



SAS® Forecast Server 14.2: Administrator's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Forecast Server 14.2: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Forecast Server 14.2: Administrator's Guide

Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

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

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

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

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

February 2020

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 trademarks of their respective companies.

14.2-P2:fsag

Contents

<i>About This Book</i>	<i>vii</i>
<i>Accessibility</i>	<i>ix</i>

PART 1 Introduction to SAS Forecast Server 1

Chapter 1 • Understanding SAS Forecast Server	3
What Is SAS Forecast Server?	3
Overview of the SAS Forecast Server Architecture	4
SAS Forecast Server Integration	6

PART 2 Installing the SAS Forecast Server 9

Chapter 2 • Pre-Installation Tasks	11
Overview of Pre-Installation Tasks	11
Verify System Requirements	12
Shared Memory Requirements for UNIX Systems	12
Standard SAS User Accounts	12
User Accounts for SAS Forecast Server	15
Determine the Location of the SAS Environment URL	16

Chapter 3 • Installing SAS Forecast Server	17
Preparing to Install and Configure a New Deployment	17
Default File Locations	17
How to Install and Configure SAS Forecast Server	18

Chapter 4 • Post-Installation Tasks for the Server	27
Minimum Requirements for Post-Installation	27
Overview of Post-Installation Tasks	27
Creating User Groups for UNIX Environments	28
Managing Roles and Capabilities	29
How to Configure the SAS Forecast Server Administrator and Users	34
Creating and Configuring a Product Environment	35
Configuring the SAS Forecast Studio Tasks	37

Chapter 5 • Post-Installation Tasks for the Middle Tier	39
Overview of Post-Installation Tasks	39
Configuring Integrated Windows Authentication (IWA)	40
Changing Configuration Properties in SAS Management Console	40
Specifying a Time-Out	40
Configuring and Using Java Web Start	41
Set the Preferred Environment in the Log On Dialog Box for SAS Forecast Studio	42
Configuring the Archive Functionality	42
Enable Server Selection for Reports	43

PART 3 Configuration Tasks 45

Chapter 6 • Creating and Configuring Libraries	47
Understanding Libraries	47
Enable Expanded Support for Libraries	49
How to Define a Pre-assigned Library in the Metadata	50
How to Define a Metadata Library That Is Assigned by SAS Forecast Server	51
Create an Automatic Environment Library	51
Create a Configured Environment Library	52
Chapter 7 • Specifying Security Permissions for Users and Groups	53
Enabling Users to Access SAS Forecast Server	53
Securing Access to SAS Forecast Server	56
Chapter 8 • Administering Reports to Use in SAS Forecast Studio	61
Overview of Reports	61
Use the Sample Reports	65
Register Reports	66
Create Reports	69
Chapter 9 • Miscellaneous Administration Tasks	85
Adding Start-Up and Shutdown Code to a Project	85
Remove a Configuration with the SAS Deployment Manager	87
Archive a Project	87
Configure LOCKDOWN Feature	88

PART 4 Using the SAS Forecast Server Clients 89

Chapter 10 • SAS Forecast Studio	91
Requirements for Starting SAS Forecast Studio	91
Running SAS Forecast Studio on Your Desktop	92
Using Java Web Start	92
Starting SAS Forecast Studio with Options	94
Adding Custom Time Intervals to SAS Forecast Studio	96
Adding a Custom Format to SAS Forecast Studio	99
Creating a Customized List of Events	101
Chapter 11 • SAS Forecast Batch Interface	105
About the SAS Forecast Server Batch Interface	106
Dictionary	112
Examples	193

PART 5 SAS Forecast Server Utility Macros 201

Chapter 12 • Using the SAS Forecast Server Utility Macros	203
About the SAS Forecast Server Utility Macros	203
Dictionary	203

PART 6 Appendixes 211

Appendix 1 • Troubleshooting SAS Forecast Server	213
Gathering Information	213
Additional Resources	217
Troubleshooting SAS Forecast Server	219
Troubleshooting the Java Web Start Client for SAS Forecast Studio	226
Troubleshooting the SAS Forecast Server Macros	227
Performance Tuning for SAS Forecast Server	228
 Recommended Reading	229
Glossary	231
Index	233

About This Book

Audience

The *SAS Forecast Server: Administrator's Guide* is intended for system administrators who need to install, configure, and optimize the SAS Forecast Server. SAS and other programming expertise are not required.

SAS Forecast Server uses the SAS Intelligence Platform, so your system administrator should be familiar with the SAS Intelligence Platform. For more information, see <http://support.sas.com/documentation/onlinedoc/intellplatform>.

The system administrator performs these tasks:

- installs and configures the required SAS Intelligence Platform software on the required operating system.
- administers the metadata for the SAS Forecast Server and SAS Data Integration Studio (if your site uses ETL processes). The system administrator maintains the metadata for servers, users, and other global resources that are required by the SAS Forecast Server and SAS Data Integration Studio.

The SAS Forecast Server administrator performs these tasks:

- administers the metadata for SAS Forecast Server. Either the SAS Forecast Server administrator or the system administrator maintains the metadata for servers, users, and other global resources that are required by the SAS Forecast Server.
- maintains the data and performs other administration tasks that enable users to analyze data.

Prerequisites

Review the system requirements documentation before you install the SAS Forecast Server to ensure that your system meets the requirements.

Note: SAS Forecast Server has internal access to SAS/OR software with limited functionality, but direct access to and the full functionality of SAS/OR software is not provided. To access the full functionality, your site must license SAS/OR.

Documentation Conventions

SASROOT

represents the SAS Foundation directory.

SAS_CONFIG

represents the SAS configuration directory.

SAS_HOME

represents the SAS installation directory

Accessibility

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

Part 1

Introduction to SAS Forecast Server

Chapter 1

Understanding SAS Forecast Server 3

Chapter 1

Understanding SAS Forecast Server

What Is SAS Forecast Server?	3
Overview of the SAS Forecast Server Architecture	4
Architecture Diagram	4
The SAS Intelligence Platform Components	4
The SAS Forecast Server Components	5
SAS Forecast Server Integration	6

What Is SAS Forecast Server?

SAS Forecast Server is a large-scale, automatic forecasting solution that enables organizations to produce huge quantities of high-quality forecasts, quickly and automatically.

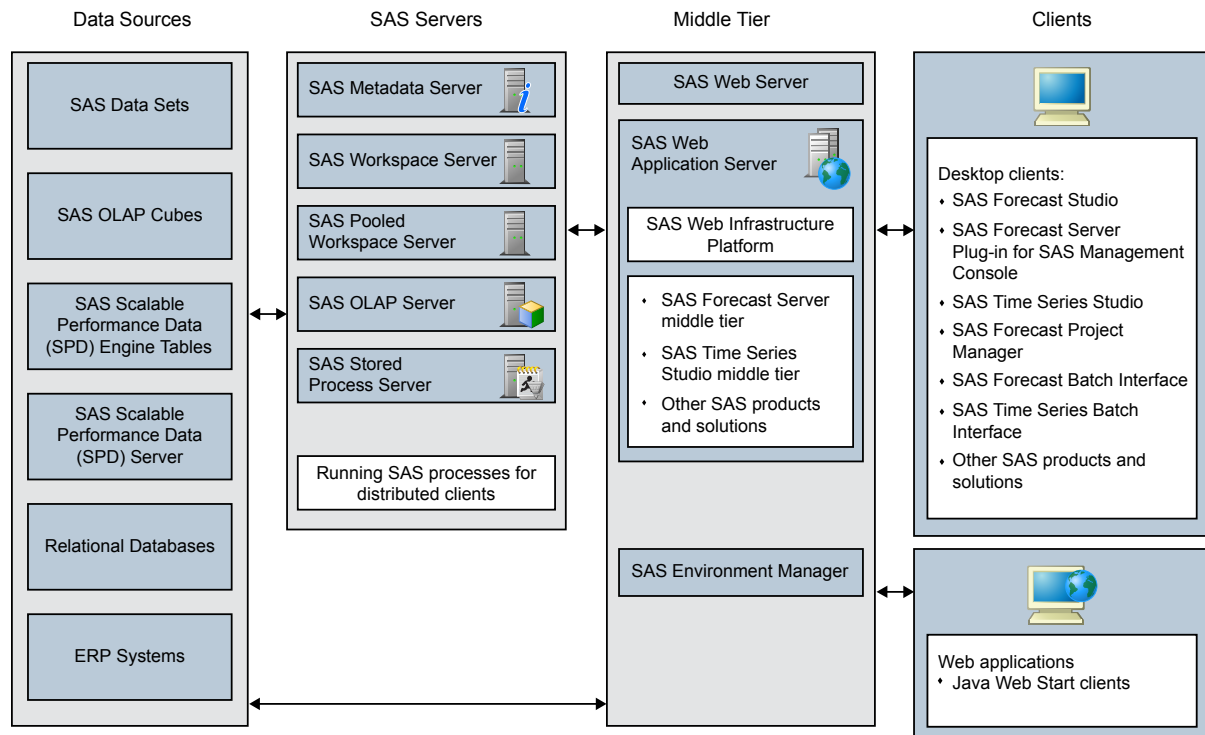
SAS Forecast Server has these main components:

- SAS Forecast Server middle tier, which is supported by the SAS Web Infrastructure Platform
- SAS Forecast Studio, which is the graphical user interface (based on Java) to the forecasting and time series analysis procedures in SAS High-Performance Forecasting and SAS/ETS software

Overview of the SAS Forecast Server Architecture

Architecture Diagram

The following figure shows how SAS Forecast Server integrates with the SAS Intelligence Platform.



The SAS Intelligence Platform Components

SAS Forecast Server works with the SAS Intelligence Platform, which is installed and configured when you deploy SAS Forecast Server.

Integration Technologies

The SAS Metadata Server provides an enterprise-level repository for SAS server configurations and application management metadata. Products such as SAS Forecast Server store metadata about users and other resources. Administrators use SAS Management Console to administer the SAS Metadata Server and the SAS server configurations. SAS Forecast Server uses the SAS Metadata Server to obtain metadata about SAS libraries, the SAS Workspace Server, and the SAS Object Spawner, and to authenticate users.

The SAS Workspace Server provides all computation and intermediate data storage services. SAS Forecast Server uses the SAS Workspace Server to execute SAS High-Performance Forecasting and to save data to SAS data sets. The SAS Forecast Server accesses the SAS Workspace Server and the SAS Object Spawner through the middle tier.

SAS Web Infrastructure Platform

The SAS Intelligence Platform provides a service-oriented framework for building enterprise applications for the middle tier. Because the middle tier for SAS Forecast Server is implemented using this framework, SAS Forecast Server has core integration support for common infrastructure needs, such as auditing and resource publishing.

The SAS Forecast Server Components

The SAS Forecast Server bundle consists of the following components:

SAS Forecast Server Procedures

In previous releases, the SAS Forecast Server Procedures were called SAS High-Performance Forecasting.

SAS Forecast Server

The SAS Forecast Server middle tier is deployed on an application server. The SAS Forecast Server middle tier is a collection of custom services that are used to support SAS Forecast Server clients and to coordinate their use of the underlying SAS Intelligence Platform.

SAS Forecast Studio

SAS Forecast Studio is a Java-based client application of SAS Forecast Server. With its graphical user interface, SAS Forecast Studio provides the user visual access to SAS Forecast Server projects. A connection to the middle tier is required by SAS Forecast Studio. The middle tier reports and manages the states of SAS Forecast Server projects.

SAS Forecast Server Plug-in for SAS Management Console

The SAS Forecast Server Plug-in for SAS Management Console provides a graphical user interface for managing the resources and content of the SAS Forecast Server middle tier. You can use the SAS Forecast Server Plug-in for SAS Management Console to perform the administrative tasks that are required to create and maintain an integrated environment. SAS Forecast Studio generates and uses the content, and the plug-in manages the project and its content. Managing includes copying an environment, moving a project, converting a project to a new release, and deleting an environment.

SAS Forecast Project Manager

The SAS Forecast Project Manager is a stand-alone management client designed to require only HTTP-based communication. The SAS Forecast Project Manager enables you to configure the project. (The SAS Forecast Project Manager has all of the management capabilities that are available in the SAS Forecast Server Plug-in for SAS Management Console.)

SAS Forecast Batch Interface

The SAS Forecast Batch Interface consists of client macros to use with the SAS Foundation. These macros send requests to the SAS Forecast Server middle tier to perform actions, including creating and managing projects and updating forecasts just like the other SAS Forecast Server clients. Most actions focus on managing projects and environments. These actions parallel actions available in the SAS Forecast Project Manager and the SAS Forecast Server Plug-in for SAS Management Console. In addition, macros for creating and running projects are included in the SAS Forecast Batch Interface. As a result, the SAS Forecast Batch Interface is the best option for scripting and scheduling.

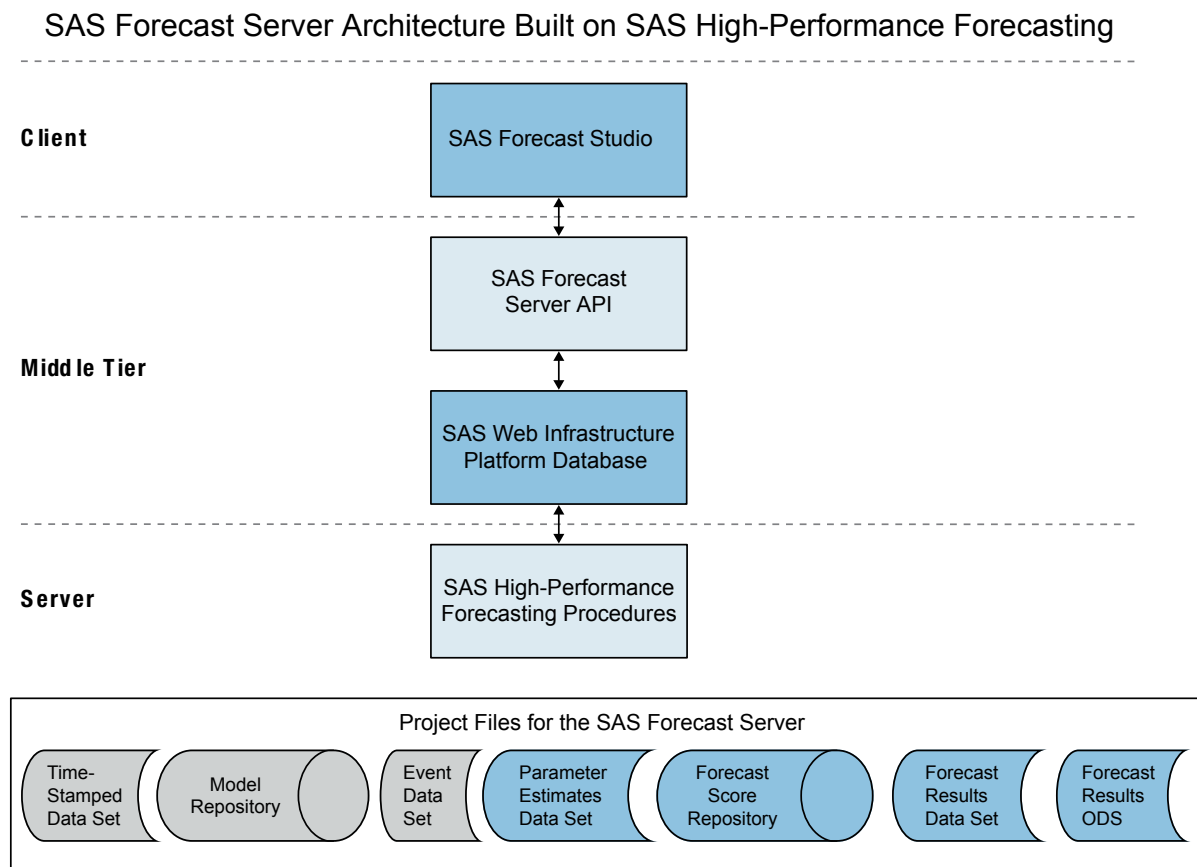
SAS Time Series Studio

SAS Time Series Studio enables you to interactively explore time series data. For more information, see *SAS Time Series Studio: Administrator's Guide*.

SAS Forecast Server Integration

The following figure shows the access points for SAS Forecast Studio, the SAS Forecast Server middle tier, and the SAS High-Performance Forecasting procedures.

Figure 1.1 The SAS Forecast Server Integration Architecture



The following conditions must be true for SAS Forecast Server to run:

- The SAS Metadata Server is running.
- The SAS Object Spawner is running.
- The SAS Web Infrastructure Platform and SAS Forecast Server are running on an application server.
- The input data for SAS Forecast Server is accessible through configured SAS libraries.

SAS Forecast Studio accesses the SAS Forecast Server middle tier (respectively) when these clients need to access the SAS Metadata Server or the SAS Workspace Server. The

SAS Forecast Server middle tier coordinates the use of the SAS Intelligence Platform (the SAS Metadata Server and the SAS Workspace Server) for the following purposes:

- access and return SAS library metadata from the SAS Metadata Server
- execute SAS High-Performance Forecasting procedures and return results
- authenticate users on the SAS Metadata Server

For more information about the SAS Intelligence Platform, see the SAS Intelligence Platform documentation set at <http://support.sas.com/documentation/onlinedoc/intellplatform>.

Part 2

Installing the SAS Forecast Server

<i>Chapter 2</i>	
Pre-Installation Tasks	<i>11</i>
<i>Chapter 3</i>	
Installing SAS Forecast Server	<i>17</i>
<i>Chapter 4</i>	
Post-Installation Tasks for the Server	<i>27</i>
<i>Chapter 5</i>	
Post-Installation Tasks for the Middle Tier	<i>39</i>

Chapter 2

Pre-Installation Tasks

Overview of Pre-Installation Tasks	11
Verify System Requirements	12
Shared Memory Requirements for UNIX Systems	12
Standard SAS User Accounts	12
Overview of User Accounts	12
Internal User Accounts	13
External User Accounts	13
Rights and Permissions Required by External User Accounts on Windows	14
Requirements for External User Accounts on UNIX	15
User Accounts for SAS Forecast Server	15
About the User Accounts for SAS Forecast Server	15
Create an Operating System Account for the Product Administrator	15
Create an Operating System Account for a SAS Forecast Studio User	16
Determine the Location of the SAS Environment URL	16

Overview of Pre-Installation Tasks

Perform the following tasks before installing SAS Forecast Server:

1. Review the system requirements documentation. For more information, see [“Verify System Requirements” on page 12](#).
2. Prepare the computing environments for user accounts and groups. Create the following external user accounts:
 - SAS Installer
 - SAS Spawned Server
 - SAS First User
 - SAS Forecast Server product administrator
 - An account for each user of SAS Forecast Server (local account or network directory service account of which the machine is a member)

For more information about user accounts, see [“Standard SAS User Accounts” on page 12](#) and [“User Accounts for SAS Forecast Server” on page 15](#).

3. If your site will be using the SAS Forecast Studio client, determine the location of the SAS environment URL. For more information, see “[Determine the Location of the SAS Environment URL](#)” on page 16.

Verify System Requirements

Review the system requirements documentation before you install the SAS Forecast Server to ensure that your system meets the requirements.

Note: In order to launch SAS Forecast Studio via the Java Web Start client, Java Runtime Environment (JRE) 1.7 or later needs to be installed on the client machine.

Shared Memory Requirements for UNIX Systems

In order to ensure enough shared memory for operating the application's database on the SAS Advanced Analytics Common Data Server, it is recommended that you extend the size of your shared memory to 6 gigabytes.

Refer to the documentation for your operating system and machine to determine the current shared memory settings. For example, on Linux for 64-bit machines, you can run the command `ipcs -l` on the command line of your operating system and note the value in the results for maximum total shared memory.

To extend the shared memory to 6 gigabytes, refer to the documentation for your operating system.

Standard SAS User Accounts

Overview of User Accounts

SAS Forecast Server uses the standard user accounts and groups that SAS software uses. See the pre-installation checklist that is included in the deployment plan for details about standard user accounts and groups. For more information about deployment plans, see “About Deployment Plans” in the *SAS Intelligence Platform: Installation and Configuration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform>. For information about creating groups and adding user accounts, see your Linux or Windows documentation.

In SAS Forecast Server, you use two types of user accounts:

- Internal user accounts are accounts that are known only to SAS. They are created and authenticated in metadata internally, rather than externally.
- External user accounts are accounts that are defined outside of SAS metadata. These accounts are local to a machine, or they are defined in a network directory service of which the machine is a member, such as Lightweight Directory Access Protocol (LDAP).

Note: The SAS Installer, SAS Spawnd Server and SAS First User (if used) external user accounts must exist on each SAS Workspace Server that has SAS Forecast Server projects before you run the SAS Deployment Wizard.

Internal User Accounts

Internal user accounts are known only to SAS. They are created and authenticated in metadata internally, rather than externally. SAS identifies an internal account by appending @saspw to the user ID. For two of the required user accounts, SAS Administrator and SAS Trusted User, the SAS Deployment Wizard prompts you to create internal user accounts.

The following table describes the default internal user accounts. (SAS internal user accounts are authenticated on the SAS Metadata Server.)

Internal User Account	User ID
SAS Administrator: The user account with privileges associated with the SAS Metadata Unrestricted Users role.	sasadm@saspw
SAS Trusted User: The user account that can impersonate other users in connections to the SAS Metadata Server. Some SAS processes use this account to communicate with the SAS Metadata Server on a client's behalf.	sastrust@saspw
SAS Anonymous Web User: The optional user account that can be used to grant web clients anonymous access to certain SAS Web Infrastructure Platform applications.	webanon@saspw

For more information about SAS internal user accounts, see [SAS 9.4 Intelligence Platform Installation and Configuration Guide](#).

External User Accounts

External user accounts are accounts that are defined outside of SAS metadata. These accounts are local to a machine, or they are defined in a network directory service of which the machine is a member, such as LDAP. SAS requires external user accounts for two purposes: installing the software and running certain SAS server processes. You must define these external user accounts before you run the SAS Deployment Wizard.

The following table describes the external user accounts for SAS Forecast Server.

External User Account	Recommended User ID
SAS Installer: The required user account that is used to install SAS. Do not use root as the SAS Installer user ID.	sas

External User Account	Recommended User ID
SAS Spawnd Server: The required user account for the process owner of SAS Stored Process Servers and SAS Pooled Workspace Servers.	sassrv
SAS First User: The optional user account that serves as the SAS First User account.	sasdemo

For more information about SAS external user accounts, see [SAS 9.4 Intelligence Platform: Installation and Configuration Guide](#).

Rights and Permissions Required by External User Accounts on Windows

Operating systems require you to assign certain rights (or permissions) to the external user accounts that are used to deploy and run SAS.

Purpose	Required Rights	Recommended User ID
SAS Installer	Administrator rights (user must be a local administrator on the machine or a member of the administrator's group)	sas
SAS Spawnd Servers	- Log on as a batch job (can be assigned by the SAS Deployment Wizard) - Act as part of the operating system	sassrv
SAS First User	Log on as a batch job (can be assigned by the SAS Deployment Wizard)	sasdemo

If you are running Windows, follow these steps to grant the **Act as part of the operating system** user right:

- From **Administrator Tools**, click **Local Security Policy**. Expand **Local Policies**, and then click **User Rights Assignment**.
- In the Policy dialog box, right-click **Act as part of the operating system**, and select **Properties**. The Act as part of the operating system Properties dialog box appears. Click **Add User or Group**.
- The Select Users, Computers, Service Accounts, or Groups dialog box appears. In the **Enter the object names to select** window, type **sassrv**. Click **OK**. Click **OK** again.

Requirements for External User Accounts on UNIX

In UNIX environments, you must ensure that the group that you designate as the primary group for the SAS Installer account also contains the SAS Spawnd Server account. This group does not have to be the primary group for the SAS Spawnd Server account. For example, you can create a group called sasusers that is the primary group for the SAS Installer user. The SAS Spawnd Server user must also be a part of the sasusers group, but it does not have to be its primary group.

User Accounts for SAS Forecast Server

About the User Accounts for SAS Forecast Server

SAS Forecast Server has one internal user account that is created during deployment. The user ID for this account is fsmeta. Like the other internal user accounts, it is identified in SAS as fsmeta@saspw. The SAS Deployment Wizard prompts for a password for this account. You specify a password, and record it for reference.

Users of SAS Forecast Server must have a valid host operating system account, and you must associate that account with a metadata user using SAS Management Console. In addition, these users must have full access to the appropriate project areas on each SAS Workspace Server. User accounts can be created as a pre-installation or post-installation task. For more information, see [“Create an Operating System Account for a SAS Forecast Studio User”](#) on page 16.

A product administrator account is a user account with additional permissions. The product administrator is not the same as a general administrator, such as the SAS administrator (sasadm@saspw). You must create the valid host operating system account for the product administrator as a pre-installation task. For more information, see [“Create an Operating System Account for the Product Administrator”](#) on page 15.

Create an Operating System Account for the Product Administrator

On each SAS Workspace Server that has SAS Forecast Server projects, create an operating system account for the product administrator of SAS Forecast Server.

If the SAS Workspace Server is running on Windows, use one of the following methods to create this operating system account:

- If you are working on a local machine, complete the following steps to create the operating system account:
 1. If you are running Windows 7, right-click the computer icon on your desktop, and select **Manage**. The Computer Management dialog box appears.
 2. In the left navigation pane, expand the **Local Users and Groups** folder. The Users folder and Groups folder appear.
 3. Right-click the **Users** folder, and select **New User**. The New User dialog box appears.
 4. In this dialog box, complete these tasks:
 - Specify a user name and password.

Note: In Windows, you cannot enter `<domain>\username`. You enter `username` only. In the SAS Deployment Wizard and SAS Management Console, you must enter `<domain>\username`.

- Clear the **User must change password at next logon** check box.
- Select the **User cannot change password** check box.
- Select the **Password never expires** check box.

Click **Create**.

5. Click **Close**.

- Define the new user (for example, `<domain>\username`) on the Microsoft Active Directory Server.

Grant the user permission **Log on as a Batch Job**.

Create an Operating System Account for a SAS Forecast Studio User

Create an operating system account for each SAS Forecast Studio user. SAS Forecast Studio runs on Windows. For more information about creating operating system accounts in Windows, see “Create an Operating System Account for the Product Administrator” on page 15.

Determine the Location of the SAS Environment URL

During the deployment of the SAS Forecast Server, you are prompted by the SAS Deployment Wizard to specify the URL location of the SAS environment file (named `sas-environment.xml`). An example is `http://server-name:port/sas/sas-environment.xml`. This file defines a set of SAS deployments at your site for the SAS Forecast Studio and batch macro client applications to use. The `sas-environment.xml` file does not need to physically exist at the URL location that you specify in the SAS Deployment Wizard during deployment. However, knowing the intended location of this URL is important and efficient because during every SAS Forecast Server client installation, you are prompted to specify this value. If you do not specify the correct URL location of the SAS environment file during deployment, then you must manually specify the URL location in a file on every client.

For more information about the structure of the SAS environment file, see “Configuring the SAS Environment File” in the *SAS Intelligence Platform: Middle-Tier Administration Guide*, located at <http://support.sas.com/documentation/onlinedoc/intellplatform>.

Note: If you include the HTTP server in your deployment, SAS automatically uses the URL for the packaged HTTP server.

Chapter 3

Installing SAS Forecast Server

Preparing to Install and Configure a New Deployment	17
Default File Locations	17
How to Install and Configure SAS Forecast Server	18

Preparing to Install and Configure a New Deployment

To install and configure a new deployment of SAS Forecast Server, use the SAS Deployment Wizard. Many of the steps in the SAS Deployment Wizard (such as creating your SAS Home directory) apply to all the SAS applications at your site. The basic process for installing and configuring a SAS product is described in the [SAS Intelligence Platform: Installation and Configuration Guide](#). Some steps in the SAS Deployment Wizard are specific to installing and configuring SAS Forecast Server. These steps are addressed in this document.

When you deploy SAS Forecast Server, you deploy all of the components that are part of the SAS Forecast Server architecture. For more information, see “[Overview of the SAS Forecast Server Architecture](#)” on page 4.

Default File Locations

The following table shows the default locations of the directories and files that are installed with SAS Forecast Server. Use this table as a reference as you proceed through the installation and deployment process.

Table 3.1 Default Locations

Directory or File	Windows Path	UNIX Path
<i>SAS-installation-directory</i> *	C:\Program Files\SASHOME	<install-dir>/SASHome
This document uses <i>SAS-installation-directory</i> to represent the SAS installation directory.		

Directory or File	Windows Path	UNIX Path
<i>SASROOT</i> This document uses <i>SASROOT</i> to represent the SAS Foundation directory.	C:\Program Files\SASHOME\SASFoundation\9.4	<install-dir>/SASHome/SASFoundation/9.4
<i>SAS-configuration-directory</i> This document uses <i>SAS-configuration-directory</i> to represent the SAS configuration directory. <i>Note:</i> This is the default value. The path might vary for individual installations.	C:\SAS\Forecast\Lev<n>	.../Forecast/Lev<n>
<i>SAS-environment-directory</i> <i>Note:</i> This is the default value. The path might vary for individual installations.	C:\SAS\Forecast\Lev<n>\AppData\SASForecastServer14.2	.../Forecast/Lev<n>/AppData/SASForecastServer14.2
<i>SAS-project-directory</i> <i>Note:</i> This is the default value. The path might vary for individual installations.	C:\SAS\Forecast\Lev<n>\AppData\SASForecastServer14.2\Projects\project-name	.../Forecast/Lev<n>/AppData/SASForecastServer14.2/Projects/project-name

*The SAS Forecast Server files that are installed in *SAS-installation-directory* are Read-Only. The working files that you should use for customizations are installed in the *SAS-configuration-directory* directory.

How to Install and Configure SAS Forecast Server

Here is the information that you need to install SAS Forecast Server:

1. Start the SAS Deployment Wizard from your SAS Software Depot. For example, on a Windows system, double-click on the setup.exe file located in your SAS Software Depot folder.
2. In the Specify Deployment Plan step, select a deployment plan for Forecast Server.

It is recommended that you use one of the standard deployment plans created by SAS. If you need a custom plan, contact your on-site SAS support personnel for assistance.

The screenshot shows the 'SAS Deployment Wizard' window. The title bar is blue with the SAS logo and standard window controls. The main area is titled 'Specify Deployment Plan' with the instruction 'Specify a customized or standard deployment plan.' There are two radio buttons: 'Specify the full path to a customized deployment plan' (unselected) and 'Select a standard deployment plan' (selected). Below the first option is a text box and a 'Browse...' button. Below the second option is a dropdown menu showing 'Forecast Server, one machine' and a 'Search' button. A section titled 'Details of the selected deployment plan:' contains a list box with the following items: 'Server, Middle Tier, and Clients', 'Required Software' (with sub-item '1. Windows Powershell'), and 'Software to Install' (with sub-items '1. SAS Deployment Agent', '2. SAS Remote Deployment Agent Client', '3. SAS Providers for OLE DB', and '4. SAS Integration Technologies Client'). At the bottom are buttons for 'Help', '< Back', 'Next >', and 'Cancel'.

SAS Deployment Wizard

Specify Deployment Plan
Specify a customized or standard deployment plan.

☐ Specify the full path to a customized deployment plan

Browse...

☒ **Select a standard deployment plan**

Forecast Server, one machine Search

Details of the selected deployment plan:

Server, Middle Tier, and Clients

Required Software

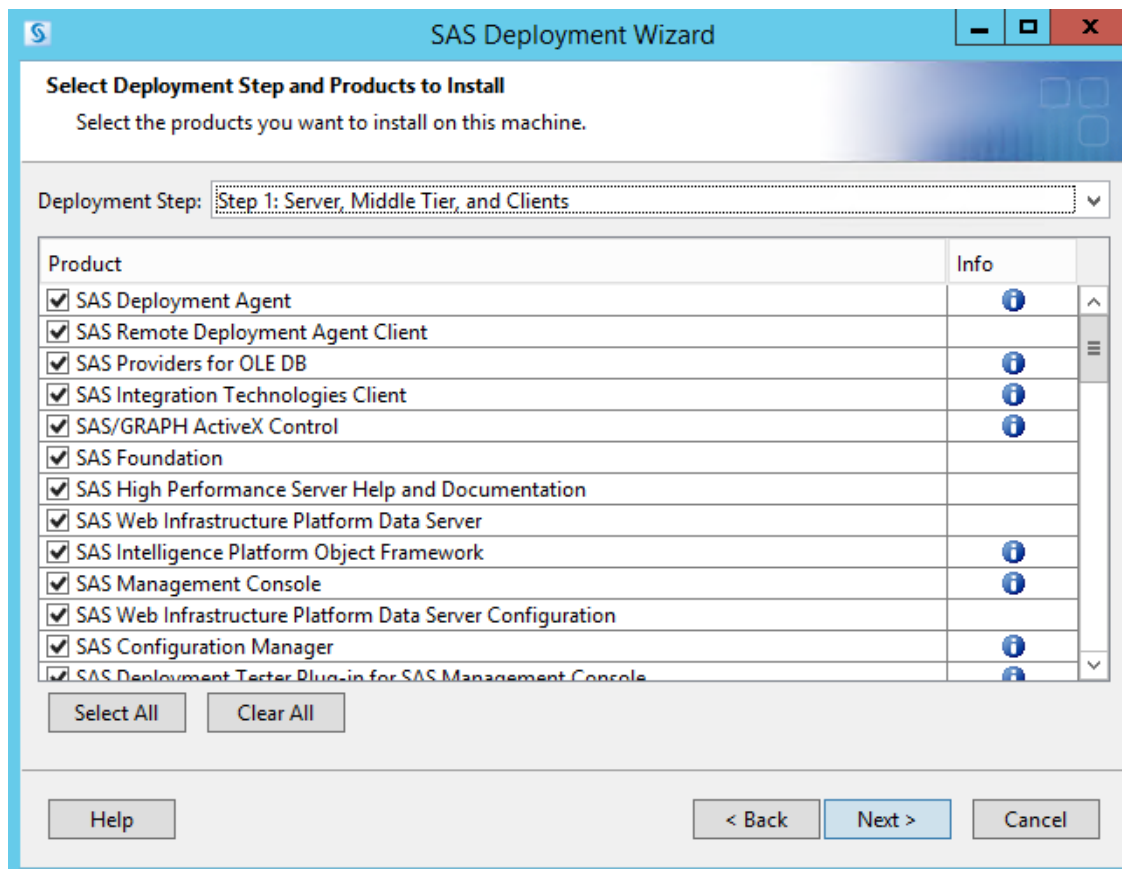
1. Windows Powershell

Software to Install

1. SAS Deployment Agent
2. SAS Remote Deployment Agent Client
3. SAS Providers for OLE DB
4. SAS Integration Technologies Client

Help < Back Next > Cancel

3. In the Select Deployment Step and Products to Install step, select **Step 1: Server, Middle Tier, and Clients (SAS Forecast Server)**.

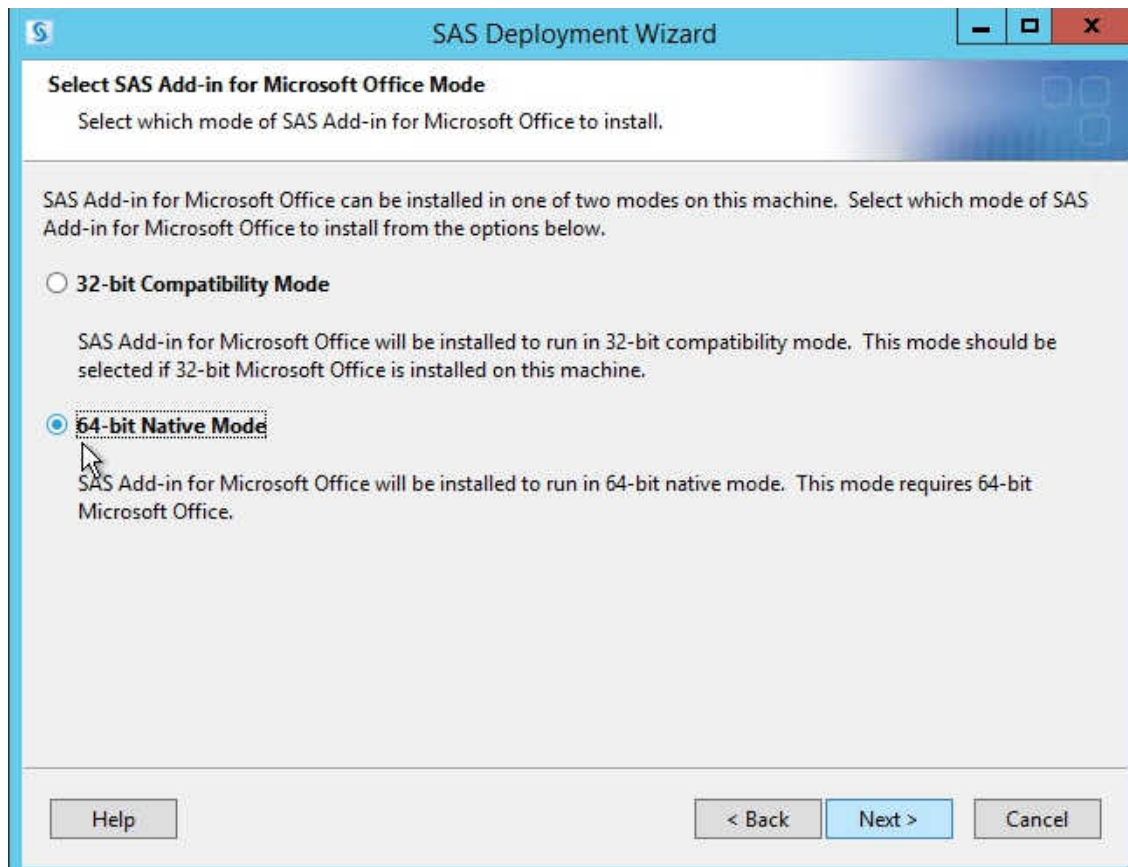


Note: Depending on what you license at your site, additional products might be installed. This list is specific to SAS Forecast Server. For a multiple machine deployment, these components could be across several machines.

Click **Next**.

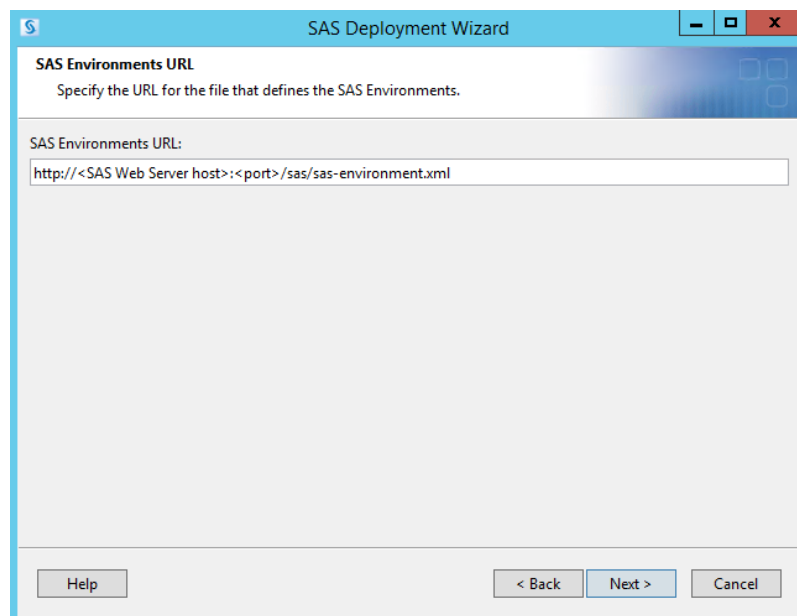
4. In the Select SAS Add-In for Microsoft Office Mode step, select whether you are running 32-bit or 64-bit Microsoft Office applications.

Note: This step is available only if your site also licenses the SAS Add-In for Microsoft Office.

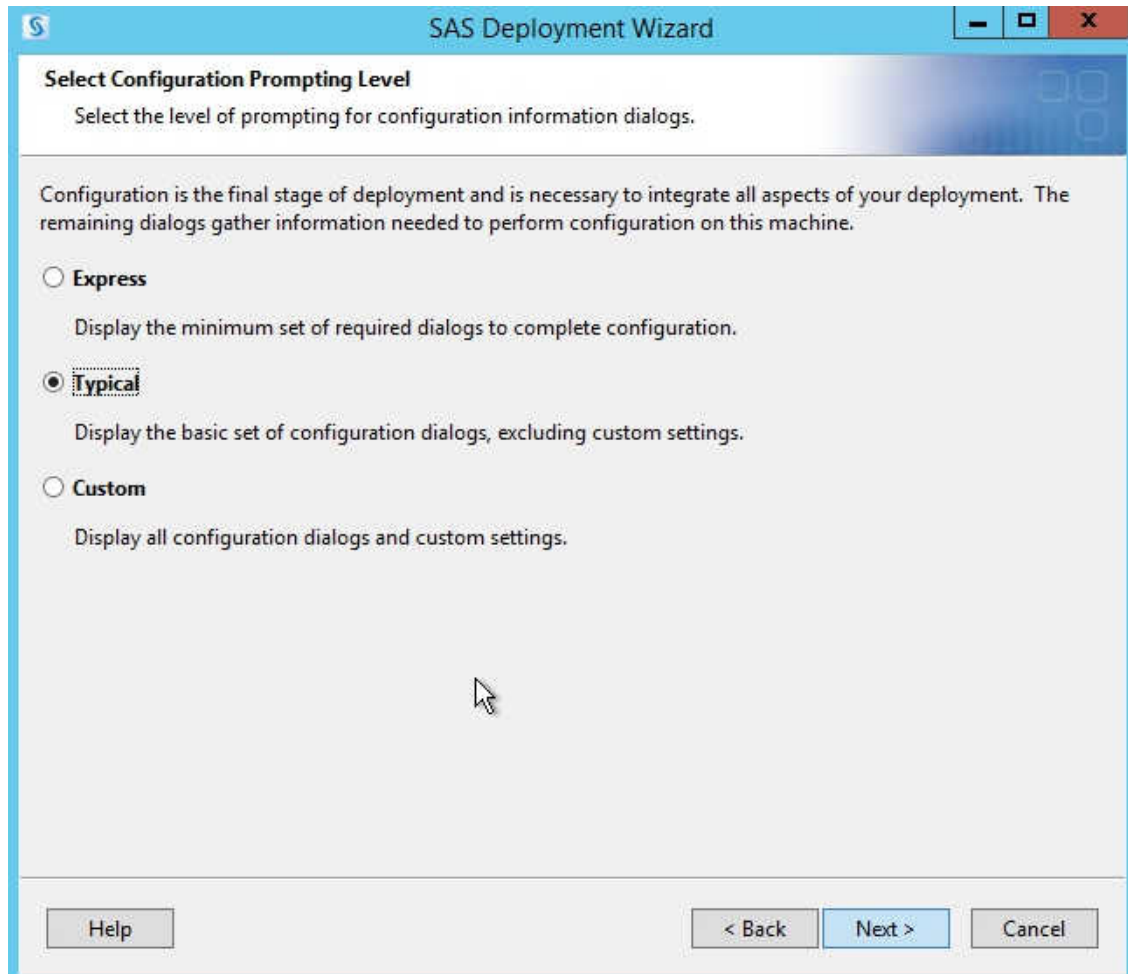


Click **Next**.

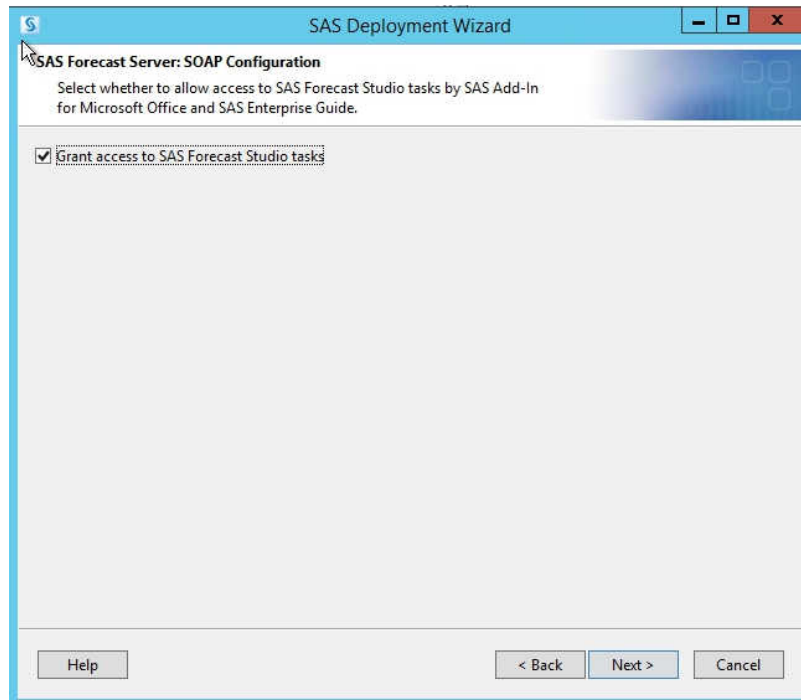
5. In the SAS Environments URL step, specify a URL location for the SAS environment file. For more information, see [“Determine the Location of the SAS Environment URL”](#) on page 16.



6. In the Select Configuration Prompting Level step, select **Typical**. Click **Next**.



7. To configure SAS Forecast Server:
 - a. In the SAS Forecast Server: SOAP Configuration step, select **Grant access to SAS Forecast Studio tasks** to allow access to these tasks by SAS Add-In for Microsoft Office and SAS Enterprise Guide. Click **Next**.

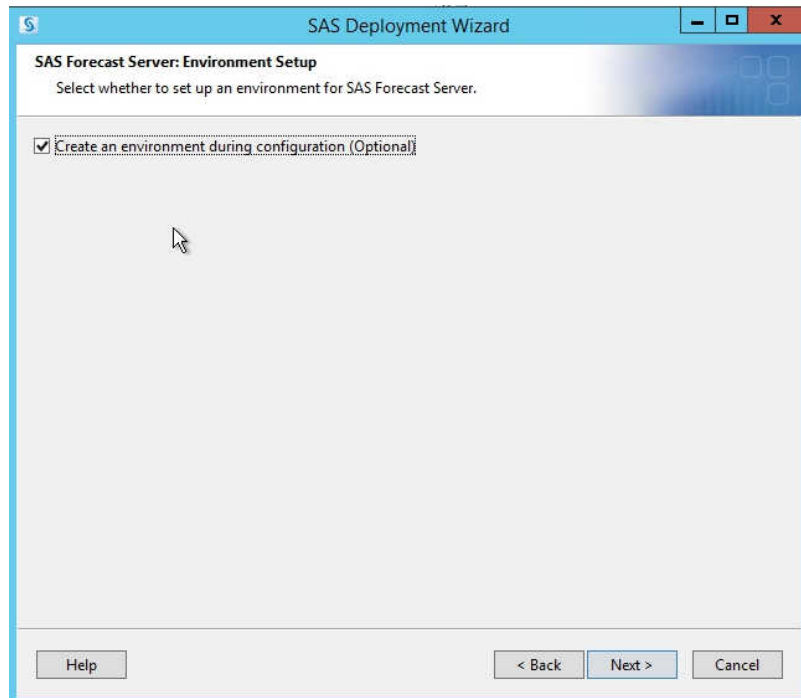


- b. In the SAS Forecast Server: Environment Setup step, you can select **Create an environment during configuration**. An environment is required as a container for SAS Forecast Server projects.

Often for UNIX environments, you have to manually configure this path on the SAS Forecast Server as a post-installation step. You can create a forecasting environment using the SAS Forecast Server Plug-ins for SAS Management Console or the SAS Forecast Project Manager. See the help for either application for more information about this step.

Click **Next**.

Note: The Environment Setup options are not available when performing a migration. The environments are created based on the migrated content.



- c. In the SAS Forecast Server: Environment Dependencies step, select a configured instance of the SAS Workspace Server that will host the environment. SAS Forecast Server depends on a configured instance of the SAS Workspace Server to successfully complete its configuration of the environment. Click **Next**.

Note: Do not use SASMeta as the workspace server in this step. For example, select SASApp instead of SASMeta.

Note: If there is a single workspace server configured in the system, then that server is assumed as the host for the environment, and this window does not appear.

- d. If you chose to create an environment, provide the following information in SAS Forecast Server: Environment Setup step:
- an environment name
 - a description
 - the root path of the SAS Forecast Server project directory (also referred to as the content location)

This field is populated with a default value that configures the environment's path under the SAS configuration directory. For example, on a Windows system, the default content location is *SAS-configuration-directory\AppData\SASForecastServer14.2*. If you intend to keep your projects in a different location, you should change this path to a location outside of the configuration directory. Specifying an outside location keeps the project data separate from the configuration data and will also make future migrations easier.

Note: The **Content Location** field refers to a path on the server tier. If your server tier and middle tier are on different hosts, and you are installing on the middle tier, then the default path that appears in the **Content Location** field still refers to a path on the server tier. This path must exist on the server tier before SAS Forecast Server is used. This information

further defines the meaning of the default path in the online Help that is associated with this page.

Click **Next**.

The screenshot shows the 'SAS Forecast Server: Environment Settings' window of the SAS Deployment Wizard. The window title is 'SAS Deployment Wizard'. The subtitle is 'SAS Forecast Server: Environment Settings'. Below the subtitle is the instruction: 'Specify the environment settings for SAS Forecast Server.' The window contains three text input fields: 'Forecast Environment Name' with the value 'Default', 'Environment Description' with the value 'Forecast Server Default Environment', and 'Content Location' with the value 'C:\SAS\Forecast\Lev1\AppData\SASForecastServer14.1'. At the bottom of the window are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

- e. In the SAS Internal Account: Forecast Server Metadata User step, specify the internal password that is used for the fsmeta internal user. This internal account is used by SAS Forecast Server to read and update metadata. Click **Next**.

The screenshot shows the 'SAS Internal Account: Forecast Server Metadata User' window of the SAS Deployment Wizard. The window title is 'SAS Deployment Wizard'. The subtitle is 'SAS Internal Account: Forecast Server Metadata User'. Below the subtitle is the instruction: 'Specify the metadata identity and the password for the Forecast Server Metadata User Identity. The Metadata User is a privileged service account used by the SAS Forecast Server to manage metadata objects.' The window contains a text input field for 'Display Name' with the value 'Forecast Server Metadata User', a text input field for 'Name' with the value 'fsmeta', a text input field for 'Internal User ID' with the value 'fsmeta@saspw', a password input field for 'Internal Password' with masked characters '*****', and a password input field for 'Confirm Internal Password' with masked characters '*****'. At the bottom of the window are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

8. In the Deployment Summary, review the list of products that you are about to install, and click **Start**.

9. In the Additional Resources step, review the additional resources and complete the manual configuration instructions in the Instructions.html file.

Click **Finish** to exit the SAS Deployment Wizard.

Chapter 4

Post-Installation Tasks for the Server

Minimum Requirements for Post-Installation	27
Overview of Post-Installation Tasks	27
Creating User Groups for UNIX Environments	28
Managing Roles and Capabilities	29
About Metadata Groups, Roles, and Capabilities	29
How to Configure the SAS Forecast Server Administrator and Users	34
Creating and Configuring a Product Environment	35
About Product Environments	35
Create a Product Environment during Deployment	35
Create a Product Environment in SAS Management Console	35
File System Permissions in LOCKDOWN Mode	36
Set File System Permissions on UNIX	36
Set File System Permissions on Windows	36
Configuring the SAS Forecast Studio Tasks	37
About the SAS Forecast Studio Tasks	37
Prerequisites for Using the SAS Forecast Studio Tasks	37
Grant Access to the SAS Forecast Studio Tasks	37

Minimum Requirements for Post-Installation

The number of post-installation and configuration tasks that you need to complete depends on your site. For example, your site might not use the Java Web Start client. If that is the case, you do not need to complete certain post-installation tasks.

Every site must complete the following steps to run SAS Forecast Server:

- create a product environment
- set the permissions for each product environment

Overview of Post-Installation Tasks

Perform the following tasks after installing SAS Forecast Server:

1. Review the `Instructions.html` file that is created by the SAS Deployment Wizard at the end of the installation process. Complete any tasks that are documented in the file.
2. Create user groups in UNIX environments. For more information, see “[Creating User Groups for UNIX Environments](#)” on page 28..
3. In SAS Management Console, create metadata user accounts for SAS Forecast Server administrators and SAS Forecast Server users. Assign users to groups and roles. For more information, see “[Managing Roles and Capabilities](#)” on page 29, and “[How to Configure the SAS Forecast Server Administrator and Users](#)” on page 34.

For more information about groups, see “Defining Groups” in the *SAS Intelligence Platform Installation and Configuration Guide* available at <http://support.sas.com/documentation/cdl/en/biig/63852/PDF/default/biig.pdf>.

4. If you did not create a product environment during installation using the SAS Deployment Wizard, then you must create a product environment. For more information, see “[Creating and Configuring a Product Environment](#)” on page 35.
5. (optional) Enable expanded support for libraries. For more information, see “[Enable Expanded Support for Libraries](#)” on page 49.

Creating User Groups for UNIX Environments

In SAS Forecast Server, you might have multiple forecasting environments. These multiple environments enable you to group forecasting projects by departments, teams, and so on. Typically, each department or team has its own UNIX user group.

To run SAS Forecast Server, you might need to set up multiple forecasting user groups. Here is an example of when you need to create multiple user groups:

- A separate business group (called `FS_LOB`) is created for each group (LOB or sub LOB) of SAS Forecast Studio users on UNIX.
- The project directory for each SAS Forecast Server environment has the `FS_LOB` group as the owner. The `FS_LOB` group has Read, Write, and Execute permissions to the project directory.
- Users can be members of multiple groups, but the `FS_LOB` group is the primary group for each user for the project directory.
- The `FS_LOB` group must include any user who might run the code from a SAS Forecast Studio project.

Here is how to set the group mask for multiple groups in a Linux environment:

```
# Added for group mask
ID=/usr/bin/id
GIDS='$ID -G'
FSGIDS="gid1 gid2 gid3"
for SGID in $GIDS; do
    for FSGID in $FSGIDS; do
        if [ $SGID -eq $FSGID ] ; then
            unmask 002
            break
        fi
    done
done
```

done
done

Managing Roles and Capabilities

About Metadata Groups, Roles, and Capabilities

Default Roles for SAS Forecast Server

SAS Forecast Server is shipped with default metadata groups and roles. When SAS Forecast Server is deployed, capabilities are already assigned to the default roles. Using SAS Management Console, you can add additional capabilities to these default roles. You can also create your own metadata groups and roles. Then you specify the capabilities for that role.

These roles are available for an out-of-the-box deployment of SAS Forecast Server:

Roles	Description
Forecast Server: Administrator	The capabilities that enable users to manage product content, such as projects, are assigned to this role.
Forecast Server: Forecaster	The capabilities that enable users to create projects and generate forecasts are assigned to this role.
Forecast Server: Analyst	The capabilities that enable the user to analyze the time series data and the results of the forecast are assigned to this role.
Forecast Server: Browser	The capabilities that enable the user to view the results of the forecasts and use reports are assigned to this role.

In addition to these SAS Forecast Server roles, you might also use the Job Execution: Job Submitter and Management Console: Advanced roles. These roles are available in SAS Management Console and are not specific to SAS Forecast Server. The Job Execution: Job Submitter role enables you to run stored processes from the Reports and Stored Processes dialog box. The Management Console: Advanced role enables you to access the product plug-ins (such as the SAS Forecast Server plug-in) in SAS Management Console.

Capabilities for SAS Forecast Server

The following table lists the capabilities for SAS Forecast Server. Use capabilities to restrict the ability of a user to request that SAS Forecast Server perform an action, such as generate forecasts or update a model specification.

Table 4.1 Client Access

Capability	Description
Forecast Studio	Enables the use of the SAS Forecast Studio client.
Management Clients	Enables the use of the SAS Forecast Server Plug-in for SAS Management Console and SAS Forecast Project Manager.
SOAP Service Bridge	Enables the use of the SAS Forecast Studio tasks that are available in SAS Enterprise Guide and the SAS Add-In for Microsoft Office.
Macro Bridge	Enables the use of the SAS Forecast Server macros (for example, %FSCOPY).

Table 4.2 General Features

Capability	Description
Administer Product	Enables you to administer the metadata for SAS Forecast Server.
Analyze Time Series	Enables the Series View in SAS Forecast Studio.
Analyze Models	Enables the Modeling View in SAS Forecast Studio.
Change Series Usage	Enables the user to specify whether a series is active. For example, this capability enables the Active Series check box in the Forecasting View and the Modeling View.
Import New Data	Enables a user to choose whether SAS Forecast Server uses the most up-to-date data when generating forecasts. For example, this capability enables the Use updated data if available check box in several dialog boxes in SAS Forecast Studio. The dialog boxes might be the Update Project Version dialog box or the Reforecast Project dialog box.

Table 4.3 Forecasts

Capability	Description
Modify Forecasts	Enables a user to forecast the project or series in SAS Forecast Studio.

Capability	Description
Reconcile Forecasts	Enables a user to specify whether SAS Forecast Server should try to reconcile the hierarchy. For example, this capability enables the Reconcile Hierarchy menu item, the Reconcile icon in the Forecasting View, and the Reconcile message. The Reconcile message is the message that appears in the workspace if there are override conflicts or unresolved nodes.
Override Forecasts	Enables a user to create, edit, or delete override values. For example, this capability enables the Overrides Calculator, the ability to lock overrides, and the ability to set scenario forecast values as Overrides in the Scenario Analysis View.

Table 4.4 Models

Capability	Description
Change Model Selection	Enables a user to change the baseline model for the series. For example, in the Modeling View, this capability enables the Set this model as forecast model and Reset to “automatic selection” links.
Modify Models	Enables a user to create, delete, copy, and edit models. This capability also enables the user to import models from a catalog and export models to a catalog.

Table 4.5 Events

Capability	Description
Change Event Usage	Enables the user to specify if an event is used in the model. For example, this capability enables the Usage in system-generated models drop-down list in the Events Properties dialog box.
Modify Events	Enables a user to create, delete, copy, and edit events. This capability also enables the user to import events from a catalog and export events to a catalog.

Table 4.6 *Environments*

Capability	Description
View Environment Details	Enables a user to view the properties of SAS Forecast Server environments.
Manage Environment Settings	Enables a user to edit the properties of SAS Forecast Server environments.
Manage Environments	Enables a user to create, delete, and rename environments.

Table 4.7 *Projects*

Capability	Description
Create Projects	Enables a user to create a new project. This capability enables the New Project Wizard.
Manage Projects	Enables a user to delete, copy, register, and unregister a project; import and unarchive a project; export and archive a project; delete project archives; and update the project version.
Manage Project Access	Enables a user to specify whether a project is shared and to change the ownership of a project. For example, this capability enables the Change Owner and Public access options in the SAS Forecast Server Plug-in for SAS Management Console.
Manage Project Settings	Enables a user to specify the hierarchy and variable settings and forecasting settings for a project. For example, this capability enables the Hierarchy and Variable Settings and Forecasting Settings dialog boxes in SAS Forecast Studio.
View Project Scripts	Enables a user to view the project code that was written to run when the project is opened and when the project is closed. For example, this capability enables the Start-up and Shutdown Code button in the New Project Wizard and in the Project Properties dialog box.
Modify Project Scripts	Enables a user to edit the project code that runs when the project is opened and when the project is closed. For example, this capability enables the user to edit the code in the Start-up and Shutdown Code dialog box.

Table 4.8 Notes

Capability	Description
View Notes	Enables the Notes panel at the bottom of the Forecasting View.
Modify Notes	Enables a user to edit the content in the Notes panel at the bottom of the Forecasting View. This capability also enables the user to create and remove notes.

Table 4.9 Reports

Capability	Description
Use Reports	Enables the Reports and Stored Processes menu item and the Reports and Stored Processes dialog box in SAS Forecast Studio.
Manage Reports	Enables a user to manage reports by using the SAS Forecast Server Plug-in for SAS Management Console. For example, this capability enables a user to deploy and remove sample reports.

Table 4.10 Scenarios

Capability	Description
View Scenarios	Enables the Scenario Analysis View in SAS Forecast Studio.
Modify Scenarios	Enables a user to create new scenarios and edit, save, or delete existing scenarios.

Table 4.11 Special Features

Capability	Description
Legacy Features	Enables legacy features. This option is provided only to aid users through transitional periods that result from design changes in the next product release. Support for these legacy features is limited. You should enable these features only at the direction of SAS Technical Support.

Capability	Description
Experimental Features	Enables experimental features that are still under development and might change (or be removed) in a future release. Also, no migration support is provided for these features. You should enable these features only at the direction of SAS Technical Support.
Debugging Features	Enables debugging features that are provided to assist customers and SAS Technical Support when problems arise at your site. You should enable these features only at the direction of SAS Technical Support.

How to Configure the SAS Forecast Server Administrator and Users

Note: On each SAS Workspace Server that contains product environments, you must have an operating system account for each user of SAS Forecast Server.

You can use the default roles that are shipped with SAS Forecast Server to quickly configure your SAS Forecast Server users. Each default role is assigned capabilities that allow access to various software features.

The defining quality of an administrator account is the Administer Product capability that grants the administrator account expanded privileges under the product security model. The Administer Product capability is granted by the Forecast Server: Administrator role.

To configure the metadata accounts:

1. Start SAS Management Console and connect as a SAS administrator (for example, sasadm@saspw).
2. Create a metadata account for the user in SAS Management Console and associate that account with the operating system account.
3. In the User Manager, right-click the user and then select **Properties**. The Properties dialog box appears.
 - a. Select the **Groups and Roles** tab.
 - b. Assign the user to a default role or a role that you created for your site.

For the SAS Forecast Server administrator, the Forecast Server: Administrator role is sufficient if the administrator account only needs to manage actions in SAS Forecast Server. However, this role does not grant access to all product features. To give the administrator access to all product features, you must select all of the following roles:

- Management Console: Advanced
- Job Execution: Job Submitter
- Forecast Server: Administrator

- Forecast Server: Analyst
- Forecast Server: Browser
- Forecast Server: Forecaster

TIP To view the capabilities for a role, right-click the role, and select **Properties** from the pop-up menu.

- Click  to move the item to the **Member of** list.
- Click **OK**.

Creating and Configuring a Product Environment

About Product Environments

Note: A product environment is different from a SAS environment. A SAS environment is used only when you log on. The product environment is used after you log on to the product and start a product session.

A product environment is a workspace for product sessions. These environments are created by the SAS Forecast Server administrator and are used only by SAS Forecast Server and its client applications. Product environments can be used to organize your projects and to control access to SAS Forecast Server projects.

If you have more than one product environment, do not use the same project directory for multiple environments. In general, a file system location should be used by a single environment. This restriction is necessary to ensure the integrity of the file system content and to properly coordinate client access. In addition, do not configure one environment to use a subdirectory within a different environment.

CAUTION:

You must create at least one product environment before you can create a project in SAS Forecast Studio. If you opted to create the default environment in the SAS Deployment Wizard, you do not need to create another environment in SAS Management Console. However, you must create a project directory for the environment and set the appropriate permissions for that project directory.

Create a Product Environment during Deployment

When you install SAS Forecast Server using the SAS Deployment Wizard, you have the option of creating a product environment. If you do not create a product environment during installation, then you must create the environment as a post-installation task. You can create this environment using SAS Management Console.

Create a Product Environment in SAS Management Console

To create a product environment, the SAS Workspace Server and the **Reports** folder require full security permissions for the Forecast Server Metadata User.

1. Open SAS Management Console and log on as the product administrator for SAS Forecast Server.
2. In the navigation tree, locate the **Application Management** node.

3. Right-click the **Forecast Server** node, and select **New Environment**. The Create Environment dialog box appears.
 4. Specify a name for the new environment.
 5. Select a workspace server (such as SASAppFS – Logical Workspace Server).
 6. In the **Location** field, specify the root path to the project directory. You might be prompted to enter credentials to connect to the SAS Workspace Server. If you are working in a UNIX environment, the user must have permission for creating directories.
- Note:* The project directory should be used only by this environment.
7. Specify the location of the **Reports** folder in the metadata. This path must already exist. If you specify a blank value, then the support for reports is disabled for that environment.
 8. Click **OK**.

File System Permissions in LOCKDOWN Mode

If the servers are in LOCKDOWN mode, you must ensure that the file paths in the product environment directory are whitelisted. For more information about LOCKDOWN mode, see the *SAS Intelligence Platform: Security Administration Guide* at [SAS Intelligence Platform: Security Administration Guide](#).

Set File System Permissions on UNIX

Users have different operating system privileges on the SAS Workspace Server. By defining a user group for SAS Forecast Server, you can assign all SAS Forecast users to the same group and grant the same permissions to all SAS Forecast Server users at one time. All SAS Forecast Server users must have Read, Write, and Execute permissions for each product environment directory that a user is permitted to use. Users also need permissions to all of the files and directories in the product environment directory. The operating system must be configured to grant these permissions as new files and directories are created. The exact details of how to do this depends on which operating system groups are defined and your site's security policies.

Set File System Permissions on Windows

In Windows operating environments, you must set file system permissions for all of the SAS Forecast Server users. To support the public settings on projects, users must have full control of and access to the environment directory and its subdirectories.

To set file system permissions on Windows for all users:

1. Open Windows Explorer and select the root directory for the SAS Forecast Server environment.
2. Right-click on this directory, and then select **Properties**.
3. Select the **Security** tab.
4. Select the users for the current machine.
5. Enable **Full Control** for the specified group of users.
6. Click **OK**.

TIP If you plan to use environments only for the convenience of organizing projects, you can create each environment directory under a common parent directory. Configure the common parent directory with the appropriate file system permissions. In this way, the child directories inherit the permissions from the parent directory, so you do not need to configure each environment.

Configuring the SAS Forecast Studio Tasks

About the SAS Forecast Studio Tasks

The Forecast Studio Create Project task, the Forecast Studio Open Project task, and the Forecast Studio Overrides Submit task enable you to create and work with SAS Forecast Studio projects in SAS Enterprise Guide and the SAS Add-In for Microsoft Office. The SAS Add-In for Microsoft Office extends the functionality of Microsoft Excel, Microsoft Word, and Microsoft PowerPoint by enabling you to access SAS analytics and SAS reporting functionality without any SAS programming experience. The SAS add-in is designed for people who are familiar with these Microsoft Office programs but who might be new to SAS.

Prerequisites for Using the SAS Forecast Studio Tasks

To use the SAS Forecast Studio tasks in the SAS add-in, you must complete these tasks:

- install SAS Forecast Server 14.2.
- install SAS Enterprise Guide or the SAS Add-In 6.1 for Microsoft Office on each client machine.
- configure SAS Forecast Server to use the SAS Add-In for Microsoft Office.

TIP To avoid having to reconfigure the product, make sure that you enable the use of the SOAP service bridge for SAS Forecast Server during deployment.

Grant Access to the SAS Forecast Studio Tasks

Note: When you initially configured SAS Forecast Server, you might have selected the **Grant access to SAS Forecast Studio tasks** in the SAS Forecast Server: SOAP Configuration step of the SAS Deployment Wizard. This option is required to enable the use of these tasks in SAS Enterprise Guide and the SAS Add-In for Microsoft Office. If you selected this option, then skip the following reconfiguration steps.

If you did not grant access to the SAS Forecast Studio tasks when you initially configured SAS Forecast Server, you must reconfigure SAS Forecast Server before you can use these tasks in SAS Enterprise Guide or the SAS Add-In for Microsoft Office. You do not need to unconfigure SAS Forecast Server first. You simply need to complete the following steps to reconfigure the product.

To reconfigure SAS Forecast Server to grant access to the SAS Forecast Studio tasks:

1. Stop the web application server. You will need to reconfigure the web application server after reconfiguring SAS Forecast Server.
2. To reconfigure SAS Forecast Server, open the SAS Deployment Wizard from your SAS Software Depot. For example, on a Windows system, double-click the setup.exe file located in the SAS Software Depot folder.

3. In the SAS Deployment Wizard, complete these steps:
 - a. In the Select Deployment Task step, select **Install SAS software**.
 - b. In the Select Deployment Type step, select **Perform a Planned Deployment**, and then select **Configure SAS Software**.
 - c. In the Select Configuration Prompting Level step, select **Typical** to display the basic set of configuration settings.
 - d. In the Select Products to Configure step, click **Clear All**, and then select **SAS Forecast Server Mid-Tier**. You must also reconfigure your web application server, so you must stop your web application server when reconfiguring the product.
 - e. In the SAS Forecast Server: SOAP Configuration step, select **Grant access to SAS Forecast Studio tasks**.
 - f. Navigate through the remaining windows in the SAS Deployment Wizard, and click **Finish**.

Chapter 5

Post-Installation Tasks for the Middle Tier

Overview of Post-Installation Tasks	39
Configuring Integrated Windows Authentication (IWA)	40
Changing Configuration Properties in SAS Management Console	40
Specifying a Time-Out	40
Specify a Time-Out for SAS Forecast Studio	40
Specify a Time-Out for the SOAP Bridge	41
Configuring and Using Java Web Start	41
Customizing the Default URL	41
How to Specify the Value of the SAS Environment URL	41
Set the Preferred Environment in the Log On Dialog Box for SAS Forecast Studio	42
Configuring the Archive Functionality	42
Enable Server Selection for Reports	43

Overview of Post-Installation Tasks

The following post-installation tasks for the middle tier are optional:

- Configure Integrated Windows Authentication (IWA). For more information, see [“Configuring Integrated Windows Authentication \(IWA\)” on page 40](#).
- Change one or more of the following configuration properties in SAS Management Console:
 - Specify a time-out value. For more information, see [“Specifying a Time-Out” on page 40](#).
 - Configure and use Java Web Start. For more information, see [“Configuring and Using Java Web Start” on page 41](#).
 - Set the preferred environment in the Log On Dialog Box for SAS Forecast Studio. For more information, see [“Set the Preferred Environment in the Log On Dialog Box for SAS Forecast Studio” on page 42](#).
 - Configure the archive functionality. For more information, see [“Configuring the Archive Functionality” on page 42](#).

- Enable server selection for reports. For more information, see “[Enable Server Selection for Reports](#)” on page 43.

Configuring Integrated Windows Authentication (IWA)

SAS Forecast Server now supports Integrated Windows Authentication (IWA). For information about how to configure IWA, see *SAS Intelligence Platform: Middle-Tier Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

For a detailed description of this functionality, see the following documentation:

- *SAS Intelligence Platform: Security Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>
- *SAS Intelligence Platform: Middle-Tier Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>

Changing Configuration Properties in SAS Management Console

Several configuration properties exist for the SAS Forecast Server middle tier. You can set these properties by using the Configuration Manager in SAS Management Console. The values of these configuration properties are cached at run time, so if you change the value of a configuration property, you must restart SAS Forecast Server. For more information about the Configuration Manager, see the SAS Management Console Help.

Note: If your site allows JMX access, then you can use the JMX calls to indicate when the application should reload the configuration properties. These signals eliminate the need for restarting SAS Forecast Server for new configuration properties to take effect. For more information, see “[Additional Resources](#)” on page 217.

Specifying a Time-Out

Specify a Time-Out for SAS Forecast Studio

By default, a SAS Forecast Studio session times out after being idle for 60 minutes. The SAS Forecast Server checks every 5 minutes to see whether the session has timed out. You can change the value of the time-out and how frequently SAS Forecast Server checks a session.

The time-out is the maximum amount of time that is allowed to elapse before SAS Forecast Studio assumes that the client session has failed. At this time, session resources are reclaimed, and any open resources (such as projects) might be corrupted. When specifying a time-out, be sure to exceed the maximum time that any client would be idle under normal operations.

To specify a new time-out value for an idle SAS Forecast Studio session:

1. Start SAS Management Console and connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the selection pane, click **SAS Forecast Server**.
6. Specify new values for the **Idle time-out limit** and **Time-out check interval** options.
Click **OK**.
7. Restart SAS Forecast Server for these changes to take effect.

Specify a Time-Out for the SOAP Bridge

The SOAP bridge is used to run the SAS Forecast Studio tasks in SAS Enterprise Guide and the SAS Add-In for Microsoft Office. By default, this connection times out after being idle for 60 minutes. The SAS Forecast Server checks every 60 seconds to see whether the service has timed out. You can override the time-out value by specifying the `com.sas.analytics.forecasting.soap.timeout` Java system property. If you set this property to a value less than or equal to 0, then the time-out is disabled. You must then restart the SOAP service to delete any old sessions.

Configuring and Using Java Web Start

Customizing the Default URL

By default, you can launch SAS Forecast Studio from `http://your-server-name:your-http-port/SASForecastServer/main.jnlp`.

To open a particular project in SAS Forecast Studio, add these parameters to your URL:

- `forecasting.launch.environment` — specifies the product environment.

TIP You are not required to specify the project parameter. You might choose to specify on the environment parameter.

- `forecasting.launch.project` — specifies the name of the project.

For example, if your URL is `http://localhost:8080/SASForecastServer/main.jnlp?forecasting.launch.environment=Default&forecasting.launch.project=Project1`, then SAS Forecast Studio opens and displays Project 1 in the default environment.

How to Specify the Value of the SAS Environment URL

To specify the link to the Java Web Start client as the SAS environment URL:

1. Start SAS Management Console and connect as a SAS administrator (for example, sasadm@saspw).

2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the selection pane, select **Forecast Studio (Java Web Start)**.
6. In the **SAS environment URL** box, specify the URL for the sas-environment.xml file.
Click **OK**.
7. Restart SAS Forecast Server for these changes to take effect.

Set the Preferred Environment in the Log On Dialog Box for SAS Forecast Studio

When users log on to SAS Forecast Studio, they must specify a SAS environment to use. You can restrict the environments that users have access to and specify a preferred environment that is selected by default.

Note: If the preferred SAS environment does not exist at run time, SAS Forecast Studio acts as if you did not specify a preferred environment.

To specify the list of SAS environments that should appear in the logon dialog box:

1. Start SAS Management Console and connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the selection pane, select **Forecast Studio (Java Web Start)**.
6. In the **Preferred SAS environment** box, specify the name of the SAS environment that should be selected in the logon dialog box.

Note: Specify the value of the NAME= attribute in the sas-environment.xml file. This name is case sensitive.

7. If the user should not be able to change the environment in the logon dialog box, set the **Lock SAS environment selection** property to **true**.
8. Click **OK**.
9. Restart SAS Forecast Server for these changes to take effect.

Configuring the Archive Functionality

Note: With the exception of the compression level, do not modify the archiving properties unless directed to by SAS Technical Support.

The archiving properties in SAS Forecast Server enable you to control the process for creating and extracting project archives.

To configure the archiving properties:

1. Start SAS Management Console and connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the selection pane, select **Forecast Server**.
6. Set these archiving properties:

Property	Description	Default Value
Compression level	Specifies how much to compress a new archive.	Default of the system level
Files to ignore (by extension)	Specifies the file types that you do not want to include in an archive.	sas7bndx
Data set files to CPORT (by extension)	Specifies data files that you want to include in the archive. A comma-separated list of values is expected.	sas7bdat
Catalog files to CPORT (by extension)	Specifies the catalog files that you want to include in the archive. A comma-separated list of values is expected.	sas7bcat
Additional CPORT option	Enables you to include additional options in the PROC CPORT statement that is used to create an archive.	(blank)

Click **OK**.

7. Restart SAS Forecast Server for these changes to take effect.

Enable Server Selection for Reports

Report executions are handled by the logical SAS Workspace Server by default. This involves creating workspace sessions, which causes a delay in report executions. As an alternative, you can use another execution server such as a SAS Stored Process Server or SAS Pooled Workspace Server. When another execution server is used, be aware that the

report execution process runs under the server identity as opposed to the current user's identity (as when using the logical SAS Workspace Server). Therefore, you must make sure that the server identity has access to all file content areas that are accessed, such as the environment content areas.

The ability to use another execution server to execute reports is disabled by default. To enable the ability, in addition to configuring user accounts and file permissions appropriately, you must set a configuration property as follows:

1. Start SAS Management Console and connect as a SAS administrator (for example, `sasadm@saspw`).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the selection pane, click **SAS Forecast Server**.
6. Specify *Enabled* for **Execute server selection for reports**.

Click **OK**.

7. Restart SAS Forecast Server for these changes to take effect. You must also restart services such as the web application server, SAS Metadata Server, and the SAS Object Spawner.

Part 3

Configuration Tasks

<i>Chapter 6</i>	
<i>Creating and Configuring Libraries</i>	<i>47</i>
<i>Chapter 7</i>	
<i>Specifying Security Permissions for Users and Groups</i>	<i>53</i>
<i>Chapter 8</i>	
<i>Administering Reports to Use in SAS Forecast Studio</i>	<i>61</i>
<i>Chapter 9</i>	
<i>Miscellaneous Administration Tasks</i>	<i>85</i>

Chapter 6

Creating and Configuring Libraries

Understanding Libraries	47
Overview of Libraries	47
External Library Assignments	48
Libraries Assigned by SAS Forecast Server	48
Authorization Checks	49
Enable Expanded Support for Libraries	49
How to Define a Pre-assigned Library in the Metadata	50
How to Define a Metadata Library That Is Assigned by SAS Forecast Server ...	51
Create an Automatic Environment Library	51
Create a Configured Environment Library	52

Understanding Libraries

Overview of Libraries

SAS Forecast Server uses SAS libraries and data sets to manage and access project data. For more information about libraries, see the “SAS Libraries” chapter in *SAS Language Reference: Concepts* and the “LIBNAME Statement” topic in *SAS DATA Step Statements: Reference*.

Note: Library names cannot begin with an underscore character. This naming convention is reserved for internal SAS libraries.

In SAS Forecast Server, libraries are either assigned by the SAS Forecast Server application or external to the SAS Forecast Server application. Library assignments that are made by the system or a user are examples of external library assignments.

Libraries can be created using either the SAS Forecast Project Manager or SAS Management Console. For information about how to create libraries in the SAS Forecast Project Manager, see the Help that is available in SAS Forecast Project Manager. The topics in this section discuss how to create libraries in SAS Management Console.

External Library Assignments

Libraries can be assigned when a server session is created or in the start-up code for a specific product environment. Because these library assignments are made outside of SAS Forecast Server, they are considered external library assignments.

Here are four main sources of external library assignments:

- All SAS libraries that are shipped with SAS are automatically assigned to each SAS session. Examples of Base SAS libraries are Sashelp, Sasuser, and Work.
- Additional libraries can be assigned by adding LIBNAME statements to your SAS or server configuration files. The scope of the configuration file (in other words, is it a SAS or server configuration file) determines the availability of the library.
- Libraries can also be assigned by including LIBNAME statements in the start-up code for an environment. These libraries are available only for the associated environment. If you assign a library in your environment's start-up code, you must include a corresponding LIBNAME CLEAR statement in the shutdown code.
- Pre-assigned libraries that are defined in the metadata and associated with a given server are automatically assigned when the server session is created.

Libraries Assigned by SAS Forecast Server

SAS Forecast Server enables you to manage and assign libraries on-demand. Even if these libraries are not currently assigned, they appear in SAS Forecast Studio like any other library. When SAS Forecast Server detects that a library is going to be accessed (for example, when the library is referenced in the code), SAS Forecast Server assigns the library before allowing that access to occur.

Due to security considerations, these libraries are not available in SAS Forecast Studio by default. SAS Forecast Server includes a configuration setting to enable the use of each type of on-demand library: manually assigned metadata libraries, configured environment libraries, and automatic environment libraries. For more information, see [“Enable Expanded Support for Libraries” on page 49](#).

Here are the three types of libraries that are assigned by SAS Forecast Server:

- Libraries that are defined in the metadata but are not pre-assigned. When SAS Forecast Server detects that one of these libraries needs to be used, the library is assigned using the META engine with the METAOUT=DATA option.
Note: The META engine requires that each data set (also referred to as a table) is registered in the metadata. If the data set is not registered, then the data set does not appear in SAS Forecast Studio. Data sets that exist only as files on the file system cannot be accessed from SAS Forecast Studio when these data sets are located in a metadata defined library that is not pre-assigned.
- Configured environment libraries are defined in the **SAS-environment-directory\Config\libs** data set. Each row of this data set represents a library assignment and contains the information required for the BASE engine:
 - the LIBNAME
 - the file system path

Note: Specify the absolute path to the library directory to prevent SAS Forecast Server from defaulting to the working directory.

- whether access to the library should be read-only

When SAS Forecast Server detects that a configured environment library needs to be used, the library is assigned using the BASE engine.

- Automatic environment libraries are implicitly defined by creating a subdirectory for each library in the ***SAS-environment-directory\Libraries*** directory. The name of this subdirectory implies the LIBNAME for the library assignment and the access settings for the library.

When SAS Forecast Server detects that an automatic environment library needs to be used, the library is assigned using the BASE engine.

The configured and automatic environment libraries enable you perform these tasks:

- manage libraries for users who have access to the file system
- define local libraries for individual environments

Authorization Checks

One key detail of library management and use is the ability to perform authorization checks on the data accesses that are being attempted by the user. How these authorization checks are performed depends on the type of library that the user is trying to access.

Two strategies are used.

- for the library to be available in SAS Forecast Studio, the SAS Intelligence Platform and the SAS Metadata Server require that a user have ReadMetadata permission on a library object. For libraries that are defined in the metadata, SAS Forecast Server requires Read metadata permission for the user to have Read access. Write metadata permission is required for the user to have Write access. Permissions can also be set for elements within the library. An element might be a data set (also referred to as a table) or a data set variable (also referred to as a column). For example, if SAS Forecast Studio tries to read a variable in a data set, then SAS Forecast Server searches for an associated Column, Table, or Library object in that order. The authorization check is performed against the first object that is found.

TIP You can create table objects for any type of metadata library by using the SAS Forecast Server plug-in for SAS Management Console. For more information, see the Help for the SAS Forecast Server plug-in.

- All other libraries are assigned using the BASE engine (for example, the configured and automatic environment libraries) or are assigned in a way that cannot be determined by SAS Forecast Studio.

Enable Expanded Support for Libraries

By default, SAS Forecast Server restricts the use of metadata libraries that are manually assigned, automatic environment libraries, and configured environment libraries. You must configure SAS Forecast Server so that users can access these libraries.

To enable this library support:

1. Open SAS Management Console as the SAS Administrator (for example, sasadm), and then connect to a metadata repository.

2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
 3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
 4. Click the **Settings** tab.
 5. In the selection pane, click **Forecast Server**.
 6. From the **Manual-assign metadata libraries**, **Automatic environment libraries**, and **Configured environment libraries** drop-down lists, specify whether SAS Forecast Server allows access to each type of library.
- Click **OK**.

How to Define a Pre-assigned Library in the Metadata

To create a metadata library that is pre-assigned:

1. Create a metadata definition for the new library.
 - a. Open SAS Management Console as the SAS Administrator (for example, sasadm), and then connect to a metadata repository.
 - b. Expand the **Data Library Manager** node, and then select **Libraries**.
 - c. Right-click the library that you want to pre-assign, and then select **Properties**.
 - d. Select the **Options** tab.
 - e. Click **Advanced Options**.
 - f. Select the **Library is Pre-assigned** check box. The selected library is assigned whenever a session is created for one of the assigned servers.
 - g. Ensure that the library is assigned to the correct SAS servers.
 - h. Click **OK**.
2. Set the metadata permissions on the new library object for your SAS Forecast Server product administrator and users.
 - a. In SAS Management Console, expand the **Data Library Manager** node, and select **Libraries**.
 - b. Right-click the library name, and then select **Properties**.
 - c. Select the **Authorization** tab.
 - d. In the **Users and Groups** area, select the product administrator and grant permissions by selecting the **Grant** boxes. Repeat this step for the SAS Forecast Server users.

Verify that the fsmeta account has ReadMetadata permission on the library. If the fsmeta account does not have this permission, SAS Forecast Server might not detect the library or might classify the library incorrectly.
3. If the SAS Object Spawner is running, stop and restart the SAS Object Spawner.

How to Define a Metadata Library That Is Assigned by SAS Forecast Server

To create a metadata library that is not pre-assigned:

1. Verify that SAS Forecast Server is configured to support metadata libraries that are manually assigned. For more information, see [“Enable Expanded Support for Libraries” on page 49](#).
2. Create a metadata definition for the new library:
 - a. Open SAS Management Console as the SAS Administrator (for example, sasadm@saspw), and then connect to a metadata repository.
 - b. Expand the **Data Library Manager** node, and then select **Libraries**.
 - c. Right-click the library that you want to pre-assign, and then select **Properties**.
 - d. Select the **Options** tab.
 - e. Click **Advanced Options**. The Advanced Options dialog box appears.
 - f. Ensure that the library is assigned to the correct SAS servers.
 - g. Click **OK**.
3. Set the metadata permissions on the new library object for the SAS Forecast Server product administrator and users.
 - a. In SAS Management Console, expand the **Data Library Manager** node, and select the **Libraries** node.
 - b. Right-click the library name, and then select **Properties**.
 - c. Select the **Authorization** tab.
 - d. In the **Users and Groups** area, select the product administrator and grant permissions by selecting the **Grant** boxes. Repeat this step for the SAS Forecast Server users.

Verify that the fsmeta account has ReadMetadata permission on the library. If the fsmeta account does not have this permission, SAS Forecast Server might not detect the library or might classify the library incorrectly.
4. Register the data sets that you want to include in this library.

Create an Automatic Environment Library

To create an automatic environment library:

1. Verify that SAS Forecast Server is configured to support automatic environment libraries. For more information, see [“Enable Expanded Support for Libraries” on page 49](#).
2. In the **SAS-environment-directory\Libraries** directory, create a new directory for the automatic environment library. The library name should be the same

as the desired libref. For example, if the library name is Hpfuser, then the name of the directory should be **hpfuser**.

Note: If the **Libraries** directory does not exist, then you need to create it.

3. Specify the security access for the library. By default, the libraries are assigned read-write access. If the library should have Read-Only access, append **.r** to the directory name. For example, the **hpfuser.r** directory creates the Hpfuser library with Read-Only access.

Now, when a user opens this product environment in SAS Forecast Studio, the new library should be available.

Create a Configured Environment Library

To create a configured environment library:

1. Verify that SAS Forecast Server is configured to support configured environment libraries. For more information, see [“Enable Expanded Support for Libraries” on page 49](#).
2. In the **SAS-environment-directory\Config** directory, create a **libs.sas7bdat** file. This data set should contain three character variables: **LIBNAME**, **Path**, and **Read-only**.

Note: If the Libs data set does not exist, SAS Forecast Server tries to create it when a user opens this product environment in SAS Forecast Studio.

3. Edit the Libs data set to include the information for the new library. For example, to define the Hpfuser library, you might use these values:
 - For the **LIBNAME** variable, specify **hpfuser**.
 - For the **Path** variable, specify **C:\mylibs\hpfuser**.
 - For the **Read-only** variable, specify **yes**.

Now, when a user opens this environment, the new library should be available.

Chapter 7

Specifying Security Permissions for Users and Groups

Enabling Users to Access SAS Forecast Server	53
Initial Users	53
Understanding the SAS Forecast Server Metadata User	54
Project Owners	55
Securing Access to SAS Forecast Server	56
Security Layers	56
Notes Regarding Permissions	56
Environment Permissions	57
Project Permissions	58
Sharing Options for Projects	59

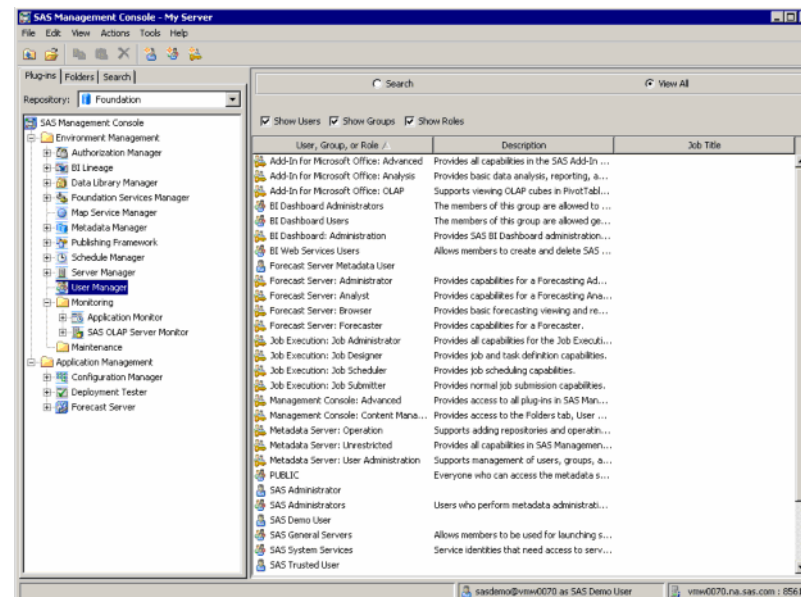
Enabling Users to Access SAS Forecast Server

Initial Users

When you install and configure the SAS Intelligence Platform and SAS Forecast Server, standard user and group definitions are added. The following display shows an example

of how these users and groups might appear in the User Manager plug-in of SAS Management Console.

Figure 7.1 Example of Standard Users and Groups Created for SAS Forecast Server



Understanding the SAS Forecast Server Metadata User

What Is the SAS Forecast Server Metadata User?

To access data that is not accessible to the current user and to update the metadata, SAS Forecast Server uses the SAS Forecast Server Metadata User (fsmeta@saspw). This account must have access to all metadata that is used by the product. For example, the WriteMetadata permission is required for any object that the product might need to modify. The ReadMetadata permission is required for any object that the product might need to use or detect.

During deployment, the SAS Forecast Server Metadata User is granted permissions to the **/System/Applications/SAS Forecast Server** metadata folder. This folder contains the metadata objects that represent the product content.

If you create additional metadata folders (folders that are outside of the **/System/Applications/SAS Forecast Server** directory) that you want to use with SAS Forecast Server, you must grant the Forecast Server Metadata User the appropriate permissions on these folders. Additional metadata folders are created to manage the report metadata objects. Special metadata objects that are used by the product (such as objects that define libraries and servers) must also be accessible. Finally, the permission settings must be sufficient to ensure that the ReadMetadata permission is granted on all other objects that are created during the deployment.

Note: A product environment cannot be associated with a server without having WriteMetadata permission on the defining server object. This security requirement is imposed by the SAS Metadata Server.

Specify User Permissions for the Forecast Server Metadata User

To grant permissions to a folder outside of the `/System/Applications/SAS Forecast Server` directory:

1. Open SAS Management Console, and log on as the SAS Administrator (for example, `sasadm@saspw`).
2. Click the **Folders** tab.
3. Locate the metadata folder that you created. For example, you might create a `/Shared Data/Forecast Server/Reports` folder to organize the reports that you use in the product.
4. Right-click on the metadata folder, and then select **Properties**. The Properties dialog box appears.
5. Click the **Authorization** tab.
6. In the **Users and Groups** area, click **Add**. The Add Users and Groups dialog box appears.
7. In the **Available Identities** list, double-click the **Forecast Server Metadata User**. The Forecast Server Metadata User is added to the **Selected Identities** list. Click **OK**.
8. In the **Users and Groups** area, select **Forecast Server Metadata User**.
9. Grant the **ReadMetadata**, **WriteMetadata**, **WriteMemberMetadata**, and **CheckinMetadata** permissions for the Forecast Server Metadata User, and then click **OK**.

TIP When appropriate, grant the **ReadMetadata**, **WriteMetadata**, **WriteMemberMetadata**, and **CheckinMetadata** permissions on the parent folder. Any child folders automatically inherit these permissions from the parent folder.

Project Owners

What Is a Project Owner?

The user who creates a project in SAS Forecast Studio is the de facto administrator for that project. Such users are referred to as project owners. In addition to the normal project capabilities, project owners can use management actions, such as the ability to change project-sharing settings and the ability to delete the project. You can transfer project ownership to other users by using either the SAS Forecast Server Plug-in for SAS Management Console or the `%FSSETOWN` macro. A project can have only one owner.

The value for the project owner is detected by comparing a generated identity token with a previously stored token value for the project. To determine the value for the project owner, either look at the SAS Forecast Studio Status page or the **Show only my token-value projects** check box in the Projects dialog box in SAS Forecast Studio.

How to Change Project Ownership

To change project ownership using the SAS Forecast Server Plug-in for SAS Management Console:

1. Open SAS Management Console. To log on, your user account must be assigned to the Forecast Server: Administrator group. You must also have a valid operating system account.

Note: Do not log on as sasadm@saspw because this account should not be configured as a product user.

2. Select the **Forecast Server** node on the **Plug-ins** tab.
3. Expand the environment node that contains the project. Then expand the **Projects** node.
4. Right-click on the desired project, and then select **Properties**. The Project Properties dialog box appears.
5. Enter the new user name in the **Owner** text box, and then click **OK**.

Note: You can also use the Access Settings batch operation to set the same owner on multiple projects. This feature is available in the SAS Forecast Server Plug-in for SAS Management Console. Right-click on the environment node, and select **Batch Operations** ⇒ **Access Settings**. For more information, see the Help for the SAS Forecast Server Plug-in for SAS Management Console.

Securing Access to SAS Forecast Server

Security Layers

Security settings in SAS Forecast Server are implemented in four layers:

- Capabilities – enforced by the product
- Metadata Permissions
 - ReadMetadata, WriteMetadata, WriteMemberMetadata, CheckinMetadata – enforced by the SAS Metadata Server
 - other permissions (such as library authorization) – enforced by the product
- Project Ownership and Sharing (also called the application security model) – enforced by the product
- File System Permissions – enforced by the operating system

Typically, permissions are checked in the order listed. For example, capabilities are checked first while file system permissions are checked last.

Notes Regarding Permissions

File System Permissions

Use the metadata and application layers when you define your security strategy. Typically, file system permissions are not checked before an operation is attempted in the product.

Note: Securing content only through the use of file system permissions is not recommended.

WriteMetadata Permissions

In SAS Forecast Server, the metadata security that is applied to the objects with product content is limited to controlling the visibility of this content by using the ReadMetadata permission. Metadata updates to these objects are performed indirectly by the SAS

Forecast Server Metadata User. The WriteMetadata settings for individual user accounts have no impact on the product behavior, so the WriteMetadata permission should be used on objects that contain product content.

What Permissions Can You Control By Using Metadata?

To secure access to metadata objects that represent the SAS Forecast Server data, you can grant or deny permissions to individuals or groups by using the **Authorization** tab in SAS Management Console.

The following table lists some of the metadata objects and their permissions.

Metadata Object	Permissions
environment	ReadMetadata – controls visibility
projects	ReadMetadata – controls visibility
libraries	ReadMetadata – controls visibility Read – controls ability to read contents Write – controls ability to write contents
tables that are registered to libraries	Read – controls ability to read contents Write – controls ability to write contents
columns in registered tables	Read – controls ability to read contents Write – controls ability to write contents
reports (stored process)	ReadMetadata – controls visibility

Note: Other permissions, such as the ability to use ReadMetadata to hide server objects, can impact product behavior. However, these permissions are usually not needed under typical operating conditions.

Environment Permissions

What Are Environments?

An environment enables you to group projects together into workspaces. It is defined by the following basic attributes:

- logical name
- host workspace server
- content path on the server

An environment defines each of the following:

- a container for projects
- a workspace for users
- a partition of a workspace server

An environment is implemented as a metadata folder, metadata object, and directory structure (in the server's file system). Only metadata and file system permissions apply

to these structures. To enable project sharing for an environment, you must configure the directory tree that is associated with the environment to grant full control to all users that are allowed to access the environment. Note that this configuration relies on the metadata settings to secure the environment.

In addition, setting the ReadMetadata permission at the environment level controls whether a user can access the projects and reports associated with that environment. When a user logs on to SAS Forecast Studio, environments for which a user does not have ReadMetadata permission do not appear as options in the Log On dialog box. Therefore, the user cannot select this environment and use its content in any SAS Forecast Server client.

Note: Each file system location in a workspace server should be used by only one environment system-wide, regardless of the number of environments or middle tiers in the system. In general, the file system location that is specified for the environment and its subdirectories should be reserved for that environment only. This restriction ensures the integrity of the file system content and the ability to properly coordinate accesses by clients.

You can create environments by using the %FSNEWENV macro or by using the SAS Forecast Server Plug-in for SAS Management Console. For more information about using the plug-in to create an environment, see the Help for the SAS Forecast Server Plug-in for SAS Management Console.

Configure Environment Permissions

To configure permissions for an environment:

1. In the file system, grant full control on the environment directory to all environment users or user groups. For easier management, a user group is recommended. For more information about user groups, see [“Creating User Groups for UNIX Environments” on page 28](#).
2. Open SAS Management Console, and log on as the SAS Administrator (for example, sasadm@saspw).
3. Click the **Folders** tab.
4. Expand the **System, Applications, SAS Forecast Server, SAS Forecast Server 14.2, and Environments** folders.
5. Right-click on the environment folder and select **Properties**. The environment Properties dialog box appears.
6. Click the **Authorization** tab.
7. Select your user (or a group of users) in the **Users and Groups** area.
8. Set the **ReadMetadata** permission for your user or group and then click **OK**.

TIP You can also set the permissions for an environment directly on the environment object in the environment folder.

Project Permissions

About Project Permissions

Projects are similar to environments in physical structure. In projects, the file is being saved to an environment subdirectory. By default, the security for the file system is specified when you create an environment. No additional security is required when

projects are created. However, you can set the ReadMetadata permission on each project object to control that project's visibility to users.

Configure Project Permissions

To configure the permissions for a project metadata object:

1. Open SAS Management Console, and log on as the SAS Administrator (for example, sasadm).
2. Click the **Folders** tab.
3. Expand the **System**, **Applications**, **SAS Forecast Server**, **SAS Forecast Server 14.2**, and **Environments** folders.
4. Select the environment folder.
5. Right-click on a project, and then select **Properties**. The *project* Properties dialog box appears.
6. Click the **Authorization** tab.
7. Select your user (or a group of users) in the **Users and Groups** area.
8. Set the **ReadMetadata** permission for your user or group, and then click **OK**.

Sharing Options for Projects

About Sharing Projects

By default, only the project owner or a SAS Forecast Server administrator can open a given project. By enabling project sharing, any user who can see a project can access the project. One way to enable sharing is to select the **Allow other users to view and edit this project** check box in the New Project wizard or the Project Properties dialog box.

Note: For migrated projects, you cannot change the sharing status of the project until the project is compatible with the current version of SAS Forecast Server. You can use the update batch operation to update multiple projects. In the SAS Forecast Server Plug-in for SAS Management Console, right-click on the environment node, and select **Batch Operations** ⇒ **Update**. For more information, see the Help for the SAS Forecast Server Plug-in for SAS Management Console.

Configure Sharing for a Project

To enable sharing for a project in the SAS Forecast Server Plug-in for SAS Management Console:

1. Open SAS Management Console. To log on, your user account must be assigned to the Forecast Server: Administrator group. You must also have a valid operating system account.
2. In the **Plug-ins** tab, expand the **Forecast Server** and the environment node that contains the project that you want to share.
3. Right-click on a project and then select **Properties**. The *project-name* Properties dialog box appears.
4. Select the **Enable sharing** check box, and then click **OK**.

For more information about how to enable project sharing in SAS Forecast Studio, see the *SAS Forecast Studio: User's Guide*.

Sharing and Groups

Although project ownership and the sharing security model do not include the explicit concept of a user group, the management structure does divide users into implied groups. For example, all users who can access a particular environment can be considered a group. Therefore, when you enable sharing on a project, you are essentially sharing the project with other members of your group. A user who does not have access to the environment where your project is stored still cannot access the project. You should use caution when copying projects that are shared to another environment. This is because the scope of the access that is allowed by the sharing option depends on how the related environment is configured.

Chapter 8

Administering Reports to Use in SAS Forecast Studio

Overview of Reports	61
What Is a Report?	61
Types of Reports	62
Manage Reports	63
Access the Reports in SAS Forecast Studio	64
Use the Sample Reports	65
Deploy the Sample Reports	65
Remove the Sample Reports	65
Register Reports	66
Preregistration Tasks	66
Register a Report	67
Create Reports	69
Types of Report Parameters	69

Overview of Reports

What Is a Report?

In the SAS Forecast Server, reports enable you to extend the capabilities of the product so that you can perform site-specific custom operations. Reports encapsulate custom logic. When the report is executed, it receives information about the run-time state of the project in SAS Forecast Studio. By collecting information about the state of the project, the SAS Forecast Server knows what the user is looking at and, consequently, what data to use when generating the report.

A report is written in the SAS programming language and is saved as a SAS stored process. A stored process is a SAS program that is stored centrally on a server. Stored processes consist of two distinct parts: the SAS code and the stored process definition that resides on a SAS Metadata Server. A client application can then execute the program and can receive and process the results. Stored processes enable you to maintain and manage code centrally, give you better control over changes, enhance security and application integrity, and ensure that every client executes the latest version of code that is available.

Stored processes are like other SAS programs, except that they have an additional feature that enables you to customize the program's execution. In a stored process, an invoking application supplies parameters when the stored process is invoked. For

example, if you have a stored process that analyzes monthly sales data, you could create a MONTH variable in the stored process. At execution time, you would supply the parameter MONTH=MAY to analyze May sales data. For more information about how to create a stored process and to invoke it in a client application, see *SAS Stored Processes: Developer's Guide* at <http://support.sas.com/documentation/onlinedoc/inttech>.

Reports are implemented using stored processes. However, not all SAS stored processes are reports. The reports in the SAS Forecast Server differ from generic stored processes in the following ways:

- The reports rely on a special infrastructure that includes the macros that are delivered with the SAS Forecast Server. As a result, reports reflect the run-time state of SAS Forecast Studio.
- SAS Forecast Server includes a custom execution mechanism for reports. In fact, most reports work only when run through this mechanism. These reports are not intended to be run outside of the SAS Forecast Server.

To distinguish reports from other stored processes, the SAS Forecast Server identifies them by the FS_REPORT keyword.

Types of Reports

In the SAS Forecast Server, you can have two types of reports.

- Several sample reports that demonstrate how reports can be used to handle a variety of tasks are provided with the SAS Forecast Server. A sample report can serve as a basis for a custom report. Simply copy the sample report, and then modify the report to meet your site's needs.

The reports are not available for use in the Reports and Stored Processes dialog box until you deploy them using the SAS Forecast Server Plug-ins for SAS Management Console. For a list and descriptions of the sample reports, see the *SAS Forecast Studio: User's Guide*.

- To meet the needs of your site, you can create custom reports.

When working with custom reports in the SAS Forecast Server, note the following constraints:

- SAS Forecast Server does not currently support the execution of reports using a SAS Stored Process Server.
- For stored processes that are compatible with SAS 9.2, each metadata object is configured to use a specific execution server. When you use multiple SAS Workspace Servers to host projects, you might need to create one metadata object for each SAS Workspace Server. You must copy the source file for the stored process to the file system of each SAS Workspace Server. In this way, each metadata object can refer to the source file that is located in its file system.
- The Reports and Stored Processes dialog box in SAS Forecast Studio lists only reports that run on the same SAS Workspace Server as the current project. Reports that exist in metadata, but that are not shown in this dialog box, are not visible to the user (for example, the ReadMetadata permission is not granted or the Use Reports capability has not been assigned to the user), are missing the FS_REPORT keyword, or are configured to use a different server.

When you register a report object in SAS Management Console for use with the SAS Forecast Server, you must use the FS_REPORT keyword to distinguish the report object

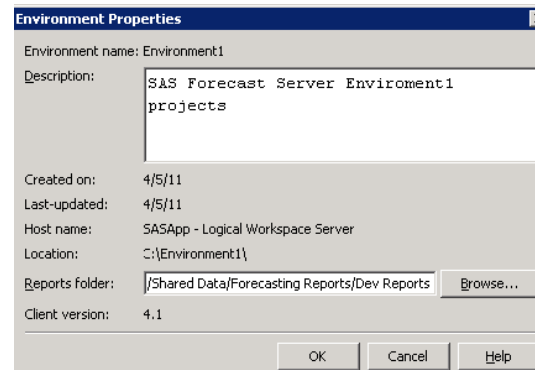
from other stored processes. The sample reports provided with the SAS Forecast Server use the FS_SAMPLE keyword to distinguish them from custom reports.

Manage Reports

To use reports in SAS Forecast Studio, you must first create a report folder. Then, you must configure your environment to use this folder location. The existence of the report folder enables the Reports and Stored Processes dialog box in SAS Forecast Studio.

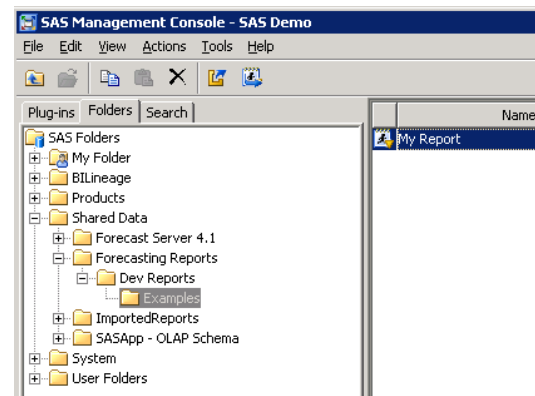
For example, suppose that you configure your environment to use the **/Shared Data/Forecasting Reports/Dev Reports** report folder as shown in the following display:

Figure 8.1 Configuring the Environment's Root Report Folder



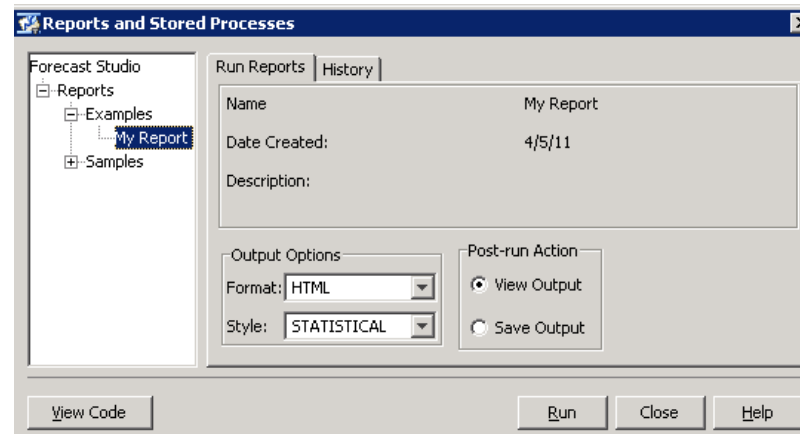
In addition, suppose that you create the **My Report** stored process metadata object at the folder location **/Shared Data/Forecasting Reports/Dev Reports/Examples**, as shown in the following display:

Figure 8.2 My Report Metadata Object



In the following example, **My Report** is displayed under the **Reports/Examples** folder in the Reports and Stored Processes dialog box.

Figure 8.3 Folder Presentation in the Reports and Stored Processes Dialog Box



The **Reports** folder at the top of the hierarchy represents the root report folder that was configured in the metadata. The root report folder path does not appear in the Reports and Stored Processes dialog box.

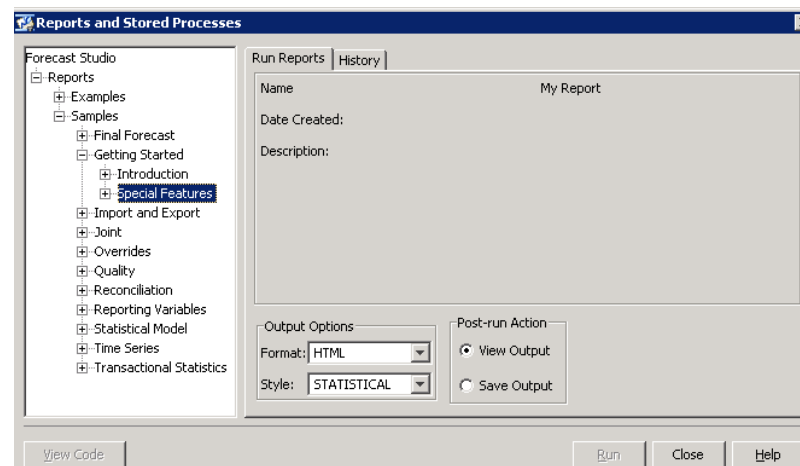
Before you can create a report for use in SAS Forecast Studio, you must let the environment know where your report folder is located. You can configure the location for the report folder in the Environment Properties dialog box.

Access the Reports in SAS Forecast Studio

You can access and execute reports when you have a project open in SAS Forecast Studio by selecting **Tools** ⇒ **Reports and Stored Processes**.

Here is an example of the Reports and Stored Processes dialog box in SAS Forecast Studio:

Figure 8.4 SAS Forecast Studio Reports and Stored Processes Dialog Box



The reports shown in the Reports and Stored Processes dialog box in SAS Forecast Studio are filtered based on whether the user is assigned the Use Reports capability, whether the ReadMetadata permission has been specified for a report, the use of the

FS_REPORT keyword, and the server constraints that you specified in the metadata. To use a report, the report must be on the same server as the project. When you run a report, the locations of the data and forecasts for the current project are passed to the stored process.

Use the Sample Reports

Deploy the Sample Reports

To deploy the sample reports:

1. Create a report folder to store your metadata objects. For example, the report folder might be `/SharedData/FSReports`.
2. Grant all permissions on the report folder to the Metadata User. Configure any additional user permissions on this folder.
 - a. In SAS Management Console, click the **Folders** tab.
 - b. Right-click on the report folder, and select **Properties**. The folder-name Properties dialog box appears.
 - c. Click the **Authorization** tab.
 - d. In **Users and Groups**, select **Forecast Server Metadata User**.

Note: If Forecast Server Metadata User is not available, click **Add**, and then select it from the Add Users and Groups dialog box.
 - e. In **Effective Permissions**, grant all of the permissions to the Forecast Server Metadata User.
 - f. Configure any additional user permissions, and then click **OK**.
3. Designate a root report folder for the environment.
 - a. Select the **Plug-ins** tab and expand the **Forecast Server** node.
 - b. Right-click the desired environment node, and then select **Properties**. The Properties dialog box appears.
 - c. In the **Reports folder** field, type the path to your reports metadata folder, or click **Browse**.
 - d. Click **OK**.
4. In SAS Management Console, click the **Plug-ins** tab.
5. Expand the **Application Management** and **Forecast Server** folders.
6. Expand the environment folder to which you want to deploy the sample reports.
7. Right-click **Reports**, and select **Deploy Samples**.

Remove the Sample Reports

To remove all sample reports from the current environment's report tree:

1. In SAS Management Console, click the **Plug-ins** tab.
2. Expand the **Application Management** and **Forecast Server** folders.

3. Expand the environment folder from which you want to remove the sample reports.
4. Right-click **Reports**, and select **Remove Samples**.
5. To remove an empty report folder from the current environment's report tree, right-click **Reports**, and select **Prune**.

Note: The **Reports** folder does not appear in an environment folder until the report tree is configured.

CAUTION:

Selecting **Remove Samples** removes any report object with the **FS_SAMPLE** keyword. Before selecting **Remove Samples**, you should remove the **FS_SAMPLE** keyword from any custom report that was created by copying a sample report.

Register Reports

Preregistration Tasks

The following tasks enable you to gather the information that you need to make a report usable in SAS Forecast Studio.

To prepare a report for registration:

1. Locate the source code for the report. The source code can be saved to a file system or in a metadata repository.

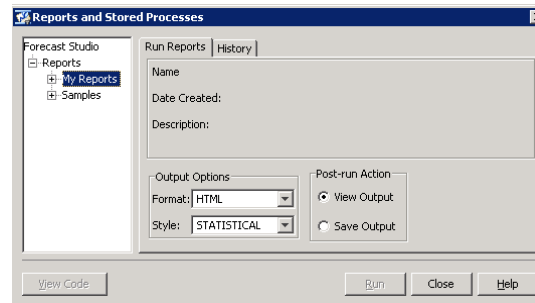
For example, the `example.sas` file is saved in the `C:\MyStoredProcesses` directory. The source repository location is `C:\MyStoredProcesses`, and the source filename is `example.sas`.

Note: The file system location must be on the same physical machine that hosts the corresponding project files for the SAS Forecast Server. Having all of these files on the same physical machine ensures that the project data is accessible to the stored process when it executes.

2. Plan the organization of your report hierarchy. If you want to display the report under its own folder group in the Reports and Stored Processes dialog box in SAS Forecast Studio, then you can create this hierarchy in the metadata using SAS Management Console. The Reports folder in the Reports and Stored Processes dialog box corresponds to the root report folder that you configured for that environment in SAS Management Console. The expandable folders in the dialog box correspond to subfolders that you create under the root report folder. For example, you can specify that the **My Reports** folder is displayed in the Reports and Stored Processes dialog box under **Reports** by creating a subfolder with that name in the root report folder.

Here is how a custom folder structure would appear:

Figure 8.5 Example Report Hierarchy

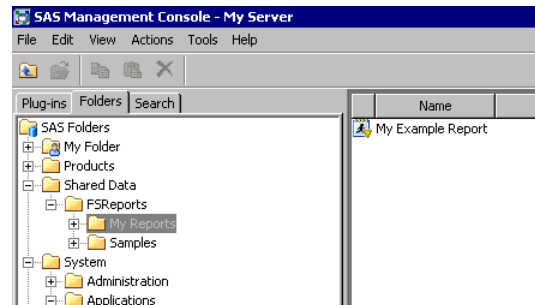


For more information about creating report folders, see the documentation for SAS Management Console at <http://support.sas.com/documentation/onlinedoc/sasmc>.

3. Select a name for the report. The report name that you specify in SAS Management Console is the metadata object name. The report name is displayed in the Reports and Stored Processes dialog box in SAS Forecast Studio. For example, if you name the report **My Example Report** in SAS Management Console, then the Reports and Stored Processes dialog box displays the **My Example Report** report.

Here is an example folder structure with a **My Reports** subfolder that contains **My Example Report**:

Figure 8.6 Report Hierarchy with an Example Report



Register a Report

For information about the basic processes and terminology that are related to using stored processes, see *SAS Stored Processes: Developer's Guide* at <http://support.sas.com/documentation/onlinedoc/inttech>.

Note: Before you change an existing stored process, it is recommended that you make a copy of the stored process metadata object and its associated source code.

Before you register a report, perform the following steps in SAS Management Console:

1. Create a report folder to store your metadata objects.
2. Write the code for the report.
3. Set permissions on the report folder so that all users can access it in SAS Forecast Studio.
 - a. In SAS Management Console, click the **Folders** tab.

- b. Right-click on the report folder, and select **Properties**. For example, the report folder might be **/Shared Data/FSReports**. The folder-name Properties dialog box appears.
 - c. Click the **Authorization** tab.
 - d. In **Users and Groups**, select a user or group.
 - e. In **Effective Permissions**, grant permissions for that user or group, and then click **OK**.
4. Configure your environment to use the report folder.
5. Create a stored process metadata object.
 - a. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
 - b. Click the **Folders** tab.
 - c. In the root report folder, right-click on any folder, and then select **New** ⇒ **Stored Process**. The New Stored Process wizard appears.
 - d. In the General step, enter the name of your stored process—**My Example Report**—and an optional description.
 - e. Next to **Keywords**, click **Add**. The Add Keyword dialog box appears.
 - f. Type **FS_REPORT**, and then click **OK**. Click **Next**.
 - g. In the Execution step, the **Application server** is the SAS Workspace Server that hosts the project files.

Note: The execution constraints for a stored process must enable you to run the stored process on the SAS Workspace Server that hosts the projects that use this stored process.

- h. For **Source code repository**, select the directory where you saved your SAS code (in this example, this value is **C:\MyStoredProcesses**). If the directory does not appear in the list, you can add a new directory location.
 - i. For **Source file**, specify **example.sas**. For **Result capabilities**, select **Package**. Click **Next**.

Note: Starting in SAS 9.3, you can save the source code with the stored process metadata. By saving the source code with the metadata, you can run the stored process on multiple application servers. When the source code is saved to a directory on the server, you can run the stored process only on the server where the source code is saved.

- j. Click **Finish** in the Parameters panel.

Note: For more information about defining parameters, see *SAS Stored Processes: Developer's Guide* at <http://support.sas.com/documentation/onlinedoc/inttech>.

Create Reports

Types of Report Parameters

What Is a Normal Report Parameter?

A normal report parameter is a stored process prompt that is completely defined in the metadata. Examples of normal parameters include specifying the title to use for the output and specifying whether to include the observation number in the results. These parameters are created using the usual interfaces for defining stored process prompts (for example, SAS Management Console). The only constraint is that a normal report parameter name cannot start with a dynamic parameter prefix. (See [Table 8.1 on page 69.](#))

What Is a Dynamic Report Parameter?

A dynamic report parameter is a parameter that is defined by options that you select from the product at run time. Because the options depend on the run-time state, they cannot be previously defined in the metadata. For example, a parameter that enables you to select an independent variable from the current project is a dynamic report parameter.

Dynamic report parameters require special support from the product to generate the list of available options. They are not extensible. To find dynamic parameters in a report, the product checks for a special prefix in the name of the text (type) parameter that matches a prefix in the following table. If a match is found, then the corresponding option list is generated and assigned to the parameter before the stored process prompt is displayed. If a match is not found, then the product assumes that the parameter is a normal report parameter and uses it as is.

Table 8.1 *Dynamic Report Parameter Prefixes*

Parameter Name	Available Options
FS_BYVAR	All project BY variables
FS_CURRENT_BYVAR	BY variables in the selected hierarchy level
FS_DEPVAR	All project dependent variables
FS_INDVAR	All project independent variables
FS_REPORTVAR	All project reporting variables
FS_EVENT	All project events
FS_FITSTAT	All statistics of fit
FS_MODELSTAT	All model statistics
FS_LIBRARY	Libraries allowing Read access

Parameter Name	Available Options
FS_WLIBRARY	Libraries allowing Write access
FS_OLAPSCHEMA	All OLAP schemas currently defined in metadata

When locating dynamic report parameters, the product has the following process:

1. Prefix matching allows multiple variables to use the same dynamic list of available options (for example, FS_INDVAR1 and FS_INDVAR2 both match FS_INDVAR). Therefore, all independent variables would be available options in the dynamic list.
2. For efficiency, only prefixes generated by truncating the name at non-letter characters are considered for when matching prefixes. For example, FS_INDVAR5 does not match the prefix FS_INDVAR because the name breaks on the letter S. However, the prefixes FS_INDVAR, FS_INDVAR_LIST, and FS_INDVAR1 all match the FS_INDVAR prefix.
3. If a required parameter has an empty list of available options, the product does not execute the stored process. This prevents a prompt that cannot be fulfilled. For example, you cannot use a required parameter named FS_INDVAR in a project that does not contain an independent variable.

What Is an Internal Report Parameter?

To help you create stored processes, the product provides internal parameters, also called predefined macro variables. Internal parameters are automatically defined and populated by the product. They do not appear in the metadata definition of the report. Internal parameters pass state information and instructions to the stored process environment. In particular, they convey information about the project and the location in the hierarchy that you are currently viewing.

Note: To ensure correct behavior, a user-defined parameter should not have the same name as an internal parameter. The naming convention used for a sample report (and generally recommended) is to prefix a metadata parameter with **FS_**. This prevents name collisions because all internal parameters have the **HPF_** prefix.

By default, the maximum length for a macro variable is 4,096. However, if the number of variables (BY, dependent, independent, reporting, and so on) is very large, you can increase the maximum length for the macro variable to 65,534 using the MVARSIZE= system option.

Note: In the following tables that describe macro variables, a variable name that ends with # indicates a sequenced variable. Unless otherwise stated, you can assume that sequencing begins with 1 for these variables. For levels, indexing starts at the top level.

Control Macro Variables

The following variables are used for basic session control:

Table 8.2 Control Macro Variables

Name	Description	Format
HPF_READ_ONLY	Project access Read-Only flag By default, stored processes have Read-Only access with respect to the project libraries. This macro variable changes access to Write. For example, the following SAS code assigns project library names with Read-Only access: <pre>%include "&HPF_INCLUDE";</pre>	Boolean
HPF_ODSDEST	Used inside a stored process to control the output destination for the report. By default, HPF_ODSDEST is set to HTML, which means that the output is in HTML format.	ODS destinations
HPF_ODSSTYLE	Used inside a stored process to control the output style.	ODS styles
HPF_MULTISELECT_VAR_COUNT	Used to specify the number of multi-value metadata parameters passed to the report.	Positive integer
HPF_MULTISELECT_VAR#	Used to specify the base name of each multi-value metadata parameter.	Multi-value metadata parameter name

Metadata Macro Variables

The following variables describe the metadata resources that are used by the SAS Forecast Server:

Table 8.3 Metadata Macro Variables

Name	Description	Format
HPF_METADATA_HOST	SAS Metadata Server host name	Host name
HPF_METADATA_PORT	SAS Metadata Server port number	Port number

Name	Description	Format
HPF_METADATA_REPNAME	Metadata repository name	Repository name

Library Macro Variables

The following variables describe the library resources that are used by the SAS Forecast Server:

Table 8.4 Library Macro Variables

Name	Description	Format
HPF_ENV_LIBNAME_COUNT	Number of local environment libraries	Positive integer
HPF_ENV_LIBNAME#	LIBNAME of each local environment library	LIBNAME library names
HPF_ENV_LIBPATH#	File system path of each local environment library	Directory path
HPF_ENV_LIBACCESS#	Access level of each local environment library	Positive integer
HPF_META_LIBNAME_COUNT	Number of manually assigned metadata libraries	Positive integer
HPF_META_LIBNAME#	LIBNAME of each manually assigned metadata library	LIBNAME library names
HPF_META_LIBID#	FQID of the metadata object that defines each manually assigned metadata library	Metadata FQID value
HPF_META_LIBACCESS#	Access level of each manually assigned metadata library	Positive integer

Basic Project Macro Variables

The following variables describe the basic characteristics of the active project:

Table 8.5 Basic Project Macro Variables

Name	Description	Format
HPF_PROJECT	Project name.	SAS name
HPF_DESC	Project description.	SAS label

Name	Description	Format
HPF_PROJECT_SERVER	Logical name of the host SAS Workspace Server.	Host name
HPF_DEFAULT_LOCATION	Base file system path to the environment directory.	File system path
HPF_PROJECT_LOCATION	File system path to the project directory.	System path
HPF_INCLUDE	File system path to the project include file. This macro variable specifies the system path and filename that contains the SAS code to assign the SAS libraries and catalogs that are associated with the project. By default, all SAS libraries and catalogs are assigned with Read-Only access (ACCESS=READONLY). For example, the following SAS code assigns project library names with Read-Only access: <pre>%include "&HPF_INCLUDE";</pre>	System filename
HPF_EVENTS	Space-delimited list of all event names.	SAS name
HPF_NUM_EVENTS	Number of events defined in the project.	Positive integer
HPF_EVENT_#	The name of individual event elements.	SAS name

Data Hierarchy and Settings Macro Variables

The following variables describe the data hierarchy and settings for the active project:

Table 8.6 Data Hierarchy and Settings Macro Variables

Name	Description	Format
HPF_INPUT_LIBNAME	SAS library reference from which the source data for the input data set was obtained.	SAS name
HPF_INPUT_DATASET	Data set from which the source data was obtained.	SAS name
HPF_NUM_BYVARS	Number of BY variables. If there are no BY variables, HPF_NUM_BYVARS is set to zero.	Positive integer
HPF_BYVAR#	Individual BY variable names listed in the n^{th} position of the ordered list of BY variables (HPF_BYVARS). The first BY variable name is stored in HPF_BYVAR1, the second in HPF_BYVAR2, and the last is stored in HPF_BYVAR&HPF_NUM_BYVARS. If there are no BY variables (&HPF_NUM_BYVARS is zero), these macro variables are not defined.	SAS name
HPF_NUM_DEPVARS	Number of dependent variables. There is always at least one dependent variable.	Positive integer
HPF_DEPVARS	Space-delimited list of all dependent variable names. The order of the dependent variable names is the same as specified in the project.	List of SAS names separated by a single space
HPF_DEPVAR#	Individual dependent variable names listed in the n^{th} position of the ordered list of dependent variables (HPF_DEPVARS). The first dependent variable name is stored in HPF_DEPVAR1, the second in HPF_DEPVAR2, and the last is stored in HPF_DEPVAR&HPF_NUM_DEPVARS. Because there is always at least one dependent variable associated with a project, HPF_DEPVAR1 is always defined.	SAS name

Name	Description	Format
HPF_NUM_INDEPVARs	Number of independent variables. If there are no independent variables, HPF_NUM_INDEPVARs is set to zero.	Positive integer
HPF_INDEPVARs	Space-delimited list of all independent variable names. The order of the independent variable names is the same as specified in the project. The macro variable is always defined. If there are no independent variables, HPF_INDEPVARs is set to NULL.	List of SAS names separated by a single space
HPF_INDEPVAR#	Individual independent variable names listed in the n^{th} position of the ordered list of independent variables (HPF_INDEPVARs). The first independent variable name is stored in HPF_INDEPVAR1, the second in HPF_INDEPVAR2, and the last is stored in HPF_INDEPVAR&HPF_NUM_INDEPVARs. If there are no independent variables (&HPF_NUM_INDEPVARs is zero), these macro variables are not defined.	SAS name
HPF_NUM_REPORTVARs	Number of reporting variables. If there are no reporting variables, then HPF_NUM_REPORTVARs is set to zero.	Positive integer
HPF_REPORTVARs	Space-delimited list of all reporting variable names. The order of the reporting variable names is the same as specified in the project. The macro variable is always defined. If there are no reporting variables, HPF_REPORTVARs is set to NULL.	List of SAS names separated by a single space
HPF_REPORTVAR#	Individual reporting variable names listed in the n^{th} position of the ordered list of reporting variables (HPF_REPORTVARs). The first reporting variable name is stored in HPF_REPORTVAR1, the second in HPF_REPORTVAR2, and the last is stored in HPF_REPORTVAR&HPF_NUM_REPORTVARs. If there are no reporting variables (&HPF_NUM_REPORTVARs is zero), these macro variables are not defined.	SAS name

Name	Description	Format
HPF_TIMEID	Time ID variable name.	SAS name
HPF_TIMEID_FORMAT	SAS format of the time ID variable.	SAS format name
HPF_SEASONALITY	Integer length of the seasonal cycle. A seasonality of 1 implies no seasonality.	Positive integer
HPF_INTERVAL	Interval of the time ID variable (for example, MONTH).	SAS time interval
HPF_DATASTART	Start date, datetime, or time value of the project. The starting time ID value of the project input data set (&HPF_LJIBNAME.&HPF_DATASET).	SAS date, datetime, or time value
HPF_DATAEND	End date, datetime, or time value of the project. The ending time ID value of the project input data set (&HPF_LJIBNAME.&HPF_DATASET).	SAS date, datetime, or time value
HPF_SETMISSING	Controls the interpretation of missing values.	
HPF_TRIMMISS	Controls whether missing values are trimmed.	
HPF_ZEROMISS	Controls the interpretation of zero values.	
HPF_NUM_LEVELS	Number of levels in the hierarchy. The levels of the hierarchy are numbered from 1 (the top of the hierarchy) to &HPF_NUM_LEVELS (the leaves of the hierarchy). If there is no hierarchy, then the number of levels is 1.	Positive integer

Name	Description	Format
HPF_LEVEL_BYVARS#	Space-delimited list of BY variable names associated with the n^{th} level, where n ranges from 1 to &HPF_NUM_LEVELS. The variables names are separated by a single space. The BY variables at level 1 (the top) are stored in HPF_LEVEL_BYVARS1. The BY variables at the lowest level (the leaves) are stored in HPF_LEVEL_BYVARS&HPF_NUM_LEVELS.	SAS name
HPF_LEVEL_DATAWHERE#	Input data filtering WHERE clause for the n^{th} level, where n ranges from 1 to &HPF_NUM_LEVELS. These WHERE clauses can be used to subset the input data sets for each level in the hierarchy to obtain information about the currently selected node. The WHERE clause at level 1 (the top) is stored in HPF_LEVEL_DATAWHERE1. The WHERE clause at the lowest level (the leaves) is stored in HPF_LEVEL_DATAWHERE&HPF_NUM_LEVELS. <i>Note:</i> You must unquote this macro variable: <pre>%unquote (&&HPF_CURRENT_DATAWHERE&n) ;</pre>	SAS WHERE clause
HPF_LEVEL_OUTWHERE#	Output data filtering WHERE clause for the n^{th} level, where n ranges from 1 to &HPF_NUM_LEVELS. These WHERE clauses can be used to subset the output data sets for each level in the hierarchy to obtain information about the currently selected node. The WHERE clause at level 1 (the top) is stored in HPF_LEVEL_OUTWHERE1. The WHERE clause at the lowest level (the leaves) is stored in HPF_LEVEL_OUTWHERE&HPF_NUM_LEVELS. <i>Note:</i> You must unquote this macro variable: <pre>%unquote (&&HPF_LEVEL_OUTWHERE&n) ;</pre>	SAS WHERE clause

Name	Description	Format
HPF_LEVEL_LIBNAME#	SAS library reference for the n^{th} level, where n ranges from 1 to &HPF_NUM_LEVELS. The SAS library reference at level 1 (the top) is stored in HPF_LEVEL_LIBNAME1. The SAS library reference at the lowest level (the leaves) is stored in HPF_LEVEL_LIBNAME&HPF_NUM_LEVELS.	SAS LIBNAME
HPF_LEVEL_NSERIES#	Number of series associated with the n^{th} level, where n ranges from 1 to &HPF_NUM_LEVELS. The number of series at level 1 (the top) is stored in HPF_LEVEL_NSERIES1. The number of series at the lowest level (the leaves) is stored in HPF_LEVEL_NSERIES&HPF_NUM_LEVELS.	Positive integer

Diagnose Macro Variables

The following variables describe the current diagnose settings for the active project:

Table 8.7 Diagnose Macro Variables

Name	Description	Format
HPF_DIAGNOSE_INTERMITTENT	Intermittency threshold values for diagnose tests.	Positive number
HPF_DIAGNOSE_SEASONTEST	Seasonality significance level for diagnose tests.	p -value

Model Selection Macro Variables

The following variables describe the current model selection settings for the active project:

Table 8.8 Model Selection Macro Variables

Name	Description	Format
HPF_SELECT_CRITERION	Statistic of fit to use for model selection.	
HPF_SELECT_HOLDOUT	Absolute number of observations to hold out for selection. Zero implies that the model fit is used for selection.	
HPF_SELECT_HOLDOUTPCT	Percentage of observations to hold out for selection.	Positive integer
HPF_SELECT_MINOBS_NON_MEAN	Minimum number of observations to require for a non-mean model.	Positive integer
HPF_SELECT_MINOBS_TREND	Minimum number of observations to require for a trend model.	Positive integer

Forecast Macro Variables

The following variables describe the current forecasting settings for the active project:

Table 8.9 Forecast Macro Variables

Name	Description	Format
HPF_LEAD	Integer number of time periods to forecast the length of the forecast horizon or lead.	Positive integer
HPF_FORECAST_ALPHA	Confidence level size.	p -value
HPF_BACK	Integer number of time periods to ignore when forecasting.	Positive integer

Reconciliation Macro Variables

The following variables describe the current reconciliation settings for the active project:

Table 8.10 Reconciliation Macro Variables

Name	Description	Format
HPF_RECONCILE_LEVEL	Index of the reconciliation level. The reconciliation level index ranges from 1 to &HPF_NUM_LEVELS, depending on the level of reconciliation.	Positive integer
HPF_RECONCILE_BYVAR	BY variable associated with the level used for reconciliation.	SAS name
HPF_RECONCILE_METHOD	Reconciliation method.	SAS name
HPF_LEVEL_RECONCILE_DATASET#	Reconciled forecast data set for each level.	
HPF_LEVEL_RECONCILE_STATISTICS#	Reconciled statistics data set for each level.	
HPF_LEVEL_RECONCILE_SUMMARY#	Reconciled summary data set for each level.	

Active Series Macro Variables

The active series macro variables differ from the other macro variables because they depend on the folder of the tree (in the forecasting hierarchy) that is currently selected. Because of this dependency, you cannot use these macro variables outside of the SAS Forecast Server.

Table 8.11 Active Series Macro Variables

Name	Description	Format
HPF_CURRENT_LEVEL	The level index number associated with the current level. The current level number ranges from 1 to &HPF_NUM_LEVELS, depending on the currently selected level of the hierarchy.	Positive integer
HPF_CURRENT_LIBNAME	The SAS library reference associated with the currently selected level of the hierarchy.	SAS LIBNAME
HPF_CURRENT_LEVEL_START	Start date, datetime, or time value of the current level. The starting time ID value of the input data set for the currently selected level of the hierarchy.	SAS date, datetime, or time value
HPF_CURRENT_LEVEL_END	End date, datetime, or time value of the current level. The ending time ID value of the input data set for the currently selected level of the hierarchy.	SAS date, datetime, or time value
HPF_CURRENT_LEVEL_NSERIES	Number of series (or nodes) associated with the currently selected level of the hierarchy.	Positive integer
HPF_CURRENT_SERIESSTART	Start date, datetime, or time value of the current node. The starting time ID value of the series for the currently selected node of the hierarchy.	SAS date, datetime, or time value
HPF_CURRENT_SERIESEND	End date, datetime, or time value of the current node. The ending time ID value of the series for the currently selected node of the hierarchy.	SAS date, datetime, or time value

Name	Description	Format
HPF_CURRENT_DATAWHERE	Input data WHERE clause for the currently selected node. This WHERE clause can be used to subset the input data set to obtain information about the currently selected node of the hierarchy. <i>Note:</i> You must unquote this macro variable: <pre>%unquote (&HPF_CURRENT_DATAWHERE) ;</pre>	SAS WHERE clause
HPF_CURRENT_OUTWHERE	Output data WHERE clause for the currently selected node. This WHERE clause can be used to subset the output data set to obtain information about the currently selected node of the hierarchy. <i>Note:</i> You must unquote this macro variable: <pre>%unquote (&HPF_CURRENT_OUTWHERE) ;</pre>	SAS WHERE clause
HPF_NUM_CURRENT_BYVARS	Number of BY variable names for the currently selected level of the hierarchy.	Positive integer
HPF_CURRENT_BYVARS	Space-delimited list of BY variable names for the currently selected level of the hierarchy. The macro variable is always defined. If there are no BY variables, HPF_CURRENT_BYVARS is set to NULL.	List of SAS names separated by a single space
HPF_CURRENT_BYVARS#	Individual BY variable names for the current level.	SAS name
HPF_CURRENT_DEPVAR	Dependent variable name associated with the currently selected node of the hierarchy. This variable is contained in the list of dependent variables (HPF_DEPVAR).	SAS name
HPF_CURRENT_HORIZON	Horizon date, datetime, or time value of the current node. The time ID value of the start of the multi-step ahead forecast for the currently selected node of the hierarchy.	SAS date, datetime, or time value

Sample Reports to Get You Started

Note: You must deploy the sample reports before you can use them in SAS Forecast Studio. For more information, see [“Use the Sample Reports” on page 65](#).

To understand how to use dynamic parameters, multi-value parameters, and manually assigned metadata libraries, see these types of sample reports in the Reports and Stored Processes dialog box in SAS Forecast Studio.

Using Dynamic Parameters

This sample report shows all special parameter names for which dynamic lists are supported and demonstrates how to configure such parameters.

Using Libraries

This sample report shows how libraries that are not pre-assigned can be used in reports using the provided macro support.

Using Package Files

This sample report shows how to add files to the generated results package. These files can be accessed by saving the report output to your local file system.

Working with List Variables

This sample report shows how list variables are now passed to programs and how to use provided macros to reformat them.

Initialize a Report

When you create a report, the header portion of the report’s source code file must contain macro calls to initialize the SAS Forecast Server workspace macros and ODS output.

```
*ProcessBody;

/*-----*/
/*- initialize the HPF stored process support      -*/
/*-----*/
%hpfstp();

/*-----*/
/*- initialize the ODS output                      -*/
/*-----*/
%HPF_InitODSOutput();

%stpbegin;
```


Chapter 9

Miscellaneous Administration Tasks

Adding Start-Up and Shutdown Code to a Project	85
Understanding Start-Up or Shutdown Code	85
How to Configure This Code to Run	86
How to Add Start-Up or Shutdown Code to Your Project	86
Editing Invalid Start-up Code	87
Remove a Configuration with the SAS Deployment Manager	87
Archive a Project	87
Configure LOCKDOWN Feature	88

Adding Start-Up and Shutdown Code to a Project

Understanding Start-Up or Shutdown Code

You can customize your project by specifying SAS code to run when the project is open or when the project is closed. For example, you specify the MPRINT system option to run when the project opens. Now, any SAS statements that are generated when the user runs the SAS Forecast Studio macros are traced for debugging.

Note: Do not use the start-up code to specify the input libraries for a project.

When you close a project, you might want SAS Forecast Studio to remove any temporary files that were created, make backups of files, or automatically export your results to an external data set.

Although the start-up and shutdown code is associated with a project, the scope of the code applies to an entire session. Therefore, it is important that the shutdown code closes any processes that were started by the start-up code. For example, in the start-up code where you specified the MPRINT system option if you do not turn off the MPRINT system option in the shutdown code and then you close the current project and open another project, SAS Forecast Studio will still be tracing any SAS statements that are generated for macro variables. You can see that the MPRINT option is still specified in the log. If you turn off the MPRINT system code in the shutdown code and then you close the current project, the tracing functionality is turned off when you open another project.

The code is saved to the startup.sas and shutdown.sas files in the ***SAS-project-directory\Config*** directory. Although you can edit these files, the code can also be modified when the project is open in SAS Forecast Studio.

How to Configure This Code to Run

Site administrators can specify whether this code should run for a SAS deployment. If the code should not run, a message stating that code execution is not permitted appears. Due to security considerations, this code is ignored by default.

To run this code:

1. Start SAS Management Console and connect as a SAS administrator (for example, sasadm@saspw).
2. In the **Plug-ins** tab, expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the selection pane, select **Forecast Server**.
6. From the **Project scripts** drop-down list, select **Execute**.
7. Restart SAS Forecast Server for these changes to take effect.

How to Add Start-Up or Shutdown Code to Your Project

When you create a project, you can add this code by using the New Project wizard. After the project is created, you can add or edit this code by using the Project Properties dialog box.

Note: For a user to add start-up or shutdown code in SAS Forecast Studio, the user must be assigned the Modify Project Scripts capability using the SAS Forecast Server plug-in in SAS Management Console. For more information, see [“Managing Roles and Capabilities” on page 29](#).

To add SAS code that runs when the project is opened or closed:

1. Open SAS Forecast Studio.
2. Select **File** ⇒ **Project Properties**. The Project Properties dialog box appears.
3. Click **Modify**. The SAS Start-up and Shutdown Code dialog box appears.
4. In the **Start-up Code** and **Shutdown Code** tabs, enter the SAS code that you want to include in the project.
5. (Optional) To update the shutdown code that is saved in memory, select the **Also update the stored shutdown code to be used during the upcoming close action** option. To view the code that is currently saved in memory, click **View the stored shutdown code**.

Note: When a project is open, the shutdown code that runs when you close the project is stored in memory. If you change the shutdown code, you must select the **Also update the stored shutdown code to be used during the upcoming close action** check box if you want your changes to be applied to the saved copy of the shutdown code.

6. Click **OK** to save your changes and to close the Project Properties dialog box.

Editing Invalid Start-up Code

If the start-up code is invalid, you might not be able to open the project. Instead, a warning message appears, and the SAS Log shows the errors in the code. To edit the start-up code, you must edit the *SAS-project-directory\Config\startup.sas* file. You cannot edit the code from the user interface.

Remove a Configuration with the SAS Deployment Manager

To remove a configuration:

1. Navigate to the SASHome directory. For example, on a Windows system, this is the `C:\Program Files\SAS\SASDeploymentManager\9.4` folder.
Double-click `sasadm.exe` to launch the SAS Deployment Manager.
2. Select **Remove existing configuration**, and then click **Next**.
3. Select a configuration directory, and then click **Next**.
4. Specify your connection information to the SAS Metadata Server, and then click **Next**.
5. Select the product that you want to unconfigure, and then click **Next**. For example, to unconfigure SAS Forecast Server, select **Forecast Server 14.2**.
6. SAS Forecast Server has metadata describing user-defined environments and projects. This metadata is stored in the SAS Metadata Server. If you are removing a configuration for SAS Forecast Server, you can also select to unregister the user content. This option removes the SAS Forecast Server metadata when you remove the configuration. In either case, the project content is not deleted when you remove the configuration. You can use the register action to re-create the removed metadata later if desired.
Note: You can use the SAS Forecast Server plug-in for SAS Management Console to register and unregister user content. For more information, see the Help for the SAS Forecast Server plug-in for SAS Management Console.
7. In the **Summary** screen, click **Start**.

Archive a Project

You can archive a project by using the SAS Forecast Server plug-in for SAS Management Console. You can also archive projects by using the %FSEXPORT macro.

To archive a project by using the SAS Forecast Server plug-in for SAS Management Console:

1. Open SAS Management Console 9.4 and connect to your profile.
2. Under the **Application Management** node, expand the **Forecast Server** node and then expand the node for the environment that contains the project that you want to archive.

3. Expand the **Projects** node and right-click a project in the navigation tree. Select **Archive** from the pop-up menu. The Archive dialog box appears.
4. (Optional) Specify a name for the archived project. By default, SAS Forecast Server uses the project name. The name must be a valid SAS name, and it cannot be the name of an existing archive.
5. (Optional) Enter a description for the archive.
6. Specify where you want to save the archive.
7. Click **OK**.

Configure LOCKDOWN Feature

Beginning in the first maintenance release for SAS 9.4, a LOCKDOWN option is included in SAS that limits the accessibility and activities of a SAS server by putting it in a locked-down state. For more information about how to configure the LOCKDOWN option, see the *SAS Intelligence Platform: Security Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Part 4

Using the SAS Forecast Server Clients

<i>Chapter 10</i>	
SAS Forecast Studio	<i>91</i>
<i>Chapter 11</i>	
SAS Forecast Batch Interface	<i>105</i>

Chapter 10

SAS Forecast Studio

Requirements for Starting SAS Forecast Studio	91
Running SAS Forecast Studio on Your Desktop	92
Using Java Web Start	92
How to Launch SAS Forecast Studio	92
Configuring the Logon Behavior for Java Web Start	92
Starting SAS Forecast Studio with Options	94
Specify the Number of Observations to Use to Detect the Time Interval	94
Specify the Sample Size for Validating BY Variable Values and Formats	95
Enable SAS Forecast Studio Options in Java Web Start	95
Adding Custom Time Intervals to SAS Forecast Studio	96
Create the Custom Time Interval	96
Example 1: StoreHours Custom Interval	97
Example 2: Creating a Custom Interval Definition Interactively	98
Additional Considerations	99
Adding a Custom Format to SAS Forecast Studio	99
Create a Custom Format	99
Create a Custom Format in Your Autoexec File	100
Additional Considerations	101
Creating a Customized List of Events	101
Create a Customized Lists of Events	101
Example: Super Bowl Events	102

Requirements for Starting SAS Forecast Studio

Before you can run the SAS Forecast Studio client, you must start the servers (for example, the SAS Metadata Server, and the web application server). For information about how to start these servers, see *SAS Intelligence Platform: System Administration Guide*.

Running SAS Forecast Studio on Your Desktop

SAS Forecast Studio runs only on the Windows operating system. To start SAS Forecast Studio on the system where it is installed, select **Start** ⇒ **Programs** ⇒ **SAS** ⇒ **SAS Forecast Studio** ⇒ **SAS Forecast Studio 14.2**.

To open SAS Forecast Studio in a particular environment or for a specific project, type **launchFile=filename** at the command prompt where *filename* is the path to a .fs file that contains the parameters for your project. An example of a filename is **C:\Documents and Settings\Europe\Sales\Products2012.fs**. The **Products2012.fs** file contains the following code:

```
environment=Default
project=Products2012
```

When you run the **launchFile=Documents and Settings\Europe\Sales\Products2012.fs**, SAS Forecast Studio opens and displays the Products2012 project.

Using Java Web Start

How to Launch SAS Forecast Studio

You do not need to install SAS Forecast Studio on every system where you need to run SAS Forecast Studio. Instead, SAS Forecast Server supports automatic downloads of SAS Forecast Studio by using Java Web Start.

To launch SAS Forecast Studio using the Java Web Start client, perform either of these steps:

- Open the **SAS Forecast Server Current Status** Web page. The default URL for this page is where the SAS Forecast Server services are deployed. An example of this URL is `http://your-server-name:port-number/SASForecastServer/Status`.

To start SAS Forecast Studio, click **Launch using Java Web Start**.

- Use the direct link to the Java Web Start client. You can launch SAS Forecast Studio from `http://your-server-name:8080/SASForecastServer/main.jnlp`.

Configuring the Logon Behavior for Java Web Start

Default Logon Behavior for the Desktop Application and Java Web Start

Each list of SAS environments has a default environment. When you log on to SAS Forecast Studio, the Log On dialog box displays the environment that was most recently used on that machine. If you are logging on to SAS Forecast Studio using the Java Web Start client, the list of SAS environments includes the **(host environment)** option. This option represents the deployment from which the Java Web Start client was downloaded.

Configuring the Logon Behavior on the Server

Unlike the desktop application, the Java Web Start client often does not have a pre-configured list of SAS environments that appear in the **SAS environment** drop-down list. In many cases, the **(host deployment)** option might be the only one available. You can configure this list of environments in several ways.

For simple configurations of SAS Forecast Server, it might be sufficient to simply specify these configuration properties for the Java Web Start client:

- the URL to the file that specifies the SAS environment options.
- the environment that should be selected by default when the Log On dialog box appears. You can specify the preferred environment in SAS Management Console.
- whether users have permission to select a different environment in the Log On dialog box. You can prevent users from changing the environment by using the **Lock SAS environment selection** property in SAS Management Console.

The disadvantage to configuring the logon behavior using these configuration properties is that the logon behavior depends only on the server where the SAS Forecast Studio client was downloaded from. Therefore, all machines and users who rely on the Java Web Start client must use the same settings.

Configuring the Logon Behavior on the Client

On the client machine, you can configure the logon behavior in either of these ways:

- Specify the URL for the sas-environment.xml file in the SAS_ENV_DEFINITION_LOCATION environment variable.
- If you ran the SAS Deployment Wizard on that machine and the URL for the sas-environment.xml file was configured during deployment, create the SASHOME environment variable. When this environment variable is defined, the desktop client automatically checks the local configuration files for a configured sas-environment.xml file to use.

If you configure the logon behavior in both ways, then the value from the SAS_ENV_DEFINITION_LOCATION environment variable is used.

Here is how you can create either of these environment variables on a machine running Windows XP. If you are running another version of Windows, then these steps might be slightly different for your operating environment.

1. On your desktop, right-click **My Computer** and select **Properties**. The System Properties dialog box appears.
2. Click the **Advanced** tab and click **Environment Variables**. The Environment Variables dialog box appears.
3. Click **New** to create a new system variable.

Environment Variable to Create	Variable Type	Variable Value
SAS_ENV_DEFINITION_LOCATION	SAS_ENV_DEFINITION_LOCATION	The location of the sas-environment.xml file. An example of this path is <code>http://server-name:port/sas/sas-environment.xml</code> .

Environment Variable to Create	Variable Type	Variable Value
SASHOME	SASHOME	The location where SAS was installed. An example of this path is C:\Program Files\SASHome.

Click **OK** to close the New System Variable dialog box.

SAS_ENV_DEFINITION_LOCATION or SASHOME now appears in the list of system variables.

4. Click **OK**.

Note: When you open a new web browser, the process for that browser uses the cached values of the environment variables. Because Java Web Start is a child of this process, the Java Web Start client also uses these cached values. For Java Web Start to recognize the new environment variables (or any changes that you make to these environment variables), you must restart your web browser and then reopen the Java Web Start client.

Starting SAS Forecast Studio with Options

Specify the Number of Observations to Use to Detect the Time Interval

You can specify the number of observations from the input data set that SAS Forecast Studio uses to detect the time interval. By default, SAS Forecast Studio uses the first 10,000 observations to detect the time interval. Usually, a few distinct time ID values are needed to detect the interval. However, data sets with many repeated values of the time ID variable sometimes require a larger sample to get enough distinct values. Increasing the number of observations might improve time interval detection when you are using data sets that contain many repeated values. Values of 100,000 and larger could cause noticeably slower performance when you are creating a new project.

To configure SAS Forecast Studio to use a customized time interval value:

1. Open the forecaststdo.ini file in a text editor. For example, in a default Windows installation, this file is located in the C:\Program Files\SAS\SASForecastStudio\14.2 directory.
2. In the .ini file, enter a new JavaArgs_*n* entry, where *n* is the number of the next argument in the list. Increment the argument number accordingly. For example, enter the following: **JavaArgs_14=-Dforecasting.interval_sample=50000**

Specify the Sample Size for Validating BY Variable Values and Formats

When you start SAS Forecast Studio, you can choose an option that specifies a default value to use for checking the validity of BY variable values and formats. This option applies when you use the new project wizard to save a project's code without running it.

The default value for the sample size is 50000. When you specify a value for this property, the first n observations of the data set are used to check for the following:

- Is any numeric BY variable continuous-valued?
- Does any BY variable have a format that maps more than one raw value to the same formatted value?

If the answer to either question is yes, then an appropriate error dialog box is displayed, and the project is not created or saved.

To configure SAS Forecast Studio to use a sample size other than 50000 to validate BY variable values and formats:

1. Open the forecaststdo.ini file in a text editor. For example, in a default Windows installation, this file is located in the `C:\Program Files\SAS\SASForecastStudio\14.2` directory.
2. In the .ini file, enter a new `JavaArgs_n` entry, where n is the number of the next argument in the list. Increment the argument number accordingly. For example, enter the following: `JavaArgs_14=-Dforecasting.by_validation_sample=75000`

Enable SAS Forecast Studio Options in Java Web Start

You can enable SAS Forecast Studio options in the Java Web Start client as follows:

1. Start SAS Management Console and connect as a SAS Administrator (for example, `sasadm@saspw`).
2. On the **Plug-ins** tab, expand the **Application Management, Configuration Manager**, and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
4. Click the **Settings** tab.
5. Click **Forecast Studio (Java Web Start)**. If there are existing arguments for **Forecast Studio (Java Web Start)** in the Other arguments window, add a semicolon to the end of each argument. Then, append each parameter minus the `-D`. The `-D` is not required when setting the parameter for the Java Web Start client. If there are no existing arguments, then just add each parameter minus the `-D`.
 - The parameter to specify the number of observations to use to detect the time interval is as follows:
`forecasting.interval_sample=50000`
 - The parameter to specify the sample size for validating BY variable values and formats is as follows:
`forecasting.by_validation_sample=75000`

6. Click **OK**.
7. Restart SAS Forecast Server for these changes to take effect.

Adding Custom Time Intervals to SAS Forecast Studio

Create the Custom Time Interval

Although SAS Forecast Server includes a variety of time intervals, your site might use custom time intervals.

To create a custom interval:

1. Using a DATA step program or a SAS editor, create the data set that defines the custom interval. The data set must include the following information:
 - a BEGIN variable that specifies the date or datetime at which each period begins. If raw data contains date values (the number of days because January 1, 1960), use these date values for the BEGIN variable and assign a date format. If the raw data contains datetime values (the number of seconds because midnight January 1, 1960), use datetime values for the BEGIN variable and assign a datetime format.

You can also define the end of each period (by using an END variable) and the seasonal cycle (by using a SEASON variable). The END variable must use the same format as the BEGIN variable. If you do not specify an END variable, then the implied value of END for each observation is one less than the value of the BEGIN variable at the next observation.

- The span of the custom interval data set must include any dates or times that are necessary for performing calculations on the time series, such as forecasting and any other operations that might extend beyond the series (such as filters).

CAUTION:

Errors occur if the date or datetime values (in the actual data, in forecasts, or in plots) are outside the range of the custom interval definition. For your custom interval, the values for the BEGIN variable must start with the earliest historical data and extend into the future for at least three forecasting horizons. For example, if the historical data includes all of the business days for 2011 and the forecasts are for the first 12 business days in 2012, the values of the BEGIN variable must range from the first business day of 2011 through the 36th business day of 2012.

2. To define the name and location of your custom interval, specify the INTERVALDS= system option in either of these files:
 - in the config file (`sasv9_usermods.cfg` or `sasv9.cfg`) that is used by the SAS Workspace Server.
 - in the autoexec file (for example, `appserver_autoexec_usermods.sas`) that is used by the SAS Workspace Server.

In SAS Forecast Server, the syntax for the INTERVALDS= system option is more restrictive than the syntax provided in the [SAS System Options: Reference](#). Here is the syntax:

```
INTERVALDS=(
interval-1=libref.dataset-name-1 <interval-n=libref.dataset-name-n>)
```

The name for the *interval* must be 1–32 characters. The name cannot contain underscores or embedded numeric characters, except that the name can end with one or more digits. For example, StoreHours12 is a valid name, and Store12Hours is an invalid name. When you specify multiple intervals, each interval name must be unique. The value of the interval is the data set that is named in *libref.dataset-name*.

libref.dataset-name specifies the libref and data set name of the file that contains the custom interval.

In this example, the StoreHours interval is associated with the StoreHoursDS data set.

```
options intervalds=(StoreHours=CustIntLib.StoreHoursDS)
```

After completing these steps, the custom interval should appear in the list of available time intervals the next time you start SAS Forecast Studio. For example, you can select a time interval when creating a new project in the New Project wizard. If a custom interval is not available, check the SAS log for errors in the definition of the custom interval.

Example 1: StoreHours Custom Interval

This example shows how to set up a custom interval for data that is recorded hourly during the hours of 9AM to 6PM Monday through Friday and 9AM to 1PM on Saturday.

Note: This example specifies the INTERVALDS system option in a configuration file.

The syntax when you specify this option in a configuration file is different from when you specify it in the OPTIONS statement. In this example, in a Windows environment, INTERVALDS is preceded by a hyphen. In addition, the option name is followed by a space rather than by an equal sign.

1. Log on to the SAS Workspace Server. Use a text editor to add the following code to the C:\SAS\Config\Lev1\SASApp\sasv9_usermods.cfg file:

```
-intervalds (StoreHours=CustIntLib.StoreHoursDS)
```

2. Start a SAS session on the workspace server. In this SAS session, complete these steps:

- a. Assign the LIBNAME CustIntLib to the directory where the data set for the custom interval (in this example, StoreHoursDS) is stored.
- b. Submit the following code to create the data set:

```
data CustIntLib.StoreHoursDS (keep=BEGIN END) ; 1
  start = '01JAN2012'D;
  stop  = '31DEC2012'D;
  do date = start to stop; 2
    dow = WEEKDAY(date);
    datetime=dhms(date,0,0,0);
    if dow not in (1,7) then
      do hour = 9 to 17;
        begin=intnx('hour',datetime,hour,'b');
        end=intnx('hour',datetime,hour,'e');
        output;
      end;
    else if dow = 7 then
      do hour = 9 to 13;
        begin=intnx('hour',datetime,hour,'b');
```

```

end=intnx('hour',datetime,hour,'e');
output;
end;
end;
format BEGIN END DATETIME.;3
run;

```

- 1 In the DATA statement, the name StoreHoursDS is specified for the data set. The KEEP= option specifies that only the BEGIN and END variables should be included in the data set.

The START= and STOP= options specify the date of the first and last observations in the data set.

- 2 The DO loop analyzes each observation from 01JAN2012 through 31DEC2012 (inclusive).
 - Use the WEEKDAY function to determine the day of the week for a specific date. By default, the days of the week are specified as 1 for Sunday, 2 for Monday, and so on. You need this information later in the DO loop to assign the correct store hours to the day of the week. For example, if the day of the week is not 1 (Sunday) or 7 (Saturday), then the store hours are 9 to 17. If the day of the week is 7 (Saturday), then the store hours are 9 to 13.
 - Use the DHMS function to return a SAS datetime value for the date.
 - Use the INTNX function to return the datetime value of the beginning of the interval that is *n* intervals from the interval that contains the given datetime value.
- 3 Finally, the FORMAT statement specifies a DATETIME. format for the values of the BEGIN and END variables.

Example 2: Creating a Custom Interval Definition Interactively

To create a custom interval definition interactively:

1. Start a SAS session on the SAS Workspace Server.
 2. Click the New Library icon in the toolbar. The New Library dialog box appears.
 3. Enter the LIBNAME that you used when you specified the INTERVALDS system option.
 4. Select the path where the custom interval is to be stored. Click **OK** to close the Select and New Library dialog boxes.
 5. In the SAS Explorer, right-click the library that you just created and click **New**. The New Member dialog box appears.
 6. Select **Table** and click **OK**. An empty table opens in Viewtable.
 7. Right-click the **A** at the type of column A and select **Column Properties**.
 8. Specify these properties:
 - Change the name from **A** to **BEGIN**.
 - Select **Numeric** as the type.
 - Specify **DATE9 .** as the format and informat.
- Click **Close**.

9. Click the first cell in the BEGIN column to edit that cell. Type **01JAN2011** and press Enter.
10. In the second cell, enter the date of the next period. Continue entering dates as needed. Then close the Viewtable window. When prompted, click **Yes** to save your changes. In the Save As window, select the library that you assigned in step 3.
11. For the member name, enter the name of the data set that you specified in the INTERVALDS system option.
12. Click **Save** to save the data set.

Additional Considerations

If you are using the SAS Forecast Server Batch Interface, you must run the macro code on a SAS Workspace Server if your project meets the following criteria:

- You are using the %FSCREATE or %FSRUNPRJ macros.
- You are using a custom time interval or custom format.
- The macros depend on LIBNAME statements in the autoexec file in order to run.

Adding a Custom Format to SAS Forecast Studio

Create a Custom Format

Although SAS Forecast Server includes a variety of formats, you might have additional formats available at your site. Using the FORMAT procedure, you create custom formats and then make these formats available in SAS Forecast Studio. For more information about this procedure, see [Base SAS Procedures Guide](#).

To create a customized format:

1. In PROC FORMAT, specify where to store the format by using the LIBRARY= option.

- To store the format in a library or catalog in the default location, write the following SAS code:

```
libname library
config-dir/Levn/SASApp/SASEnvironment/SASFormats;
proc format library=library;
...
```

- To store the user-defined formats in the **C:\myfmts** location on the SAS server, write the following SAS code:

```
libname library "c:\myfmts";
proc format library=library;
...
```

By default, the catalog name for this library is **Formats** (unless you changed the default value to a different catalog name).

2. Use the VALUE statement to, define the format. For example, the following code defines the \$LINE format. The special code **Line1**, **Line2**, and so on, are converted to the corresponding product line.

```
value $ line
'Line1'='Product Line1'
'Line2'='Product Line2'
'Line3'='Product Line3'
'Line4'='Product Line4'
'Line5'='Product Line5';
```

3. Define the format library for the SAS Workspace Server that is used by SAS Forecast Server.
 - If the customized format is stored in the default library (**SAS-configuration-directory/Lev1/SASApp/SASEnvironment/SASFormats**), then no further modifications are required.
 - If the customized format is in a different location, you must configure SAS to search the **C:\myfmts** library in addition to the default library.

To search a different format library:

1. Open the configuration file in the following default location:

Windows

```
SAS-configuration-directory\Levn\SASApp
\sasv9_usermods.cfg
```

UNIX

```
SAS-configuration-directory/Levn/SASApp/
sasv9_usermods.cfg
```

2. In the sasv9_usermods.cfg file, use the SET= system option to define the library definition. In the following code, the SET= option associates the FSFMTS libref with the **C:\myfmts** directory. Then set the FMTSEARCH= system option to FSFMTS, so that SAS Forecast Server searches this library.

```
-set FSFMTS ("c:\myfmts")
-fmtsearch (FSFMTS)
```

When you restart the SAS Workspace Server, the system resolves any references to custom formats that are stored in **C:\myfmts**.

Create a Custom Format in Your Autoexec File

Another way that you can define a custom format is in your autoexec file.

To create a custom format in the autoexec file:

1. Open the autoexec file in the following default location:

Windows

```
SAS-configuration-directory\Levn\SASApp
\appserver_autoexec_usermods.sas
```

UNIX

```
SAS-configuration-directory/Levn/SASApp/  
appserver_autoexec_usermods.sas
```

2. Add the SAS code to define and create the format.

Here is an example:

```
/* First, define the library where format will be placed */  
libname test "C:\temp";  
  
/* Use the OPTIONS statement to specify that SAS Forecast Server needs to look  
in library=test for the format */  
options fmtsearch = (test);  
  
/* Use PROC FORMAT to create the format. */  
proc format library=test;  
  value $ Region  
    'Region1' = 'RegionOne'  
    'Region2' = 'RegionTwo'  
    'Region3' = 'RegionThree'  
    'Region4' = 'RegionFour';  
run;  
  
/* Use a DATA step to create the formatted data set. This data set does not  
need to be in the same library as the format. */  
data test.pricedata_format;  
  set sashelp.pricedata;  
  format regionName $ Region.;  
run;
```

3. Save the changes to the autoexec file.

Additional Considerations

If you are using the SAS Forecast Server Batch Interface, you must run the macro code on a SAS Workspace Server if your project meets the following criteria:

- You are using the %FSCREATE or %FSRUNPRJ macros.
- You are using a custom time interval or custom format.
- The macros depend on LIBNAME statements in the autoexec file in order to run.

Creating a Customized List of Events

Create a Customized Lists of Events

SAS Forecast Studio is shipped with a list of predefined events. However, you might need to add to this list of events or create a customized list of events for your company. To define an event, you must create the event using the HPFEVENTS procedure and then use the DATEKEYS procedure to associate a date key, label, and locale with this

event. The results are output to a data set. The date key forms the name of the event. The EVENTDS system option is defined (preferably in the SAS configuration file) to point to one or more of these event data sets, so that this list of events appears in SAS Forecast Studio.

To create a customized list of events:

1. Using a DATA step program or a SAS editor, create the data set that defines the events.
2. To define the name and location of your event data set, specify the EVENTDS= system option in the configuration file (**sasv9_usermods.cfg** or **sasv9.cfg**) that is used by the SAS Workspace Server.

After completing these steps, the customized list of events should appear in the list of available time intervals the next time you start SAS Forecast Studio. For example, you can select a time interval when creating a new project in the New Project wizard. If a custom interval is not available, check the SAS log for errors in the definition of the custom interval.

Example: Super Bowl Events

1. Start a SAS session on the workspace server, and run the following code:

```
proc datekeys;
  datekeydef SuperBowl =          '15JAN1967'D '14JAN1968'D '12JAN1969'D '11JAN1970'D
                                '17JAN1971'D '16JAN1972'D '14JAN1973'D '13JAN1974'D '12JAN1975'D
                                '18JAN1976'D '09JAN1977'D '15JAN1978'D '21JAN1979'D '20JAN1980'D
                                '25JAN1981'D '24JAN1982'D '30JAN1983'D '22JAN1984'D '20JAN1985'D
                                '26JAN1986'D '25JAN1987'D '31JAN1988'D '22JAN1989'D '28JAN1990'D
                                '27JAN1991'D '26JAN1992'D '31JAN1993'D '30JAN1994'D '29JAN1995'D
                                '28JAN1996'D '26JAN1997'D '25JAN1998'D '31JAN1999'D '30JAN2000'D
                                '28JAN2001'D '03FEB2002'D '26JAN2003'D '01FEB2004'D '06FEB2005'D
                                '05FEB2006'D '04FEB2007'D '03FEB2008'D '01FEB2009'D '07FEB2010'D
                                '06FEB2011'D '05FEB2012'D '03FEB2013'D '02FEB2014'D
  / PULSE=DAY locale= 'en_US'
    label="Super Bowl Sunday";
  datekeykey Christmas / locale= 'en_US'
    label="Christmas Day";
  datekeykey Thanksgiving / locale= 'en_US'
    label="Thanksgiving Day in US";
  datekeykey TurkeyDay=N4W5NOVYR / locale= 'en_US'
    label="Alternate Name for Thanksgiving";
  datekeydata out=holiday condense;
run;
proc print data=holiday;
run;

options eventds=(holiday);

title 'LIST option gives a list of available datekeys- with defaults';
proc datekeys;
  datekeydata out=holidaylist LIST;
run;
proc print data=holidaylist;
```



```

run;

title 'LIST option gives a list of available datekeys - without defaults';
proc datekeys;
    datekeydata out=holidaylist LIST NODEFAULTS;
run;
proc print data=holidaylist;
run;

data ts(keep=date);
    do i=1 to 120;
        date=INTNX('MONTH','01JAN2000'D,i-1);
        output;
    end;
    format date DATE.;
run;
title 'Using system option custom datekeys are allowed';
proc hpfevents;
    id date interval=month;
    eventkey SuperBowl;
    eventkey Xmas=Christmas;
    eventdata out=myevents condense;
    eventdummy out=mydummies;
run;
proc print data=myevents;
run;
proc print data=mydummies;
run;

```

2. Open the configuration file in the following default location:

Windows

***SAS-configuration-directory\Levn\SASApp
 \sasv9_usermods.cfg***

UNIX

***SAS-configuration-directory/Levn/SASApp/
 sasv9_usermods.cfg***

3. In the sasv9_usermods.cfg file, use the EVENTSDS= system option to point to the event data sets.

EVENTSDS= ("c:\eventslist")

Chapter 11

SAS Forecast Batch Interface

About the SAS Forecast Server Batch Interface	106
Disclaimer and Warning	106
Notes and Restrictions	106
Overview	106
Installation and Availability	107
General Usage	107
Working with SAS Forecast Server	108
Working with SAS Forecasting for Desktop	109
Encrypting Your Passwords	109
Using Special Characters	110
Summary of the Macros	110
Dictionary	112
%FSADDEVT Macro	112
%FSCLEAR Macro	116
%FSCOPY Macro	116
%FSCREATE Macro	118
%FSDELARC Macro	132
%FSDELENV Macro	134
%FSDELEVT Macro	134
%FSDELPRJ Macro	136
%FSEVTREQ Macro	137
%FSEXPALL Macro	138
%FSEXPOR Macro	140
%FSEXPSET Macro	141
%FSGETENV Macro	142
%FSGETPRJ Macro	144
%FSGETURP Macro	145
%FSIMPALL Macro	146
%FSIMPORT Macro	147
%FSLOAD Macro	148
%FSLOGIN Macro	149
%FSLOGOUT Macro	151
%FSLOOP Macro	151
%FSMIGALL Macro	153
%FSMIGPRJ Macro	154
%FSMOVE Macro	155
%FSNEWENV Macro	157
%FSREGENV Macro	158
%FSPRJAGG Macro	159
%FSPUBLISH Macro	160
%FSREGPRJ Macro	161

%FSREN Macro	162
%FSRMOVR Macro	163
%FSRUNPRJ Macro	167
%FSRUNRPT Macro	168
%FSSETDAT Macro	170
%FSSETDEF Macro	172
%FSSETOVR Macro	173
%FSSETOWN Macro	176
%FSSETPUB Macro	177
%FSUNREG Macro	178
%FSUNRENV Macro	179
%FSUPDATE Macro	180
%FSVER Macro	192
Examples	193
Example 1: Migrating SAS Forecast Server 4.1 Projects to SAS Forecast Server 14.2	193
Example 2: In-Place Migration of SAS Forecast Server 4.1 Projects to SAS Forecast Server 14.2	195
Example 3: Create a SAS Forecast Server Project	195
Example 4: Archive a SAS Forecast Server Project	196
Example 5: Add Overrides to a SAS Forecast Server Hierarchical Project	197
Example 6: Remove Overrides from a SAS Forecast Server Hierarchical Project	198
Example 7: Add Overrides to a SAS Forecast Server Non-Hierarchical Project ..	199

About the SAS Forecast Server Batch Interface

Disclaimer and Warning

The macros described in this chapter are for internal use by SAS Forecast Server software only. These macros use undocumented aspects of SAS Forecast Server software that could change or be removed in future releases without notice.

Modification or use of these macros for anything other than their intended purpose is not allowed. SAS Institute will not support user-written SAS programs that rely on any such use of these macros or on the undocumented aspects of SAS Forecast Server that these macros invoke.

Notes and Restrictions

Using macros in the SAS Forecast Server Batch Interface is not supported when the server is in a locked-down state. For more information about the LOCKDOWN statement, see *SAS Intelligence Platform: Security Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform/index.html>.

Overview

The SAS Forecast Server Batch Interface is available for the SAS Forecast Server and SAS Forecasting for Desktop. Macros in the SAS Forecast Server Batch Interface provide a SAS language interface to the SAS Forecast Server and SAS Forecasting for Desktop. Effectively, the macros serve as an alternative client to SAS Forecast Studio, SAS Forecast Studio for Desktop, the SAS Forecast Server Plug-ins for SAS Management Console, SAS Forecast Project Manager, and SAS Forecast Project

Manager for Desktop. Some macros enable you to create projects with options identical to those available in SAS Forecast Studio and SAS Forecast Studio for Desktop. However, most of the macros focus on the management of environments and projects.

Installation and Availability

When your site licenses the SAS Forecast Server or SAS Forecasting for Desktop, the SAS Forecast Server Batch Interface is automatically included in all SAS Foundation installations. The SAS Forecast Server Batch Interface component is installed in the extension directory forecastbat. On Windows, the macros in the SAS Forecast Server Batch Interface reside in an autocall library in SASROOT/forecastbat/sasmacro. On UNIX, they reside in the SASFoundation/9.4/sasautos directory.

General Usage

Whether you are working with the SAS Forecast Server or SAS Forecasting for Desktop, the basic pattern of usage is the same. Here are the general usage steps:

1. Call the %FSLOGIN macro to begin a product session.
2. Call one or more of the other SAS Forecast Server Batch Interface macros to perform actions in the product session.
3. Call the %FSLOGOUT macro to end the product session.

For example, the %FSGETENV macro can be used with the PRINT=YES argument to retrieve the list of available forecasting environments and print them to the SAS LISTING output. To make this call, the code would be similar to the following:

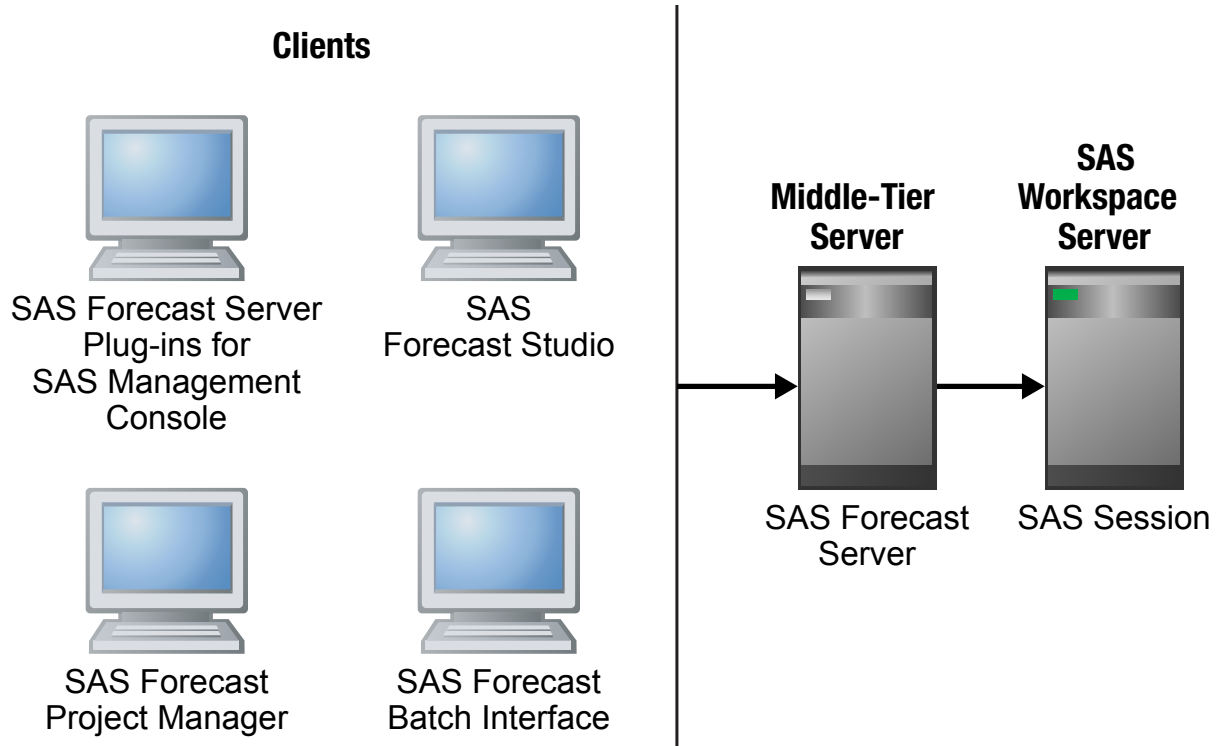
```
%FSLOGIN(arguments)
%FSGETENV(print=YES)
%FSLOGOUT()
```

In this example, the arguments for %FSLOGIN are intentionally omitted because they depend on the type of deployment. There is more information about these arguments in the subsections that follow.

Note: Prior to release 12.1, each SAS Forecast Server Batch Interface macro created and ended a dedicated product session during the course of its call. This required %FSLOGIN information to be included in the arguments for each macro. As of release 12.1, these macro arguments are no longer supported. SAS scripts that were created for releases prior to 12.1 must be updated to use %FSLOGIN and %FSLOGOUT macros instead.

Working with SAS Forecast Server

The following figure shows how the SAS Forecast Server Batch Interface macros interact with the other components of SAS Forecast Server.

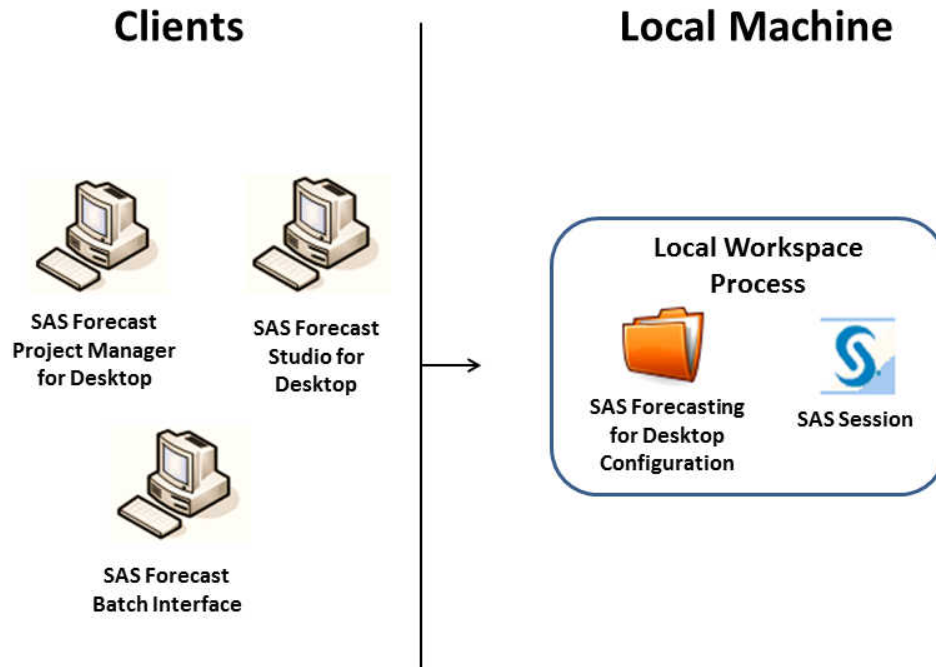


As with the other SAS Forecast Server clients, the SAS Forecast Server Batch Interface uses SAS environments to identify the SAS deployments that are available. As a result, the same configuration requirements apply to the SAS Forecast Server Batch Interface. SAS Forecast Server Batch Interface macros must have access to a `sas-environment.xml` file that defines the list of SAS environments or an error message is displayed. For the login macro, `%FSLOGIN`, the `SASENVIROMENT` argument is used to identify the selected SAS environment by name. As a result, the value for this argument is the symbolic name of the SAS environment. This symbolic name is specified by the `NAME=` attribute in the `sas-environment.xml` file. The symbolic name is used by the clients that do not have a graphical user interface because a symbolic name is independent of your locale. (Do not use the `DESC=` attribute to specify the SAS environment. The `DESC=` attribute is the label that appears when the SAS environment is displayed in a list, but is a locale-sensitive value.)

Note: `%FSLOGIN` is the only macro that refers to SAS environments. All other environments in the SAS Forecast Server Batch Interface are forecasting environments.

Working with SAS Forecasting for Desktop

The following figure shows how the SAS Forecast Server Batch Interface macros interact with the other components of SAS Forecasting for Desktop.



Because SAS Forecasting for Desktop can use your operating system identity directly, creating a product session is trivial. The %FSLOGIN call needs to indicate that only a desktop session is requested, which is done by passing the DESKTOP=YES argument. Here is the complete code for the previous example for SAS Forecasting for Desktop:

```
%FSLOGIN(desktop=YES)
%FSGETENV(print=YES)
%FSLOGOUT()
```

Encrypting Your Passwords

It is not recommended that you use plain text passwords in the code for your macro variables. Instead, use PROC PWENCODE to obtain an encrypted password.

For example, to encrypt the password "SASmetal," submit the following code in the SAS Program Editor:

```
proc pwencode in="SASmetal";
run;
```

Copy the encrypted password from the SAS log and paste this password into your macro code.

For more information about the PWENCODE procedure, see the *Base SAS Procedures Guide*.

Using Special Characters

Special characters, such as the apostrophe, are not allowed when you specify a directory path for a parameter. For example, if you specify an environment path of `C:\sas\Forecast Directory\14.2\Test` for the %FSNEWENV macro, the environment cannot be created, and an error message appears.

Summary of the Macros

Note: The SAS Forecasting for Desktop license does not permit the use of the %FSCREATE macro. Instead, projects must be created using the SAS Forecast Studio for Desktop client. In addition, the %FSRUNRPT macro does not apply to SAS Forecasting for Desktop deployments because stored processes are not available in desktop mode.

The following table lists the SAS Forecast Server macros.

Macro Name	Description
%FSADDEVT	Defines an event and adds it to the event repository for the specified project.
%FSCLEAR	Clears project information currently stored in global macro variables.
%FSCOPY	Copies a SAS Forecast Server project to a new destination.
%FSCREATE	Creates a new SAS Forecast Server project in batch mode. <i>Note:</i> If you license SAS Forecasting for Desktop, the %FSCREATE macro is not available. It is available only if you license SAS Forecast Server. For more information about SAS Forecast Studio for Desktop, see the <i>SAS Forecast Studio: User's Guide</i> .
%FSDELARC	Deletes an archived SAS Forecast Server project.
%FSDELENV	Deletes an existing product environment.
%FSDELEVT	Deletes specified events from the event repository.
%FSDELPRJ	Deletes an existing SAS Forecast Server project.
%FSEVTREQ	Sets the required attributes on events in the project's event repository.
%FSEXPALL	Exports all SAS Forecast Server projects to archive files.
%FSEXPOR	Exports a single SAS Forecast Server project to an archive file.
%FSEXPSET	Exports the project settings to a local file.

Macro Name	Description
%FSGETENV	Retrieves the metadata about the product environments.
%FSGETPRJ	Retrieves the metadata about the SAS Forecast Server projects.
%FSGETURP	Creates a log file that lists the names of any unregistered projects in a specified environment.
%FSIMPALL	Imports all SAS Forecast Server projects listed in a data set from archived files.
%FSIMPORT	Imports a SAS Forecast Server project from an archived file.
%FSLOAD	Opens an existing SAS Forecast Server project, and loads global macro variables that describe the project.
%FSLOGIN	Enables you to create a new session for a specific instance of the middle tier. This option is supported both by SAS Forecast Server and by SAS Forecasting for the Desktop.
%FSLOGOUT	Closes a session on the specified instance of the middle tier.
%FSLOOP	Enables you to iterate through a given data set and invokes a callback macro with the values from each row.
%FSMIGALL	Migrates all existing SAS Forecast Server projects to the current version of SAS Forecast Server.
%FSMIGPRJ	Migrates an existing SAS Forecast Server project to the current version of SAS Forecast Server.
%FSMOVE	Moves a SAS Forecast Server project to a new destination.
%FSNEWENV	Creates a new SAS Forecast Server environment.
%FSRJAGG	Opens existing SAS Forecast Server projects and aggregates the top level reconciled forecasts.
%FSPUBLSH	Opens an existing SAS Forecast Server project, loads global macro variables, which describe the project, and publishes the project to a predefined SAS library
%FSREGENV	Registers an existing directory structure as an environment, registering all projects found within.
%FSREGPRJ	Registers a project in metadata.
%FSREN	Renames a single SAS Forecast Server project.
%FSRMOVR	Removes ALL overrides for all series specified in the SERIESDS= data set.

Macro Name	Description
%FSRUNPRJ	Opens an existing SAS Forecast Server project, and runs the project at a given stage.
%FSRUNRPT	Runs the specified forecasting report and saves the results to a local directory.
%FSSETDAT	Changes the input data source for a project.
%FSSETDEF	Stores a default value for an argument.
%FSSETOVR	Adds the overrides specified in the OVRDS= data set to a project.
%FSSETOWN	Assign the owner of a project.
%FSSETPUB	Determines whether a SAS Forecast Server project should allow public (shared) access to all users of its product environment .
%FSUNREG	Unregisters an existing SAS Forecast Server project from the metadata server.
%FSUNRENV	Unregisters an environment.
%FSUPDATE	Updates the settings of an existing SAS Forecast Studio project.
%FSVER	Prints the version of SAS Forecast Server to the log and initializes a macro variable that contains this value.

The %FSCREATE and %FSRUNRPT macros are not applicable to SAS Forecasting for Desktop.

Note: The SAS macros are automatically installed with SAS Forecast Server. For example, on Windows, the default installation location is the **C:\Program Files\SAS\SASFoundation\9.4\forecastbat\sasmacro** directory.

Dictionary

%FSADDEVT Macro

The %FSADDEVT macro defines an event and adds it to the event repository for the specified project. All types of events and their attributes can be defined.

Syntax

```
%FSADDEVT (EVENTNAME=,EVENTTYPE= [,options])
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSADDEVT macro if no default values are currently stored. The required arguments are separated by commas.

EVENTNAME= *event-name*

specifies the name of the event to be created.

EVENTTYPE= POINT | RAMP | LEVELSHIFT | TEMPORARYCHANGE | COMBINED

specifies the type of event. For more information about each type of event, see the *SAS Forecast Studio: User's Guide*.

Note: POINT events are called pulse events in SAS Forecast Studio.

Options

The following options can be used with the %FSADDEVT macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

EVENTCHANGEPARAMETER = *value*

specifies the change for a temporary change event. This option does not have any effect on other event types. Valid values range from 0 to 1 (inclusive). The default value is 0.5.

EVENTCOMBINATION=*event-name1 event-name2*

specifies the events to include in the combined event. This option is valid only when the event type is COMBINED. Event names are delimited by a space.

EVENTCOMBINATIONRULE= ADD | MAX | MIN | MINNZ | MINMAG | MULT
specifies which combination rule to use when the defined event has several values that overlap in the same time period. Here is a description of the valid values:

ADD	add the overlapping values
MAX	use the maximum value
MINNZ	use the minimum nonzero value
MINMAG	use the value with the least magnitude
MULT	multiply the overlapping values

EVENTDESCRIPTION=*text*

specifies a description of the event.

EVENTDIFF = YES | TRUE | NO | FALSE | 1 | 0

specifies whether to use the same differencing value that is used in the model. This option applies only to ARIMA models. The default value is **YES**.

EVENTOCCURRENCESAFTER = *value*

specifies the duration of the component that occurs after the event. For example, EVENTOCCURRENCESAFTER=4 specifies that the event ends four periods after the time at which you specified for the event. If you specified more than one date or time for the event, the EVENTOCCURRENCESAFTER option applies to all of them.

EVENTOCCURRENCESBEFORE = *value*

specifies the duration of the component that occurs before the event. For example, EVENTOCCURRENCESBEFORE=4 specifies that the event starts four periods before the date and time that you specified for the event. If you specified more than one date or time for the event, the EVENTOCCURRENCESBEFORE option applies to all of them.

EVENTOCCURRENCESAFTERGROWTH = YES | TRUE | NO | FALSE | 1 | 0

specifies the slope for the component that occurs after the event. This option is valid only if the event type is RAMP or TEMPORARYCHANGE. The YES | TRUE values indicate growth, and the NO | FALSE values indicate decay.

EVENTOCCURRENCESBEFOREGROWTH = YES | TRUE | NO | FALSE | 1 | 0

specifies the slope for the component that occurs before the event. This option is valid only if the event type is RAMP or TEMPORARYCHANGE. The YES | TRUE values indicate growth, and the NO | FALSE values indicate decay.

EVENTPERIODINTERVAL = *time-interval*

specifies the time interval by which the event repeats infinitely.

EVENTPERIODICFROMSTART = *date-time-value*

specifies the starting point for the recurrence. The interval is specified by the EVENTPERIODINTERVAL option.

EVENTPERIODICFROMSTARTINTERVAL = *time-interval*

specifies the time interval for an event that repeats infinitely. The starting point is specified by the EVENTPERIODICFROMSTART option.

EVENTPERIODICITYEND = *date-time-value*

specifies the end date or time for the periodic interval. When you specify the EVENTPERIODICITYSTART, EVENTRANGEINTERVAL, and EVENTPERIODICITYEND options, then the periodicity for the recurrence is finite.

EVENTPERIODICITYOCCURENCES = *integer-value*

specifies the number of times that the event recurs. The recurrence starts at the date or time specified by the value of the EVENTPERIODICITYSTART option and occurs at the interval specified by the value of the EVENTRANGEINTERVAL option. If you specify this option and the EVENTPERIODICITYEND option, the value of the EVENTPERIODICITYEND option is ignored.

EVENTPERIODICITYSTART = *date-time-value*

specifies the start date or time for the periodic interval. When you specify the EVENTPERIODICITYSTART, EVENTRANGEINTERVAL, and EVENTPERIODICITYEND options, then the periodicity for the recurrence is finite.

EVENTRANGEINTERVAL = *time-interval*

specifies the time interval at which the event repeats. The start and end dates are specified by the EVENTPERIODICITYSTART and EVENTPERIODICITYEND options.

EVENTREQUIRED = NO | MAYBE | YES | UNDEF

specifies whether the event should be used when SAS Forecast Server automatically fits a model. The default value is **NO**. Here are the descriptions of the valid values:

NO	specifies that the events be included in the model as long as the event parameters are significant and the increment of the value of criterion exceeds the specified threshold.
MAYBE	specifies that the events be included in the model as long as the event parameters are significant.
YES	specifies that the events be included in the model as long as the model can be diagnosed.
UNDEF	specifies that the events not be included in the model.

EVENTTIMINGSHIFT = *integer-value*

specifies how long to delay the onset of the event. This delay is calculated from the start date or time of the event. Negative values can be used to specify an onset before the start date or time.

EVENTTIMINGVALUES = *value1 value2 . . .*

specifies the dates and times of the event. Values must be separated by a space. Here are the valid values:

- Dates in the ddmmmyyyy format (for example, 01JAN2012)
- Date-time values in the ddmmmyyyy:hh:mm:ss format (for example, 01JAN2012:04:21:15)
- Event keywords, including holiday names and seasonal events. For a list of these keywords, see the “Event Definitions” topic in the HPFEVENTS procedure chapter in the *SAS High-Performance Forecasting: User’s Guide*.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on [page 170](#).

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSADDEVT global macro variable indicates whether a new event was created or whether it failed due to errors.

Example

```
%fsaddevt (projectname=Shoe_Sales_2012,
           environment=Default,
           eventname=evt2,
           eventtype=ramp,
           eventdescription=sample event,
           eventTimingValues=01MAR1955,
           eventRequired=YES
           )
```

%FSCLEAR Macro

The %FSCLEAR macro clears project information currently stored in global macro variables. Use the %FSLOAD macro to add project information to a global macro variable.

Syntax

```
%FSCLEAR ()
```

Example

```
%fsclear()
```

%FSCOPY Macro

The %FSCOPY macro copies a SAS Forecast Server project to a new destination. Both destinations must run the same version or a later version of SAS Forecast Server.

Syntax

```
%FSCOPY (SOURCEPROJECTNAME=, DESTINATIONPROJECTNAME=,  
SOURCEENVIRONMENT=, DESTINATIONENVIRONMENT= [ , options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

You must specify either a source environment and a destination environment or a source project and a destination project. You cannot specify both an environment and a project at the same time. Required arguments are separated by commas.

SOURCEPROJECTNAME= *source-project-name*

specifies the name of the SAS Forecast Server project on the source host. The name must be a valid SAS name. If you do not specify this option, then the default value for the project is used. A global default value for the project name can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

DESTINATIONPROJECTNAME= *destination-project-name*

specifies the name of the SAS Forecast Server project on the destination host. The name must be a valid SAS name. If you do not specify this option, then the default value for the project is used. A global default value for the project name can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

SOURCEENVIRONMENT= *environment-name*

specifies the name of the SAS Forecast Server environment to be used as source.

This option is required only if there are multiple environments available to the user.

If omitted, the current stored default value is used. The default value is **Default**. A global default value for environment can be specified with the %FSSETDEF macro.

For more information, see “%FSSETDEF Macro” on page 172.

DESTINATIONENVIRONMENT= *environment-name*

specifies the name of the SAS Forecast Server environment to be used as destination.

This option is required only if there are multiple environments available to the user.

If omitted, the current stored default value is used. The default value is **Default**. A global default value for environment can be specified with the %FSSETDEF macro.

For more information, see “%FSSETDEF Macro” on page 172.

Options

The following options can be used with the %FSCOPY macro. Options must follow the required arguments and are separated by commas.

ARCHIVEFOLDER= *directory-name*

specifies the directory where to save the archive. It is recommended that you specify a value for this directory. The ARCHIVEFOLDER= and

REMOTEARCHIVEFOLDER= options are closely related.

- If you specify the ARCHIVEFOLDER= option but you do not specify a value for the REMOTEARCHIVEFOLDER= option, SAS Forecast Server assumes that the directories for the archive folder and the remote archive folder are the same. For this to work, the source and destination environments must be using the same server. If these environments are on different servers, an error results.
- If you specify a value for the REMOTEARCHIVEFOLDER= option, you must specify a value for the ARCHIVEFOLDER= option.
- If you do not specify a value for the ARCHIVEFOLDER= or REMOTEARCHIVEFOLDER= option, both options are set to the Work directory of the source environment. This option is valid only when you are copying or moving a project from one destination to another on the same server. If the source environment and the destination environment are on different servers, an error results.

CPORT= TRUE | FALSE | YES | NO | 1 | 0

exports data sets and catalogs using CPORT. You must specify this option if the source and destination environments are running on different operating systems. The default is CPORT=NO.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

REMOTEARCHIVEFOLDER= *directory-name*

specifies the directory where the archived projects of the source SAS Workspace Server can be found on the destination SAS Workspace Server (for example, \ **sourceserver\SAS\ForecastStudio\Archives**). If you specify a remote archive folder, you must specify a value for the ARCHIVEFOLDER= option. For more information about how the REMOTEARCHIVEFOLDER= and ARCHIVEFOLDER= options work together, see the description for the ARCHIVEFOLDER= option.

Note: The user must have Read and Write privileges to the REMOTEARCHIVEFOLDER= directory.

TMPARCHIVENAME= *project-name*
 specifies the name of the temporary archived project. The default is _fs_tmp_archive. The name must not match an existing archive.

Results

The %FSCOPY global macro variable indicates whether the %FSCOPY macro terminates successfully or encounters errors: &FSCOPY= SUCCESS | ERROR.

Example

```
%fscopy(sourceprojectname=Shoe_Sales,
destinationprojectname=Shoe_Sales_2012,
sourceEnvironment=Default,
destinationEnvironment=Default,
)
```

%FSCREATE Macro

The %FSCREATE macro creates a new SAS Forecast Server project in batch mode.

Note: If you license SAS Forecasting for Desktop, the %FSCREATE macro is not available. It is available only if you license SAS Forecast Server. For more information about SAS Forecasting for Desktop, see the *SAS Forecast Studio: User's Guide*.

Syntax

```
%FSCREATE (PROJECTNAME=, DATA=, ID=, VAR= [ , options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSCREATE macro if no default values are currently stored. The required arguments are separated by commas.

PROJECTNAME=*project-name*
 specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this argument, then the default value for the project name is used. The value of this default cannot be customized with the %FSSETDEF macro.

DATA= *SAS-data-set*
 specifies the name of the input SAS data set. The value that you enter for this argument is not case sensitive.

ID=*variable*
specifies the time ID variable

VAR= *variable*
specifies one or more dependent variables. Multiple variable names can be specified only if HIERARCHY= NO | FALSE. Multiple variable names are separated by a space.

Options

The following options can be used with the %FSCREATE macro. Options must follow the required arguments, and are separated by commas.

ACCUMULATE= TYPE [TYPE(*var1var2*) TYPE(*var3var4*) ...]
specifies the accumulation options for the dependent, input, and reporting variables.

Here are the valid values for TYPE:

NONE	specifies that no accumulation occurs; the ID variable values must be equally spaced with respect to the frequency.
TOTAL	specifies that observations are accumulated based on the total sum of their values. For independent variables, the default for the ACCUMULATE= and AGGREGATE= options is TOTAL when you create a project using the %FSCREATE macro. However, if you create a project using the SAS Forecast Studio interface, the default values for accumulation and aggregation for independent variables is Average of values .
AVERAGE AVG	specifies that observations are accumulated based on the average of their values.
MINIMUM MIN	specifies that observations are accumulated based on the minimum of their values.
MEDIAN MED	specifies that observations are accumulated based on the median of their values.
MAXIMUM MAX	specifies that observations are accumulated based on the maximum of their values.
N	specifies that observations are accumulated based on the number of nonmissing observations.
NMISS	specifies that observations are accumulated based on the number of missing observations.
NOBS	specifies that observations are accumulated based on the number of observations.
FIRST	specifies that observations are accumulated based on the first of their values.
LAST	specifies that observations are accumulated based on the last of their values.
STDDEV STD	specifies that observations are accumulated based on the standard deviation of their values.
CSS	specifies that observations are accumulated based on the corrected sum of squares of their values.

USS specifies that observations are accumulated based on the uncorrected sum of squares of their values.

Here are some examples:

- To use an average accumulation for all variables, specify **ACCUMULATE=AVG**.
- To use a total accumulation for all variables except for *var1*, which uses an average accumulation, specify **ACCUMULATE= TOTAL AVG(*var1*)**.
- To use a total accumulation for all variables except *var1* and *var2*, which use an average accumulation, and *var3*, which accumulates the observations based on standard deviations, specify **ACCUMULATE=TOTAL AVG(*var1 var2*) STD(*var3*)**.

ADJUST= *var1(var2var3)* [/operation=(pre,post)]

Specifies the adjustment options for the dependent variables. If not specified, the default value for pre-adjust and post-adjust is NONE.

Here are the valid pre-adjust and post-adjust values:

- NONE
- ADD
- SUBTRACT
- MULTIPLY
- DIVIDE
- MIN
- MAX

Here are some examples:

- To adjust the sale variable with the values of the price1 variable, specify **ADJUST=sale(price1)**.
- To adjust the sale variable with the values of the price1 variable and to adjust the price variable with the values of the price2 variable, specify **ADJUST=sale(price1) price(price2)**.
- To adjust the sale variable with the values of the price1 and price2 variables, specify **ADJUST=sale(price1 price2)**.
- To adjust the sale variable with the values of the price1 variable before any forecasts are generated, specify **ADJUST=sale(price1) / operation= (ADD, NONE)**.

AGGREGATE= TYPE [TYPE(*var1var2*) TYPE(*var3var4*) ...]

specifies the aggregation options for the dependent, input, and reporting variables. Aggregation is valid only if HIERARCHY=YES. See the ACCUMULATE option for valid values and examples.

Note: For the dependent variable, the only valid values are TOTAL | AVERAGE | AVG.

ALLOWNEGATIVE= YES | TRUE | NO | FALSE | 1 | 0

specifies whether negative forecasts are allowed. The default value is NO.

ALPHA= *n*

specifies the confidence level for the series. The default value is 0.05, which is a 95% confidence level.

ARIMAX= YES | TRUE | NO | FALSE | 1 | 0

specifies whether ARIMA models should be considered. The default value is YES.

BACK= *n*

specifies the out-of-sample range (the number of periods from the end). The default value is 0.

BASENAME= *model-name*

specifies the basename to be used for system-generated models. If not specified, then the default name is used by system for system-generated models. If an invalid SAS name is specified, then the setting is ignored and the default basename is used instead.

The default basename is generated automatically by SAS High-Performance Forecasting.

BY= *variable*

specifies the BY variables. Multiple variable names must be separated by a space.

COMBINE = YES | TRUE | NO | FALSE | 1 | 0

specifies whether SAS Forecast Server should automatically generate a combined model for the set of time series models that is generated from the diagnosis of each time series. The default value is **NO**.

COMBINECRITERION = *selection-criterion*

specifies the statistic of fit to use when ranking the combined models. The default value is **RMSE**.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMAX option, the UCM option, or both the ARIMAX and UCM options are set to YES or TRUE.

Here are the possible values for the selection criterion:

- AADJRSQ — Amemiya's adjusted R-squared
- ADJRSQ — adjusted R-squared
- AIC — Akaike information criterion
- AICC — Akaike information criterion, finite sample size corrected
- APC — Amemiya's prediction criterion
- GMAPE — geometric mean percent error
- GMAPES — geometric mean absolute error percent of standard deviation
- GMAPPE — geometric mean predictive error
- GMASPE — geometric mean symmetric percent error
- GMRAE — geometric mean relative absolute error
- MAE — mean absolute error
- MAPE — mean absolute percent error
- MAPES — mean absolute error percent of standard deviation
- MAPPE — mean absolute predictive symmetric percent error
- MASE — mean absolute scaled error
- MAXERR — maximum error

- MAXPE — maximum percent error
- MAXPPE — maximum predictive percent error
- MAXRE — maximum relative error
- MAXSPE — maximum symmetric percent error
- MDAPE — median absolute percent error
- MDAPES — median absolute percent error percent of standard deviation
- MDAPPE — median absolute predictive percent error
- MDASPE — median absolute symmetric percent error
- MDRAE — median relative absolute errors
- ME — mean error
- MINERR — minimum error
- MINPE — minimum percent error
- MINPPE — minimum predictive percent error
- MINRE — minimum relative error
- MINSPE — minimum symmetric percent error
- MPE — mean percent error
- MPPE — mean predictive percent error
- MRAE — mean relative absolute error
- MRE — mean relative error
- MSPE — mean symmetric percent error
- MSE — mean square error
- RMSE — root mean square error
- RSQUARE — R-squared
- RWRSQ — random walk R-squared
- SBC — Schwarz Bayesian information criterion
- SMAPE — mean absolute symmetric percent error
- SSE — sum of squared errors
- UMSE — unbiased mean square error
- URMSE — unbiased root mean square error

COMBINEENCOMPASS = OLS | HLN

specifies the forecast encompassing test to use to remove any models that contain redundant information. The default value is **OLS**.

Note: This option is valid only if the following conditions are met:

- the COMBINE option is set to YES or TRUE
- the ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE

Here are the valid values:

- OLS uses an ordinary least squares (OLS) regression test to estimate pairwise encompassing between candidate forecasts.
- HLN uses the Harvey-Leybourne-Newbold (HLN) test to estimate pairwise encompassing between candidate forecasts.

COMBINEMETHOD = AVERAGE | AICC | OLS | ERLS | LAD | NERLS | NRLS | RMSEWGT

specifies the method for determining the combination weights. These combination weights are used to calculate the weighted average of the candidate forecasts. The default method is **AVERAGE**.

Note: This option is valid only if the following conditions are met:

- the COMBINE option is set to YES or TRUE
- the ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE

Here are the valid values:

- AVERAGE computes the simple average of the forecasts that you selected for combination. This is the default.
- AICC uses the Akaike's information criterion to compute the combination weights based on corrected AIC weights.
- OLS uses ordinary least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast residuals.
- ERLS uses equally restricted least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast subject to the constraint that the weights sum to 1.
- LAD uses a measure of fit based on the least absolute deviation to compute the combination weights.
- NERLS uses nonnegative, equality restricted least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast residuals subject to the constraints that the weights sum to 1 and be nonnegative.
- NRLS uses nonnegative restricted least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast residuals subject to the constraints that the weights be nonnegative.
- RANKWGT assigns weights using the rank of the candidate forecasts when combination is performed. The weights must sum to 1. If not, the last value in the list is adjusted, and a warning is issued. The weights are assigned by ranking the candidate forecasts from best to worst. The best uses the first weight, and so on. The set of weights used is normalized to account for candidates that fail to forecast or for candidates that are omitted from the final combination because of any exclusion tests.
- RMSEWGT uses the weights of the root mean square error to compute the combination weights. The weights are normalized to sum to 1.

COMBINEMISSMODE = MISSING | RESCALE

specifies the method for treating missing values in the forecast combination. In a given time series across the combination ensemble, one or more combination contributors can have a missing value. This setting determines the treatment of those in the final combination for such time indices. The default value is **MISSING**.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMAX option, the UCM option, or both the ARIMAX and UCM options are set to YES or TRUE.

MISSING generates a missing combined forecast at each time index with one or more missing contributors. This method is the default for AICC weights, OLS weights, restricted least squares weights, and LAD weights. You cannot rescale the combination weights if you selected the OLS-based regression test to estimate pairwise encompassing. In this test, the estimated weights are not constrained to sum to one.

RESCALE rescales the combination weights for the nonmissing contributors at each time index to sum to 1. This method is the default for simple average, user-specified weights, ranked user weights, ranked weights, and root mean square error (RMSE) weights.

COMBINEMISSPCT = *integer*

specifies a threshold for the percentage of missing values in the combination estimation region. This threshold is used to exclude a candidate forecast from consideration in the final combination. By default, no missing percentage test is performed on candidate forecasts. Valid values are 1 to 100.

The COMBINEMISSPCT option and the HORMISSPERCENT option are independent of each other. You can specify one or both of these options.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMAX option, the UCM option, or both the ARIMAX and UCM options are set to YES or TRUE.

COMBINESTDERR = DIAG | ESTCORR

specifies the method for computing the prediction error variance series. This series is used to compute the prediction standard error, which in turn is used to compute confidence bands on the combined forecast. The default value is **DIAG**.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE.

CREATEOUTCOMPONENT= YES | NO

specifies whether to create the component data set. The default value is **YES**, and the data set is created. A component data set is created for each BY variable in the project. Because a component data set can also contain several other variables, these data sets could be quite large. As a result, a component data set could require additional disk space and computing time. If you do not want to create this data set, set CREATEOUTCOMPONENT=NO. If you suppress this data set, then you do not see the effects of events or outliers in the plot and data table in the Forecasting View.

CREATEOUTINDEP= YES | NO

specifies whether to create the OUTINDEP component data set. The default value is **NO**, and no data set is created. If the value is YES, then the data set is created.

CRITERION= *options*

specifies the model selection criterion (statistic of fit) to be used to select from several candidate models. The default value is MAPE. For more values, see the COMBINECRITERION option.

DESCRIPTION= *text*

specifies the project description.

DETECTOUTLIERS= YES | TRUE | NO | FALSE | 1 | 0

specifies whether outliers in the data should be detected when fitting an ARIMA model. The default value is **NO**.

DISAGGREGATION= PROPORTIONS | EQUALSPLIT

specifies the disaggregation method for reconciliation. The default value is **PROPORTIONS**.

ENCOMPASSALPHA=*value*

specifies the significance level (or alpha value) for the forecast encompassing test. Valid values are 0 to 1 (inclusive). The default value is **0.05**. See also the COMBINEENCOMPASS option.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE.

ENDZEROSMAXNUM = *integer*

specifies the maximum number of trailing zeros for a nonzero model. The default value is 0.

ENDZEROSMAXPCT = *integer*

specifies the maximum percentage of trailing zeros for a nonzero model relative to the number of nonzero values in the entire series. The default value is 0.

ENDZEROSMINOBS = *integer*

specifies a threshold for the series length that is required to enable the ENDZERO test. The default value is 0.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

ESM= YES | TRUE | NO | FALSE | 1 | 0

specifies whether exponential smoothing models (ESM) should be considered. The default value is **YES**.

ESMONLYATLEVEL= *BY- variable*

for hierarchical projects, specifies the highest level to fit the exponential smoothing models (ESM). Use this option to improve performance for projects with large hierarchies. The argument specified must be the name of a BY variable. By default, no level is specified for fitting the ESM models.

EVENTSDATASET=*SAS-data-set*

specifies the name of the events repository data set to include in the project during project creation. The value must be in the format *libname.datasetname*.

Note: The library of the events repository data set must be a pre-assigned library. This means that a library that is defined dynamically such as Work or a library that is assigned on demand will not work.

You can specify usage parameters for the events as follows:

- UNDEF—attribute is unknown. The default behavior is to not use the event.
- NO—do not use the event.
- MAYBE—use the event at the discretion of the diagnostic routines.
- YES—use the event.

The default usage parameter for events is UNDEF. If you want to provide a usage parameter other than UNDEF for events, then you must provide it in a separate data set with the same name as the events repository data set with a “2” appended to the end. For example, to provide additional usage parameters for an events repository data set that is named *testlib.MyEventsRepository*, then you must provide the additional usage parameters in a data set that is named *testlib.MyEventsRepository2*.

HIERARCHY= YES | TRUE | NO | FALSE | 1 | 0

specifies whether the BY variables should be considered hierarchical. The default value is **NO**.

HOLDOUT= *n* | NO | FALSE

specifies the number of periods to be used as the holdout sample for model selection. If you do not want to specify a holdout sample, set this option to NO or FALSE.

By default, the HOLDOUT= option is set to NO. If you specify a value for the HOLDOUTPCT= option but do not specify a value for the HOLDOUT= option, an error message appears because these two options are in conflict.

HORMISSPCT=*integer*

specifies a threshold for the percentage of missing forecast values in the combination horizon. This threshold is used to exclude a candidate forecast from consideration in the final combination. Valid values range from 1 to 100 (inclusive).

This option is different from the COMBINEMISSPCT option, which is for the estimation region. The COMBINEMISSPCT option and the HORMISSPERCENT option are independent of each other. You can specify one or both of these options.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE.

HOLDOUTPCT= *n* | NO | FALSE

specifies the maximum percentage of the series length to be used as the holdout sample for model selection. The holdout percentage guards against errors that are caused by holdout samples that are too large for the selected time series. For example, suppose a series has a length of 100 periods (excluding any leading or trailing missing values). If the HOLDOUT= option is set to 20 and the HOLDOUTPCT= option is set to 10, the holdout sample contains 10 periods.

If you do not specify a value for the HOLDOUTPCT= option, the default value is 5.

IDFORMAT= *date*

specifies the date-and-time format of the values in the time ID variable.

IMPORTSETTINGSPATH=*full-path-to-file*

specifies the file that contains the project settings to use in the new project. You must specify the full path to the macro file on the machine that invokes the macro. For

example, this could be a file created by %FSEXPSET. When you set the IMPORTSETTINGSPATH= option, the project settings in this file are used to initialize the values for the project. If you set any project settings by using arguments in the %FSCREATE macro, these arguments are processed after the IMPORTSETTINGSPATH= option. This is to enable project settings to be adjusted individually if necessary after the initial import of the project settings from the file.

INPUT= *variable*

specifies the input (or independent) variables. Separate multiple variable names by a space.

INTERMITTENT= *n* | NO | FALSE

specifies a number greater than 1 that is used to determine whether a time series is intermittent. The default value is 2. To disable this test, set this option to NO or FALSE.

INTERVAL= *interval-measure*

specifies the time interval of the time ID variable. If a time interval is not specified, SAS Forecast Server detects the time interval from the data. Any SAS interval name or abbreviation can be used.

Here are the SAS interval names that you can use: SECOND, MINUTE, HOUR, DAY, WEEKDAY, WEEK, WEEKV, TENDAY, SEMIMONTH, MONTH, R445MON, R454MON, R544MON, QTR, R445QTR, R454QTR, R544QTR, SEMIYEAR, YEAR, YEARV, R445YR, R454YR, and R544YR.

When specifying a SAS interval, remember the following:

- Do not include the DT prefix for datetime values. This prefix is automatically added when needed.
- Do not specify the multiplier or shift on the INTERVAL= option. To specify a multiplier, use the TIMEMULTIPLIER= option. To specify a shift, use the TIMESHIFT= option to specify a shift parameter.
- To specify the weekend days for the WEEKDAY interval, use the WEEKENDDAYS= option.

You can also use custom intervals that are defined by the INTERVALDS system option.

LEAD= *n*

specifies the number of periods into the future in which multiple step forecasts are made. The larger the horizon value, the larger the prediction error variance at the end of the horizon. The default value is 12.

MAXNUMOUTLIERS= *n*

specifies the maximum number of outliers to include in ARIMAX models. The default value is 2.

MAXPCTOUTLIERS= *n*

specifies the maximum number of outliers to include in ARIMAX models as a percentage of the series length, not including beginning, and ending missing values. The default value is 2.

If you specify both the MAXNUMOUTLIERS and MAXPCTOUTLIERS options, then SAS Forecast Server uses the smaller value. For example, you set MAXNUMOUTLIERS=5 and MAXPCTOUTLIERS=10. The maximum number of the outliers is 5% or 10% of the series length, whichever is smaller.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary

and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

MINOBSTREND= *n*

specifies the minimum number of observations needed for a trend model. The default value is 2.

MINOBSNONMEAN= *n*

specifies the minimum number of observations needed for a non-mean model. The default value is 1.

MINOBSSEASON= *n*

is used to determine the number of observations required for a seasonal model. The value of the MINOBSSEASON option is multiplied by the seasonal cycle length. In order for a seasonal model to be fit to a series, the model must have more observations than the calculated value. Models with fewer observations will not be fit.

The value of MINOBSSEASON must be greater than or equal to 1. The default value is 2.

MODELSELECTIONLIST= *SAS-data-set*

specifies a model selection list that contains models that you can use. The default value is **SASHELP.HPFDFLT.TSFSSELECT**.

PUBLICACCESS= YES | TRUE | NO | FALSE | 1 | 0

specifies whether the project can be opened by other users. The default value is **NO**.

RECONCILIATION= TOPDOWN | TD | BOTTOMUP | BU | MIDDLEOUT(*BY-variable*) | MO(*BY-variable*) | NO | FALSE

specifies whether SAS Forecast Server reconciles the hierarchy. By default, SAS Forecast Server reconciles the hierarchy using the top-down reconciliation method (TOPDOWN or TD). To reconcile the hierarchy by a middle level, you must specify the BY variable for the level where the reconciliation starts.

Here are some examples of how to specify this option:

- To specify a top-down reconciliation, use RECONCILIATION=TOPDOWN.
- To specify a middle-out reconciliation that starts at the *regionName* variable, use RECONCILIATION=MIDDLEOUT(*regionName*).
- If you do not want to reconcile the hierarchy, specify RECONCILIATION=NO (or FALSE).

REPORTING= *variable*

specifies the reporting (or auxiliary) variables for the project. Separate multiple variable names by a space.

REQUIRED = *type(mod) [var1 var2 var3] type(mod) [var1 var2 var3]*

specifies whether to include the input variables in the model, where

- type = YES|NO|MAYBE
- mod=POSITIVE|NEGATIVE

Note: The values for the REQUIRED= option are case sensitive. In your macro code, use uppercase letters.

- NO — specifies that the input variables be included in the model as long as their parameters are significant and the increment of the value of criterion exceeds a threshold. The default is REQUIRED=NO.

- **MAYBE** — specifies that the input variables be included in the model as long as their parameters are significant.
- **YES** — specifies that the input variables be included in the model as long as the model does not fail to be diagnosed.

Either the **POSITIVE** or **NEGATIVE** option with parentheses can follow *type*. This is optional. For example, specifying **REQUIRED=YES(POSITIVE)** drops the input variable from the model if its coefficient is negative, while specifying **REQUIRED=YES(NEGATIVE)** implies the opposite. The specification of **POSITIVE** or **NEGATIVE** does not mean that constraints are imposed during the estimation of the variable's coefficient in the model.

Here are some examples:

- **REQUIRED=MAYBE YES[*var1*]** — All variables (except variable1) with significant parameters are included in the model. variable1 is included in the model as long as the model does not fail to be diagnosed.
- **REQUIRED=MAYBE NO[*var1 var2*] YES(POSITIVE)[*var3*]** — Variables 1 and 2 are included in the model as long as their parameters are significant and the increment of the value of criterion exceeds a threshold. Variable 3 is included in the model if its coefficient is positive and the model does not fail to be diagnosed. All remaining variables are included in the model as long as their parameters are significant.

RUN= YES | TRUE | NO | FALSE | 1 | 0

specifies whether to produce forecasts when the project is created. The default value is **YES**. The **NO** option is equivalent to the **Save SAS code to produce forecasts later** option in the New Project wizard in SAS Forecast Studio. If you select this option, the SAS code (.sas file in the project folder) for the desired task (for example, **CREATE_PROJECT_IMPORT_DATA.sas**) should be submitted when forecasts are needed. If the %FSRUNPRJ macro is submitted without first submitting this code, it automatically selects the task and submits the code.

SASENVIRONMENT = *environment-name*

specifies the symbolic name of the SAS environment where the middle tier for the SAS Forecast Server is saved. This symbolic name is specified by the **NAME=** attribute in the sas-environment.xml file and is case sensitive. The default value is **Default**.

SEASONALITY= *n*

specifies the length of a season. When possible, SAS Forecast Server determines the seasonal cycle length from the time ID variable. You can specify a seasonal cycle length other than the default if you want to model a cycle in the data. For example, if your data contains a 13-week cycle, then use **SEASONALITY=13**.

SEASONTEST= *n* | NO | FALSE

specifies the sensitivity of the seasonality test. You can specify no seasonality test or specify a significance probability value for the test. Series with strong seasonality have small test probabilities. A significance probability value of 0 always implies seasonality. A significance probability value of 1 always implies no seasonality.

The default value is **0.01**. To disable the test, set this option to **NO** or **FALSE**.

SETMISSING= 0 | MISSING | AVG | MIN | MED | MAX | FIRST | LAST | PREV | NEXT

specifies how to replace the missing values in the data. The default value is **MISSING**.

- **0** - The missing values are set to 0.

- MISSING - The missing values are set to missing. This is the default value.
- AVG - The missing values are set to the accumulated average value.
- MIN - The missing values are set to the accumulated minimum value.
- MED - The missing values are set to the accumulated median value.
- MAX - The missing values are set to the accumulated maximum value.
- FIRST - The missing values are set to the accumulated first nonmissing value.
- LAST - The missing values are set to the accumulated last nonmissing value.
- PREV - The missing values are set to the previous accumulated nonmissing value. Missing values at the beginning of the accumulated series remain missing.
- NEXT - The missing values are set to the next accumulated nonmissing values. Missing values at the end of the accumulated series remain missing.

START= *date*

specifies the first date to use in data preparation. The date format is ddmmyyyy and the date-and-time format is ddmmyyyy:hh:mm:ss. The default value is the earliest date in the input data set.

TIMEMULTIPLIER= *n*

specifies the multiplier for the time interval that you specified by using the INTERVAL= option (for example, for two-week periods (WEEK2), use TIMEMULTIPLIER=2 with INTERVAL=WEEK).

TIMESHIFT= *n*

specifies the offset for the time interval that you specified in the INTERVAL= option (for example, for one-week periods starting on the second weekday (WEEK.2), use TIMESHIFT=2 with INTERVAL=WEEK).

TRANSBOXCOX= *n*

specifies the Box-Cox value if the transformation type (specified by the TRANSTYPE option) is BOXCOX. Valid values range from -5 to 5 (inclusive). The default is 1.

TRANSOPT= MEAN | MEDIAN

specifies how to calculate the forecasts for the system-generated models. The default is MEAN.

TRANSTYPE= AUTO | LOG | NONE | SQRT | LOGISTIC | BOXCOX

specifies the transformation to apply to the dependent variable or independent variable. This transformation applies to the system-generated models.

Here is a description for each valid value:

AUTO	automatic transformation. SAS Forecast Studio uses the model selection criteria to determine whether to apply a logarithmic transformation or no transformation.
LOG	logarithmic transformation. This option is not available for negative series.
NONE	no transformation. This is the default.
SQRT	square-root transformation.
LOGISTIC	logistic transformation.
BOXCOX	Box-Cox transformation. If you select this type of transformation, then you can specify a value for this transformation using the TRANSBOXCOX option.

TRIMMISS= NONE | LEFT | RIGHT | BOTH

specifies how missing values are removed from the accumulated time series. The default value is **NONE**. If the project is hierarchical with reconciliation, only NONE is allowed.

Here is a description of each valid value:

- NONE The missing values are kept.
- LEFT The beginning missing values are removed.
- RIGHT The ending missing values are removed.
- BOTH Both the beginning and ending missing values are removed.

UCM= YES | TRUE | NO | FALSE | 1 | 0

specifies whether SAS Forecast Server should automatically generate an unobserved component model (UCM). The default value is **NO**.

WEEKENDDAYS= 1 | 2 | 3 | 4 | 5 | 6 | 7

specifies which days are the weekend (or inactive) days in the week. Valid only when INTERVAL=WEEKDAY. The days are specified as numbers (1 through 7) representing the days of the week (Sunday through Saturday).

Here are some examples:

- To specify Saturday and Sunday as the weekend (or inactive) days in the week, use WEEKENDDAYS=17
- To specify Friday and Saturday as the weekend days, use WEEKENDDAYS=67.

ZEROMISS= NONE | LEFT | RIGHT | BOTH

specifies how beginning or ending zero values (or both) are interpreted in the accumulated time series. The default value is **NONE**.

Here is a description of each valid value:

- NONE The beginning and ending zeros are unchanged.
- LEFT The beginning zeros are set to missing.
- RIGHT The ending zeros are set to missing.
- BOTH Both the beginning and ending zeros are set to missing.

Additional Considerations

If you are using the SAS Forecast Server Batch Interface, you must run the macro code on a SAS Workspace Server if your project meets the following criteria:

- You are using the %FSCREATE or %FSRUNPRJ macros.
- You are using a custom time interval or custom format.
- The macros depend on LIBNAME statements in the autoexec file in order to run.

Results

The %FSCREATE global macro variable indicates whether the %FSCREATE macro finishes successfully or encounters errors: &FSCREATE = SUCCESS | ERROR.

A series of global macro variables are populated after a SAS Forecast Server project is loaded. These macro variables describe the project. Using these macro variables, you can write SAS code that is generic enough to use across multiple projects. For example, you can use this code to explore and use data in your SAS Forecast Server projects. You can also use these macro variables in SAS Stored Processes.

Example

```
%fscree (projectname=Shoe_Sales_2012,
environment=Default,
data=sashelp.pricedata,
id=date,
by=regionName productLine productName,
hierarchy=YES,
var=sale,
input=price discount,
reporting=price1 price2 price3,
accumulate=TOTAL AVG(price) NONE(price1 price3),
aggregate=NONE total(sale) MIN(price1 price3),
disaggregation=EQUALSPLIT,
reconciliation=NO,
publicaccess=YES,
required=YES(POSITIVE) [price] MAYBE[discount]
setmissing=last,
zeromiss=left,
trimmiss=right,
back=1,
start=01Feb1998,
interval=MONTH,
seasonality=12,
idformat=MMYY.,
timemultiplier=1,
timeshift=1,
arimax=YES,
esm=YES,
modelselectionlist=SASHELP.HPFDFLT.TSFSSELECT,
detectoutliers=YES,
intermittent=NO,
seasontest=0.3,
holdout=NO,
holdoutpct=2,
minobstrend=2,
minobsnonmean=2,
criterion=MAPE,
lead=24,
alpha=0.10,
allownegative=NO,
description=foo bar stuff,
transopt=MEDIAN,
transtype=boxcox,
transboxcox=2
)
```

%FSDelARC Macro

The %FSDelARC macro deletes an archived SAS Forecast Server project.

Syntax

```
%FSDelARC (ARCHIVENAME= [ , options ])
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSDELARC macro if no default values are currently stored.

ARCHIVENAME= *archived-project-name*

specifies the filename of the archive that you want to delete the project from (for example, myarchive.far). If the archive name does not end with .far, then the extension is automatically assumed.

Options

The following options can be used with the %FSDELARC macro. Options must follow the required arguments and are separated by commas.

ARCHIVEFOLDER= *archive-folder-name*

specifies the directory containing the archive to delete. If not specified, the default location is assumed.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME=*project-name*

specifies the name of the SAS Forecast Server project that is associated with the archive. You must specify this option only if you do not specify the ARCHIVEFOLDER= option. The value of this default cannot be customized with the %FSSETDEF macro.

Results

The %FSDELARC global macro variable indicates whether the %FSDELARC macro finishes successfully or encounters errors: &FSDELARC = SUCCESS | ERROR.

Example

```
%fsdelarc (projectname=Shoe_Sales_2012,
archive=Archive_Shoe_Sales_2012,
)
```

%FSDELENV Macro

The %FSDELENV macro deletes an environment. When you delete an environment, the metadata, and file system content are deleted.

Syntax

```
%FSDELENV (ENVIRONMENT= [, options ])
```

Details

Required Arguments

The following argument must be used with the %FSDELENV macro.

ENVIRONMENT= *environment-name*

specifies the name of the SAS Forecast Server environment. For security reasons, this argument has no default, and must be explicitly entered. Using the %FSSETDEF macro to set a global default value is not applicable in this instance.

Options

The following argument is optional for the %FSDELENV macro. Options must follow the required arguments and are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSDELENV global macro variable indicates whether the %FSDELENV macro finishes successfully or encounters errors: &FSDELENV = SUCCESS | ERROR.

Example

```
%fsdelenv (enviroment=Default
)
```

%FSDELEVT Macro

The %FSDELEVT macro deletes selected events from the event repository.

Syntax

```
%FSDELEVT (EVENTNAMES= [, options])
```


Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSDELEVT macro if no default values are currently stored.

EVENTNAMES= *name1 name2*

specifies the events that you want to delete. Only one event name is required. Separate multiple names by a space.

Options

The following options can be used with the %FSDELEVT macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

EVENTINTEGRITYRULE= NONE | CHECK | FORCE

determines how events are deleted when those events are in a combination event or model selection lists. The default value is **NONE**.

NONE No integrity checks are performed. The events are removed from the event repository unconditionally.

CHECK SAS Forecast Server checks combination events and model selection lists for these events. An error appears if any of the events are found.

FORCE The events are removed from combination events, model selection lists, and the event repository.

Note: Running the %FSDELEVT macro with EVENTINTEGRITYRULE= option causes the model selection lists that contain the event to be inconsistent. To make all the model selection lists consistent, run the %FSRUNPRJ macro with the METHOD=DIAGNOSE option. For more information, see “%FSRUNPRJ Macro” on page 167.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Results

The %FSDELEVT global macro variable indicates whether the %FSDELEVT macro terminates successfully or encounters errors: &FSDELEVT= SUCCESS | ERROR.

Example

```
%fsdelevt(projectname=Shoe_Sales_2012,
          eventnames=evt2 evt4,
          environment=Default
        )
```

%FSDELPRJ Macro

The %FSDELPRJ macro deletes one project, and if you choose, any related archives in the default archiving folder.

Syntax

%FSDELPRJ (PROJECTNAME= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument must be used with the %FSDELPRJ macro if no default values are currently stored.

PROJECTNAME=*project-name*
specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this argument, then the default value for the project name is used. The value of this default cannot be customized with the %FSSETDEF macro.

Options

The following options can be used with the %FSDELPRJ macro. Options must follow the required arguments and are separated by commas.

DELETEARCHIVES= YES | TRUE | NO | FALSE | 1 | 0
specifies whether the project archives in the default archive location are to be deleted. By default, all archives are deleted with the project.

ENVIRONMENT = *environment-name*
specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

NOWARN= YES | NO

specifies whether to suppress errors if the SAS Forecast Server project does not exist. The default is **NO**.

Results

The %FSDELPRJ global macro variable indicates whether the %FSDELPRJ macro finishes successfully or encounters errors: &FSDELPRJ = SUCCESS | ERROR.

Example

```
%fsdelprj (projectname=Shoe_Sales_2012,
deleteArchives=YES,
)
```

%FSEVTREQ Macro

The %FSEVTREQ macro sets the required attributes on events in the project's event repository.

Syntax

%FSEVTREQ (EVENTNAME=, EVENTREQUIRED= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required used with the %FSEVTREQ macro if no default values are currently stored. The required arguments are separated by commas.

EVENTNAME=*event-name*

specifies the name of the event. To set attributes for all events in the project's event repository, use the **_ALL_** keyword in place of *event-name*.

EVENTREQUIRED= YES | MAYBE | NO | UNDEF

specifies whether the event must be included in the model.

Here are the descriptions of the valid values:

YES	specifies to include the event in the model as long as the model can be diagnosed.
-----	--

MAYBE	specifies to include the event in the model if the parameters of the event are significant.
NO	specifies to include the event in the model if the parameters of the event are significant and the value of the criterion exceeds a specified threshold.
UNDEF	specifies not to include the event in the model.

Options

The following options can be used with the %FSEVTREG macro. Options must follow the required arguments and are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Results

The %FSEVTREG global macro variable indicates whether the %FSEVTREG macro finishes successfully or encounters errors: &FSEVTREG = SUCCESS | ERROR.

Example

```
%fsevtreg (projectname=Shoe_Sales_2012,
eventname=_ALL_,
eventrequired=undef
)
```

%FSEXPALL Macro

The %FSEXPALL macro exports all SAS Forecast Server projects to archived files. This macro should be used only by users who are assigned the Administer Product capability in SAS Management Console.

Syntax

```
%FSEXPALL ( [ options ] )
```

Details

Options

The following options can be used with the %FSEXPALL macro. Options must follow the required arguments and are separated by commas.

ARCHIVEFOLDER= *directory-name*

specifies the directory where the archive is to be saved. If not specified, the archives are saved to project subfolders in the default location for the environment.

CPORT= YES | TRUE | NO | FALSE | 1 | 0

exports data sets and catalogs using CPORT. Use this argument if you unarchive the project on a different platform. The default is **NO**.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

NAMESUFFIX= *string*

specifies a suffix to the name of the archive. The suffix must make the archive name unique. If an archive with the same name exists, then archiving fails for that project. The default suffix is **_fs_export**.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

OUT= *SAS-data-set*

specifies the fully qualified name of the data set that contains information about the archived projects. The format is *library.dataset*.

SUBFOLDERS= YES | TRUE | NO | FALSE | 1 | 0

specifies whether archives are saved in subfolders corresponding to project names. If **NO**, all archives are saved in the same location. To use this option, you must specify the ARCHIVEFOLDER option. The default is **NO**.

PRINT = YES | TRUE | 1 | NO | FALSE | 0

specifies whether to print the output data set. The default is **NO**.

Results

The %FSEXPALL global macro variable indicates whether the %FSEXPALL macro finishes successfully or encounters errors: &FSEXPALL = SUCCESS | ERROR.

The OUT= option produces a SAS data set that contains the following variables:

NAME	specifies the project name.
CREATED	specifies the date-and-time when the project is created.
MODIFIED	specifies the date-and-time when the project is last modified.
CREATEDBY	specifies the user ID that created the project initially.
OWNEDBY	specifies the user ID of the project's owner.
LOCKED	equals 1 if the project is locked (currently opened by another user).

CANOPEN	equals 1 if the user can open the project.
CANDELETE	equals 1 if the user can delete the project.
ISPUBLIC	equals 1 if the project has public access.
ARCHIVENAME	specifies the name of the exported project's archive file.
ARCHIVEFAIL	equals 1 if an error was detected during archiving the project.

In addition, archives for all projects are created in the default archive folder. The name of the archives is the name of the project followed by the NAMESUFFIX= parameter.

Example

```
%fsexpall (out=work.projects,
namesuffix=_fs_export31,
environment=default
)
```

%FSEXPORT Macro

The %FSEXPORT macro exports a SAS Forecast Server project to an archive file.

Syntax

%FSEXPORT (ARCHIVENAME=, DESCRIPTION= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSEXPORT macro if no default values are currently stored. The required arguments are separated by commas.

ARCHIVENAME= *archived-project-name*

specifies the filename of the archive that you want to export the project into (for example, myarchive). Do not specify the file extension. It is automatically assumed to be .far. Use the ARCHIVEFOLDER= option to specify a file path to the archive file.

DESCRIPTION= *text*

specifies a description to assign to the archive. The description is recorded in the Manifest.fs file within the archive. If a comma is present in the description string, the description must be enclosed in quotation marks.

Options

The following options can be used with the %FSEXPSET macro. Options must follow the required arguments and are separated by commas.

ARCHIVEFOLDER= *directory-name*

specifies the directory where the archive is to be saved. If not specified, the default location is used.

CPORT= YES | TRUE | NO | FALSE | 1 | 0

exports data sets and catalogs using CPORT. Use this argument if you unarchive the project on a different platform. The default is **NO**.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Results

The global macro variable %FSEXPSET indicates whether the %FSEXPSET macro finishes successfully or encounters errors: &FSEXPSET = SUCCESS | ERROR.

Also, an archived project with the ARCHIVENAME= filename is created in the default archive folder.

Example

```
%fsexport( projectname=Shoe_Sales_2011,
archivename=Archive_Shoe_Sales_2011,
description=Project pd1,
user=sasuser,
password=saspass
)
```

%FSEXPSET Macro

The %FSEXPSET macro exports the project settings to a local file.

Syntax

%FSEXPSET (OUTPATH= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument must be used with the %FSEXPSET macro if no default values are currently stored.

OUTPATH= *full-path-of-output-file*
specifies the path and filename of the local file. These files use an .fps extension.

Options

The following options can be used with the %FSEXPSET macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*
specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*
identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*
specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Results

The global macro variable %FSEXPSET indicates whether the %FSEXPSET macro finishes successfully or encounters errors: &FSEXPSET = SUCCESS | ERROR.

Example

```
%fsexpset (outpath=C:\ForecastServer\Projects\Settings.fps,
environment=Default,
projectname=Shoe_Sales_2012,
)
```

%FSGETENV Macro

The % FSGETENV macro retrieves information about the SAS Forecast Server environments. Only an administrative user can run this macro.

Syntax

%FSGETENV ([options])

Details

Options

You can use the following options with the %FSGETENV macro. Options must follow the required arguments and are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

OUT= *SAS-data-set*

specifies the name of the data set that contains the program results. The default value is Work.FSEnv.

PRINT = YES | TRUE | 1 | NO | FALSE | 0

specifies whether to print the output data set. The default is NO.

Results

The %FSGETENV global macro indicates whether the %FSGETENV macro finishes successfully or encounters errors: &FSGETENV = SUCCESS | ERROR.

The OUT= option produces a SAS data set that contains the following variables:

NAME	specifies the environment name.
DESCRIPTION	specifies the description of the environment.
CREATED	specifies the date-and-time when the environment was created.
MODIFIED	specifies the date-and-time when the environment was last modified.
HOSTNAME	specifies the name of the workspace server used by the environment.
REPORTSROOT	specifies the SAS metadata folder where the stored process metadata objects are stored.
CONTENTPATH	specifies the file system path associated with the environment.
VERSION	specifies the version of SAS Forecast Server that the environment uses.

Example

```
%fsgetenv (out=work.getprojects,
print=YES
)
```

%FSGETPRJ Macro

The %FSGETPRJ macro retrieves information about the SAS Forecast Server projects.

Syntax

%FSGETPRJ ([options])

Details

Options

You can use the following options with the %FSGETPRJ macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

OUT= *SAS-data-set*

specifies the name of the data set that contains the program results. The default value is Work.FSPRj.

PRINT = YES | TRUE | 1 | NO | FALSE | 0

specifies whether to print the output data set. The default is NO.

Results

The %FSGETPRJ global macro indicates whether the %FSGETPRJ macro finishes successfully or encounters errors: &FSGETPRJ=SUCCESS | ERROR.

The OUT= option produces a SAS data set that contains the following variables:

NAME	specifies the project name.
DESCRIPTION	specifies the description of the project.
VERSION	specifies the version of SAS Forecast Server for the project.
LIBRARY	specifies the library for the project's input data set.
DATASET	specifies the name of the project's input data set.
CREATED	specifies the date-and-time when the project was created.
MODIFIED	specifies the date-and-time when the project was last modified.
CREATEDBY	specifies the project identity of the person who created the project initially which in some cases is the same as the user ID.

MODIFIEDBY	specifies the project identity of the last person who modified the project which in some cases is the same as the user ID.
OWNEDBY	specifies the project identity of the project's owner which in some cases is the same as the user ID.
LOCKED	equals 1 if the project is locked (currently opened by another user).
CANOPEN	equals 1 if the user can open the project.
CANDELETE	equals 1 if the user can delete the project.
ISPUBLIC	equals 1 if the project is available to all users.

Example

```
%fsgetprj (out=work.getprojects,
print=YES
)
```

%FSGETURP Macro

The %FSGETURP macro creates a list of unregistered project names in a specified environment. To register the projects, you can use the project names generated by the %FSGETURP macro as input to the %FSREGPRJ macro.

Syntax

%FSGETURP ([options])

Details

Options

You can use the following options with the %FSGETURP macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

OUT= *SAS-data-set*

specifies the name of the data set that contains the program results. The default value is Work.FSPRj.

PRINT = YES | TRUE | 1 | NO | FALSE | 0

specifies whether to print the output data set. The default is NO.

Results

The %FSGETURP global macro variable indicates whether the %FSGETURP macro finishes successfully or encounters errors: &FSGETURP = SUCCESS | ERROR.

Example

```
%fsgeturp (environment=Default,
out=sales.unregistered,
print=YES
)
```

%FSIMPALL Macro

The %FSIMPALL macro imports all of the SAS Forecast Server archived files listed in the PROJECTDS= data set. This macro should be used only by users who have the Administer Product capability.

Syntax

%FSIMPALL (PROJECTDS=, ARCHIVEFOLDER= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSIMPALL macro if no default values are currently stored. The required arguments are separated by commas.

PROJECTDS= *SAS-data-set*

specifies the SAS data set that contains the list of the SAS Forecast Server projects to import. It can be a fully qualified data set name in the form library.dataset. The PROJECTDS= data set is created when you specify the OUT= option of the %FSEXPALL macro.

The PROJECTDS= data set must be a SAS data set that contains the following variables:

Name

specifies the project name.

archivename

specifies the name of the archive.

archivefail

(optional) equals 1 if an error was detected during archiving. If the Archive Fail variable is included, only the archives where the value of the Archive Fail variable is 0 are imported.

ARCHIVEFOLDER= *directory-name*

specifies the folder that contains the archived projects (for example, C:\SAS\ForecastStudio\Archives).

Options

You can use the following options with the %FSIMPALL macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

SUBFOLDERS= YES | TRUE | NO | FALSE | 1 | 0

specifies whether archives are stored in subfolders corresponding to project names within the specified archive folder location. The default value is NO, and all archives are assumed to be located in the archive folder location.

Results

The %FSIMPALL global macro variable indicates whether the %FSIMPALL macro finishes successfully or encounters errors: &FSIMPALL = SUCCESS | ERROR.

Note: %FSIMPALL does not migrate 2.1 projects to the current version. However, you can use the %FSMIGALL macro to accomplish that. For an example that uses %FSIMPALL, see “%FSMIGALL Macro” on page 153.

Example

```
%fsimpall (projectds=work.projects,
archivefolder=\\sourceserver\SAS\ForecastStudio\Archives
)
```

%FSIMPORT Macro

The %FSIMPORT macro imports a SAS Forecast Server archived file.

Syntax

%FSIMPORT (ARCHIVEPATH= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSIMPORT macro if no default values are currently stored.

ARCHIVEPATH= *directory-name*
specifies the location of the archived project.

Options

You can use the following options with the %FSIMPALL macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*
specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*
identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*
specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Results

The %FSIMPORT global macro variable indicates whether the %FSIMPORT macro finishes successfully or encounters errors: &FSIMPORT = SUCCESS | ERROR.

Example

```
%fsimport (projectname=Shoe_Sales_2011,
archivepath=C:\SAS\ForecastStudio\Archives\pd1\Archive_Shoe_Sales_2011
)
```

%FSLOAD Macro

The %FSLOAD macro opens an existing SAS Forecast Server project and loads global macro variables that describe the project.

Note: The %FSLOAD macro requires that forecasts have been generated for the project. If the project was created using the RUN=NO option in the %FSCREATE macro, the %FSLOAD macro displays an error message. Use the %FSRUNPRJ macro to run the project.

Syntax

```
%FSLOAD ( [ options ] )
```

Details

Options

You can use the following options with the %FSLOAD macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 170.

Results

The %FSLOAD global macro variable indicates whether the %FSLOAD macro finishes successfully or encounters errors: &FSLOAD = SUCCESS | ERROR.

A series of global macro variables are populated after a SAS Forecast Server project is loaded. These macro variables describe the project. Using these macro variables, you can write SAS code that is generic enough to use across multiple projects. For example, you can use this code to explore and use data in your SAS Forecast Server projects. If you include %put _user_ in the example code, the values generated by the %FSLOAD macro appear in the log.

Example

```
%fsload (projectname=Shoe_Sales_2012,
environment=default
)
%put _user_
```

%FSLOGIN Macro

The %FSLOGIN macro enables you to create a new session for a specific instance of the middle tier. Each middle tier is permitted to have at most one session at any given time, allowing the middle tier to be used to identify the session. This option is supported both by SAS Forecast Server and by SAS Forecasting for Desktop.

Note: %FSLOGIN stores information about the most recent session in a global variable for use by other macros.

Syntax

`%FSLOGIN (DESKTOP=,USER=, PASSWORD= [ , options ] )`

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSLOGIN macro if no default values are currently stored. The required arguments are separated by commas.

`DESKTOP= YES|NO|TRUE|FALSE|0|1`

specifies whether the instance of the middle tier is being run in SAS Forecast Server or in SAS Forecasting for Desktop. This setting is used only the first time a session is created for that instance of the middle tier. By default, this option is set to NO, and the new session is created for SAS Forecast Server.

`USER= user-name`

specifies the user name that you use to log on to SAS Forecast Studio. This option is required when `DESKTOP=NO`, which is the default setting.

`PASSWORD= password`

specifies the password that you use to log on to SAS Forecast Studio. This option is required when `DESKTOP=NO`, which is the default setting.

Options

You can use the following options with the %FSLOGIN macro. Options must follow the required arguments and are separated by commas.

`SASENVIRONMENT= environment-name`

specifies the symbolic name of the SAS environment where the middle tier for SAS Forecast Server is saved. This symbolic name is specified by the NAME= attribute in the sas-environment.xml file and is case sensitive. The default value is **default**.

`MIDTIER= label`

identifies which middle tier to use. This argument should typically be omitted and the default values used. In such cases, you must give each middle tier a unique label (name) so that the later macro calls can be properly directed to the correct sessions. If the MIDTIER= option is not specified, this label is chosen automatically based on the type of the middle tier. For SAS Forecast Server, the default label is "fsmain". For SAS Forecasting for Desktop, the default label is "fsdesk". Middle tiers using different names operate independently, even if they have the same type.

Results

The %FSLOGIN global macro indicates whether the %FSLOGIN macro finishes successfully or encounters errors: `&FSLOGIN=SUCCESS | ERROR`.

Example

```
%fslogin (user=sasdemo,
password=Password1,
```



```

sasEnvironment=default,
desktop=NO
)

```

%FSLOGOUT Macro

The %FSLOGOUT macro closes a session on the specified instance of the middle tier.

Syntax

%FSLOGOUT ([options])

Details

Options

You can use the following option with the %FSLOGOUT macro.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSLOGOUT global macro indicates whether the %FSLOGOUT macro finishes successfully or encounters errors: &FSLOGOUT=SUCCESS | ERROR.

%FSLOOP Macro

The %FSLOOP macro enables you to iterate through a given data set and invokes a callback macro with the values from each row.

Syntax

%FSLOOP (DATA=,CALLBACK=,VARLIST= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSLOOP macro if no default values are currently stored.. The required arguments are separated by commas.

DATA=*data-set-name*

specifies the data set that you want the %FSLOOP macro to use.

CALLBACK=*name-of-handler-macro*

specifies the handler macro to invoke for each element. The macro must implement the expected signature.

VARLIST=*var-name1 var-name2 . . . var-name_n*

specifies the variable names in the data set. Variable names must be separated by a space. The macro specified in the CALLBACK= argument is expected to accept each row variable through a macro variable of the same name, unless the VARARGS= optional argument is used to map the data set variables to other macro arguments.

Options

You can use the following options with the %FSLOOP macro. Options must follow the required arguments and are separated by commas.

ARGDATA=*name-of-data-set*

specifies the data set that contains static argument values. These additional arguments are passed on each generated macro call. The data set must be formatted for two character columns named Name and Value. By default, no static arguments are used.

VARARGS=*macro_varname1 macro_varname2 . . . macro_varname_n*

specifies the variable names for the macro input that corresponds to the column names in the VARLIST= argument. If you do not specify the VARARGS= argument, the variable names are assumed to match the column names in the input data set. If VARLIST=NAME DESCRIPTION but the macro wants the input variables to be X= and Y=, then you should specify VARARGS= X Y.

Results

The %FSLOOP global macro indicates whether the %FSLOOP macro finishes successfully or encounters errors: &FSLOOP=SUCCESS | ERROR.

Example

```
%FSLOGIN (desktop=no,
user=sasdemo,
password=Password1
)
* Example of using FSGETPRJ to get project information and put results in work.prjlst;

%FSGETPRJ(
environment=Default,
out=work.prjlst)

* Example using arguments data set to fill extra parameters;

data args;
length name $32;
length value $128;

name='environment';
```

```

value='Default';
output;

name='isPublic';
value='true';
output;
run;

%FSLOOP(
  data=work.prjlst,
  callback=FSSETPUB,
  varlist=name,
  varargs=project,
  argdata=args
)

%FSLOGOUT()

```

%FSMIGALL Macro

The %FSMIGALL macro updates all registered projects to the current version of SAS Forecast Server.

Note: This macro updates only projects created with SAS Forecast Server 2.1 or later. You cannot use this macro to update projects created with SAS Forecast Server 1.4 directly to SAS Forecast Server 14.2. You must first upgrade these projects to use SAS Forecast Server 2.1.

Syntax

%FSMIGALL (OUT= [, options])

Details

Options

You can use the following options with the %FSMIGALL macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

OUT= *SAS-data-set*

specifies the fully qualified name of the data set that contains information about the archived projects. The format is library.dataset. The default value is Work.FSMig.

PRINT = YES | TRUE | 1 | NO | FALSE | 0

specifies whether to print the output data set. The default is NO.

Results

The %FSMIGALL global macro indicates whether the %FSMIGALL macro finishes successfully or encounters errors: &FSMIGALL=SUCCESS | ERROR.

The OUT= option produces a SAS data set that contains the following variables:

NAME	specifies the project name.
CREATED	specifies the date-and-time when the project is created.
MODIFIED	specifies the date-and-time when the project is last modified.
CREATEDBY	specifies the user ID that created the project initially.
OWNEDBY	specifies the user ID of the project's owner.
LOCKED	equals 1 if the project is locked (currently opened by another user).
CANOPEN	equals 1 if the user can open the project.
CANDELETE	equals 1 if the user can delete the project.
ISPUBLIC	equals 1 if the project is available to all users.
ISMIGRATED	equals 1 if the project was successfully migrated.

Example

```
%fsmigall (out=work.migrated,
print=YES
)
```

%FSMIGPRJ Macro

The %FSMIGPRJ macro updates an existing SAS Forecast Server project to the current version of SAS Forecast Server.

Note: This macro updates only projects created with SAS Forecast Server 2.1 or later. You cannot use this macro to update projects created with SAS Forecast Server 1.4 directly to SAS Forecast Server 4.1. You must first upgrade these projects to use SAS Forecast Server 2.1.

Syntax

```
%FSMIGPRJ ( [ options ] )
```

Details

Options

You can use the following options with the %FSMIGPRJ macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 170.

Results

The %FSMIGPRJ global macro indicates whether the %FSMIGPRJ macro finishes successfully or encounters errors: &FSMIGPRJ=SUCCESS | ERROR.

Example

```
%fsmigprj (projectname=Shoe_Sales_2012,
environment=default
)
```

%FSMOVE Macro

The %FSMOVE macro moves a SAS Forecast Server project to a new destination.

Syntax

```
%FSMOVE ( [ options ] )
```

Details

Options

You can use the following options with the %FSMOVE macro. Options must follow the required arguments and are separated by commas.

ARCHIVEFOLDER = *directory-name*

specifies the directory to save the archive in. If not specified, the default location is used.

CPORT= YES | TRUE | NO | FALSE | 1 | 0

exports data sets and catalogs using CPORT. Needed if the source and destination are running on different operating systems. The default is **NO**.

DESTINATIONENVIRONMENT= *environment-name*

specifies the name of the SAS Forecast Server environment to be used as destination. This option is required only if there are multiple environments available to the user. If omitted, the current stored default value is used. The default value is **Default**. A global default value for environment can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

DESTINATIONPROJECTNAME= *destination-project-name*

specifies the name of the SAS Forecast Server project in the destination environment. The name must be a valid SAS name. A global default value for the project name can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

REMOTEARCHIVEFOLDER= *project-name*

specifies the folder location of the archived project. This location must be accessible by the destination environment (for example, \\remotehost\SAS\ForecastStudio\Archives).

SOURCEENVIRONMENT= *environment-name*

specifies the name of the SAS Forecast Server environment to be used as source. This option is required only if there are multiple environments available to the user. If omitted, the current stored default value is used. The default value is **Default**. A global default value for environment can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

SOURCEPROJECTNAME= *source-project-name*

specifies the name of the SAS Forecast Server project in the source environment. The name must be a valid SAS name. A global default value for the project name can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

TMPARCHIVENAME= *value*

specifies the name of the temporary archived project. The default is **_fs_tmp_archive**. The name must not match an existing archive.

Results

The %FSMOVE global macro indicates whether the %FSMOVE macro finishes successfully or encounters errors: &FSMOVE=SUCCESS | ERROR.

Example

```
%fsmove (sourceprojectname=Shoe_Sales,
destinationprojectname=Shoe_Sales_2012,
remoteArchiveFolder=C:\SAS\ForecastStudio\Archives,
sourceEnvironment=Default,
destinationEnvironment=Default,
)
```

%FSNEWENV Macro

The %FSNEWENV macro creates a new SAS Forecast Server environment.

Syntax

```
%FSNEWENV (ENVIRONMENT=, PATH= [ , options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSNEWENV macro if no default values are currently stored. The required arguments are separated by commas.

ENVIRONMENT=*environment-name*

specifies the name of the SAS Forecast Server environment. The value for this argument is case sensitive. The default environment is **Default**. The value of this default cannot be customized with the %FSSETDEF macro.

PATH=*file path*

specifies the absolute file path to the base content directory.

Options

You can use the following options with the %FSNEWENV macro. Options must follow the required arguments and are separated by commas.

DESC=*text*

specifies the environment description.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

REPORTSPATH= *UNIX-style path*

specifies a UNIX absolute path to the Base SAS Folder in the metadata (for example, **/MyContent/Reports**). Searches for reports start in this directory. This argument is used only when you are creating an environment for a networked middle tier.

WSSERVER= *workspace server-name*

specifies the name of the logical workspace server. The default value is **SASAPP - Logical Workspace Server**.

Results

The %FSNEWENV global macro indicates whether the %FSNEWENV macro finishes successfully or encounters errors: &FSNEWENV=SUCCESS | ERROR.

Example

```
%fsnewenv (environment=Default,
desc=A new environment,
path=C:\FSEnvironment,
reportspath=/MyContent/Reports,
wssserver=SASApp - Logical Workspace Server
)
```

%FSREGENV Macro

The %FSREGENV macro registers an existing directory structure as an environment and registers all projects in that directory.

Syntax

%FSREGENV (ENVIRONMENT=, PATH= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSREGENV macro if no default values are currently stored. The required arguments are separated by commas.

ENVIRONMENT=*environment-name*

specifies the name of the SAS Forecast Server environment. The value for this argument is case sensitive. The default environment is **Default**. Using the %FSSETDEF macro to set a global default value is not applicable in this instance.

PATH=*file path*

specifies the absolute file path to the base content directory on the specified workspace server.

Options

You can use the following options with the %FSREGENV macro. Options must follow the required arguments and are separated by commas.

DESC=*text*

specifies the environment description.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary

and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

REGISTERPROJECTS= YES | TRUE | NO | FALSE | 1 | 0
specifies whether to register in metadata any existing projects in the environment. The default value is **YES**.

REPORTSPATH= *UNIX-style path*
specifies a UNIX absolute path to the Base SAS Folder in metadata (for example, /**MyContent/Reports**). Searches for reports start in this directory.

WSSERVER= *workspace server-name*
specifies the name of the logical workspace server. The default value is **SASAPP - Logical Workspace Server**.

Results

The %FSREGENV global macro indicates whether the %FSREGENV macro finishes successfully or encounters errors: &FSREGENV=SUCCESS | ERROR.

Example

```
%fsregenv (environment=Default,
desc=A new environment,
path=C:\FSEnvironment,
reportspath=/MyContent/Reports,
wsserver=SASApp - Logical Workspace Server,
registerprojects=no
)
```

%FSPRJAGG Macro

The %FSPRJAGG macro opens existing Forecast Server projects and aggregates the top level reconciled forecasts.

Syntax

%FSPRJAGG (PROJECTS=,OUTFOR=,AGGFOR= [, options])

Details

Required Arguments

The following arguments are required with the %FSPRJAGG macro. The required arguments are separated by commas.

PROJECTS=*project-list*
specifies a list of Forecast Server project names. The value must be a valid SAS name. .

OUTFOR=*data- set-name*
specifies the SAS data set name that contains the result.

AGGFOR=*data- set-name*
specifies the SAS data set name that contains the aggregate forecast result.

Options

You can use the following options with the %FSPRJAGG macro. Options must follow the required arguments and are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

Results

The %FSPRJAGG global macro indicates whether the %FSPRJAGG macro finishes successfully or encounters errors: &FSPRJAGG=SUCCESS | ERROR.

Example

```
%fsprjagg (projects=projecta projectb projectc,
outfor=outdata,
aggfor=aggforecastdata
)
```

%FSPUBLISH Macro

The %FSPUBLISH macro opens an existing Forecast Server project, loads global macro variables, which describe the project, and publishes the project to a predefined SAS library. The macro must be submitted from a SAS session on the server.

Syntax

%FSPUBLISH (PROJECTS=,OUTFOR=,AGGFOR= [, options])

Details**Options**

You can use the following options with the %FSPUBLISH macro. Options are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for

ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

OUT=*output-data-set*

specifies the name of the output data set. The default is the project name, and the data set is stored in the temporary WORK library. A two-level name must be specified to create permanent output. The libref (first level) must be defined in metadata (see **Data Library Manager** in SAS Management Console). The user must have permission to write metadata to the shared data folder.

Results

The %FSPUBLISH global macro indicates whether the %FSPUBLISH macro finishes successfully or encounters errors: &FSPUBLISH=SUCCESS | ERROR.

Example

```
%fspublish (projectname=Shoe_Sales_2012,out=public.pd1)
```

%FSREGPRJ Macro

The %FSREGPRJ macro registers project information in metadata.

Syntax

```
%FSREGPRJ (PROJECTNAME= [ options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSREGPRJ macro if no default values are currently stored. The required arguments are separated by commas.

PROJECTNAME=*project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this argument, then the current stored default value for the project name is used. The value of this default cannot be customized with the %FSSETDEF macro.

Options

You can use the following options with the %FSREGPRJ macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSREGPRJ global macro indicates whether the %FSREGPRJ macro finishes successfully or encounters errors: &FSREGPRJ=SUCCESS | ERROR.

Example

```
%fsregprj (projectname=Shoe_Sales_2012,
environment=Default
)
```

%FSREN Macro

The %FSREN macro renames a SAS Forecast Server project.

Syntax

%FSREN (NEWPROJECTNAME= [options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSREN macro if no default values are currently stored.

NEWPROJECTNAME= *project-name*

specifies the new name of the SAS Forecast Server project. The name must be a valid SAS name.

Options

You can use the following options with the %FSREN macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

NOWARN= YES | NO

specifies whether to suppress errors if the SAS Forecast Server project does not exist. The default value is **NO**.

Results

The %FSREN global macro indicates whether the %FSREN macro finishes successfully or encounters errors: &FSREN=SUCCESS | ERROR.

Example

```
%fsren (projectname=Shoe_Sales,  
newprojectname=Shoe_Sales_2012  
)
```

%FSRMOVR Macro

The %FSRMOVR macro removes ALL overrides for all series specified in the SERIESDS= data set.

Syntax

%FSRMOVR (SERIESDS=,PROJECTNAME= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default

value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSRMOVR macro if no default values are currently stored. The required arguments are separated by commas.

SERIESDS= *data set*

specifies the SAS data set that contains the list of series. This can be a fully qualified data set name in the form of library.dataset.

The SERIESDS= data set contains the BY variables that describe the series. The following variables are optional.

BY variable	specifies a BY variable that describes the series. Add a column to the data set for each BY variable. (See the example below.) Include all BY variables, starting from the highest level down to the lowest level at which overrides are to be applied. For example, if the hierarchy of the project is (REGION, PRODUCT), and PRODUCT is the lowest level at which overrides are to be applied, then the list of BY variables in the SERIESDS= data set needs to include at least the variables REGION and PRODUCT. The data set does not need to be sorted in any way. However, it might be convenient for it to be sorted in the same order as the project data (that is, by the BY variables and time ID variable). The special character (.) indicates a missing value and that a series is at a non-leaf level or the root node.
DATE	(Required) specifies the time ID. It needs to be in the same format and frequency of the project time ID. (For example, if the project time ID has a date or datetime format, then the DATE column should have a matching date or datetime format.)
OVERRIDE	(Numeric) specifies the override value.
OLOCK	is a binary value. The value 0 = locked override. The value 1 (or any other number but 0) = unlocked override.
SELECTLOCK	(Optional) is a character variable. Possible values are ALL: all overrides are removed; LOCKED: only locked overrides are removed; and UNLOCKED: only unlocked overrides are removed. The default is ALL. If a value is not specified or the variable is not present, then all overrides are removed for the series.
FROM	(Optional) is a time ID value. Overrides are removed only from this date on. If the missing value character (.) is specified or the variable is not present, overrides are removed from the beginning of the series forecasts.
TO	(Optional) is a time ID value. Overrides are removed only until this date. If the missing value character (.) is specified or the variable is not present, overrides are removed until the end of the series forecasts.
DEPENDENT	(For non-hierarchical projects only) specifies the value of the depend series where the override is to be added.

Suppose the hierarchy consists of REGION, PRODUCT, and PRODUCTNAME. This is an example of a SERIESDS= data set.

REGION	PRODUCT	PRODUCTNAME	SELECTLOCK	FROM	TO
Region1	.	.	all	01JAN2003	31DEC2003
Region1	Line1	Product1	UNLOCKED	.	31DEC2003
Region1	Line1	Product2	LOCKED	01JAN2003	.

In the first line, the missing value character (.) in the PRODUCT and PRODUCTNAME columns indicates that the override applies to non-leaf series Region1. This means that all overrides are removed for all series with REGION=Region1 from 01JAN2003 to 31DEC2003. The hierarchy consists of REGION, PRODUCT, and PRODUCTNAME. Therefore, when the values for REGION, PRODUCT, and PRODUCTNAME are all present, it indicates that the series is at leaf level or the lowest level of the hierarchy. This means that in the second line, all UNLOCKED overrides are removed for all series with REGION=Region1, PRODUCT=Line1, and PRODUCTNAME=Product1 from the start date of the horizon to 31DEC2003. The missing value (.) for FROM in the second line indicates to use the start date of the horizon. In the third line, all LOCKED overrides are removed for all series with REGION=Region1, PRODUCT=Line1, and PRODUCTNAME=Product2 from 01JAN2003 until the end date of the horizon. The missing value (.) for TO in the third line indicates to use the end date of the horizon.

PROJECTNAME=*project-name*

specifies the SAS Forecast Studio project from which the overrides are to be removed. The project-name must be a valid SAS name. If you do not specify this option, the current stored default value for the project-name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Options

You can use the following options with the %FSRMOVR macro. Options must follow the required arguments. Options are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

RECONCILE= YES | NO

specifies whether the project should be reconciled after removing the overrides. The default is RECONCILE=YES. If you do not reconcile the project, the final forecasts do not reflect the changes in the override values. You can also use the %FSRUNPRJ macro with the METHOD=RECONCILE option to reconcile the project. For more information, see “%FSRUNPRJ Macro” on page 167.

OUTFAILED= *data set*

specifies the SAS data set that contains information about the series in the SERIESDS= data set for which no overrides were removed.

The OUTFAILED= data set contains the BY variables that describe the series and variables, as follows:

BY variable	specifies a BY variable that describes the series. A column exists in the data set for each BY variable.
DATE	specifies the time ID.
FAILCODE	numeric code that specifies the reason for the failure.
FAILREASON	string that describes the reason for the failure.

The supported values for FAILCODE and FAILREASON are as follows:

FAILCODE	FAILREASON
1	No matching series in the project
2	Existing override in OUTOVRD not replaced

Results

The %FSRMOVR global macro indicates whether the %FSRMOVR macro finishes successfully or encounters errors: &FSRMOVR= SUCCESS | ERROR.

Example

```
/*Create a SERIESDS= data set to remove overrides from hierarchical project*/
data _seriesds;
infile datalines delimiter= ",";
format regionName $7. productLine $5. productName $9.
      selectlock $8. from date9. to date9.;
informat from date9. to date9.;
input  regionName productLine productName selectlock from to ;
cards;
Region1,...,all,01JAN2003,31DEC2003
Region1,...,all,01JAN2003,31DEC2003
Region1,Line1,..,all,01JAN2003,31DEC2003
Region1,Line1,Product1,all,01JAN2003,31DEC2003
Region1,Line1,INVALID,all,01JAN2003,31DEC2003
Region1,Line1,Product2,all,01JAN2003,31DEC2003
;
data _seriesds;
set _seriesds;
if missing(regionName) then regionName=".";
if missing(productLine) then productLine=".";
if missing(productName) then productName=".";
run;
%fsrmovr(seriesds=_seriesds,
        projectName=pricedata_hier,
        environment=default,
        reconcile=YES,
        outfailed=outovr
        )
```

%FSRUNPRJ Macro

The %FSRUNPRJ macro opens an existing SAS Forecast Server project and runs the project at a given stage.

Syntax

```
%FSRUNPRJ ( METHOD=, IMPORTDATA= [ , options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSRUNPRJ macro if no default values are currently stored. The required arguments are separated by commas.

METHOD= CREATE | DESTRUCTIVE-DIAGNOSE | DIAGNOSE | SELECT | FIT | FORECAST | RECONCILE | OVERRIDES | CURRENT
specifies the stage when the project opens in SAS Forecast Studio.

IMPORTDATA= YES | TRUE | NO | FALSE | 1 | 0
specifies whether to update the data in the project with any new data in the input data set. The default value is **NO**, and any changes to the input data source are not included in the project.

Options

You can use the following options with the %FSRUNPRJ macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*
specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

LEAD= *n*
specifies the number of periods into the future in which multi-step forecasts are made. The default value is 12.

MIDTIER = *label*
identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*
specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the

project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on [page 170](#).

RETAINCHOOSE= YES | TRUE | NO | FALSE | 1 | 0

clears any user-specified model selections and resets the series to the model automatically selected by SAS Forecast Studio. The default value is **YES**.

Additional Considerations

If you are using the SAS Forecast Server Batch Interface, you must run the macro code on a SAS Workspace Server if your project meets the following criteria:

- You are using the %FSCREATE or %FSRUNPRJ macros.
- You are using a custom time interval or custom format.
- The macros depend on LIBNAME statements in the autoexec file in order to run.

Results

The %FSRUNPRJ global macro indicates whether the %FSRUNPRJ macro finishes successfully or encounters errors: &FSRUNPRJ=SUCCESS | WARNING | ERROR.

Note: A warning message is displayed if there are override conflicts or reconciliation failures.

A series of global macro variables are populated after a SAS Forecast Server project is loaded. These macro variables describe the project. Using these macro variables, you can write SAS code that is generic enough to use across multiple projects. For example, you can use this code to explore and use data in your SAS Forecast Server projects.

Example

```
%fsrunprj (projectname=Shoe_Sales_2012,
method=FORECAST,
importdata=NO,
retainchoose=YES
)
```

%FSRUNRPT Macro

The %FSRUNRPT macro runs the specified report and saves the output to a local directory. %FSRUNRPT works only with SAS Forecast Server deployments.

Syntax

%FSRUNRPT (REPORTPATH=,OUTPATH= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default

value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSRUNRPT macro if no default values are currently stored. The required arguments are separated by commas.

OUTPATH=*directory-path*

specifies the local directory where to save the output. An example is **C:\temp**.

REPORTPATH= *UNIX-style path*

specifies a UNIX absolute path to the Base SAS Folder in the metadata (for example, **/MyContent/Reports**). Searches for reports start in this directory. This argument is used only when you are creating an environment for a networked middle tier.

Options

You can use the following options with the %FSRUNRPT macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

ODSDEST=*ODS-destination*

specifies the ODS format for the output. Any ODS format works (for example, HTML, RTF, PDF, and so on). The default format is HTML.

ODSSTYLE=*ODS-style*

specifies the ODS style to apply to the results. The default style is Journal.

ODSOPTIONS=OPTIONS(*ODS-option*)

specifies the ODS options for the output where *ODS_option* is DELIMITER. There is no default.

Examples are as follows:

- odsoptions=options(Delimiter=';')
- odsoptions=options(Delimiter='09'x)

OUTFILE=*name-of-output-file*

specifies the name of the file that contains the output. By default, the name is assigned by the server.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see [“%FSSETDAT Macro” on page 170](#).

PROMPTDATA=*name-of-data-set*

specifies the data set that contains the parameter values for prompts. The data set must be formatted for two character columns named Name and Value. By default, any prompts are ignored.

PROMPTVALUEDELIM=*delimiter*

specifies the delimiter used when providing values for multi-value prompts. The default delimiter is a space.

SAVELOG= YES | TRUE | NO | FALSE | 1 | 0

specifies whether to save the log from the report. The log is saved in the same location as the output files. The default value is FALSE.

SERVERTYPE=WKS | PWKS | STP

specifies the server type to use when executing the report. The accepted values are WKS (workspace server), PWKS (pooled workspace server), and STP (stored process server). The default value is WKS.

The pooled workspace server and stored process server run under a server identity. The ability to select a workspace server is provided through a configuration property available from SAS Management Console. For more information, see [Enable Server Selection for Reports on page 39](#).

UNIT=*directory-path*

specifies the path to the focus unit. By default, the root unit has the focus.

UNITDELIM=*delimiters*

specifies the delimiters for the unit path. By default, the delimiters are ~ and |.

Results

The %FSRUNRPT global macro indicates whether the %FSRUNRPT macro finishes successfully or encounters errors: &FSRUNRPT=SUCCESS |ERROR.

Example

```
%fsrunrpt (reportpath=Samples/Getting Started
/Introduction/Introduction to Stored Processes,
outpath=C:\temp,
environment=Default,
projectname=Shoe_Sales_2012,
odsdest=RTF,
odsoptions=options(Delimiter='$'),
serverType=PWKS
)
```

%FSSETDAT Macro

The %FSSETDAT macro enables you to change the input data set after the project has been created.

Syntax

```
%FSSETDAT ( DATA= [ , options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSSETDAT macro if no default values are currently stored.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

DATA= *input-data-set*

specifies the new input data set. The value must be in the *libname.memname* format.

Options

You can use the following options with the %FSSETDAT macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSSETDAT global macro variable indicates whether the %FSSETDAT macro terminates successfully or encounters errors: &FSSETDAT= SUCCESS | ERROR.

Example

```
%fssetdat (data=fslib.fsprj,
projectname=Shoe_Sales_2012,
midtier=server1,
environment=Default
)
```

%FSSETDEF Macro

The %FSSETDEF macro stores a default value for an argument so that it can be used by later calls to the other macros.

Note: Currently, you can use the %FSSETDEF macro to store default values for the ENVIRONMENT= and PROJECTNAME= arguments (and variants) of most macros.

Syntax

%FSSETDEF (MIDTIER=,KEY=,VALUE=)

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

All arguments are required and must be separated by commas.

KEY=*key-name*

specifies the key to associate with the value specified in the VALUE variable. This key specifies what type of variable the default is being set for. Here are the valid key values:

environment	specifies that the value is for an environment variable (for example, ENVIRONMENT, SOURCEENVIRONMENT, DESTINATIONENVIRONMENT).
project	specifies that the value is for a project name variable (for example, PROJECTNAME, SOURCEPROJECTNAME, DESTINATIONPROJECTNAME).

MIDTIER=*label*

identifies which middle tiers can use this stored default. The special “*” value can be used to indicate a default for all middle tiers. A * default is used only when the middle tier has not been assigned an explicit default value. If you are not explicitly setting the MIDTIER= value, the value automatically assigned by %FSLOGIN should be passed. For more information, see “%FSLOGIN Macro” on page 149..

VALUE=*default-argument-value*

specifies the value to store with the key. This value is then used as the default argument value as appropriate by macros that support that configured default.

Options

There are no optional arguments for the %FSSETDEF macro.

Results

The %FSSETDEF global macro indicates whether the %FSSETDEF macro finishes successfully or encounters errors: &FSSETDEF=SUCCESS | ERROR | WARNING.

Example

```
%fssetdef (midtier=fsmain,
key=projectname,
value=sales
)
```

%FSSETOVR Macro

The %FSSETOVR macro adds the overrides specified in the OVRDS= data set to a project.

Syntax

```
%FSSETOVR (OVRDS=,PROJECTNAME= [, options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSSETOVR macro if no default values are currently stored. The required arguments are separated by commas.

OVRDS= data set

specifies the SAS data set that contains the list of overrides. This can be a fully qualified data set name in the form of library.dataset.

The OVRDS= data set needs to contain the BY variables that describe the series and variables as follows:

BY variable	specifies a BY variable that describes the series. Add a column to the data set for each BY variable. (See the example below.) Include all BY variables, starting from the highest level down to the lowest level at which overrides are to be applied. For example, if the hierarchy of the project is (REGION, PRODUCT), and PRODUCT is the lowest level at which overrides are to be applied, then the list of BY variables in the OVRDS= data set needs to include at least the variables REGION and PRODUCT. The data set does not need to be sorted in any way. However, it might be convenient for it to be sorted in the same order as the project data (that is, by the BY variables and time ID variable). The special character (.) indicates a missing value and that a series is at a non-leaf level or the root node.
DATE	(Required) specifies the time ID. It needs to be in the same format and frequency of the project time ID. (For example, if the project time ID has a date or datetime format, then the DATE column should have a matching date or datetime format.)

OVERRIDE	(Required and numeric) specifies the override value.
OLOCK	(Required) is a binary value. The value 0 = locked override. The value 1 (or any other number but 0) = unlocked override.
REPLACE	(Optional) is a binary value. The value 0 = do not replace existing override. The value 1 = replace existing override. The default is 1. If REPLACE=1, any existing override in the corresponding OUTOVRD data set is replaced with the value in the OVRDS= data set. If REPLACE=0, the value in the OUTOVRD data set is not replaced. If the REPLACE variable is not specified, it is assumed to be equal to 1 for all overrides in the data set. (See the OUTFAILED= data set.)
DEPENDENT	(For non-hierarchical projects only) specifies the value of the depend series where the override is to be added.

Suppose the hierarchy consists of REGION and PRODUCT. This is an example of an OVRDS= data set.

REGION	PRODUCT	DATE	OVERRIDE	OLOCK
region1	green shirt	01/01/2010	500	0
region1	red shirt	01/01/2010	500	0
region1	.	01/01/2010	2500	1
.	.	01/01/2010	5000	0

The hierarchy consists of REGION and PRODUCT. Therefore, when the values for REGION and PRODUCT are both present, it indicates that the series is at leaf level or the lowest level of the hierarchy. This means that in the first line, the locked override value of 500 replaces the forecast value of all series with REGION=region1 and PRODUCT=green shirt for the month of January 2010. In the third line, the missing value (.) in the PRODUCT column indicates that the override applies to non-leaf series region1. This means that the unlocked override value of 2500 replaces the forecast value of all series with REGION=region1 for the month of January 2010. In the fourth line, the missing value character (.) in the REGION and PRODUCT columns indicates that the override applies to the root level of all series. This means that the locked override value of 5000 replaces the forecast value of all series for the month of January 2010. Because the REPLACE variable has not been specified, all overrides replace any existing overrides.

PROJECTNAME=*project-name*

specifies the SAS Forecast Studio project from which the overrides are to be applied. The project-name must be a valid SAS name. If you do not specify this option, the current stored default value for the project-name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see [“%FSSETDAT Macro” on page 170](#).

Options

You can use the following options with the %FSSETOVR macro. Options must follow the required arguments. Options are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

RECONCILE= YES | NO

specifies whether the project should be reconciled after applying the overrides. The default is RECONCILE=YES. If you do not reconcile the project, the final forecasts do not reflect the changes in the override values. You can also use the %FSRUNPRJ macro with the METHOD=RECONCILE option to reconcile the project. For more information, see “%FSRUNPRJ Macro” on page 167.

OUTFAILED= *data set*

specifies the SAS data set that contains information about overrides in the OVRDS= data set that were not applied to the project.

The OUTFAILED= data set contains the BY variables that describe the series and variables, as follows:

BY variable specifies a BY variable that describes the series. A column exists in the data set for each BY variable.

DATE specifies the time ID.

FAILCODE numeric code that specifies the reason for the failure.

FAILREASON string that describes the reason for the failure.

The supported values for FAILCODE and FAILREASON are as follows:

FAILCODE	FAILREASON
1	No matching series in the project
2	Existing override in OUTOVRD not replaced

Results

The %FSSETOVR global macro indicates whether the %FSSETOVR macro finishes successfully or encounters errors: &FSSETOWN= SUCCESS | ERROR.

Example

```
/*Create a ovrds= data set for a non-hierarchical project using PRICEDATA.*/
/* The dependent column is included because the project is non-hierarchical.*/
data _ovrds_nh;
infile datalines delimiter= ",";
format regionName $7. productLine $5. productName $9.
date date9. dependent $5.;
informat date date9.;
input regionName productLine productName date OVERRIDE OLOCK dependent;
cards;
Region1,Line1,Product1,01JAN03,420.08563234,0,sale
Region1,Line1,Product1,01FEB03,441.66454424,0,sale
Region1,Line1,Product1,01MAR03,449.95226719,0,price
Region1,Line1,Product1,01APR03,438.08158443,0,price
Region1,Line1,.,01JAN03,1198.7663702,0,sale
```

```

Region1,Line1,,01FEB03,1277.9928575,0,sale
Region1,Line1,,01MAR03,1366.9543514,0,price
Region1,Line1,,01APR03,1274.8691429,0,price
Region1,,,01JAN03,1198.7663702,0,sale
Region1,,,01FEB03,1277.9928575,0,sale
Region1,,,01MAR03,1366.9543514,0,price
Region1,,,01APR03,1274.8691429,0,price
Region1,Line1,INVALID,01JAN03,420.08563234,0,sale
;
data _ovrds_nh;
set _ovrds_nh;
if missing(regionName) then regionName=".";
if missing(productLine) then productLine=".";
if missing(productName) then productName=".";
run;

%fssetovr(ovrds=_ovrds_nh,
         projectName=pricedata_nonhier,
         environment=default,
         reconcile=YES,
         outfailed=test_outfailed_nonhierarchy
        )

```

%FSSETOWN Macro

The %FSSETOWN macro assigns an owner to a SAS Forecast Server project.

Syntax

%FSSETOWN (OWNER= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following arguments are required with the %FSSETOWN macro if no default values are currently stored. The required arguments are separated by commas.

OWNER= *user-name*

specifies the user name that owns the SAS Forecast Server project.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

Options

You can use the following options with the %FSSETOWN macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSSETOWN global macro indicates whether the %FSSETOWN macro finishes successfully or encounters errors: &FSSETOWN=SUCCESS| ERROR.

Example

```
%fssetown (projectname=Shoe_Sales_2012,
owner=sastrust,
environment=Default
)
```

%FSSETPUB Macro

The %FSSETPUB macro enables public (shared) access to a SAS Forecast Server project for all users of the environment that contains that project.

Syntax

%FSSETPUB (ISPUBLIC= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSSETPUB macro if no default values are currently stored. The required arguments are separated by commas.

ISPUBLIC= TRUE | FALSE | YES | NO | 1 | 0

specifies whether the project should be available to all users.

Options

You can use the following options with the %FSSETPUB macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see [“%FSSETDEF Macro” on page 172](#).

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see [“%FSSETDAT Macro” on page 170](#).

Results

The %FSSETPUB global macro indicates whether the %FSSETPUB macro finishes successfully or encounters errors: &FSSETPUB=SUCCESS| ERROR.

Example

```
%fssetpub (project=Shoe_Sales_2012,
ispublic=YES,
environment=Default
)
```

%FSUNREG Macro

The %FSUNREG macro unregisters an existing SAS Forecast Server project from the metadata server. The project files will not be deleted from the disk. If needed, you can register the project again from the project files.

Syntax

```
%FSUNREG (PROJECTNAME= [ , options ] )
```

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default

value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSUNREG macro if no default values are currently stored.

PROJECTNAME=*project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this argument, then the current stored default value for the project name is used. The value of this default cannot be customized with the %FSSETDEF macro.

Options

You can use the following options with the %FSREGPRJ macro. Options must follow the required arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSUNREG global macro indicates whether the %FSUNREG macro finishes successfully or encounters errors: &FSUNREG=SUCCESS | ERROR.

Example

```
%fsunreg (projectname=Shoe_Sales_2012,
environment=Default
)
```

%FSUNRENV Macro

The %FSUNRENV macro unregisters a SAS Forecast Server environment. Unregistering an environment removes the existing SAS Forecast Server projects and environment from the metadata server without deleting the project files from the disk. You can use the %FSREGENV macro variable to create an environment and register its projects.

Syntax

%FSUNRENV (ENVIRONMENT= [, options])

Details

Required Arguments

Note: Some required arguments do not have to be specified by the user in the macro call for the requirement to be satisfied. In some cases, when a user does not specify a required argument, the macro attempts to resolve it with a default value previously stored in the macro run-time environment. If the required argument has a default value (either the system default value or a value set with %FSSETDEF), then the argument does not have to be specified in the macro.

The following argument is required with the %FSUNRENV macro if no default values are currently stored.

ENVIRONMENT=*environment-name*

specifies the name of the SAS Forecast Server environment. For security reasons, this argument has no default, and must be explicitly entered. Using the %FSSETDEF macro to set a global default value is not applicable in this instance.

Options

You can use the following options with the %FSUNRENV macro. Options must follow the required arguments and are separated by commas.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

Results

The %FSUNRENV global macro indicates whether the %FSUNRENV macro finishes successfully or encounters errors: &FSUNRENV=SUCCESS | ERROR.

Example

```
%fsunrenv (environment=Default,
midtier=server1
)
```

%FSUPDATE Macro

The %FSUPDATE macro updates the settings of an existing SAS Forecast Server project.

Note: Most of the arguments supported by the %FSCREATE macro are supported. The exceptions are BY, DATA, HIERARCHY, ID, RUN, and VAR. DATA can be changed using the %FSSETDAT macro.

Syntax

```
%FSUPDATE ( [ options ] )
```

Details

Options

The following options can be used with the %FSUPDATE macro. Options must follow the required arguments, and are separated by commas.

ACCUMULATE= TYPE [TYPE(var1var2) TYPE(var3var4) ...]

specifies the accumulation options for the dependent, input, and reporting variables.

TIP In SAS Forecast Studio, the **Set accumulation to the value used for aggregation** option is checked by default. If the ACCUMULATE= or AGGREGATE= arguments are used, the **Set accumulation to the value used for aggregation** option is not selected. This setting persists, so the next time the project is opened in SAS Forecast Studio, the **Set accumulation to the value used for aggregation** option still is not selected.

Here are the valid values for TYPE:

NONE	specifies that no accumulation occurs; the ID variable values must be equally spaced with respect to the frequency.
TOTAL	specifies that observations are accumulated based on the total sum of their values. This is the default value.
AVERAGE AVG	specifies that observations are accumulated based on the average of their values.
MINIMUM MIN	specifies that observations are accumulated based on the minimum of their values.
MEDIAN MED	specifies that observations are accumulated based on the median of their values.
MAXIMUM MAX	specifies that observations are accumulated based on the maximum of their values.
N	specifies that observations are accumulated based on the number of nonmissing observations.
NMISS	specifies that observations are accumulated based on the number of missing observations.
NOBS	specifies that observations are accumulated based on the number of observations.
FIRST	specifies that observations are accumulated based on the first of their values.
LAST	specifies that observations are accumulated based on the last of their values.
STDDEV STD	specifies that observations are accumulated based on the standard deviation of their values.
CSS	specifies that observations are accumulated based on the corrected sum of squares of their values.
USS	specifies that observations are accumulated based on the uncorrected sum of squares of their values.

Here are some examples:

- To use an average accumulation for all variables, specify **ACCUMULATE=AVG**.

- To use a total accumulation for all variables except for var1, which uses an average accumulation, specify **ACCUMULATE= TOTAL AVG(var1)**.
- To use a total accumulation for all variables except var1 and var2, which use an average accumulation, and var3, which accumulates the observations based on standard deviations, specify **ACCUMULATE=TOTAL AVG(var1 var2) STD(var3)**.

ADJUST= var1(var2var3) [/operation=(pre,post)]

Specifies the adjustment options for the dependent variables. If not specified, the default value for pre-adjust and post-adjust is NONE.

Here are the valid pre-adjust and post-adjust values:

- NONE
- ADD
- SUBTRACT
- MULTIPLY
- DIVIDE
- MIN
- MAX

Here are some examples:

- To adjust the sale variable with the values of the price1 variable, specify **ADJUST=sale(price1)**.
- To adjust the sale variable with the values of the price1 variable and to adjust the price variable with the values of the price2 variable, specify **ADJUST=sale(price1) price(price2)**.
- To adjust the sale variable with the values of the price1 and price2 variables, specify **ADJUST=sale(price1 price2)**.
- To adjust the sale variable with the values of the price1 variable before any forecasts are generated, specify **ADJUST=sale(price1) / operation=(ADD, NONE)**.

AGGREGATE= TYPE [TYPE(var1var2) TYPE(var3var4) ...]

specifies the aggregation options for the dependent, input, and reporting variables. Aggregation is valid only if HIERARCHY=YES. See the ACCUMULATE option for valid values and examples.

Note: For the dependent variable, the only valid values are TOTAL | AVERAGE | AVG.

TIP In SAS Forecast Studio, the **Set accumulation to the value used for aggregation** option is checked by default. If the ACCUMULATE= or AGGREGATE= arguments are used, the **Set accumulation to the value used for aggregation** option is not selected. This setting persists, so the next time the project is opened in SAS Forecast Studio, the **Set accumulation to the value used for aggregation** option still is not selected.

ALLOWNEGATIVE= YES | TRUE | NO | FALSE | 1 | 0

specifies whether negative forecasts are allowed. The default value is NO.

ALPHA= n

specifies the confidence level for the series. The default value is 0.05, which is a 95% confidence level.

ARIMAX= YES | TRUE | NO | FALSE | 1 | 0

specifies whether ARIMA models should be considered. The default value is YES.

BACK= *n*

specifies the out-of-sample range (the number of periods from the end). The default value is 0.

COMBINE = YES | TRUE | NO | FALSE | 1 | 0

specifies whether SAS Forecast Server should automatically generate a combined model for the set of time series models that is generated from the diagnosis of each time series. The default value is **NO**.

COMBINECRITERION = *selection-criterion*

specifies the statistic of fit to use when ranking the combined models. The default value is **RMSE**.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMAX option, the UCM option, or both the ARIMAX and UCM options are set to YES or TRUE.

Here are the possible values for the selection criterion:

- AADJRSQ — Amemiya's adjusted R-squared
- ADJRSQ — adjusted R-squared
- AIC — Akaike information criterion
- AICC — Akaike information criterion, finite sample size corrected
- APC — Amemiya's prediction criterion
- GMAPE — geometric mean percent error
- GMAPES — geometric mean absolute error percent of standard deviation
- GMAPPE — geometric mean predictive error
- GMASPE — geometric mean symmetric percent error
- GMRAE — geometric mean relative absolute error
- MAE — mean absolute error
- MAPE — mean absolute percent error
- MAPES — mean absolute error percent of standard deviation
- MAPPE — mean absolute predictive symmetric percent error
- MASE — mean absolute scaled error
- MAXERR — maximum error
- MAXPE — maximum percent error
- MAXPPE — maximum predictive percent error
- MAXRE — maximum relative error
- MAXSPE — maximum symmetric percent error
- MDAPE — median absolute percent error
- MDAPES — median absolute percent error percent of standard deviation
- MDAPPE — median absolute predictive percent error

- MDASPE — median absolute symmetric percent error
- MDRAE — median relative absolute errors
- ME — mean error
- MINERR — minimum error
- MINPE — minimum percent error
- MINPPE — minimum predictive percent error
- MINRE — minimum relative error
- MINSPE — minimum symmetric percent error
- MPE — mean percent error
- MPPE — mean predictive percent error
- MRAE — mean relative absolute error
- MRE — mean relative error
- MSPE — mean symmetric percent error
- MSE — mean square error
- RMSE — root mean square error
- RSQUARE — R-squared
- RWRSQ — random walk R-squared
- SBC — Schwarz Bayesian information criterion
- SMAPE — mean absolute symmetric percent error
- SSE — sum of squared errors
- UMSE — unbiased mean square error
- URMSE — unbiased root mean square error

COMBINEENCOMPASS = OLS | HLN

specifies the forecast encompassing test to use to remove any models that contain redundant information. The default value is **OLS**.

Note: This option is valid only if the following conditions are met:

- the COMBINE option is set to YES or TRUE
- the ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE

Here are the valid values:

OLS uses an ordinary least squares (OLS) regression test to estimate pairwise encompassing between candidate forecasts.

HLN uses the Harvey-Leybourne-Newbold (HLN) test to estimate pairwise encompassing between candidate forecasts.

COMBINEMETHOD = AVERAGE | AICC | OLS | ERLS | LAD | NERLS | NRLS | RMSEWGT

specifies the method for determining the combination weights. These combination weights are used to calculate the weighted average of the candidate forecasts. The default method is **AVERAGE**.

Note: This option is valid only if the following conditions are met:

- the COMBINE option is set to YES or TRUE
- the ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE

Here are the valid values:

AVERAGE	computes the simple average of the forecasts that you selected for combination. This is the default.
AICC	uses the Akaike's information criterion to compute the combination weights based on corrected AIC weights.
OLS	uses ordinary least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast residuals.
ERLS	uses equally restricted least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast subject to the constraint that the weights sum to 1.
LAD	uses a measure of fit based on the least absolute deviation to compute the combination weights.
NERLS	uses nonnegative, equality restricted least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast residuals subject to the constraints that the weights sum to 1 and be nonnegative.
NRLS	uses nonnegative restricted least squares to compute the combination weights. The results minimize the l_2 norm of the combined forecast residuals subject to the constraints that the weights be nonnegative.
RANKWGT	assigns weights using the rank of the candidate forecasts when combination is performed. The weights must sum to 1. If not, the last value in the list is adjusted, and a warning is issued. The weights are assigned by ranking the candidate forecasts from best to worst. The best uses the first weight, and so on. The set of weights used is normalized to account for candidates that fail to forecast or for candidates that are omitted from the final combination because of any exclusion tests.
RMSEWGT	uses the weights of the root mean square error to compute the combination weights. The weights are normalized to sum to 1.

COMBINEMISSMODE = MISSING | RESCALE

specifies the method for treating missing values in the forecast combination. In a given time series across the combination ensemble, one or more combination contributors can have a missing value. This setting determines the treatment of those in the final combination for such time indices. The default value is **MISSING**.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMAX option, the UCM option, or both the ARIMAX and UCM options are set to YES or TRUE.

MISSING	generates a missing combined forecast at each time index with one or more missing contributors. This method is the default for AICC weights, OLS weights, restricted least squares weights, and LAD
---------	---

weights. You cannot rescale the combination weights if you selected the OLS-based regression test to estimate pairwise encompassing. In this test, the estimated weights are not constrained to sum to one.

RESCALE rescales the combination weights for the nonmissing contributors at each time index to sum to 1. This method is the default for simple average, user-specified weights, ranked user weights, ranked weights, and root mean square error (RMSE) weights.

COMBINEMISSPCT = *integer*

specifies a threshold for the percentage of missing values in the combination estimation region. This threshold is used to exclude a candidate forecast from consideration in the final combination. By default, no missing percentage test is performed on candidate forecasts. Valid values are 1 to 100.

The COMBINEMISSPCT option and the HORMISSPERCENT option are independent of each other. You can specify one or both of these options.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMAX option, the UCM option, or both the ARIMAX and UCM options are set to YES or TRUE.

COMBINESTDERR = DIAG | ESTCORR

specifies the method for computing the prediction error variance series. This series is used to compute the prediction standard error, which in turn is used to compute confidence bands on the combined forecast. The default value is **DIAG**.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE.

CRITERION = *options*

specifies the model selection criterion (statistic of fit) to be used to select from several candidate models. The default value is MAPE. For more values, see the COMBINECRITERION option.

DESCRIPTION = *text*

specifies the project description.

DETECTOUTLIERS = YES | TRUE | NO | FALSE | 1 | 0

specifies whether outliers in the data should be detected when fitting an ARIMA model. The default value is **NO**.

DISAGGREGATION = PROPORTIONS | EQUALSPLIT

specifies the disaggregation method for reconciliation. The default value is **PROPORTIONS**.

ENCOMPASSALPHA = *value*

specifies the significance level (or alpha value) for the forecast encompassing test. Valid values are 0 to 1 (inclusive). The default value is **0.05**. See also the COMBINEENCOMPASS option.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE.

ENDZEROSMAXNUM = *integer*

specifies the maximum number of trailing zeros for a nonzero model. The default value is 0.

ENDZEROSMAXPCT = *integer*

specifies the maximum percentage of trailing zeros for a nonzero model relative to the number of nonzero values in the entire series. The default value is 0.

ENDZEROSMINOBS = *integer*

specifies a threshold for the series length that is required to enable the ENDZERO test. The default value is 0.

ENVIRONMENT = *environment-name*

specifies the name of the product environment. The value for this option is case sensitive. The default environment is Default. A global default value for ENVIRONMENT can be specified with the %FSSETDEF macro. For more information, see “%FSSETDEF Macro” on page 172.

ESM= YES | TRUE | NO | FALSE | 1 | 0

specifies whether exponential smoothing models (ESM) should be considered. The default value is **YES**.

ESMONLYATLEVEL= *BY- variable*

for hierarchical projects, specifies the highest level to fit the exponential smoothing models (ESM). Use this option to improve performance for projects with large hierarchies. The argument specified must be the name of a BY variable. By default, no level is specified for fitting the ESM models.

HOLDOUT= *n* | NO | FALSE

specifies the number of periods to be used as the holdout sample for model selection. If you do not want to specify a holdout sample, set this option to NO or FALSE.

HORMISSPCT=*integer*

specifies a threshold for the percentage of missing forecast values in the combination horizon. This threshold is used to exclude a candidate forecast from consideration in the final combination. Valid values range from 1 to 100 (inclusive).

This option is different from the COMBINEMISSPCT option, which is for the estimation region. The COMBINEMISSPCT option and the HORMISSPERCENT option are independent of each other. You can specify one or both of these options.

Note: This option is valid only if the following conditions are met:

- The COMBINE option is set to YES or TRUE.
- The ARIMA option, the UCM option, or both the ARIMA and UCM options are set to YES or TRUE.

HOLDOUTPCT= *n* | NO | FALSE

specifies the maximum percentage of the series length to be used as the holdout sample for model selection. The holdout percentage guards against errors that are caused by holdout samples that are too large for the selected time series. For example, suppose a series has a length of 100 periods (excluding any leading or trailing missing values). If the HOLDOUT= option is set to 20 and the HOLDOUTPCT= option is set to 10, the holdout sample contains 10 periods.

If you do not specify a value for the HOLDOUTPCT= option, the default value is 5.

IDFORMAT= *date*

specifies the date-and-time format of the values in the time ID variable.

INPUT= *variable*

specifies the input (or independent) variables. Separate multiple variable names by a space.

INTERMITTENT= *n* | NO | FALSE

specifies a number greater than 1 that is used to determine whether a time series is intermittent. The default value is 2. To disable this test, set this option to NO or FALSE.

INTERVAL= *interval-measure*

specifies the time interval of the time ID variable. If a time interval is not specified, SAS Forecast Server detects the time interval from the data.

Here is the list of valid values:

- for SAS date values: YEAR, SEMIYEAR, QTR, MONTH, SEMIMONTH, TENDAY, WEEK, WEEKDAY, DAY, YEARV, R445YR, R454YR, R544YR, R445QTR, R454QTR, R544QTR, R445MON, R454MON, R544MON, and WEEKV
- for SAS datetime values: HOUR, MINUTE, and SECOND

Note: Add a DT prefix to any of the date intervals to create the datetime interval.

Abbreviations or aliases of these intervals are not valid values. You can add multiplier and shift parameters to any of these intervals. For the WEEKDAY interval, you can add values for the weekend day. Custom interval names are also allowed. For more information, see “Time Intervals” in the *SAS/ETS User’s Guide*.

LEAD= *n*

specifies the number of periods into the future in which multiple step forecasts are made. The larger the horizon value, the larger the prediction error variance at the end of the horizon. The default value is 12.

MAXNUMOUTLIERS= *n*

specifies the maximum number of outliers to include in ARIMAX models. The default value is 2.

MAXPCTOUTLIERS= *n*

specifies the maximum number of outliers to include in ARIMAX models as a percentage of the series length, not including beginning, and ending missing values. The default value is 2.

If you specify both the MAXNUMOUTLIERS and MAXPCTOUTLIERS options, then SAS Forecast Server uses the smaller value. For example, you set MAXNUMOUTLIERS=5 and MAXPCTOUTLIERS=10. The maximum number of the outliers is 5% or 10% of the series length, whichever is smaller.

MIDTIER = *label*

identifies which middle tier session to use. This option is needed only when you are using multiple middle tier sessions in the same script, which is usually unnecessary and should be avoided. If the MIDTIER= option is not specified, SAS Forecast Server uses the most recently created session if it is still open. If no such session exists or the session has been closed, then an error is reported.

MINOBSTREND= *n*

specifies the minimum number of observations needed for a trend model. The default value is 2.

MINOBSNONMEAN= *n*

specifies the minimum number of observations needed for a non-mean model. The default value is 1.

MINOBSSEASON= *n*

is used to determine the number of observations required for a seasonal model. The value of the MINOBSSEASON option is multiplied by the seasonal cycle length. In order for a seasonal model to be fit to a series, the model must have more observations than the calculated value. Models with fewer observations are not fit.

The value of MINOBSSEASON must be greater than or equal to 1. The default value is 2.

MODELSELECTIONLIST= *SAS-data-set*

specifies a model selection list that contains models that you can use. The default value is **SASHELP.HPFDFLT.TSFSSELECT**.

PROJECTNAME = *project-name*

specifies the name of the SAS Forecast Server project. The name must be a valid SAS name. If you do not specify this option, the current stored default value for the project name is used. A global default value for PROJECTNAME can be specified with the FSSETDEF macro. For more information, see “%FSSETDAT Macro” on page 170.

PUBLICACCESS= YES | TRUE | NO | FALSE | 1 | 0

specifies whether the project can be opened by other users. The default value is **NO**.

RECONCILIATION= TOPDOWN | TD | BOTTOMUP | BU | MIDDLEOUT(*BY-variable*) | MO(*BY-variable*) | NO | FALSE

specifies whether SAS Forecast Server reconciles the hierarchy. By default, SAS Forecast Server reconciles the hierarchy using the top-down reconciliation method (TOPDOWN or TD). To reconcile the hierarchy by a middle level, you must specify the BY variable for the level where the reconciliation starts.

Here are some examples of how to specify this option:

- To specify a top-down reconciliation, use RECONCILIATION=TOPDOWN.
- To specify a middle-out reconciliation that starts at the *regionName* variable, use RECONCILIATION=MIDDLEOUT(*regionName*).
- If you do not want to reconcile the hierarchy, specify RECONCILIATION=NO (or FALSE).

REPORTING= *variable*

specifies the reporting (or auxiliary) variables for the project. Separate multiple variable names by a space.

REQUIRED = *type(mod) [var1 var2 var3] type(mod) [var1 var2 var3]*

specifies whether to include the input variables in the model, where

- *type* = YES|NO|MAYBE
- *mod*=POSITIVE|NEGATIVE

Note: The values for the REQUIRED= option are case sensitive. In your macro code, use uppercase letters.

- NO — specifies that the input variables be included in the model as long as their parameters are significant and the increment of the value of criterion exceeds a threshold. The default is REQUIRED=NO.
- MAYBE — specifies that the input variables be included in the model as long as their parameters are significant.
- YES — specifies that the input variables be included in the model as long as the model does not fail to be diagnosed.

Either the POSITIVE or NEGATIVE option with parentheses can follow *type*. This is optional. For example, specifying REQUIRED=YES(POSITIVE) drops the input variable from the model if its coefficient is negative, while specifying REQUIRED=YES(NEGATIVE) implies the opposite. The specification of POSITIVE or NEGATIVE does not mean that constraints are imposed during the estimation of the variable's coefficient in the model.

Here are some examples:

- REQUIRED=MAYBE YES[*var1*] — All variables (except variable1) with significant parameters are included in the model. variable1 is included in the model as long as the model does not fail to be diagnosed.
- REQUIRED=MAYBE NO[*var1 var2*] YES(POSITIVE)[*var3*] — Variables 1 and 2 are included in the model as long as their parameters are significant and the increment of the value of criterion exceeds a threshold. Variable 3 is included in the model if its coefficient is positive and the model does not fail to be diagnosed. All remaining variables are included in the model as long as their parameters are significant.

SEASONALITY= *n*

specifies the length of a season. When possible, SAS Forecast Server determines the seasonal cycle length from the time ID variable. You can specify a seasonal cycle length other than the default if you want to model a cycle in the data. For example, if your data contains a 13-week cycle, then use SEASONALITY=13.

SEASONTEST= *n* | NO | FALSE

specifies the sensitivity of the seasonality test. You can specify no seasonality test or specify a significance probability value for the test. Series with strong seasonality have small test probabilities. A significance probability value of 0 always implies seasonality. A significance probability value of 1 always implies no seasonality.

The default value is 0.01. To disable the test, set this option to NO or FALSE.

SETMISSING= 0 | MISSING | AVG | MIN | MED | MAX | FIRST | LAST | PREV | NEXT

specifies how to replace the missing values in the data. The default value is **MISSING**.

- 0 - The missing values are set to 0.
- MISSING - The missing values are set to missing. This is the default value.
- AVG - The missing values are set to the accumulated average value.
- MIN - The missing values are set to the accumulated minimum value.
- MED - The missing values are set to the accumulated median value.
- MAX - The missing values are set to the accumulated maximum value.
- FIRST - The missing values are set to the accumulated first nonmissing value.
- LAST - The missing values are set to the accumulated last nonmissing value.
- PREV - The missing values are set to the previous accumulated nonmissing value. Missing values at the beginning of the accumulated series remain missing.
- NEXT - The missing values are set to the next accumulated nonmissing values. Missing values at the end of the accumulated series remain missing.

START= *date*

specifies the first date to use in data preparation. The date format is ddmmmyyyy and the date-and-time format is ddmmmyyyy:hh:mm:ss. The default value is the earliest date in the input data set.

TIMEMULTIPLIER= *n*

specifies the multiplier for the time interval that you specified by using the **INTERVAL=** option. You could have specified the multiplier in the **INTERVAL=** option. For example, if you specified **INTERVAL=WEEK2**, then the time interval is **WEEK** and the multiplier is 2.

TIMESHIFT= *n*

specifies the offset for the time interval that you specified in the **INTERVAL=** option. You could have specified this offset in the **INTERVAL** option. For example, if you specified **INTERVAL=WEEK.2**, then the time interval is **WEEK**, and the time shift is 2 periods.

TRANSBOXCOX= *n*

specifies the Box-Cox value if the transformation type (specified by the **TRANSTYPE** option) is **BOXCOX**. Valid values range from -5 to 5 (inclusive). The default is **1**.

TRANSOPT= MEAN | MEDIAN

specifies how to calculate the forecasts for the system-generated models. The default is **MEAN**.

TRANSTYPE= AUTO | LOG | NONE | SQRT | LOGISTIC | BOXCOX

specifies the transformation to apply to the dependent variable or independent variable. This transformation applies to the system-generated models.

Here is a description for each valid value:

AUTO	automatic transformation. SAS Forecast Studio uses the model selection criteria to determine whether to apply a logarithmic transformation or no transformation.
LOG	logarithmic transformation. This option is not available for negative series.
NONE	no transformation. This is the default.
SQRT	square-root transformation.
LOGISTIC	logistic transformation.
BOXCOX	Box-Cox transformation. If you select this type of transformation, then you can specify a value for this transformation using the TRANSBOXCOX option.

TRIMMISS= NONE | LEFT | RIGHT | BOTH

specifies how missing values are removed from the accumulated time series. The default value is **NONE**. If the project is hierarchical with reconciliation, only **NONE** is allowed.

Here is a description of each valid value:

NONE	The missing values are kept.
LEFT	The beginning missing values are removed.
RIGHT	The ending missing values are removed.
BOTH	Both the beginning and ending missing values are removed.

UCM= YES | TRUE | NO | FALSE | 1 | 0

specifies whether SAS Forecast Server should automatically generate an unobserved component model (UCM). The default value is **NO**.

WEEKENDDDAYS= 1 | 2 | 3 | 4 | 5 | 6 | 7

specifies which days are the weekend (or inactive) days in the week. Valid only when INTERVAL=WEEKDAY. The days are specified as numbers (1 through 7) representing the days of the week (Sunday through Saturday).

Here are some examples:

- To specify Saturday and Sunday as the weekend (or inactive) days in the week, use WEEKENDDDAYS=17
- To specify Friday and Saturday as the weekend days, use WEEKENDDDAYS=67.

The weekend (or inactive) days can also be specified using the INTERVAL option. For example, INTERVAL=WEEKDAY67W is the same as setting INTERVAL=WEEKDAY and WEEKENDDDAYS=67.

ZEROMISS= NONE | LEFT | RIGHT | BOTH

specifies how beginning or ending zero values (or both) are interpreted in the accumulated time series. The default value is **NONE**.

Here is a description of each valid value:

NONE	The beginning and ending zeros are unchanged.
LEFT	The beginning zeros are set to missing.
RIGHT	The ending zeros are set to missing.
BOTH	Both the beginning and ending zeros are set to missing.

Results

The %FSUPDATE global macro variable indicates whether the %FSUPDATE macro finishes successfully or encounters errors: &FSUPDATE = SUCCESS | ERROR.

A series of global macro variables are populated after a SAS Forecast Server project is loaded. These macro variables describe the project. Using these macro variables, you can write SAS code that is generic enough to use across multiple projects. For example, you can use this code to explore and use data in your SAS Forecast Server projects.

Example

```
%fsupdate (projectname=Shoe_Sales_2012,
reconciliation=YES,
publicaccess=NO
)
```

%FSVER Macro

The %FSVER macro prints the version of SAS Forecast Server to the log and initializes a macro variable that contains this value.

Syntax

%FSVER ()

Details

Required Arguments

There are no required arguments for the %FSVER macro.

Results

The output from this macro is FSVERSION, which lists the version number,

Examples

Example 1: Migrating SAS Forecast Server 4.1 Projects to SAS Forecast Server 14.2

%FSEXPALL, %FSIMPALL, %FSMIGALL, and %FSMIGPRJ Macros

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

This example explains how to batch update projects created with SAS Forecast Server 4.1 to a SAS Forecast Server 14.2 middle tier host installation on a separate machine. This example assumes the following:

- Both middle tier hosts are running Microsoft Windows.
- All SAS products are installed in their default installation folders.
- The name of the machine hosting the SAS Forecast Server 4.1 middle tier is fs41.
- The name of the machine hosting the SAS Forecast Server 14.2 middle tier is fs142.
- The project archive folder on fs41 is **C:\SAS\ForecastServer\Archives**.
- The project archive folder is accessible on fs141 using the **\\fs41\SAS\ForecastServer\Archives** path.
- The SAS administrator user ID is fsadm on both machines.

To migrate the SAS Forecast Server 4.1 projects:

1. The SAS Forecast Server macros should have been automatically installed in SAS Forecast Server 4.1. If they are not installed for some reason, install the macros on the SAS Forecast Server 4.1 middle tier. The macros are automatically installed on the SAS Forecast Server 14.2 middle tier. For more information about installing these macros for the 4.1 release, see the *SAS Forecast Server 4.1 Administrator's Guide*.
2. Archive the projects on fs41:
 - a. Open SAS in Display Manager mode on fs41.
 - b. Define the library where the OUT= data set is written. Both hosts must have access to the library. This example uses the Archive folder on fs41.

```
libname migrate 'C:\SAS\ForecastStudio\Archives\';
```

- c. Submit the following SAS code to execute the %FSEXPALL macro and export (archive) all projects:

```
%fsexpall (user=sasdemo,
password=Password1,
out=migrate.projects,
namesuffix=_fs_export41,
environment=Default
)
```

TIP Use the NAMESUFFIX= parameter to make the archive names unique. If an archive with the same name exists, then the archiving process fails for that project.

- d. To verify that the projects were archived successfully, open the migrate.projects data set and review the ARCHIVEFAIL variable. ARCHIVEFAIL equals 0 if the project was successfully archived. For more information about the global macro variables for the %FSEXPALL macro, see “%FSEXPALL Macro” on page 138.

3. Import all projects to fs141:

- a. Open SAS in Display Manager mode on fs141.
b. Define the library that contains the PROJECTDS= data set.

```
libname migrate '\\fs41\SAS\ForecastStudio\Archives\';
```

- c. Run the %FSIMPALL macro to import all projects. This macro creates the project data folders and registers the projects with the SAS Metadata Server.

```
%fslogin (user=sasdemo,
password=Password1,
desktop=NO
)
%fsimpall (projectds=migrate.projects,
archivefolder=\\fs21\SAS\ForecastStudio\Archives,
environment=Default
)
%fslogout ()
```

4. Migrate all projects to the current version using the %FSMIGALL macro:

```
%fslogin (user=sasdemo,
password=Password1,
desktop=NO
)
%fsmigall (out=work.migration,
environment=Default
)
%fslogout ()
```

TIP As an alternative to %FSMIGALL, you can use the %FSMIGPRJ macro to migrate single projects. For more information, see “%FSMIGPRJ Macro” on page 154. If a project has not been migrated, then SAS Forecast Studio prompts you to do so the first time the project is opened.

TIP You can also migrate single projects or migrate projects in batch using the SAS Forecast Server Plug-in for SAS Management Console or SAS Forecast Project Manager.

5. To verify that all of the projects were successfully migrated, open the work.migration data set and review the ISMIGRATED variable. ISMIGRATED equals 1 if the project was successfully migrated.

Example 2: In-Place Migration of SAS Forecast Server 4.1 Projects to SAS Forecast Server 14.2

%FSMIGALL Macro

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

If you have upgraded the SAS Forecast Server middle tier on the current machine to SAS Forecast Server 14.2, you can use the %FSMIGALL macro to migrate all of the projects registered in the SAS Metadata Server.

Note: Before following this example, you must create a new environment to point to the old environment location. You can use the SAS Forecast Server Plug-in for SAS Management Console or SAS Forecast Project Manager to register the environment. For more information, see the Help for the SAS Forecast Server Plug-in for SAS Management Console in SAS Management Console or the Help for the SAS Forecast Project Manager..

Alternatively, you can use the %FSMIGPRJ macro to migrate single projects. For more information, see [“%FSMIGPRJ Macro” on page 154](#). If a project has not been migrated, then SAS Forecast Studio prompts you to do so the first time the project is opened.

Note: You can also migrate single projects or migrate projects in batch using the SAS Forecast Server Plug-in for SAS Management Console or the SAS Forecast Project Manager

To migrate all projects using %FSMIGALL:

1. Open SAS in Display Manager mode on the SAS server machine where the macros are installed.
2. Submit the following SAS code:


```
%fsmigall(out=work.migration,
          environment=Default,
          )
```
3. To verify that all of the projects were successfully migrated, open the work.migration data set and review the ISMIGRATED variable. ISMIGRATED equals 1 if the project was successfully migrated.

Example 3: Create a SAS Forecast Server Project

%FSCREATE Macro

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You

must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

In this example, the %FSCREATE macro is used to create a SAS Forecast Server project.

The SASHELP.PRICEDATA data set is used to create a hierarchical project. The hierarchy of the project is defined by the following variables:

- regionName
- productLine
- productName

This example also uses the following variables to create the project:

- date is the time ID variable.
- sale is used to model the total sale as a function of the price variable and the discount variable.
- price1-price3 are used as the reporting variables.

The project is reconciled in a middle-out fashion starting from the productLine level, and the disaggregation method is “equal split of the difference”.

You can accept the default values for all other project creation options.

Note: You can use the sasdemo user ID to create the project. However, the project must be publicly available so that other users can open it and modify it.

To create a project using the %FSCREATE macro:

1. Open SAS in Display Manager mode on the SAS server machine where the macros are installed.
2. Run the macro for creating a project by submitting the following SAS code:

```
%fscreeate (projectname=mypricedataprf,
environment=Default,
data=sashelp.pricedata,
id=date,
by=regionName productLine productName,
hierarchy=YES,
var=sale,
input=price discount,
reporting=price1 price2 price3,
aggregate=NONE total(sale),
disaggregation=EQUALSPLIT,
reconciliation=BOTTOMUP,
publicaccess=YES
)
```

For more information, see [“%FSCREATE Macro” on page 118](#).

Example 4: Archive a SAS Forecast Server Project

%FSEXPORT Macro

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You

must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

This example exports a SAS Forecast Server project to an archive file using the %FSEXPORT macro.

This example assumes the following:

- pd1 is the project's name.
- ArchPD1 is the archive's name.
- Project pd1 is the archive's description.
- There is only one SAS Forecast Server environment.
- The middle tier server is your local machine, so the ENVIRONMENT= and HOST= parameters are not specified.

To archive a project using the %FSEXPORT macro:

1. Open SAS in Display Manager mode on a SAS server machine.
2. Submit the following SAS code:

```
%fsexport (projectname=pd1,
archivename=ArchPD1,
description=Project pd1,
archivefolder=C:\SAS\ForecastServer\Archives,
environment=Default
)
```

For more information, see [“%FSEXPORT Macro” on page 140](#).

Example 5: Add Overrides to a SAS Forecast Server Hierarchical Project

%FSSETOVR Macro

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

This example adds an override to a hierarchical project.

This example assumes the following:

- pricedata_hier is an existing hierarchical project.

The hierarchy of the project is defined by the following BY variables:

- regionName
- productLine
- productName
- OVRDS is an existing data set that contains the overrides for the project.

The data set contains the following variables in addition to the BY variables:

- the time ID column. If the time ID variable in the project is STARTDATE, then the OVRDS= data set should have a STARTDATE column.
- OVERRIDE

- OLOCK

For more information about how to create this data set, see “OVRDS= data set” on [page 173](#).

To run the example code:

1. Open SAS in SAS Display Manager mode on a SAS server machine.
2. Submit the following SAS code:

```
/*Add override to hierarchical project */
%fssetovr(ovrds=_ovrds,
projectName=pricedata_hier,
environment=default,
reconcile=YES,
outfailed=_outfailed_hierarchy
)

/*Print the data set to see results of override */
proc print data=_outfailed_hierarchy;
run;
```

Example 6: Remove Overrides from a SAS Forecast Server Hierarchical Project

%FSRMOVR Macro

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

This example removes overrides from a hierarchical project.

This example assumes the following:

- pricedata_hier is an existing hierarchical project.
The hierarchy of the project is defined by the following BY variables:
 - regionName
 - productLine
 - productName
- SERIESDS is an existing data set that contains the list of series for which the overrides are to be removed.

The data set contains the following variables:

- regionName
- productLine
- productName
- SELECTLOCK
- FROM time ID value
- TO time ID value

For more information about how to create this data set, see [“SERIESDS= data set” on page 164](#).

To run the example code:

1. Open SAS in SAS Display Manager mode on a SAS server machine.
2. Submit the following SAS code:

```
/*Remove override from hierarchical project */
%fsrmovr(seriesds=_seriesds,
projectName=pricedata_hier,
environment=default,
reconcile=YES,
outfailed=_outfailedremove_hierarchy
)
/*Print the data set to see results of override */
proc print data=_outfailedremove_hierarchy;
run;
```

Example 7: Add Overrides to a SAS Forecast Server Non-Hierarchical Project

%FSSETOVR Macro

Note: For examples that use release 12.1 or later of SAS Forecast Server, you must execute the %FSLOGIN macro before the example code to create a new session. You must execute the %FSLOGOUT macro after the example code to close the session. For more information, see [%FSLOGIN and %FSLOGOUT macros on page 149](#).

This example adds an override to a non-hierarchical project.

This example assumes the following:

- pricedata_nonhier is an existing non-hierarchical project.
- _ovrds is an existing data set that contains the overrides for the project.

The data set contains the following variables in addition to the BY variables:

- the time ID column. If the time ID variable in the project is STARTDATE, then the OVRDS= data set should have a STARTDATE column.
- OVERRIDE
- OLOCK
- DEPENDENT

Note: The DEPENDENT column is required for non-hierarchical projects. It specifies the value of the depend series where the override is to be added.

For more information about how to create this data set, see [“OVRDS= data set” on page 173](#).

To run the example code:

1. Open SAS in Display Manager mode on a SAS server machine.
2. Submit the following SAS code:

```
/*Add override to non-hierarchical project */
```

```
%fssetovr(ovrds=_ovrds,  
projectName=pricedata_nonhier,  
environment=default,  
reconcile=YES,  
outfailed=_outfailed_nonhierarchy  
)  
  
/*Print the dataset to see results of override */  
proc print data=_outfailed_nonhierarchy;  
run;
```

Part 5

SAS Forecast Server Utility Macros

Chapter 12

Using the SAS Forecast Server Utility Macros 203

Chapter 12

Using the SAS Forecast Server Utility Macros

About the SAS Forecast Server Utility Macros	203
Dictionary	203
HPF_HIER_ROLLUP	203
HPF_ImportEventsFromExcel	205
HPF_PART_EQUAL	207
HPF_PART_SPLIT	209

About the SAS Forecast Server Utility Macros

Utility macros can be used to manage your SAS Forecast Server projects. Before running any of the utility macros, you first must run the appropriate HPF macro for AUTOCALL initialization.

Dictionary

HPF_HIER_ROLLUP

The HPF_HIER_ROLLUP macro performs hierarchical aggregation of forecasting data sets.

Note: Before you run the HPF_HIER_ROLLUP macro, you first must run the %HPFHIER() macro.

Syntax

```
%HPF_HIER_ROLLUP(DATASET=,BYVARS=,TIMEID=,INTERVAL=,
HORIZONSTART=,OUTSET= [,options])
```

Details

Required Arguments

The following arguments must be used with the HPF_HIER_ROLLUP macro. The required arguments are separated by commas.

DATASET = *data-set-name*

specifies the name of the input data set that contains the forecasts at the lowest level of aggregation. The data set must have the standard form for a forecasting data set (for example, OUTFOR= data set). The name must be a valid SAS name and in the *library.data-set-name* format.

BYVARS = *variable-names*

specifies the BY variables in the input data set. A name must be a valid SAS name.

TIMEID = *variable-name*

specifies the time ID variable in the input data set. The value must be a valid SAS date, time, or datetime. The name must be a valid SAS name.

INTERVAL = *interval-name*

specifies the time interval for the time ID variable. The name must be a valid SAS time interval.

HORIZONSTART = *date-or-time-value*

specifies the start of the forecast horizon. The value must be a valid SAS date, time, or datetime.

OUTSET = *data-set-name*

specifies the name of the output data set that contains the forecasts at the highest level of aggregation. The data set must have the standard form for a forecasting data set (for example, OUTFOR= data set). The name must be a valid SAS name and in the *library.data-set-name* format.

Options

You can use the following options with the HPF_HIER_ROLLUP macro. Options must follow arguments and are separated by commas.

ACTUAL = *variable-name*

specifies the name of the variable in the input data set that contains the actual values to be forecast. The default value is ACTUAL.

PREDICT = *variable-name*

specifies the name of the variable in the input data set that contains the predicted time. The default value is PREDICT.

DEPVAR = *variable-name*

specifies the name of the variable that is used to rename the ACTUAL variable for display purposes. The default value is ACTUAL.

AGGREGATE = TOTAL | AVERAGE

specifies the statistic that aggregates the forecasts in the input data set. The default value is TOTAL.

SETMISS = MISSING | *number*

specifies the missing value interpretation of an actual value in the aggregate time series. The default value is MISSING.

ZEROMISS = NONE | LEFT | RIGHT | BOTH

specifies the leading or trailing zero interpretation of an actual value in the aggregate time series. The default value is NONE.

MINOBS = *non-negative-integer*

specifies the minimum number of observations in a non-mean model for the forecasts of the aggregate time series. The default value is 2.

LEAD = *non-negative-integer*

specifies the forecast horizon of the lead. The default value is 12.

Results

The HPF_RETURN_CODE global macro indicates whether the HPF_HIER_ROLLUP macro finished successfully or encountered errors.

&HP_RETURN_CODE = SUCCESS | ERROR

Example

In this example, the HPFENGINE procedure is used to generate forecasts at the lowest level of aggregation. The HPF_HIER_ROLLUP macro is used to aggregate the forecasts using bottom-up reconciliation.

```
proc hpfengine data=sashelp.pricedata out=_NULL_ outfor=lowest
  task=select(minobs=4) lead=24;
  by region line product;
  id date interval=month horizonstart='01JAN2004'd;
  forecast sale;
run;

%hpfhier();/*AUTOCALL Initialization*/

%HPF_Hier_RollUp( dataset=lowest,
                  byvars=region line product,
                  timeid=date,
                  interval=month,
                  horizonstart='01JAN2004'd,
                  depvar=sale,
                  minobs=4, lead=24, outset=forecasts);
```

HPF_ImportEventsFromExcel

The HPF_ImportEventsFromExcel macro imports an Excel file that contains event descriptions into an event definition data set.

Notes: SAS/ACCESS Interface to PC Files must be licensed to run this macro.
Before you run the HPF_ImportEventsFromExcel macro, you first must run the %HPFEXCEL() macro.

Syntax

```
%HPF_ImportEventsFromExcel(INFILE=,SHEET=,
  DBMS=,DATASET= [,options])
```

Details

Required Arguments

The following arguments must be used with the HPF_ImportEventsFromExcel macro. The required arguments are separated by commas.

INFILE = *Excel-filename*

specifies the name of the Excel file that contains the event descriptions. Only one worksheet from an Excel file can be imported at a time. If the Excel file contains multiple worksheets, then multiple macro calls are required to import all of the worksheets.

SHEET = *worksheet-name*

specifies the name of the worksheet in the Excel file that contains the event descriptions. This option is not required if the Excel file specified in the INFILE option contains only one worksheet.

DBMS = *identifier*

specifies the type of data to import. This option is not required if the file extension of the Excel file is XLS or XLSX.

- If the DBMS= option is not set and the file extension of the Excel file is XLS, then DBMS=XLS is implied.
- If the DBMS= option is not set and the file extension of the Excel file is XLSX, then DBMS=XLSX is implied.

For more information about the DBMS= option, see “The IMPORT Procedure” in the *Base SAS Procedures Guide*.

DATASET = *SAS-data-set-name*

specifies the event repository (which is saved as a SAS data set) that will contain the event descriptions for use in the EVENTDATA IN= statement of PROC HPFEVENTS or in the INEVENT= option of PROC HPFDIAGNOSE or PROC HPFENGINE.

Options

You can use the following options with the HPF_ImportEventsFromExcel macro. Options must follow arguments and are separated by commas.

DATE = *column-name*

specifies the name of the column in the Excel file that contains the date information. Date information is used for the _STARTDATE_ variable of the event definition data set.

- If a value is not specified for DATE=, then the _STARTDATE_ column is used if it exists.
- If a _STARTDATE_ column does not exist, then the STARTDATE column is used if it exists.
- If neither a _STARTDATE_ nor a STARTDATE column exists, then the DATE column is used if it exists.
- _STARTDATE_ is not a required variable in an event definition data set. The date information can be specified in the _STARTDT_, _STARTOBS_, or the _KEYNAME_ column. However, each event should have a valid value for the _STARTDATE_, _STARTDT_, _STARTOBS_, or _KEYNAME_ variable for the resulting event definition data set to be processed by PROC HPFEVENTS.

ORIGINAL = OUTLOOK

specifies that the Excel file was created by exporting data from a Microsoft Outlook calendar. In this case, the date is formatted as text. It requires special input formatting.

DESCRIPTION = *column-name*

specifies the name of the column to use to create an event name.

- If DESCRIPTION= is not specified, then the value of the _NAME_, NAME, or DESCRIPTION column is used. Columns are listed in order of precedence.
- If the _LABEL_ column does not exist, then the value in the DESCRIPTION= column is used for the _LABEL_ column.
- For the _NAME_ variable, the DESCRIPTION= column is processed to form a valid SAS variable name for event processing. For the _LABEL_ variable, the _LABEL_ column is used as specified. For example, **Christmas Day** creates an event named **ChristmasDay** with the label **Christmas Day** because spaces are not allowed in SAS variable names (but they are allowed in labels).
- If no column value exists for the DESCRIPTION= value, then the event definition data set cannot be created.

Other Columns Processed

If any of the following columns exist in the Excel file, then they are processed as SAS variables for the event definition data set: _CLASS_, _KEYNAME_, _ENDDATE_, _DATEINTRVL_, _STARTDT_, _ENDDT_, _DTINTRVL_, _STARTOBS_, _ENDOBS_, _OBSINTRVL_, _TYPE_, _VALUE_, _PULSE_, _DUR_BEFORE_, _DUR_AFTER_, _SLOPE_BEFORE_, _SLOPE_AFTER_, _SHIFT_, _TCPARM_, _RULE_, _PERIOD_, _LABEL_. The values of the columns are used as observations in the corresponding SAS variables for PROC HPFEVENTS. This makes it possible to export an event definition data set into Excel, and then import the Excel file back into SAS.

Results

The HPF_RETURN_CODE global macro indicates whether the HPF_ImportEventsFromExcel macro finished successfully or encountered errors.

&HPF_RETURN_CODE = SUCCESS | ERROR

Example

```
%hpfxcel();/*AUTOCALL Initialization*/

%HPF_ImportEventsFromExcel(infile=C:\importEvents\Chinese2010.xlsx,
    dataset=c
);
```

HPF_PART_EQUAL

The HPF_PART_EQUAL macro partitions an input data set into equally sized partitions.

Note: Before you run the HPF_PART_EQUAL macro, you first must run the %HPFPART() macro.

Syntax

```
%HPF_PART_EQUAL(DATASET=,BYVARS,NPARTS=,PARTSET=[,options]);
```

Details

Required Arguments

The following arguments must be used with the HPF_PART_EQUAL macro. The required arguments are separated by commas.

BYVARS = *variables*

specifies the variable to use in the partitioning.

DATASET = *data-set-name*

specifies the name of the input data set that you want to partition. The name must be a valid SAS name and in the *library.data-set-name* format.

NPARTS = *integer*

specifies the number of partitions. This value must be a positive integer.

PARTSET = *data-set-name*

specifies the name of the data set whose contents are partitioned. The name must be a valid SAS name and in the *library.data-set-name* format.

Options

You can use the following options with the HPF_PART_EQUAL macro. Options must follow arguments and are separated by commas.

BASENAME = *SAS-name*

specifies the base name of the partitioned data set or view that will be indexed using the NPARTS= option. For example, if BASENAME=SEGMENT and NPARTS=3, the base names of the partitioned data sets or views are SEGMENT1, SEGMENT2, and SEGMENT3. The default is BASENAME=PART.

LIBREF = *library-reference*

specifies the name of the library reference that contains the partitioned data sets or views. The name must be a valid SAS library reference. The default is LIBREF=WORK.

VIEW = NO | YES

specifies whether the partition creates SAS data views or SAS data sets. If VIEW=YES, SAS data views are created. If VIEW=NO, SAS data sets are created. The default is NO.

Results

The HPF_RETURN_CODE global macro indicates whether the HPF_PART_EQUAL macro finished successfully or encountered errors.

```
&HPF_RETURN_CODE = SUCCESS | ERROR
```

Example

```
%hpfpert();/*AUTOCALL Initialization*/

%HPF_PART_EQUAL(dataset=sashelp.pricedata,byvars=regionName productLine,
  nparts=3,partset=partition);
%put HPF_RETURN_CODE=&HPF_RETURN_CODE;
%put HPF_RETURN_MESSAGE=&HPF_RETURN_MESSAGE;
```

HPF_PART_SPLIT

The HPF_PART_SPLIT macro partitions an input data set using a variable in the input data set.

Note: Before you run the HPF_Part_Split macro, you first must run the %HPFPART() macro.

Syntax

```
%HPF_PART_SPLIT(DATASET=,SPLITVAR=,PARTSET=[,options]);
```

Details

Required Arguments

The following arguments must be used with the HPF_PART_SPLIT macro. The required arguments are separated by commas.

DATASET = *data-set-name*

specifies the name of the input data set that you want to partition. The name must be a valid SAS name and in the *library.data-set-name* format.

PARTSET = *data-set-name*

specifies the name of the data set whose contents are partitioned. The name must be a valid SAS name and in the *library.data-set-name* format.

SPLITVAR = *variable-name*

specifies the name of the variable in the input data set that you want to use to partition. The name must be a valid SAS name.

Options

You can use the following options with the HPF_PART_SPLIT macro. Options must follow arguments and are separated by commas.

LIBREF = *library-reference*

specifies the name of the library reference that will contain the partitioned data sets or views. The name must be a valid SAS library reference. The default is LIBREF=WORK.

VALIDVALUES = NO | YES

specifies whether the values for the SPLITVAR= variable are valid SAS names. If VALIDVALUES=YES, the names for the partitioned data sets are generated by the values of the SPLITVAR= variable. If VALIDVALUES=NO, the names for the partitioned data sets are generated by indexing the values of the SPLITVAR= variable. The default is NO.

VIEW = NO | YES

specifies whether the partition creates SAS data views or SAS data sets. If VIEW=YES, SAS data views are created. If VIEW=NO, SAS data sets are created. The default is NO.

Results

The HPF_RETURN_CODE global macro indicates whether the HPF_PART_SPLIT macro finished successfully or encountered errors.

```
&HPF_RETURN_CODE = SUCCESS | ERROR
```

Example

```
%hpfpart();/*AUTOCALL Initialization*/

%HPF_PART_SPLIT(dataset=sashelp.pricedata,splitvar=region,partset=partition);
%put HPF_RETURN_CODE=&HPF_RETURN_CODE;
%put HPF_RETURN_MESSAGE=&HPF_RETURN_MESSAGE;
```

Part 6

Appendixes

<i>Appendix 1</i>	
<i>Troubleshooting SAS Forecast Server</i>	213

Appendix 1

Troubleshooting SAS Forecast Server

Gathering Information	213
Overview	213
Operating System Environmental and Configuration Information	214
Problem Description	214
Sample Test Data	215
Log Files	215
Additional Resources	217
The Status Page for SAS Forecast Server	217
The Status Page for the SAS Forecast Server: SOAP Service	218
JMX Beans	218
Troubleshooting SAS Forecast Server	219
SAS License Is Expiring	219
Project Owner Cannot Access Project	220
How to Unlock a SAS Forecast Studio Project	220
Client Fails or Freezes during Start-Up Due to Initial Environment Selection	224
Cluster Node Continues to Hold Locks after a Failure	225
Troubleshooting the Java Web Start Client for SAS Forecast Studio	226
Java Version Missing for Java Web Start	226
JNLP File Is Not Signed	226
Enable the JWS Cache	226
Cannot Select a SAS Environment from the Log On Dialog Box	226
SAS Forecast Server Cannot Locate Any Compatible SAS Environments at Logon Time	227
Troubleshooting the SAS Forecast Server Macros	227
Classpath Variable Is Not Set	227
log4j System Property Is Not Initialized	227
Performance Tuning for SAS Forecast Server	228

Gathering Information

Overview

When you are troubleshooting unexpected application behavior, it is important to isolate and describe the problem and the context in which it occurs. Here are the general classes of information that can expedite resolution of a technical problem:

- operating system environmental and configuration information
- detailed problem description
- log files
- other files or screen shots
- sample test data

Use the following table to help gather information. Providing this information helps SAS Technical Support reproduce and fix your problem.

Table A1.1 Information Gathering Checklist

Task	Done
Details of your operating environment.	[]
Detailed description of the problem (including what it takes to reproduce it).	[]
Sample data that would help reproduce the problem.	[]
Obtain log files.	[]
Full Java stack trace from the error page.	[]

Operating System Environmental and Configuration Information

If you request help from SAS Technical Support, then providing the following information about your installation can result in faster resolution of the problem:

- hardware platform, operating environment (including SAS version number), amount of physical memory, and number of processors
- JDK version
- JRE version
- the SAS Forecast Server version number and patch level
- the configuration settings for SAS Forecast Server. You can get this information from the **Settings** and **Advanced** tabs in the Configuration Manager in SAS Management Console.
- server language and locale

Note: You must provide the preceding information only once, unless it has changed from previous reports.

Problem Description

Provide a scenario description that includes as much information as possible. Include a description of the general task that you are trying to accomplish, your role and permissions, and what has happened during the SAS session. Provide details such as the following:

- Are you working with new data or updating existing data?
- How easy is the problem to reproduce?
- What browser and version are you using?
- Is the problem locale-specific? If so, which locales are having problems?

Sample Test Data

If possible, capture the information entered that caused the problem. In certain situations, SAS Technical Support might request your data load files so that they can better replicate your operating environment.

Log Files

View SAS Forecast Studio Log

To view the latest information in the SAS Forecast Studio log, select **Tools** ⇒ **SAS Log**.

Note: The SAS Forecast Studio log is typically for the latest project. If you open or work on another project, then the current log might be overwritten. Therefore, if you want to obtain the SAS Forecast Studio log, do so immediately after you close SAS Forecast Studio.

You can also view the log on the file system.

- In Windows, the default location for the SAS Forecast Studio logs is in the **C:\users\userid** folder.

If you do not find the SAS Forecast Studio logs at that location, the default location might have been altered. In that case, check the new location specified in

```
JavaArgs_N=-Duser.home=<new location>
```

in the configuration settings file (forecaststdo.ini) at this location: **<SAS-installation-directory>\SASForecastStudio\14.2\forecaststdo.ini**

- In LINUX, the default location for the SAS Forecast Studio logs is in the user's home directory on the compute server where the SAS Forecast Server environment runs. For example: **/home/user**

If you do not find the SAS Forecast Studio logs at that location, the default location might have been altered. In that case, check the new location specified in

```
-Duser.home=<new location>
```

in the JREOPTIONS section of the sasv9.cfg file located on the compute server: **SASROOT/9.4/sasv9.cfg**

Set Logging Options

To include additional information in the log:

1. Start SAS Management Console and connect as a SAS administrator (sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.

4. Click the **Settings** tab.
5. In the selection pane, click **Forecast Server**.
6. In the **Forecast Server > Logging** section, set these options:
 - Disable filtering. By default, messages about internal activities are omitted from the SAS log.
 - Enable the inclusion of JDBC messages. By default, JDBC messages are not included in the SAS log.

Click **OK**.

Enable SAS Workspace Server Logs

SAS Workspace Server logs can be generated by SAS Forecast Server. When there is an issue in your SAS Forecast Studio project, SAS Workspace Server logs might give insight about the cause of the issue.

To enable SAS Workspace Server logs in SAS 9.2, 9.3, and 9.4, log off from SAS Forecast Studio and complete these steps.

For UNIX, on the server where SAS Forecast Server resides, go to `.../Lev1/SASApp/WorkspaceServer` and complete these steps.

1. Back up logconfig.xml: `cp logconfig.xml logconfig.orig.xml`
2. Back up logconfig.trace.xml: `cp logconfig.trace.xml logconfig.trace.orig.xml`
3. Enable logging: `rm logconfig.xml cp logconfig.trace.xml logconfig.xml`
4. Log on to SAS Forecast Studio and run the project.
5. A workspace server log begins as soon as you log on to SAS Forecast Studio, and remains open until you log off. The log is located in the `.../Lev1/SASApp/WorkspaceServer/Logs` directory.

For Windows, on the server where SAS Forecast Server resides, go to: `C:\SAS\{config name}\Lev1\SASApp\WorkspaceServer` and complete these steps.

1. Back up logconfig.xml: `rename logconfig.xml logconfig.orig.xml`
2. Back up logconfig.trace.xml: `rename logconfig.trace.xml logconfig.trace.orig.xml`
3. Enable logging: `rename logconfig.trace.xml logconfig.xml`
4. Log on to SAS Forecast Studio and run the project.
5. A workspace server log begins as soon as you log on to SAS Forecast Studio, and remains open until you log off. The log is located in the `C:\SAS\config name\Lev1\SASApp\WorkspaceServer\Logs` directory.

Note: Additional information is displayed in the workspace server logs when you add the following two options to the sasv9_usermods.cfg file.

```
-FULLSTIMER
-MSGLEVEL1
```

- The FULLSTIMER option collects performance statistics on each SAS step.
- The MSGLEVEL1 option collects information about index processing, MERGE statements, and sorting.

The sasv9_usermods.cfg file is located in the directory that corresponds to your operating system.

- UNIX: `.../Lev1/SASApp/WorkspaceServer/`
- WINDOWS: `C:\SAS\config name\Lev1\SASApp\WorkspaceServer\`

Java Stack Traces

When sending information to SAS Technical Support, send the complete full text of the Java stack trace. The logs for SAS Forecast Server are in the standard location for the application container. An example of this path is `SAS_CONFIG\Web\Logs`. Log files for SAS Forecast Server are in the form `SASForecastServer*.log`. SAS Technical Support prefers that you do not send a screen shot of this information because often the screen shots do not include the full text of the trace.

Additional Resources

The Status Page for SAS Forecast Server

From the status page for SAS Forecast Server, you can validate the status of your server, confirm your licensed product, and determine whether your product versions are compatible. This page also provides details about current sessions including the resources that are being used. The default URL for this page is where the SAS Forecast Server services are deployed. An example of this URL is `http://your-server-name:port-number/SASForecastServer/Status`.

Here is an example of a status page for SAS Forecast Server:

ACTIVE NODE SESSIONS			
User	Capabilities	Details	Resources
User Display Name User Name User Product Identity	☒ Permitted Action ☐ Restricted Action	System Session ID Product Session ID Session Description Creation Timestamp Last-Access Timestamp Elapsed Time Since Last Access Active User Requests	Current Workspace Server Current Forecasting Environment Currently Open Project Currently Open Data Specification Currently Held Lock Active User Requests
sasfsadm sasfsadm sasfsadm	☒ FSForecastStudio ☒ FSForecastPlugin ☒ FSSOAPBridge ☒ FSMacroBridge ☒ FSAdministerProduct ☒ FSAnalyzeTimeSeries ☒ FSAnalyzeModels	252fab17-757b-4ff0-9f1a-894541ae95fc 37e3970c-aaf4-4944-9d05-73b67eed7f44 Forecast Studio Nov 11, 2016 4:30:40 PM Nov 11, 2016 4:30:52 PM 0 minutes 0	SASApp - Logical Workspace Server Test_1 Test_1 [READER]

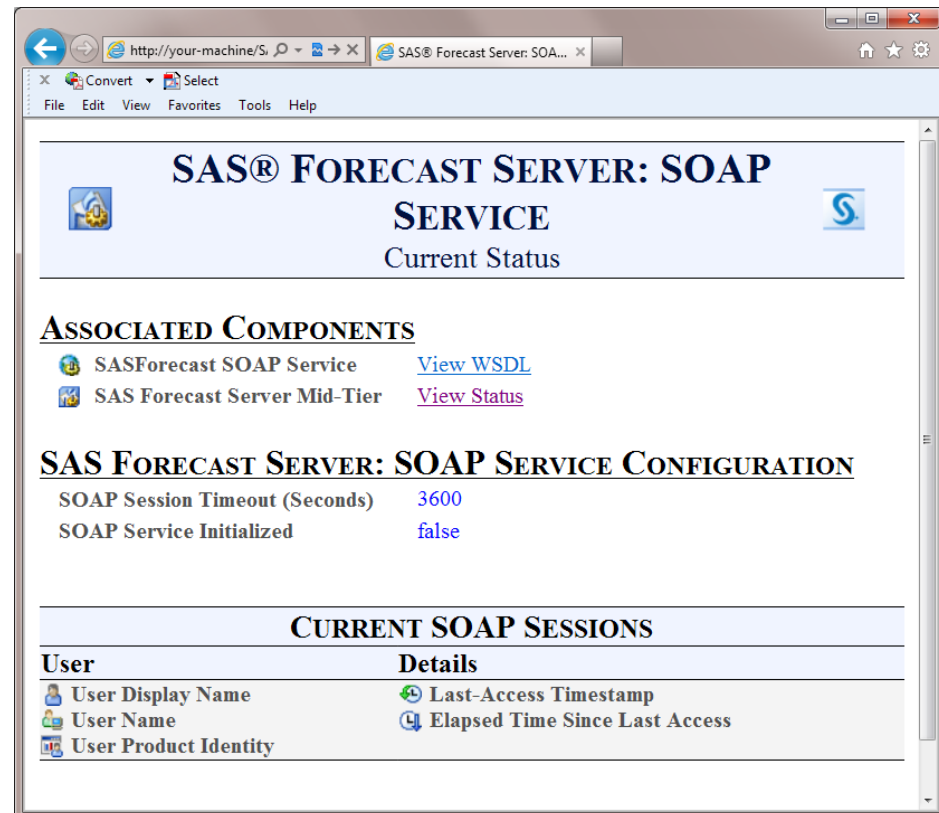
The SAS Forecast Server Current Status Web page provides the following information:

- links so that you can launch SAS Forecast Studio and SAS Forecast Project Manager.
- a link to the status of the SAS Forecast SOAP Service.
- general information, such as what version of SAS Forecast Server you are running.
- the configuration properties for the middle tier.
- the configuration properties for the Java Web Start client.
- the status of your servers, such as the SASApp - Logical Workspace Server
- for each of the current sessions, a summary of the actions (or capabilities) that are available to each user. For more information, see [“Managing Roles and Capabilities” on page 29](#).
- information about the node that you are running on in a clustered environment.

The Status Page for the SAS Forecast Server: SOAP Service

The SOAP service is required in order to run the SAS Forecast Studio tasks in the SAS Add-In for Microsoft Office or SAS Enterprise Guide. You can access the status page for this service from the status page for SAS Forecast Server. For more information, see “The Status Page for SAS Forecast Server” on page 217.

Here is an example of the status page for the SOAP service:



The SAS Forecast Server: SOAP Service Current Status Web page provides the following information:

- a link to the WSDL for the SOAP Service
- a link to the status of SAS Forecast Server
- the configuration properties for the SOAP service
- details about each of the current sessions

JMX Beans

Although the SAS Forecast Server Status page provides a summary of the run-time state of the middle tier, you cannot modify the state of the middle tier from this page. The SAS Forecast Server provides a simple JMX bean that you can use to modify the state. The JMX bean provides some details about the run-time state, but these details are directly related to the actions that are supported by the JMX bean.

These actions are supported by the JMX bean:

- General operation:
 - **isInitialized()**—reports whether the middle tier has completed the initialization that is performed at start-up.
 - **reloadConfiguration()**—reloads the product configuration from the metadata and attempts to apply any changes to the running application.
Note: Configuration information is also cached in the Configuration Service. This service must be reloaded before the **reloadConfiguration()** call to avoid getting the configuration information from an old cache.
- Server validation:
 - **getServerStatus()**—returns the validation information for all tested SAS Workspace Servers.
 - **refreshServer(server-name)**—clears the stored validation information for the specified SAS Workspace Server. Clearing this information results in the server being retested when the server is next accessed.
 - **refreshAllServers()**—clears the stored validation information for all SAS Workspace Servers.
- Session management:
 - **getSessions()**—reports all active sessions in the middle tier.
 - **killSession(ID)**—forces the specified session to close.
Note: Use this action only to close sessions that clients are no longer associated with. For example, you might use this action when a client process must be manually terminated.
 - **removeProcessMarker(ID)**—removes the indicator that specifies that the cluster node is still running. You can use this action to signal that the cluster node associated with the specified ID has failed so that the remaining nodes will recover any orphaned resources (such as removing stale locks).

TIP After executing a JMX call, you can confirm your change by using the SAS Forecast Server Status page.

Troubleshooting SAS Forecast Server

SAS License Is Expiring

If you install SAS Forecast Server as an add-on product to an existing SAS 9.4 deployment, you might see error messages about your SAS license expiring. When you are adding SAS Forecast Server to an existing SAS 9.4 deployment or you are renewing the software license for SAS Forecast Server, you must update the SAS Installation Data (SID) file in the metadata data.

For more information, see the software renewal instructions for your operating environment. These instructions are available from the Install Center at <http://support.sas.com/documentation/installcenter/94/index.html>.

- [Software License Renewal Instructions: SAS 9.4 for Microsoft Windows and Windows for x64 Planned Deployments](#)

- [Software License Renewal Instructions: SAS 9.4 for UNIX Environments Planned Deployment](#)

Project Owner Cannot Access Project

If you cannot access a SAS Forecast Server project that you should own, you might want to check the owner information that is stored with the project. SAS Forecast Server stores the identity of the owner as a token value that is derived from the user ID. However, the token value might not be identical to the user ID.

To determine the owner of the project, see these locations in SAS Forecast Studio:

- the Project Properties dialog box
- the **Owner** column of the table in the Projects dialog box

You can also determine the owner of the project by using the SAS Forecast Server plug-in for SAS Management Console and using the SAS macros.

To view the token value for the user who is currently logged in, see these locations:

- the active sessions table on the SAS Forecast Server Status page
- the **Show my (*identity-token*) projects only** check box in the Projects dialog box

Note: In some cases, a token mismatch can result when migrating projects from a previous release. Prior to SAS Forecast Server 4.1, the user ID at the logon prompt was used to generate these tokens. Starting in SAS Forecast Server 4.1, the name attribute of the metadata identity is used instead. In cases where the user ID and metadata name differ, the values of the project owner must be updated during the migration process.

How to Unlock a SAS Forecast Studio Project

The fastest way to unlock a SAS Forecast Studio project is to simply restart the server on which SAS Forecast Server resides. However, if you use that method, you must go through the process of restarting the server every time a project becomes locked in SAS Forecast Studio.

Follow the steps below to download and use the JConsole tool to unlock a SAS 9.4 SAS Forecast Studio project without having to restart the server each time.

1. Install the appropriate Java Development Kit (JDK) for your operating environment. To download the JDK, see [Java SE — Downloads | Oracle Technology Network](#).

Running JConsole from the middle-tier server where you add the Java Management Extensions (JMX) JVM parameters (covered in step 2) is recommended. This eliminates any connection issues when trying to connect to the middle tier from another system. If the middle tier is deployed on a Windows system, install the JDK and run JConsole on your middle-tier Windows system. If the middle tier is deployed on a UNIX system, install the JDK and run JConsole on the middle-tier UNIX system.

2. Add the following JVM parameters to the SAS Web Application Server instance that SAS Forecast Server is deployed with.

If you only have SASServer1_1, then SAS Forecast Server is deployed with SASServer1_1. Otherwise SAS Forecast Server is deployed with SASServer11_1. The following directory paths are for **SASServer1_1**. If SAS Forecast Server is

deployed with **SASServer11_1**, then change **SASServer11_1** to **SASServer11_1**.

Note: It is a best practice to make a backup of your configuration files before changing them.

- In Windows environments, open the wrapper.conf file in the **SAS-configuration-directory\Lev#\Web\WebAppServer\SASServer11_1\Conf** directory. Add the following JVM parameters to the Java Additional Parameters list:

```
wrapper.java.additional.nn=-Dcom.sun.management.jmxremote
wrapper.java.additional.nn=-Dcom.sun.management.jmxremote.port=number
wrapper.java.additional.nn=-Dcom.sun.management.jmxremote.authenticate=false
wrapper.java.additional.nn=-Dcom.sun.management.jmxremote.ssl=false
wrapper.java.additional.nn=-Djava.rmi.server.hostname=localhost
```

Each JVM parameter is numbered, as denoted by *nn*. So, for example, if the last JVM parameter in the list is this:

```
wrapper.java.additional.48=-Dsas.app.repository.path="C:/SAS/SASHome/SASVersionedJarRepository/eclipse"
```

Add the new parameters after this one, incrementing the *nn* in each new parameter by one:

```
wrapper.java.additional.49=-Dcom.sun.management.jmxremote
wrapper.java.additional.50=-Dcom.sun.management.jmxremote.port=7969
wrapper.java.additional.51=-Dcom.sun.management.jmxremote.authenticate=false
wrapper.java.additional.52=-Dcom.sun.management.jmxremote.ssl=false
wrapper.java.additional.53=-Djava.rmi.server.hostname=localhost
```

- In UNIX environments, open the setenv.sh file in the **SAS-configuration-directory/Lev#/Web/WebAppServer/SASServer11_1/bin** directory and add the following parameters to the end of the JVM_OPTS list before the last double quotation mark:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=number
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Djava.rmi.server.hostname=localhost
```

Here is an example of parameters that the JVM_OPTS list could start and end with:

```
JVM_OPTS="-Xmx4096m -Xss256k -Xms1024m ...
...
...
-Dsas.app.repository.path=/opt/SAS94M5/sashome/SASVersionedJarRepository/eclipse"
```

In this case, add each parameter to the end of the list just before the final quotation mark, separated by spaces and with no line breaks included.

```
JVM_OPTS="-Xmx4096m -Xss256k -Xms1024m ...
...
-Dsas.app.repository.path=/opt/SAS94M5/sashome/SASVersionedJarRepository/eclipse -Dcom.sun.management
.jmxremote -Dcom.sun.management.jmxremote.port=7969 -Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=localhost"
```

Note: For SASServer1_1, the port number is 6969. If SAS Forecast Server is installed using a multiple server configuration, then the server is SASServer11_1, and the port number is 7969. For more information about the default ports for SAS 9.4, see [SAS Intelligence Platform: Installation and Configuration Guide](#).

3. Restart the SAS Web Application Server.

- In Windows environments, restart the SASServer1_x WebAppServer service on which SAS Forecast Server runs.
- In UNIX environments, see “Using the tcruntime-ctl Command” in [SAS Intelligence Platform: Middle-Tier Administration Guide](#).

4. Open the SAS Forecast Server Status page. Append this code to the URL in the address bar:

```
?debugRequests=true&debugSessions=true
```

Now the URL in the address bar should be similar to the following: `http://server-name/SASForecastServer/Status??debugRequests=true&debugSessions=true`.

Refresh the page.

At the bottom of the SAS Forecast Server Status page, locate the product session ID for the locked SAS Forecast Studio project. You will use this product session ID to kill this session.

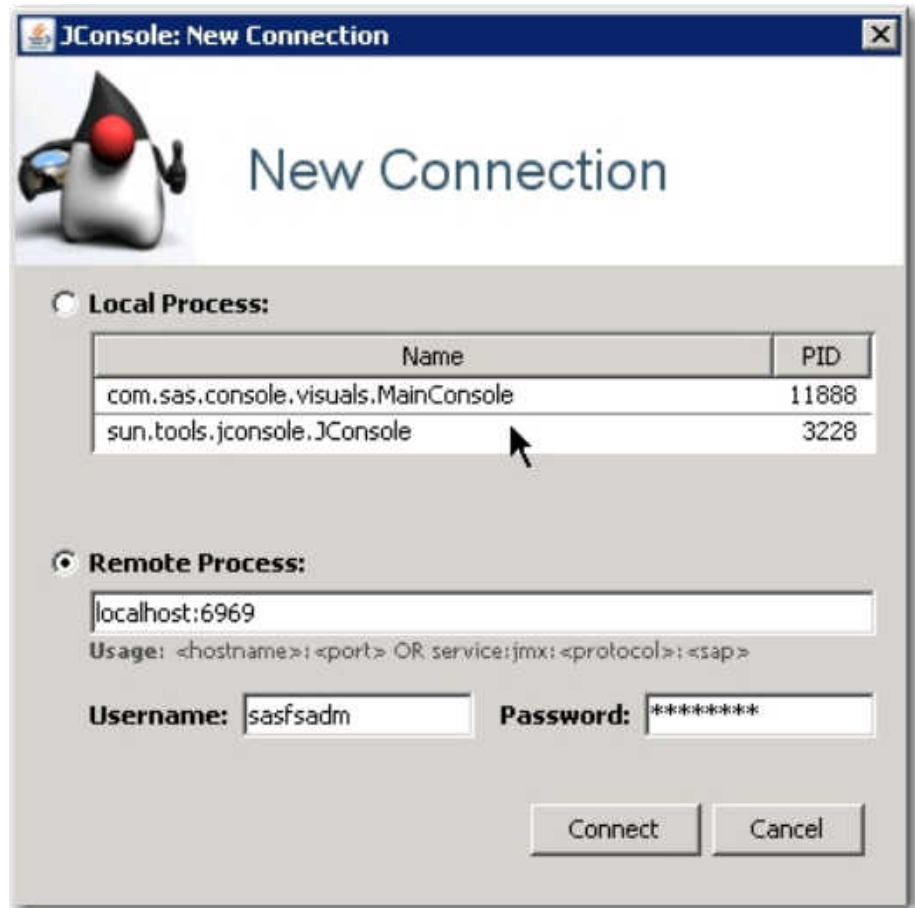
Here is an example of a status page for SAS Forecast Server:

ACTIVE NODE SESSIONS			
User	Capabilities	Details	Resources
User Display Name User Name User Product Identity	☒ Permitted Action ☐ Restricted Action	System Session ID Product Session ID Session Description Creation Timestamp Last Access Timestamp Elapsed Time Since Last Access Active User Requests	Current Workspace Server Current Forecasting Environment Currently Open Project Currently Open Data Specification Currently Held Lock Active User Requests
sasfsadm sasfsadm sasfsadm	☒ FSForecastStudio ☒ FSForecastPlugin ☒ FSSOAPBridge ☒ FSMacroBridge ☒ FSAdministerProduct ☒ FSAnalyzeTimeSeries ☒ FSAnalyzeModels	252fab17-757b-4f0b-9f1a-894541ae95fc 37e3970c-aaf4-4944-9d05-73f67ee7f44 Forecast Studio Nov 11, 2016 4:30:40 PM Nov 11, 2016 4:30:52 PM 0 minutes 0	SASApp - Logical Workspace Server Test_1 Test_1:[READER]

5. Launch the Java Monitoring and Management Console (JConsole) tool from the JDK directory.

- In Windows environments, double-click the jconsole file in the **C:\Program Files\Java\jdk1.7.0_67\bin** directory.
- In UNIX environments, open this tool by running `./jconsole` from the **jdk1.7.0_71/bin** directory.

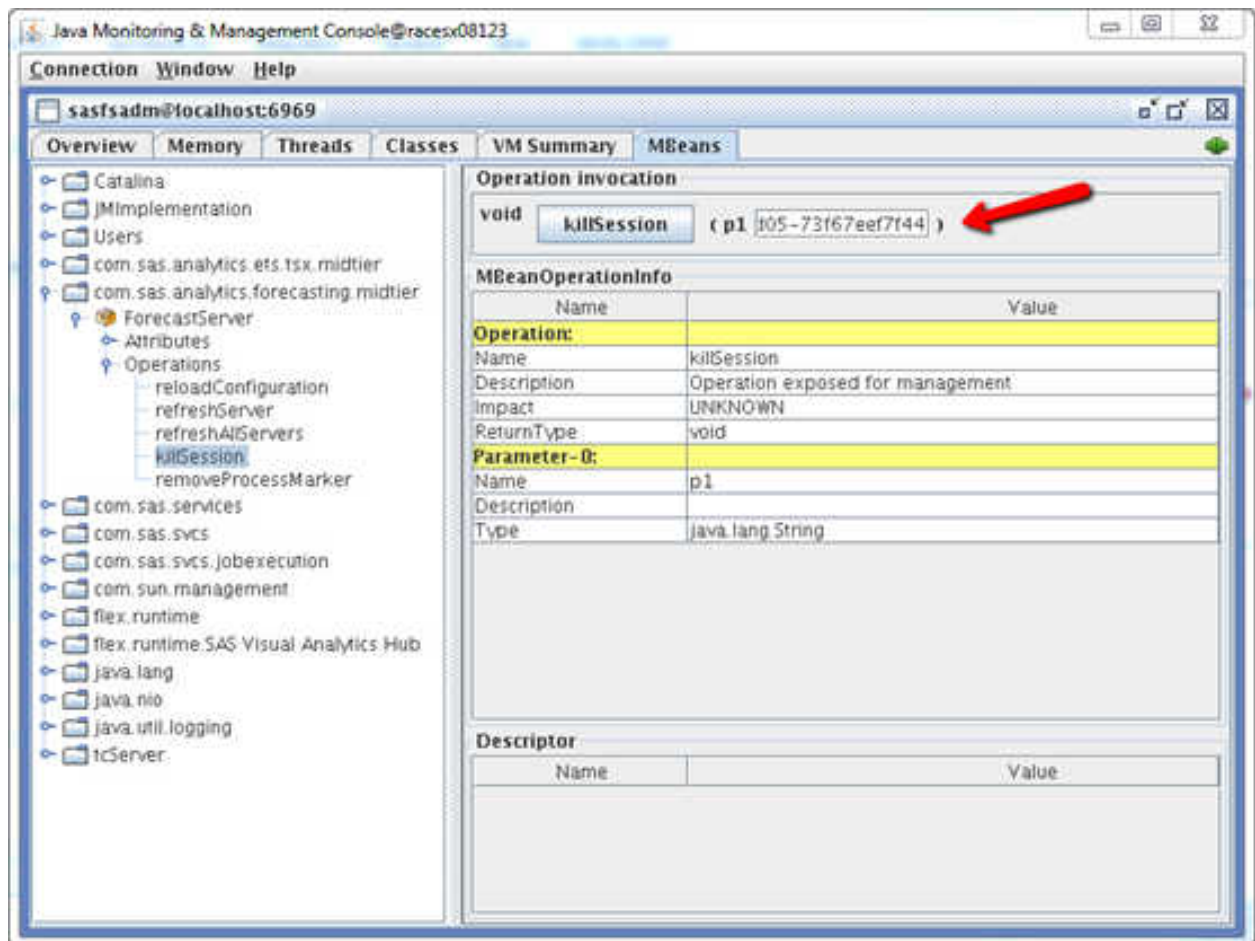
6. In the JConsole New Connection dialog box, select **Remote Process**.



7. Log on using the server name and port. For SASServer1_1, use 6969 as the port number. For SASServer11_1, use 7969 as the port number. Use the logon credentials for a user who has access to both the operating system and to SAS Forecast Server.

Note: If you receive a message that the secure connection failed and your application is not configured with Secure Sockets Layer, click **Insecure**.

8. In the Java Monitoring and Management Console (JConsole), click the **MBeans** tab.
9. In the directory tree, select **com.sas.analytics.forecasting.midtier** ⇒ **Operations** ⇒ **killsession**.



10. At the bottom of the SAS Forecast Server Status page, locate the product session ID for the locked SAS Forecast Studio project.
11. On the **MBeans** tab in the JConsole window, replace the ID string in the **killSession** field with the product session ID. Click **killSession**.

The locked SAS Forecast Studio project is removed from the bottom of the SAS Forecast Server Status page because the project is now unlocked.

Client Fails or Freezes during Start-Up Due to Initial Environment Selection

To perform the initialization process, clients of SAS Forecast Server must select a working product environment during start-up. However, the initialization process occurs in SAS Forecast Server before the user is granted control. As a consequence, SAS Forecast Server arbitrarily selects a product environment.

There are two known scenarios where this behavior can cause a failure during start-up:

- The selected environment has an AuthDomain for which no stored credentials are available. In this scenario, the initialization process executes before the credentials challenge prompt is configured. Because no valid credentials are available, a run-time error can occur rather than the expected run-time challenge prompt.

Note: Because of architectural limitations, run-time challenge prompts are supported only by some of the clients. Some clients can fail because a run-time challenge

prompt is not even supported. In that scenario, credentials must be stored to enable the proper use of that environment.

- The selected environment is improperly configured (for example, it uses an invalid directory path). In this scenario, misconfiguration prevents the initialization process from completing normally.

Note: Misconfigured environments can also lead to failures when you take management actions to correct them or remove them, depending on the nature of the error. If you encounter a product environment that does not respond properly to normal management actions, contact SAS Technical Support for assistance.

In either scenario, the clients allow the user to control the working environment used during the initialization process as a way of avoiding problems. Add `forecasting.environment.default=environment-name` to the VM arguments for the client to set the initial working environment.

Cluster Node Continues to Hold Locks after a Failure

In a clustered environment, if a cluster node experiences a failure, then any environments and projects being accessed through that node might be left in a locked state.

When multiple nodes are accessing content, the processes must be synchronized to coordinate who is currently accessing data at any point in time. For example, if a session on a particular node is currently running the Pricedata project, then the sessions on the other cluster nodes should not be allowed to access the Pricedata project at the same time. This is to avoid data corruption that could occur if multiple sessions access the same content simultaneously.

Access to content must be synchronized both within cluster nodes and across cluster nodes. To accomplish the latter, each resource has a lock marker that shows which (if any) cluster node currently has a lock on that resource. There is also a process marker that indicates that a cluster node is still running. When the cluster node process terminates normally, its process marker is removed and all locks are released. However, in the event that a cluster node process crashes, the lock marker and process marker are not removed. They continue to block access to the associated resources. In this case, you must manually remove the process marker to signal that the failed cluster node process is no longer running.

To manually remove a marker, you must know the identifier for the failed cluster node process. In a clustered environment, each cluster node has its own individual Status page. A Mid-Tier Process Coordination Status section is on the Status page for each node. This section includes information about any locks currently held by cluster node processes. The identifier for each node is listed in this section. Usually, this information is sufficient to identify the failed cluster node process based on knowing the resources that are inaccessible. With the Mid-Tier Node ID from this table, you can invoke the JMX action `removeProcessMarker` to remove the process marker. Once this is done, the remaining cluster node processes handle the cleanup work, including unlocking the resources.

In this example, the process marker is 7d158a86-4a3e-46fd-b036-9beb155e4976.

MID-TIER PROCESS COORDINATION STATUS	
Mid-Tier Process	Active Resources
Mid-Tier Node ID	Currently Held Lock
7d158a86-4a3e-46fd-b036-9beb155e4976	Default:Pricedata Default:[READER]

For more information about JMX, see “Additional Resources” on page 217.

Troubleshooting the Java Web Start Client for SAS Forecast Studio

Java Version Missing for Java Web Start

Java Web Start does not work because the required version of JRE is not installed on the client tier. For SAS 9.4 products, the required Java version 1.7 or later. To download the JRE, see <http://support.sas.com/resources/thirdpartysupport/v93>.

JNLP File Is Not Signed

When you launch the Java Web Start client, you see a warning that the JNLP file is not signed. For SAS Forecast Server, the JNLP files are dynamically generated and therefore cannot be signed. You can ignore this warning.

Enable the JWS Cache

The Java Web Start client does not work properly if the Java Web Start cache is disabled. By default, the cache is enabled. If this caching is disabled for any reason, complete these steps to turn caching on:

1. On Windows 7, select **Start** ⇒ **Settings** ⇒ **Control Panel** ⇒ **Programs** ⇒ **Java**.
2. Click the **General** tab.
3. In the **Temporary Internet Files** section, click **Settings** and select the **Keep temporary files on my computer** check box.

Cannot Select a SAS Environment from the Log On Dialog Box

During the logon process, SAS Forecast Server looks for a URL that references a sas-environment.xml file. This file contains the list of SAS environments that a user can select from during logon. SAS Forecast Server searches these locations in order of precedence:

1. the `sas.env.definition.location` (a Java system property).
2. the `env.definition.location` (a Java system property).
3. the `SAS_ENV_DEFINITION_LOCATION` environment variable for the operating system.
4. the configured URL that is stored in the `SASENVIRONMENTSURL` property in %
{`sas.home`}/`sassw.config`. (`sas.home` is a Java system property.)
5. the configured URL that is stored in the `SASENVIRONMENTSURL` property in
%`SASHOME`/`sassw.config`. (`SASHOME` is an environment variable for the operating system.) During the deployment of SAS Forecast Studio, you are prompted for the value to store in the `sassw.config` file. If the SAS Forecast Server client cannot locate this URL, verify that the information in the `sassw.config` file is correct.

Launching SAS Forecast Studio using Java Web Start does not require this URL to be configured. However, if the URL has been configured on either the client machine or in the configuration properties for Java Web Start, then SAS Forecast Server might generate an error. To resolve this error, look in the previous locations and verify that the valid URL is specified.

SAS Forecast Server Cannot Locate Any Compatible SAS Environments at Logon Time

During the logon process, SAS Forecast Server runs a compatibility test on each SAS environment. Primarily, this test is used to ensure that each of the SAS environments that a user can select from the Log On dialog box are associated with a compatible deployment of SAS Forecast Server. Only SAS environments that pass this test appear in the Log On dialog box.

If no SAS environments pass this test, an error message states that no SAS environments could be found. If you see this message, verify the following requirements:

- The sas-environment.xml file that you are using includes at least one SAS environment with a SAS Forecast Server deployment.
- The version of the SAS Forecast Server deployment matches the version of the client.
- The servers associated with the deployment are running and responsive.

Troubleshooting the SAS Forecast Server Macros

Classpath Variable Is Not Set

You do not have to set the classpath variable before using the macros. If you see the following note while executing the macros, you can safely ignore it.

NOTE: Could not initialize classpath. Classpath variable is not set.

log4j System Property Is Not Initialized

You do not need to configure the log4j logging service for Java. If you see the following warning while executing the macros, you can safely ignore it.

log4j:WARN No appenders could be found for logger java-class-name
log4j:WARN Please initialize the log4j system properly.

Performance Tuning for SAS Forecast Server

For scalability issues, slow system performance, or failures caused by memory settings, you can change the default settings for SAS Forecast Studio and Java Web Start in the following tiers:

- SAS Forecast Studio client tier
- SAS server tier

To improve performance:

1. Change the maximum heap size. By default, this value is **512m**. For information about formatting and value constraints, see the JRE documentation.

You can change this value in either of these ways:

- Change the **`Xmx`** value in the .ini files.
- by setting the **Maximum heap size** property in SAS Management Console.
 1. Start SAS Management Console and connect as a SAS administrator.
 2. In the **Plug-ins** tab, expand the **Configuration Manager** and **SAS Application Infrastructure** nodes.
 3. Right-click the **Forecast Server 14.2** node and select **Properties**. The Forecast Server 14.2 Properties dialog box appears.
 4. Click the **Settings** tab.
 5. In the selection pane, click **Virtual Machine Options**.
 6. Specify a value for the maximum heap size. Click **OK**.
- 2. Increase the JVM memory for the server, middle tier, and client by specifying the **`-Xmx1g`** command.

The Windows configuration files are at the following locations:

- Client (SAS Forecast Studio): **`SAS_HOME\SASForecastStudio\14.2\forecaststdo.ini`**
- JRE variables **`SAS_HOME\wrapper.conf`**

For more information about changing your system settings to improve system performance, see the section about best practices for configuring the middle tier in the *SAS Intelligence Platform: Middle-Tier Administration Guide* at <http://support.sas.com/documentation/onlinedoc/intellplatform>.

Recommended Reading

- *SAS Forecast Server: Migration Guide*
- *SAS Forecast Studio: User's Guide*
- Administration documentation for the SAS Intelligence Platform at <http://support.sas.com/documentation/onlinedoc/intellplatform>.

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books

Glossary

client application

an application that runs on a client machine.

client tier

the portion of a distributed application that requests services from the server tier. The client tier typically uses a small amount of disk space, includes a graphical user interface, and is relatively easy to develop and maintain.

data set

See SAS data set.

descriptor information

information about the contents and attributes of a SAS data set. For example, the descriptor information includes the data types and lengths of the variables, as well as which engine was used to create the data. SAS creates and maintains descriptor information within every SAS data set.

environment

a virtual container of run-time settings for SAS Forecast Server client sessions.

foundation services

See SAS Foundation Services.

Integrated Object Model server

See IOM server.

IOM server

a SAS object server that is launched in order to fulfill client requests for IOM services. Short form: IOM 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.

middle tier

in a SAS business intelligence system, the architectural layer in which Web applications and related services execute. The middle tier receives user requests, applies business logic and business rules, interacts with processing servers and data servers, and returns information to users.

object spawner

a program that instantiates object servers that are using an IOM bridge connection. The object spawner listens for incoming client requests for IOM services. When the spawner receives a request from a new client, it launches an instance of an IOM server to fulfill the request. Depending on which incoming TCP/IP port the request was made on, the spawner either invokes the administrator interface or processes a request for a UUID (Universal Unique Identifier).

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.

SAS Management Console

a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Server

a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories.

SAS Workspace Server

a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces.

server tier

in a SAS business intelligence system, the tier in which the SAS servers execute. Examples of such servers are the SAS Metadata Server, the SAS Workspace Server, the SAS Pooled Workspace Server, the SAS Stored Process Server, and the SAS OLAP Server. These servers are typically accessed either by clients or by Web applications that are running in the middle tier.

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.

spawner

See object spawner.

Index

Special Characters

%FSADDEVT macro [112](#)
 %FSCLEAR macro [116](#)
 %FSCOPY macro [116](#)
 %FSCREATE macro [118](#), [195](#)
 %FSDELARC macro [132](#)
 %FSDELENV macro [134](#)
 %FSDELEVT macro [134](#)
 %FSDELPRJ macro [136](#)
 %FSEVTREQ macro [137](#)
 %FSEXPALL macro [138](#), [193](#)
 %FSEXPOR macro [140](#), [196](#)
 %FSEXPSET macro [141](#)
 %FSGETENV macro [142](#)
 %FSGETPRJ macro [144](#)
 %FSGETURP macro [145](#)
 %FSIMPALL macro [146](#), [193](#)
 %FSIMPORT macro [147](#)
 %FSLOAD macro [148](#)
 %FSLOOP macro [151](#)
 %FSMIGALL macro [153](#), [193](#), [195](#)
 %FSMIGPRJ macro [154](#), [193](#)
 %FSMOVE macro [155](#)
 %FSNEWENV macro [157](#)
 %FSPRJAGG macro [159](#)
 %FSPUBLISH macro [160](#)
 %FSREGENV macro [158](#)
 %FSREGPRJ macro [161](#)
 %FSREN macro [162](#)
 %fsrmovr
 remove overrides example [197](#), [198](#), [199](#)
 %FSRUNPRJ macro [167](#)
 %FSSETDAT macro [170](#)
 %fssetovr
 add overrides example [197](#), [198](#), [199](#)
 %FSSETOVR macro [163](#), [173](#)
 %FSSETOWN macro [176](#)
 %FSSETPUB macro [177](#)
 %FSUNREG macro [178](#)
 %FSUNRENV macro [179](#)

A

accounts
 SAS user [12](#)
 add overrides
 add overrides [163](#), [173](#)
 architecture
 diagram [4](#)
 overview [4](#)
 archives
 creating [87](#)
 deleting [132](#), [136](#)
 importing [146](#), [147](#)
 archiving projects [42](#)
 authorization for libraries [49](#)
 automatic environment libraries [48](#)
 automatic environment library
 creating [51](#)

B

batch interface [5](#), [106](#)
 BY variables
 validating [95](#)

C

capabilities [29](#)
 Classpath variable [227](#)
 clustering
 held locks after failure [225](#)
 code
 invalid [87](#)
 components
 client tier [5](#)
 middle tier [5](#)
 SAS Forecast Project Manager [5](#)
 SAS Time Series Studio [6](#)
 configuration properties
 archiving projects [42](#)
 SAS environment URL [41](#)
 session time-out [40](#)
 setting [40](#)
 URL for Java Web Start [41](#)
 configurations

- removing 87
- configured environment libraries 48
- configured environment library
 - creating 52
- custom formats 99
- custom time intervals 96
 - examples 97, 98

D

- data sets
 - changing 170
- default file locations 17

E

- environment
 - initial selection causes failure 224
- environments
 - attributes 142
 - configuring 35
 - creating 35, 157
 - deleting 134
 - overview 35, 57
 - permissions 57, 58
 - registering 158
 - unregistering 179
- events
 - adding 112
 - creating 137
 - deleting 134
- external library assignments 48
- external user accounts 13
 - required rights and permissions 14

F

- formats
 - creating a custom format 99
 - validating 95

H

- HPF_HIER_ROLUP 203
- HPF_IMPORTEventsFromExcel 205
- HPF_PART_EQUAL 207
- HPF_PART_SPLIT 209

I

- importing an Excel file 205
- installation
 - See also* post-installation tasks
 - See also* pre-installation tasks
 - how to 17
 - process overview 11, 27

- internal user accounts 13
- INTERVALDS= system option 96

J

- Java stack traces 217
- Java Web Start
 - configuring 41
 - configuring login 92
 - JRE 226
 - JWS cache 226
 - starting 92
 - URL 41
- JMX access 40
- JMX beans 218
- JNLP file 226
- JWS cache 226

L

- libraries
 - assigned by SAS Forecast Server 48
 - authorization 49
 - automatic environment 48
 - configured environment 48
 - external library assignments 48
 - not pre-assigned 51
 - overview 47
 - preassigned 50
- LOCKDOWN mode 36
- log4j system property 227
- logging options 215
- logging server for Java 227
- logs
 - SAS Forecast Server 215, 217
 - SAS Workspace Server 216

M

- macro variables
 - clearing 116
- macros 106
 - hierarchical aggregation 203
 - HPF_ImportEventsFromExcel 205
 - HPF_PART_EQUAL 207
 - HPF_PART_SPLIT 209
 - importing an Excel file 205
 - partitioning input data 207, 209
 - passwords 109
 - utility 203
- metadata
 - registering 161
 - unregistering 178, 179
- middle tier 5

O

operating system accounts
 UNIX environments 28
 users 15
 owners
 projects 163, 173, 176

P

permissions
 environments 57, 58
 file system 36, 56
 metadata 57
 SAS Forecast Server Metadata User 55
 user access 53
 WriteMetadata 56
 post-installation tasks
 See also installation
 create environment 35
 middle tier 40, 43
 pre-installation tasks
 See also installation
 administrative user account
 recommendations 15
 project owners 220
 changing 55
 overview 55
 project scripts 85
 projects
 accessing 220
 archiving 138, 140, 196
 attributes 144
 copying 116
 creating 118, 195
 deleting 132, 136
 exporting 138, 140, 196
 exporting settings 141
 importing 146, 147
 loading 148
 migrating 193
 moving 155
 opening 167
 owners 176
 permissions 58
 registering 145, 153, 158, 161
 renaming 162
 running 167
 sharing 59, 177
 start-up and shutdown code 85
 unregistered project names 145
 unregistering 178
 upgrading 154
 properties
 configuration 40

R

reports
 about reports 61
 about stored processes 61
 accessing in SAS Forecast Studio 64
 active series macro variables 80
 basic project macro variables 72
 control macro variables 71
 custom reports 69
 data hierarchy and settings macro
 variables 73
 diagnose macro variables 79
 dynamic report parameters 69
 forecast macro variables 79
 internal report parameters 70
 keywords 62
 managing 63
 metadata macro variables 71
 model selection macro variables 79
 normal report parameters 69
 parameter 69
 preregistration tasks 66
 reconciliation macro variables 80
 registering 66, 67
 required macro calls 83
 sample reports 65
 special report features 83
 roles 29
 configuring users 34

S

SAS Add-In for Microsoft Office 37
 SAS Administrator account 13
 SAS Anonymous Web User account 13
 SAS Demo User account 14
 SAS Deployment Manager 87
 SAS Enterprise Guide 37
 SAS environment
 Log On dialog box 42
 URL 41
 SAS environments
 compatibility 227
 Log On dialog box 226
 SAS Forecast Project Manager
 about 5
 SAS Forecast Server
 architecture 4
 capabilities 29
 components 5
 initial users 53
 logs 215, 217
 middle tier 5
 overview 3
 performance tuning 228
 roles 29

- security layers 56
- Status page 217
- SAS Forecast Server macro
 - migrating projects 193
- SAS Forecast Server macros 5
 - Classpath variable 227
 - migrating projects 195
- SAS Forecast Server Metadata User
 - overview 54
 - permissions 55
- SAS Forecast Server Plug-in for SAS
 - Management Console 5
- SAS Forecast Server Procedures 5
- SAS Forecast Studio 5
 - reports 64
 - SAS environment 42
 - session time-out 40
 - starting 92
- SAS Forecast Studio tasks
 - accessing 37
 - overview 37
 - prerequisites 37
 - SOAP bridge 37
- SAS Installer account 13
- SAS Intelligence Platform
 - components 4
 - integration 6
- SAS Management Console
 - configuration properties 40
- SAS Metadata Server 4, 91
- SAS Object Spawner 4
- SAS Remote Services 91
- SAS Services Application 4
- SAS Spawned Servers account 14
- SAS Trusted User account 13
- SAS Web Infrastructure Platform 4
- SAS Workspace Server 4
 - logs 216
- security
 - file system permissions 56
 - layers 56
 - WriteMetadata 56
- select server for reports 43
- servers

- required 91
- session time-out 40
- shared memory 12
- shutdown code
 - projects 85
- SOAP bridge 37, 41
 - connection time-out 41
- start-up code
 - invalid 87
 - projects 85
- Status page 217
- system requirements 12

T

- time intervals
 - custom 96
 - detecting 94
- troubleshooting
 - gathering information 213
 - JRE 226
 - system performance 228

U

- UNIX environments
 - operating system accounts 28
 - permissions 36
- UNIX operating environments
 - shared memory 12
- user accounts 12
 - product administrator 15
 - users of SAS Forecast Studio 15
- users
 - configuring 34
- utility macros 203

W

- Web application server 91
- Windows environments
 - permissions 36
- WriteMetadata permissions 56