Java 12, 167
    - adding to portal Web application (example) 282
    - column layout 176
    - configuring logging 170
    - customizing 173
    - customizing display for viewers 175
    - data explorations 174, 175, 179
    - default panel 177
    - header and footer styles 177
    - HTTP session timeout interval 14
    - installing 167
    - performance improvement 171
    - redeploying 14, 171
    - upgrading information maps 169
    - viewing cubes 169
  - SAS Web Report Studio 10, 101
    - See also* report scheduling and distribution
    - access to reports 134
    - adding content for report creators 123
    - adding to portal Web application (example) 282
    - administrative files 105
    - administrative tasks 102
    - authorization layers 135
    - configuring HTTP Server for static content 86, 87, 88
    - configuring logs 109
    - customizing report styles 140
    - data sources 123
    - fonts for 127
    - images for 125
    - importing reports 128
    - input resources 123
    - performance improvement 113
    - portal configuration after redistributing 353
    - pre-generated reports 153
    - protecting data 138
    - redeploying 14, 116
    - roles 130
    - setting up users 129
    - storage for reporting 119
    - stored processes for 124
    - using as default report viewer 353
  - SAS Web Report Viewer 10
    - configuring HTTP Server for static content 86, 87, 88
    - redeploying 14
    - redistributing 352
  - scalability 74
  - scheduling reports
    - See* report scheduling and distribution
  - Secure Sockets Layer (SSL) 42
    - configuring SAS Information Delivery Portal for 44
    - configuring servlet container for 42

- configuring Web applications for 42
- considerations for use 82
- importing distributed certificates 43
- security
  - criteria for configuration 72
  - dashboards 309
  - ESRI server 371
  - planning for implementation 20
  - pre-generated batch reports 136
  - reports access 134
  - row-level 137
- servers
  - defining for ESRI 371
  - permissions for 48
  - redistributing 347
  - required application servers 363
  - start-up order 13
- service deployment configurations 336
  - accessing from SAS Metadata Server 342
  - accessing from XML files 343
  - reimporting 339
  - storing metadata 337
- servlet container
  - configuring for SSL 42
  - deploying themes to 329
  - tuning 64
- servlets
  - detecting changes in 64
- session affinity 80
- session timeout interval 14
- shared dashboard portlets 313
- shared pages 246
- sharing portal content 226
- single sign-on 24
  - metadata server authentication 27
  - trusted Web authentication 29
- SQL procedure
  - creating recipient lists 162
- SSL (Secure Sockets Layer)
  - See* Secure Sockets Layer (SSL)
- start-up order for servers 13
- starting Web applications 13
- static content
  - configuring Apache cache control for 96
  - configuring HTTP Server for 86
  - deployed in HTTP server proxy 77
- static snapshot reports 154
- storage containers
  - for reporting 120
- stored processes
  - administrative tasks for the portal 296
  - converting SAS programs to 124
  - execution in the portal 295
  - for SAS Web Report Studio 124
  - generation of alerts in the portal 298
  - graphs in output 124
  - in portal Web application 294
  - non-streaming 295
  - output styles 124
- surrogate metadata identity 129
- synchronized objects
  - custom report styles 148
- syndication channels
  - adding to portal Web application 286

## T

- tables
  - adding disclaimer text to 139
  - CSS formats for 142
- temporary files
  - protecting 138
- text
  - CSS formats for 147
- themes 180
  - changing default 332
  - compressing files 328
  - deleting custom themes 333
  - deploying to HTTP servers 330
  - deploying to servlet container 329
  - migrating to HTTP servers 331
- thread-pool properties 69
- timeout interval 14
- trusted Web authentication 83
  - for single sign-on 29
  - logon format 366
  - sample sequence 30
  - setting up 32

## U

- updating metadata for portal permissions 212
- users
  - configured on system 359
  - defined in metadata 360
  - permissions 23
  - planning accounts 21
  - planning for portal 220
  - portal Web application administration 193
  - required 359
  - SAS Web Report Studio 129
  - setting up for Web authentication 38

## W

- Web applications
  - deployed across J2EE application server cluster 79
  - in single J2EE application server 75
  - overview of administrative tasks 5
  - prerequisites for administering 4
  - starting 13
- WebDAV graph portlets 264
- WebDAV server 12
  - metadata for 345
  - protecting report content 137
- WebLogic
  - configuring a cluster of J2EE application servers 84
  - modifying XML files for Web authentication 36
- WebSphere
  - JVM arguments 66
  - setting thread-pool properties 69
  - tuning 66
  - tuning values for AIX 69
- WebSphere Application Server
  - enabling SSL 43
  - updating configuration for Web authentication 38
- worker threads 65



**X**

## XML files

accessing service deployment configurations 343

## Xythos WebFile Server (WFS)

adding files 275

authorization 231

configuring for Web authentication 40

content services for portal Web application 189



# Your Turn

---

We welcome your feedback.

- If you have comments about this book, please send them to **[yourturn@sas.com](mailto: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](mailto:suggest@sas.com)**.



# SAS® Publishing delivers!

Whether you are new to the workforce 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.

## SAS® Press Series

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from the SAS Press Series. Written by experienced SAS professionals from around the world, these books deliver real-world insights on a broad range of topics for all skill levels.

[support.sas.com/saspress](http://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: 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, and hard-copy books.

[support.sas.com/publishing](http://support.sas.com/publishing)

## SAS® Learning Edition 4.1

Get a workplace advantage, perform analytics in less time, and prepare for the SAS Base Programming exam and SAS Advanced Programming exam with SAS® Learning Edition 4.1. This inexpensive, intuitive personal learning version of SAS includes Base SAS® 9.1.3, SAS/STAT®, SAS/GRAPH®, SAS/QC®, SAS/ETS®, and SAS® Enterprise Guide® 4.1. Whether you are a professor, student, or business professional, this is a great way to learn SAS.

[support.sas.com/LE](http://support.sas.com/LE)



THE  
POWER  
TO KNOW®





