

# **SAS® Forecast Server 12.1**

## **Administrator's Guide**



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® Forecast Server 12.1: Administrator's Guide*. Cary, NC: SAS Institute Inc.

**SAS® Forecast Server 12.1: Administrator's Guide**

Copyright © 2012, 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 Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

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

August 2012

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

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

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

---

# Contents

<i>About This Book</i> . . . . .	<i>vii</i>
<i>Accessibility and Compatibility Features</i> . . . . .	<i>ix</i>
<i>Recommended Reading</i> . . . . .	<i>xi</i>

## PART 1 Introduction to the SAS Forecast Server 1

<b>Chapter 1 • Understanding the SAS Forecast Server</b> . . . . .	<b>3</b>
What Is the 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

<b>Chapter 2 • Pre-Installation Tasks</b> . . . . .	<b>11</b>
Create an Operating System Account for Product Administrators and Users . . . . .	11
Install Your Web Application Server . . . . .	12
Create and Configure a Forecasting Environment . . . . .	13
Determine the Location of the SAS Environment URL . . . . .	14
<b>Chapter 3 • Installing the SAS Forecast Server</b> . . . . .	<b>15</b>
About the Deployment of the SAS Forecast Server . . . . .	15
How to Install and Configure the SAS Forecast Server . . . . .	15
<b>Chapter 4 • Post-Installation Tasks</b> . . . . .	<b>21</b>
Minimum Requirements for Post-Installation . . . . .	21
Create a Forecasting Environment . . . . .	21
Set Permissions for Forecasting Environments in UNIX Operating Environments . . . . .	22
Manage Roles and Capabilities . . . . .	23

## PART 3 Configuration Tasks 31

<b>Chapter 5 • Specifying Security Permissions for Users and Groups</b> . . . . .	<b>33</b>
Enable Users to Access the SAS Forecast Server . . . . .	33
Secure Access to the SAS Forecast Server . . . . .	36
<b>Chapter 6 • Administering Reports to Use in SAS Forecast Studio</b> . . . . .	<b>41</b>
Overview of Reports . . . . .	41
Use the Sample Reports . . . . .	45
Register Reports . . . . .	46
Create Reports . . . . .	48
<b>Chapter 7 • Creating and Configuring Libraries</b> . . . . .	<b>63</b>

Libraries . . . . .	63
Enable Expanded Support for Libraries . . . . .	65
How to Define a Pre-assigned Metadata Library . . . . .	66
How to Define a Metadata Library That Is Assigned by the SAS Forecast Server . . . . .	67
How to Define an Automatic Environment Library . . . . .	67
How to Define a Configured Environment Library . . . . .	68
<b>Chapter 8 • Customizing Project Code, Time Intervals, Formats, and Events . . . . .</b>	<b>69</b>
Add Start-Up and Shutdown Code to a Project . . . . .	69
Create Custom Time Intervals . . . . .	71
Create a Custom Format . . . . .	74
Create a Customized List of Events . . . . .	75
<b>Chapter 9 • Miscellaneous Administration Tasks . . . . .</b>	<b>79</b>
Change Configuration Properties in SAS Management Console . . . . .	79
Specify a Time-Out . . . . .	79
Configure a SAS Environment URL . . . . .	80
Set the Preferred SAS Environment in the Logon Dialog Box for SAS Forecast Studio . . . . .	81
Configure the Archive Functionality . . . . .	81
Remove a Product Configuration with the SAS Deployment Manager . . . . .	82
Archive a Project . . . . .	83
 <b>PART 4 Using the SAS Forecast Server Clients 85</b>	
<b>Chapter 10 • SAS Forecast Studio . . . . .</b>	<b>87</b>
Requirements for Starting SAS Forecast Studio . . . . .	87
Run SAS Forecast Studio from a Local Installation . . . . .	87
Use Java Web Start . . . . .	88
Start SAS Forecast Studio with Options . . . . .	90
<b>Chapter 11 • SAS Forecast Batch Interface . . . . .</b>	<b>93</b>
About the SAS Forecast Batch Interface . . . . .	94
Dictionary . . . . .	99
<b>Chapter 12 • SAS Forecast Studio Tasks in Other SAS Software . . . . .</b>	<b>169</b>
About the SAS Forecast Studio Tasks . . . . .	169
Prerequisites for Using the SAS Forecast Studio Tasks . . . . .	169
Grant Access to the SAS Forecast Studio Tasks . . . . .	170
 <b>PART 5 SAS Forecast Server Utility Macros 171</b>	
<b>Chapter 13 • Using the SAS Forecast Server Utility Macros . . . . .</b>	<b>173</b>
About the SAS Forecast Server Utility Macros . . . . .	173
Dictionary . . . . .	173
 <b>PART 6 Appendixes 181</b>	
<b>Appendix 1 • Troubleshooting the SAS Forecast Server . . . . .</b>	<b>183</b>

Gather Information .....	184
Additional Resources .....	186
Troubleshoot the SAS Forecast Server .....	189
Troubleshoot the Java Web Start Clients .....	190
Troubleshoot the Logon Process .....	191
Troubleshoot the SAS Forecast Server Plug-ins for SAS Management Console . . .	192
Troubleshoot the SAS Forecast Batch Interface .....	193
Performance Tuning for the SAS Forecast Server .....	194
<b>Glossary</b> .....	<b>197</b>
<b>Index</b> .....	<b>199</b>



# 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. For more information, see <http://support.sas.com/documentation/installcenter/en/ikforecastofrsr/65717/HTML/default/index.html>.

*Note:* You must have SAS/OR software licensed to access its full functionality. The 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.

## Documentation Conventions

*!SASROOT*

represents the SAS Foundation directory.

*SAS\_CONFIG*

represents the SAS configuration directory.

*SAS\_HOME*

represents the SAS installation directory



# Accessibility and Compatibility Features

SAS Forecast Server has been tested with assistive technology tools. It includes accessibility and compatibility features that improve the usability of the product for users with disabilities with some exceptions. These features are related to accessibility standards for electronic information technology that were adopted by the US Government under Section 508 of the US Rehabilitation Act of 1973 (2008 draft proposal initiative update). For more information about the accessibility of the SAS Forecast Server, see the *SAS Forecast Studio: User's Guide*.

For detailed information about the accessibility of the SAS Forecast Server, send e-mail to **[accessibility@sas.com](mailto:accessibility@sas.com)** or call SAS Technical Support.

**x** *Accessibility and Compatibility Features*

# 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 [support.sas.com/bookstore](http://support.sas.com/bookstore). If you have questions about which titles you need, please contact a SAS Publishing Sales Representative:

SAS Publishing Sales  
SAS Campus Drive  
Cary, NC 27513-2414  
Phone: 1-800-727-3228  
Fax: 1-919-677-8166  
E-mail: [sasbook@sas.com](mailto:sasbook@sas.com)  
Web address: [support.sas.com/bookstore](http://support.sas.com/bookstore)



## **Part 1**

---

# Introduction to the SAS Forecast Server

### *Chapter 1*

### **Understanding the SAS Forecast Server ..... 3**



## Chapter 1

# Understanding the SAS Forecast Server

---

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

---

## What Is the SAS Forecast Server?

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

The SAS Forecast Server has two primary components:

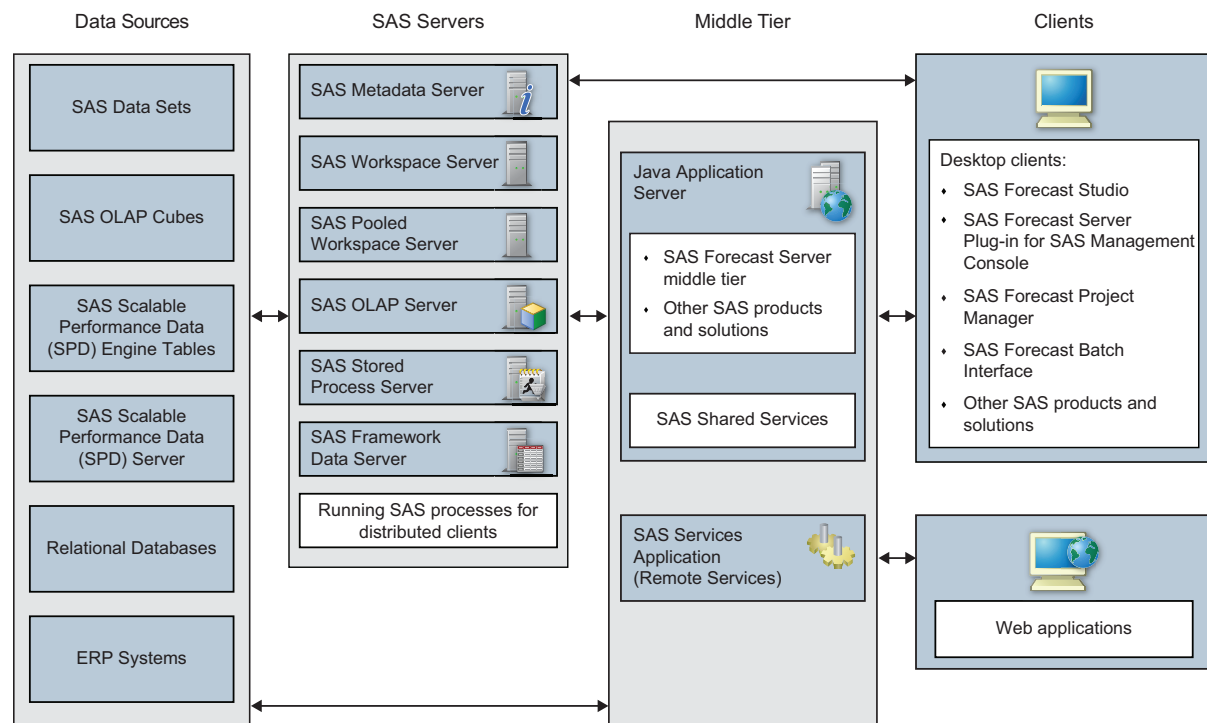
- SAS Forecast Server Mid-Tier, which is supported by the SAS Web Infrastructure Platform Database
- SAS Forecast Studio, which is the graphical user interface (based on Java) to the forecasting and time series analysis procedures in SAS High-Performance Forecasting and SAS/ETS software

## Overview of the SAS Forecast Server Architecture

### Architecture Diagram

The following figure shows how SAS Forecast Studio and the SAS Forecast Server Mid-Tier fit with the SAS Intelligence Platform.

**Figure 1.1** The SAS Forecast Server Architecture



### The SAS Intelligence Platform Components

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

#### SAS Metadata Server

The SAS Metadata Server provides an enterprise-level repository for SAS server configurations and application management metadata. Products such as the 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. The 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. A SAS System installation is required.

#### The SAS Workspace Server and the SAS Object Spawner

The SAS Workspace Server provides all computation and intermediate data storage services. A SAS System installation is required.



The SAS Forecast Server uses the SAS Workspace Server to execute SAS High-Performance Forecasting procedures 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 Database

The SAS Forecast Server supports deployment to each application server that is supported in SAS 9.3.

#### SAS Services Application

The SAS Services Application deploys a set of services called Remote Services. Remote Services are used by the SAS Information Delivery Portal, the SAS Stored Process Web application, and other Web applications. The SAS Services Application must be started before you start your Web application server.

## **The Components of the SAS Forecast Server**

The SAS Forecast Server bundle consists of the following components:

#### SAS Forecast Server

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

#### SAS Forecast Studio

SAS Forecast Studio is the primary client application of the 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 state of the SAS Forecast Server project.

*Note:* In addition, SAS Forecast Server clients require a connection to the middle tier.

#### SAS Forecast Server Plug-ins for SAS Management Console

The SAS Forecast Server Plug-ins for SAS Management Console provide a graphical user interface for managing the resources and content of the SAS Forecast Server Mid-Tier. You can use the SAS Forecast Server Plug-ins 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-ins manage 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-ins 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 Mid-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-ins for SAS

Management Console. In addition, macros for creating and running projects are included in the batch interface. As a result, the SAS Forecast Batch Interface is the best option for scripting and scheduling.

#### SAS Time Series Studio Mid-Tier

The SAS Time Series Studio Mid-Tier is deployed on an application server. The SAS Time Series Studio Mid-Tier is a collection of custom services that are used to support the SAS Time Series Studio Client.

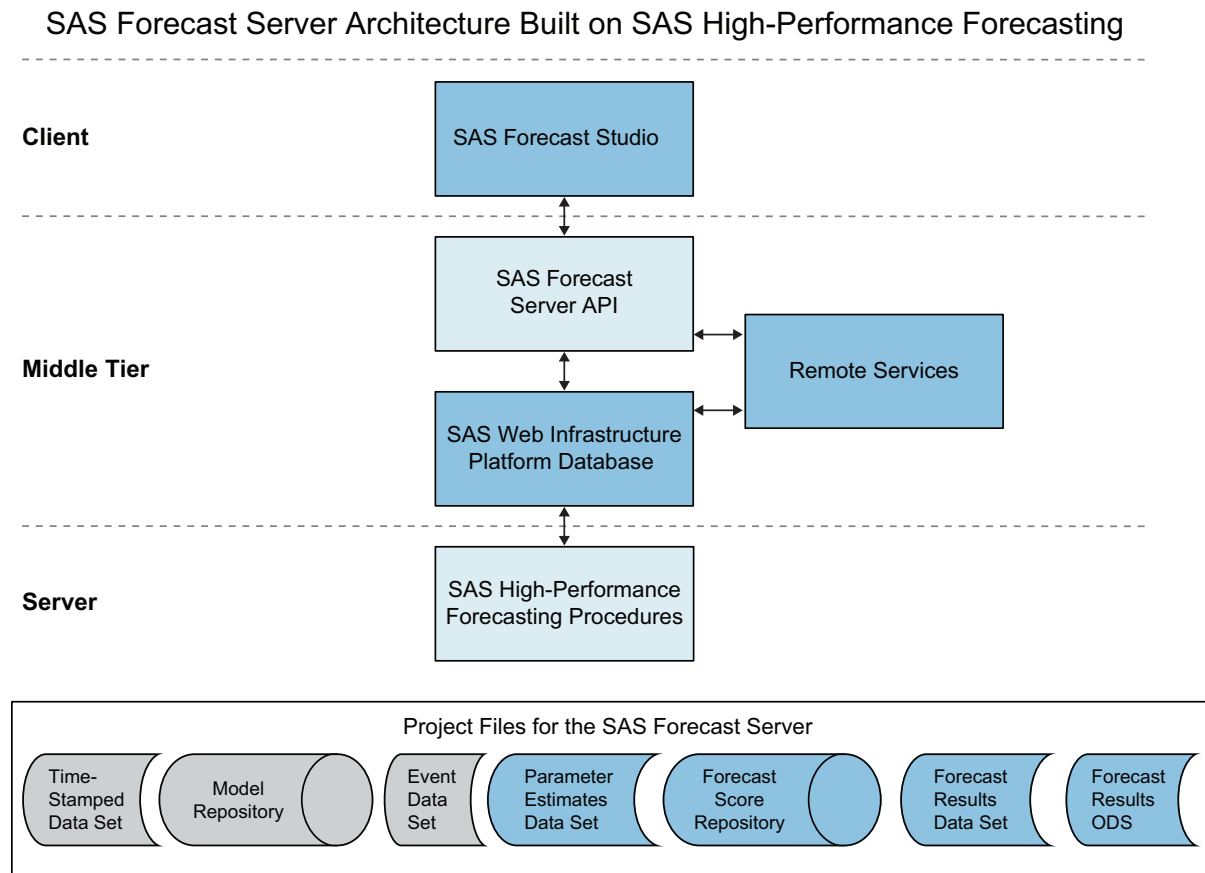
#### SAS Time Series Studio Client

The SAS Time Series Studio Client is experimental for the release 12.1. It requires a 64-bit Windows middle tier. It enables you to interactively explore time series data. You can interactively structure the time series data from several hierarchical and frequency perspectives. You can interactively query (or subset) the time series data using hierarchical queries, graphical queries, filtering queries, or manual selection. Given a target series, you can interactively search for similar series. After retrieving time series data, you can analyze the data using common statistical time series analysis techniques. After understanding the patterns in the selected time series, the data can be exported for subsequent analysis such as forecasting, econometric analysis, pricing analysis, risk analysis, time series mining, and other analyses related to time series data.

---

## SAS Forecast Server Integration

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

**Figure 1.2** The SAS Forecast Server Integration Architecture

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

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

SAS Forecast Studio accesses the SAS Forecast Server Mid-Tier when it needs to access the SAS Metadata Server or the SAS Workspace Server. The SAS Forecast Server Mid-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>	
<b>Pre-Installation Tasks</b> .....	<i>11</i>
<i>Chapter 3</i>	
<b>Installing the SAS Forecast Server</b> .....	<i>15</i>
<i>Chapter 4</i>	
<b>Post-Installation Tasks</b> .....	<i>21</i>



## Chapter 2

# Pre-Installation Tasks

---

<b>Create an Operating System Account for Product Administrators and Users . . .</b>	<b>11</b>
About the User Accounts for the SAS Forecast Server . . . . .	11
Create an Operating System Account for the Product Administrator . . . . .	12
Create an Operating System Account for a SAS Forecast Studio User . . . . .	12
<b>Install Your Web Application Server . . . . .</b>	<b>12</b>
<b>Create and Configure a Forecasting Environment . . . . .</b>	<b>13</b>
About Forecasting Environments . . . . .	13
How You Can Create Forecasting Environments . . . . .	13
Requirements for a New Forecasting Environment . . . . .	14
<b>Determine the Location of the SAS Environment URL . . . . .</b>	<b>14</b>

---

## Create an Operating System Account for Product Administrators and Users

### *About the User Accounts for the SAS Forecast Server*

SAS Forecast Server has two types of user accounts:

#### product administrator

A product administrator is specific to the SAS Forecast Server. A product administrator account is not the same as a general administrator account, such as the SAS administrator (sasadm@saspw) account. A product administrator must have a valid host operating system account, and you must associate that account with a metadata user using SAS Management Console.

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 12](#).

#### SAS Forecast Studio user

A SAS Forecast Studio user must have a valid host operating system account, and you must associate that account with a metadata user using SAS Management Console. In addition, a SAS Forecast Studio user must have full access to the appropriate project areas on each SAS Workspace Server.

You can create regular a user account for the SAS Forecast Server as a post-installation task. For more information, see [“Manage Roles and Capabilities” on page 23](#).

### 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 12.

---

## Install Your Web Application Server

Before you install the SAS Forecast Server, you must have installed a Web application server. For more information about how to install and configure a Web application server so that it works with SAS, see <http://support.sas.com/resources/thirdpartysupport/v93/appservers/index.html>.



---

## Create and Configure a Forecasting Environment

### About Forecasting Environments

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

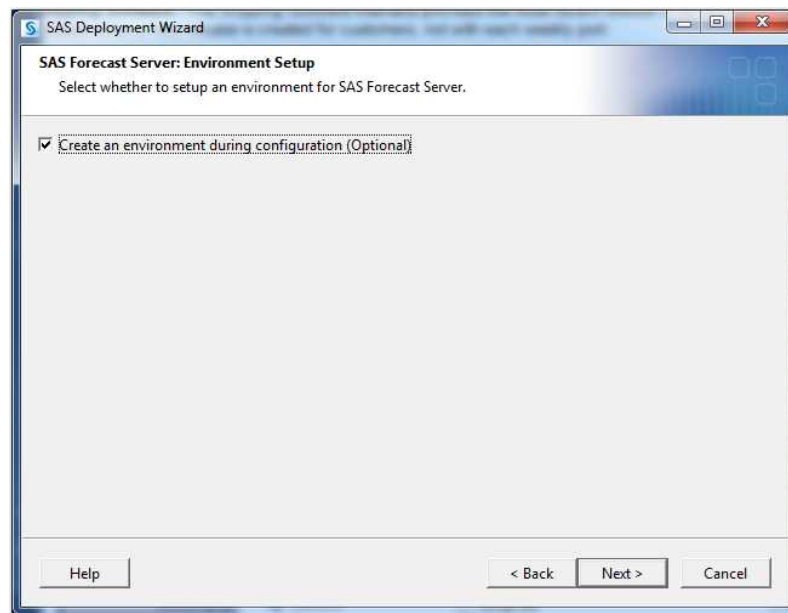
A forecasting environment is a product workspace for a product session. A forecasting environment is created by the SAS Forecast Server product administrator, and it is used only by the SAS Forecast Server and its client applications. Forecasting environments can be used to organize your projects and to control access to SAS Forecast Server projects.

### How You Can Create Forecasting Environments

You must create at least one forecasting environment before you can create a project in SAS Forecast Studio.

You can create a forecasting environment using one of the following ways:

- You can create a forecasting environment when you install the SAS Forecast Server by using the SAS Forecast Server: Environment Setup step in the SAS Deployment Wizard. For more information, see [“How to Install and Configure the SAS Forecast Server”](#) on page 15.



- After the SAS Forecast Server is installed, you can create a forecasting environment using the SAS Forecast Server Plug-ins for SAS Management Console or the SAS Forecast Project Manager. For more information, see the online Help for the SAS Forecast Server Plug-ins for SAS Management Console or the online Help for the SAS Forecast Project Manager.

### Requirements for a New Forecasting Environment

All product users who need to access the forecasting environment must have full access to the file system content in that environment directory, including the root directory and any subdirectories. A product administrator needs to set permissions based on the specific operating environment. A product administrator should verify that every user has permission to access any new files that are created in the environment directory.

---

## 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://<your HTTP server>/sas-environment.xml`. This file defines a set of SAS deployments at your site for SAS Forecast Server 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>.

## Chapter 3

# Installing the SAS Forecast Server

---

About the Deployment of the SAS Forecast Server .....	15
How to Install and Configure the SAS Forecast Server .....	15

---

## About the Deployment of the SAS Forecast Server

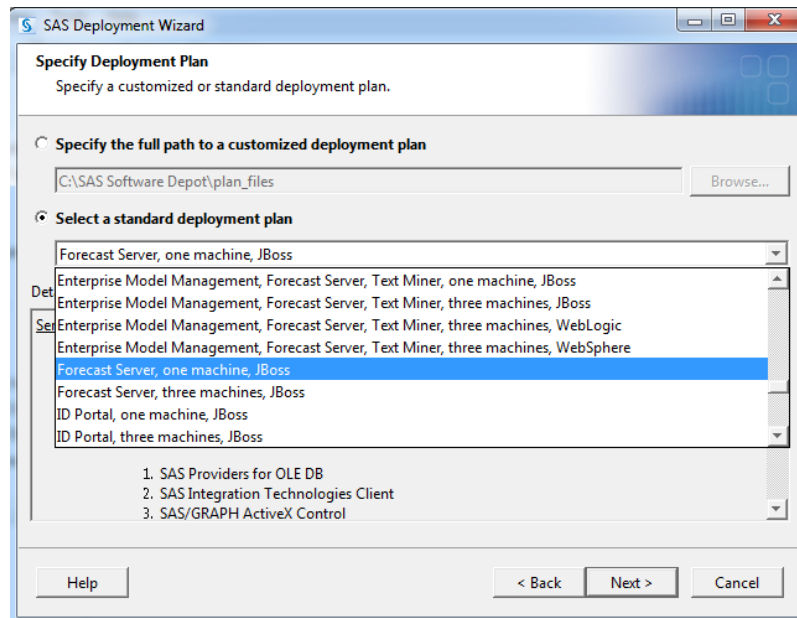
When you deploy the SAS Forecast Server, you deploy all of the components that are part of the SAS Forecast Server architecture. For more information, see [“Overview of the SAS Forecast Server Architecture” on page 4](#).

---

## How to Install and Configure the SAS Forecast Server

To install and configure the SAS Forecast Server, you use the SAS Deployment Wizard and follow the basic process described in the *SAS Intelligence Platform: Installation and Configuration Guide*. Some steps in the SAS Deployment Wizard are specific to installing and configuring the SAS Forecast Server. Here is the information that you need to install the SAS Forecast Server:

1. Start the SAS Deployment Wizard from your SAS Software Depot. For example, on a Windows system, double-click the setup.exe file located in your SAS Software Depot folder.
2. In the Specify Deployment Plan step, select your deployment plan. It is recommended that you use one of the standard deployment plans created by SAS. If you need a custom plan, contact your SAS support personnel for assistance.

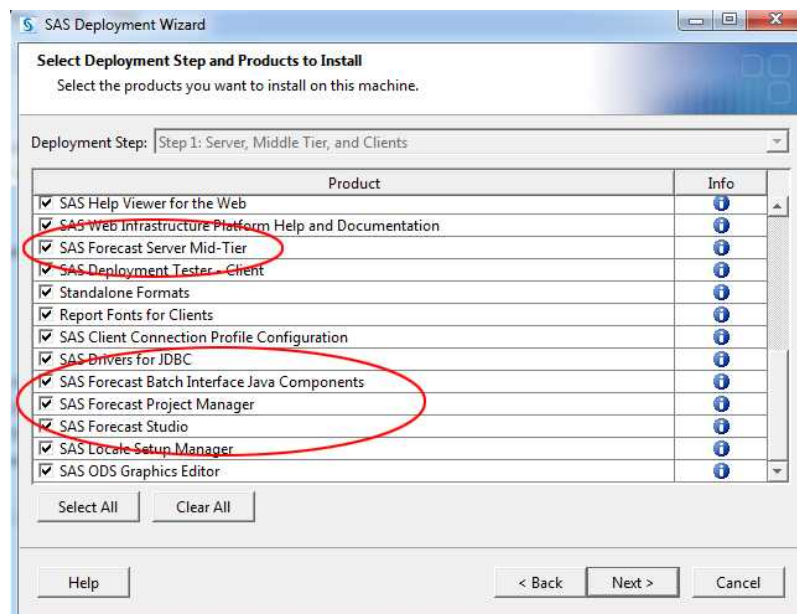


*Note:* To install the experimental release of SAS Time Series Studio Client, you must have a custom plan. For more information, contact your SAS support personnel.

3. In the Select Deployment Step and Products to Install step, verify the products that you want to install on your machine. Depending on what software you license at your site, additional products might be installed. For a multiple-machine deployment, products and components can be installed on several machines.

In this example, the following SAS Forecast Server components are being installed on a single machine:

- SAS Forecast Server Mid-Tier
- SAS Forecast Batch Interface Java Components
- SAS Forecast Project Manager
- SAS Forecast Studio

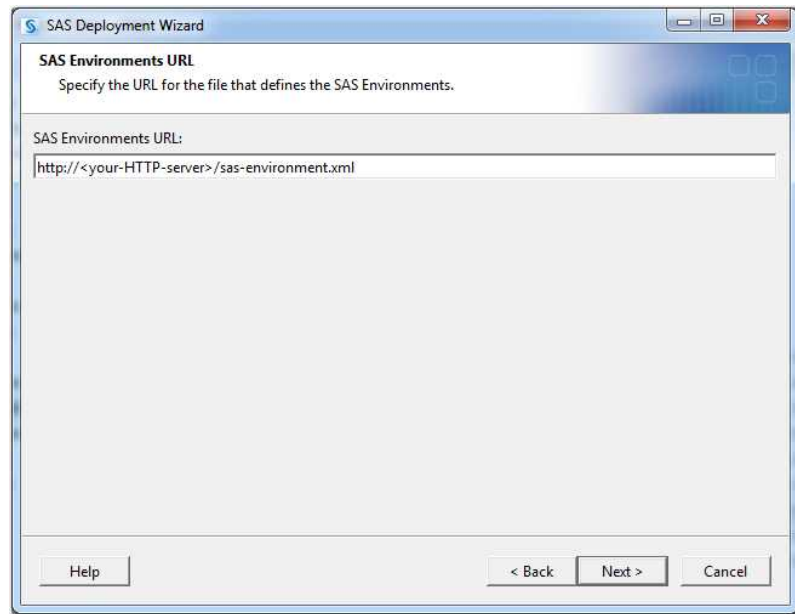


Click **Next**.

4. In the Select Microsoft Office Applications step, select the Microsoft Office applications in which you want to run the SAS Add-In for Microsoft Office. Click **Next**.

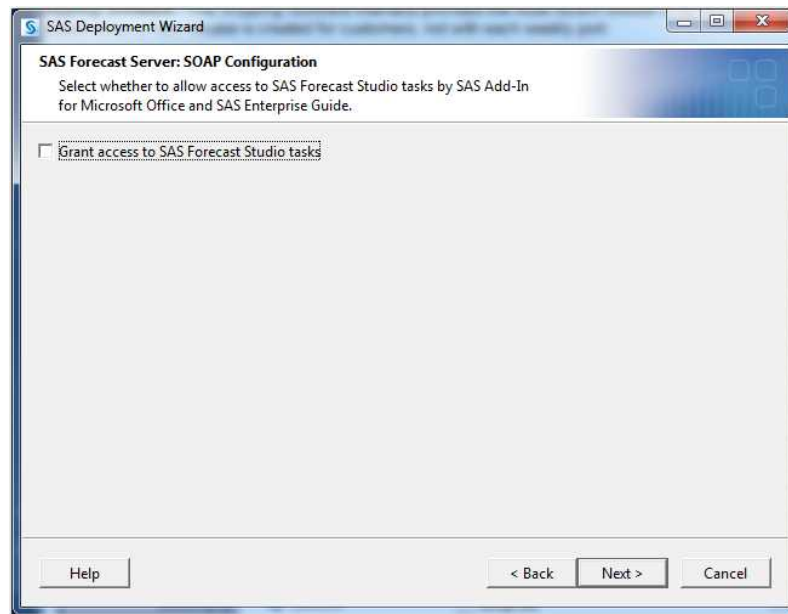
*Note:* This step is available only if your site licenses the SAS Add-In for Microsoft Office.

5. In the SAS Environments URL step, specify the URL location of the SAS environment file. For more information, see [“Determine the Location of the SAS Environment URL”](#) on page 14.

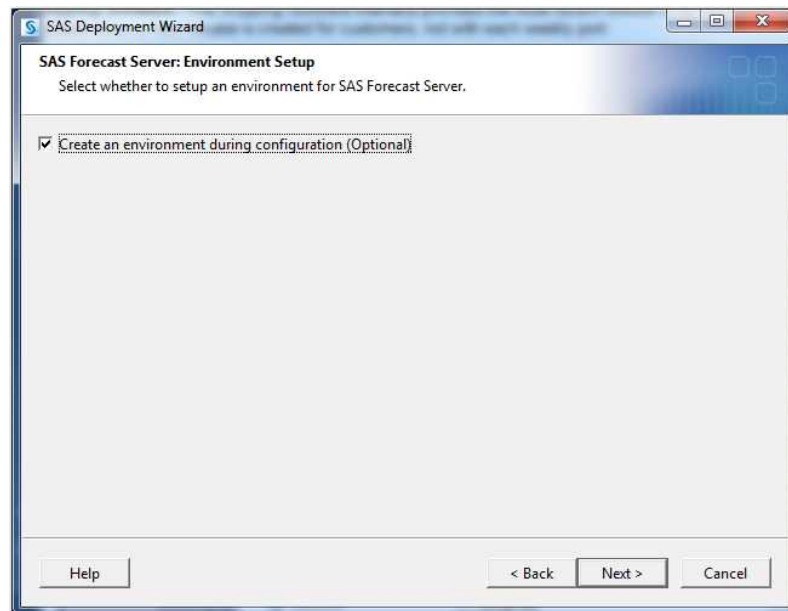


6. In the SAS Forecast Server: SOAP Configuration step, select **Grant access to SAS Forecast Studio tasks** to allow the SAS Add-In for Microsoft Office and SAS Enterprise Guide to access SAS Forecast Studio tasks. For more information, see [“SAS Forecast Studio Tasks in Other SAS Software”](#) on page 169.

Click **Next**.



7. In the SAS Forecast Server: Environment Setup step, you can select to **Create an environment during configuration**. Selecting this option enables you to create a forecasting environment during configuration, not as a post-installation step. A forecasting environment is required. It serves as a container for SAS Forecast Server projects and should not be confused with a SAS environment. If you choose to create a forecasting environment as a post-installation step, you must create at least one forecasting environment in a SAS Forecast Server client before using SAS Forecast Studio. Click **Next**.



If you select the **Create an environment during configuration** check box, registration information for the Default forecasting environment is created.

The screenshot shows the 'SAS Forecast Server: Environment Setup' dialog box. It 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\Config\Lev1\AppData\SASForecastServer12.1'. At the bottom, there are buttons for 'Help', '< Back', 'Next >', and 'Cancel'.

*Note:* In this example, the content location refers to a path on the SAS Workspace Server. The path that you specify for the content location must meet the requirements for a new forecasting environment. As a result, you might have to manually configure this path on the SAS Forecast Server as a post-installation step. For more information about forecasting environments and their requirements, see [“About Forecasting Environments” on page 13](#).

8. In the SAS Internal Account: Forecast Server Metadata User step, specify the password for the metadata user.

The screenshot shows the 'SAS Internal Account: Forecast Server Metadata User' dialog box. It contains several text input fields: 'Display Name' with the value 'Forecast Server Metadata User', 'Name' with the value 'fsmeta', 'Internal User ID' with the value 'fsmeta@saspw', 'Internal Password' (empty), and 'Confirm Internal Password' (empty). At the bottom, there are buttons for 'Help', '< Back', 'Next >', and 'Cancel'.

9. In the Deployment Summary, review the products that you are about to install on your machine, and click **Start**.
10. 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

---

<b>Minimum Requirements for Post-Installation</b> . . . . .	<b>21</b>
<b>Create a Forecasting Environment</b> . . . . .	<b>21</b>
<b>Set Permissions for Forecasting Environments in UNIX</b>	
<b>Operating Environments</b> . . . . .	<b>22</b>
Use Operating System Groups to Assign Permissions . . . . .	22
Conditions for the SAS Forecast Server User Group . . . . .	22
Update the SAS Scripts to Grant Permissions to the SAS	
Forecast Server User Group . . . . .	22
<b>Manage Roles and Capabilities</b> . . . . .	<b>23</b>
About Metadata Groups, Roles, and Capabilities . . . . .	23
How to Configure the SAS Forecast Server Users . . . . .	28

---

## 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 or the sample reports that are provided with the SAS Forecast Server. If that is the case, you do not need to complete certain post-installation tasks.

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

- create a forecasting environment
- set the permissions for each forecasting environment
- specify the roles and capabilities for each user of the SAS Forecast Server

---

## Create a Forecasting Environment

When you install the SAS Forecast Server using the SAS Deployment Wizard, you have the option of creating a default forecasting environment. If you do not create a forecasting environment during installation, then you must create the environment as a post-installation task. You can create this environment using the SAS Forecast Server Plug-ins for SAS Management Console or the SAS Forecast Project Manager.

For more information, see [“Create and Configure a Forecasting Environment”](#) on page 13.

---

## Set Permissions for Forecasting Environments in UNIX Operating Environments

### *Use Operating System Groups to Assign Permissions*

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

### *Conditions for the SAS Forecast Server User Group*

If you are working in a UNIX operating environment, the following conditions must be met:

- A group of SAS Forecast Server users is created for the UNIX operating environment. The logon ID of each SAS Forecast Server user must be in this group. The group must include any user who might run code that is created from a SAS Forecast Server project in a SAS session.
- A user can be a member of multiple groups, but the SAS Forecast Server user group is the primary group for each user.
- SAS scripts are updated to grant permissions to each SAS Forecast Server user on the SAS Workspace Server and the SAS Stored Process Server. For more information, see [“Update the SAS Scripts to Grant Permissions to the SAS Forecast Server User Group”](#) on page 22.
- Each forecasting environment directory has the correct ownership, and the SAS Forecast Server user group members have Read, Write, and Execute permissions.

### *Update the SAS Scripts to Grant Permissions to the SAS Forecast Server User Group*

Using the UMASK option, you can grant permissions to a SAS Forecast Server user on a conditional basis if the user is a member of the SAS Forecast Server user group.

*Note:* The following example might require changes to match your server configuration. In particular, this example could result in changed permissions to other SAS files such as OLAP cubes. For example, if you are working with multiple UNIX user groups, and your site has a SAS OLAP Server, you must ensure that the account under which the SAS OLAP Server runs maintains the Read and Execute permissions to OLAP cubes.

To grant permissions to the SAS Forecast Server user group:

1. On each SAS Workspace Server, open `<config-dir>/Lev1/SASApp/appservercontext_env_usermods.sh`.
2. Enter the configuration information for your operating environment. Here is the general format of this code:

*Note:* The following code uses grave accents, not quotation marks.

```
CMD=<your-operating-system-path>
CURR_GID=`eval $CMD -g`
GID=<solution-group-id>
if [$CURR_GID -eq $GID]; then umask 002fi
```

- a In **CMD=<your-operating-system-path>**, specify the full path on your SAS Workspace Server where the ID command is stored. You can get this information by submitting a **which id** or **whence id** command.
- b In **GID=<solution-group-id>**, specify the group ID. Type **id** to get the GID and UID information.
- c A value of 002 is recommended for the **umask** option.

Here are code examples for each UNIX environment where the SAS Forecast Server is supported:

Operating Environment	Sample Code
AIX HP 64 (HP-RISC) HP 64 Itanium	<pre>CMD=/usr/bin/id CURR_GID=`eval \$CMD -g` GID=201 if [\$CURR_GID -eq \$GID]; then umask 002 fi</pre>
Solaris 64 Solaris X64	<pre>CMD=/usr/xpg4/bin/id CURR_GID=`eval \$CMD -g` GID=201 if [\$CURR_GID -eq \$GID]; then umask 002 fi</pre>
Linux	<pre>#!/bin/bash CMD=/usr/bin/id CURR_GID=`eval \$CMD -g` GID=500 if ["\$CURR_GID" -eq "\$GID"]; then umask 002 fi</pre>

## Manage Roles and Capabilities

### About Metadata Groups, Roles, and Capabilities

#### Default Roles for the SAS Forecast Server

SAS Forecast Server provides default metadata roles. When the SAS Forecast Server is deployed, capabilities are already assigned to the default metadata roles. Using SAS

Management Console, you can add capabilities to these default roles. You can create your own metadata roles and specify capabilities for those roles.

The following roles are provided with the SAS Forecast Server:

Role	Description
Forecast Server: Administrator	Manages product content, such as projects.
Forecast Server: Forecaster	Creates projects and generates forecasts.
Forecast Server: Analyst	Analyzes the time series data and the results of the forecast.
Forecast Server: Browser	Views the results of the forecasts and uses reports.

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

### **Capabilities in the SAS Forecast Server**

The following table lists the capabilities in the SAS Forecast Server. Use these capabilities to prevent a user from requesting that the SAS Forecast Server perform a specific action, such as generate a forecast or update a model specification.

*Note:* Granting a capability to a user does not necessarily ensure access to an action.

Other settings (such as project configuration, project ownership, and metadata permissions) can limit access to an action.

**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-ins for SAS Management Console and the SAS Forecast Project Manager.
SOAP Service Bridge	Enables the use of 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 Batch Interface (for example, the FSCOPY macro).

**Table 4.2** General Features

Capability	Description
Administer Product	<p>Enables the user to administer the metadata for and the projects in the SAS Forecast Server.</p> <p>Users who are assigned this capability are exempt from the SAS Forecast Server's internal security checks, which are based on project ownership and whether the project is shared. However, these users are still subject to other restrictions imposed by other capabilities and metadata permissions.</p>
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 <b>Active series</b> check box in the Forecasting View and the Modeling View.
Import New Data	<p>Enables the user to choose whether to update the project when the SAS Forecast Server determines that changes have been made to the input data set. For example, this capability selects the <b>Use updated data if available</b> check box in several dialog boxes in SAS Forecast Studio, such as the Update Project Version dialog box and the Reforecast Project dialog box.</p> <p>Users who are not assigned this capability cannot update the input data set for the project. Instead, the project uses the version of the data that was available when the project was created.</p>

**Table 4.3** Forecasts

Capability	Description
Modify Forecasts	Enables the user to forecast the project or series in SAS Forecast Studio.
Reconcile Forecasts	<p>Enables the user to reconcile the hierarchy. For example, this capability enables the <b>Reconcile Hierarchy</b> menu item, the Reconcile Hierarchy icon in the Forecasting View, and the <b>Reconcile</b> message (that appears in the workspace if there are override conflicts or unresolved nodes).</p>

Capability	Description
Override Forecasts	Enables the user to create, edit, or delete override values. For example, this capability enables the Override calculator, the ability to lock overrides, and ability to select <b>Set Scenario Forecast Values as Overrides</b> in the Scenario Analysis View.

**Table 4.4** Models

Capability	Description
Change Model Selection	Enables the user to change the baseline model for the series. For example, in the Modeling View, this capability enables the <b>Set this model as forecast model</b> and <b>Reset to “automatic selection”</b> links.
Modify Models	Enables the user to create, delete, copy, and edit models. This capability 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 whether an event is used in the model. For example, this capability enables the <b>Usage in system-generated models</b> menu in the Events Properties dialog box.
Modify Events	Enables the user to create, delete, copy, and edit events. This capability 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 the user to view the properties of SAS Forecast Server environments.
Manage Environment Settings	Enables the user to edit the properties of SAS Forecast Server environments.
Manage Environments	Enables the user to create, delete, and rename environments.

**Table 4.7** Projects

Capability	Description
Create Projects	Enables the user to create a new project. This capability enables the New Project wizard.
Manage Projects	Enables the 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 the user to specify whether a project is shared and to change the ownership of a project. For example, this capability enables the <b>New Owner</b> and <b>Share</b> options in the SAS Forecast Server Plug-ins for SAS Management Console.
Manage Project Settings	Enables the 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 the user to view the project code that runs when the project is opened and when the project is closed. For example, this capability enables the <b>Start-up and shutdown Code</b> button in the New Project wizard and in the Project Properties dialog box.
Modify Project Scripts	Enables the 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 SAS 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 the user to edit the content in the Notes panel at the bottom of the Forecasting View. This capability enables the user to create and remove notes.

**Table 4.9** Reports

Capability	Description
Use Reports	Enables the <b>Reports and Stored Processes</b> menu item and the Reports and Stored Processes dialog box in SAS Forecast Studio.
Manage Reports	Enables the user to manage reports. 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 the user to create new scenarios and to edit, save, and delete existing scenarios.

**Table 4.11** Special Features

Capability	Description
Legacy Features	Enables legacy features. This capability is provided only to help users through transitional periods that result from design changes in the next release. Support for these legacy features is limited. You should enable legacy features only if directed by SAS Technical Support.
Experimental Features	Enables experimental features that are still under development. These experimental features might change or be removed in a future release. No migration support is provided for these experimental features. You should enable experimental features only if directed by SAS Technical Support.
Debugging Features	Enables debugging features that are provided to help users and SAS Technical Support when problems arise. You should enable debugging features only if directed by SAS Technical Support.

### How to Configure the SAS Forecast Server Users

*Note:* On each SAS Workspace Server that contains forecasting environments, you must have an operating system account for each SAS Forecast Server user. For more



information, see [“Create an Operating System Account for Product Administrators and Users”](#) on page 11.

You can use the default metadata roles that are provided with the SAS Forecast Server to quickly configure your SAS Forecast Server users. Each default metadata role is assigned capabilities that enable access to features.

The following steps configure an administrator account for the SAS Forecast Server by using a default role. In this example, all SAS Forecast Server roles are listed.

The defining quality of an administrator account is the Administer Product capability that grants the administrator account expanded privileges in the product security model. In this example, the Administer Product capability is granted by assigning the Forecast Server: Administrator role.


To configure the metadata roles:

1. Start SAS Management Console. 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 **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.

The Forecast Server: Administrator role is sufficient if the administrator account only needs to manage actions in the SAS Forecast Server. However, the Forecast Server: Administrator role does not grant access to all product features. To give the administrator account 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**.

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



## Part 3

---

# Configuration Tasks

<i>Chapter 5</i>	
<b><i>Specifying Security Permissions for Users and Groups</i></b> .....	<b><i>33</i></b>
<i>Chapter 6</i>	
<b><i>Administering Reports to Use in SAS Forecast Studio</i></b> .....	<b><i>41</i></b>
<i>Chapter 7</i>	
<b><i>Creating and Configuring Libraries</i></b> .....	<b><i>63</i></b>
<i>Chapter 8</i>	
<b><i>Customizing Project Code, Time Intervals, Formats, and Events</i></b> ..	<b><i>69</i></b>
<i>Chapter 9</i>	
<b><i>Miscellaneous Administration Tasks</i></b> .....	<b><i>79</i></b>



## Chapter 5

# Specifying Security Permissions for Users and Groups

---

<b>Enable Users to Access the SAS Forecast Server</b> . . . . .	<b>33</b>
Initial Users . . . . .	33
The SAS Forecast Server Metadata User . . . . .	34
Project Owners . . . . .	35
<b>Secure Access to the SAS Forecast Server</b> . . . . .	<b>36</b>
Security Layers . . . . .	36
Notes Regarding Permissions . . . . .	36
Environment Permissions . . . . .	37
Project Permissions . . . . .	38
Sharing Options for Projects . . . . .	39

---

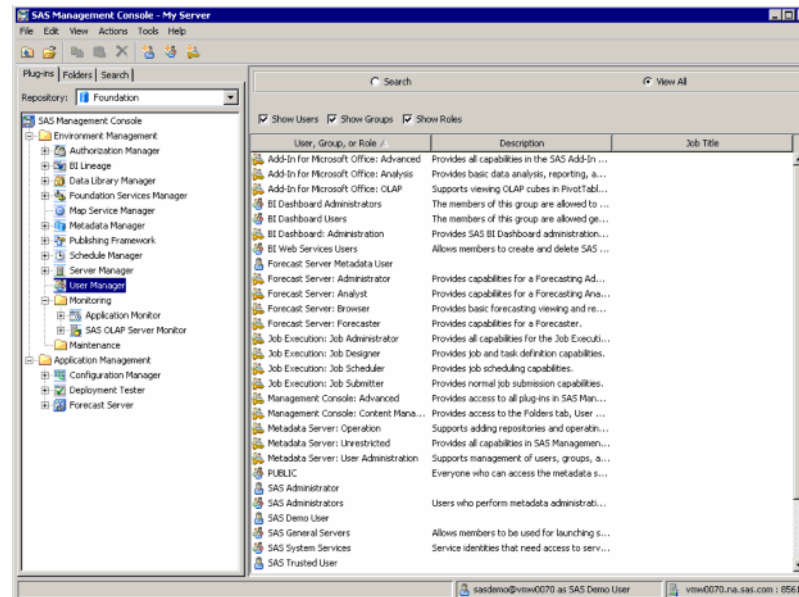
## Enable Users to Access the SAS Forecast Server

### *Initial Users*

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

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

**Display 5.1** Example of Standard Users and Groups Created for the SAS Forecast Server



## The SAS Forecast Server Metadata User

### What Is the SAS Forecast Server Metadata User?

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

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

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

*Note:* No product environment can be associated with a server without having WriteMetadata permission to the defining server object. This security requirement is imposed by the SAS Metadata Server. However, the SAS Forecast Server does not modify server objects.

### **Specify User Permissions for the Metadata User**

To grant the Metadata User the appropriate permissions to a folder outside of the **/System/Applications/SAS Forecast Server** directory:

1. Start SAS Management Console. Connect as a SAS administrator (for example, `sasadm@saspw`).
2. Click the **Folders** tab.
3. Locate the metadata folder that you created.
4. Right-click on the metadata folder, and then select **Properties**. The Properties dialog box appears.
5. Click the **Authorization** tab.
6. Next to **Users and Groups**, click **Add**. The Add Users and Groups dialog box appears.
7. In the **Available Identities** list, double-click **Forecast Server Metadata User**. The Forecast Server Metadata User is added to the **Selected Identities** list. Click **OK**.
8. In **Users and Groups**, select **Forecast Server Metadata User**.
9. Grant the **ReadMetadata**, **WriteMetadata**, **WriteMemberMetadata**, and **CheckInMetadata** permissions to the Forecast Server Metadata User, and then click **OK**.

**TIP** When appropriate, grant the **ReadMetadata**, **WriteMetadata**, **WriteMemberMetadata**, and **CheckInMetadata** permissions to the parent folder. A child folder automatically inherits these permissions from its 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. This user is referred to as the project owner. In addition to the default capabilities, a project owner can perform management actions, such as changing project-sharing settings and deleting a project. You can transfer project ownership to another user by using the SAS Forecast Server Plug-ins for SAS Management Console, the SAS Forecast Project Manager, or the FSSETOWN macro. A project can have only one owner.

The value for the project owner is detected by comparing a generated identifier for the user (called an identity token) with a previously stored token value for the user. To determine the previously stored token value for the user who is currently logged in, either look at the session record on the SAS Forecast Studio Status page or select the **Show my (token-value) projects only** check box in the Projects dialog box in SAS Forecast Studio.

*Note:* In many cases, the identity token for a user matches the user name.

### **How to Change Project Ownership**

You can change project ownership using the SAS Forecast Server Plug-ins for SAS Management Console or the SAS Forecast Project Manager. For more information, see the online Help for the SAS Forecast Server Plug-ins for SAS Management Console or the online Help for SAS Forecast Project Manager.

## Secure Access to the SAS Forecast Server

### Security Layers

Security settings in the SAS Forecast Server are implemented in four layers.

- capabilities, which are enforced by the SAS Forecast Server
- metadata permissions
  - ReadMetadata, WriteMetadata, WriteMemberMetadata, and CheckInMetadata, which are enforced by the SAS Metadata Server
  - other permissions, such as library authorizations, which are enforced by the SAS Forecast Server
- project ownership and sharing, also called the application security model, which are enforced by the SAS Forecast Server
- file system permissions, which are enforced by the operating system

Typically, permissions are checked in the order in which they are listed. For example, capabilities are checked first and file system permissions are checked last. In order for a user to be able to access project content, all four layers of security must permit this access.

### Notes Regarding Permissions

#### File System Permissions

Use metadata and SAS Forecast Server permissions when you define your security strategy. Typically, file system permissions are not checked before an operation is attempted in a product, which could result in run-time errors.

*Note:* Only using file system permissions to secure content is not recommended.

#### WriteMetadata Permissions

In the SAS Forecast Server, the metadata security that is applied to an object with product content is limited to controlling the visibility of this content using the ReadMetadata permission. Metadata updates to an object are performed indirectly by the Metadata User. The WriteMetadata settings for an individual user account have no impact on the product's behavior, so WriteMetadata permission should be granted so that a user can see an object with product content.

#### What Permissions Can You Control Using Metadata?

To secure access to metadata objects that represent SAS Forecast Server data, you can grant or deny permissions to a user or group 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



Metadata Object	Permissions
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 using the ReadMetadata permission to hide server objects) can impact the product's behavior. However, these other permissions are usually not needed under typical operating conditions.

## Environment Permissions

### About Environment Permissions

An environment is implemented as a metadata folder, a metadata object, or a directory structure (in the server file system). Only metadata and file system permissions apply to an environment. To enable project sharing within an environment, you must grant full control to all users who are allowed to access the environment. Metadata permissions should be used to secure the environment.

The ReadMetadata permission at the environment level controls whether a user can access the projects and reports within an environment. When a user logs on to SAS Forecast Studio, environments to which he or she does not have ReadMetadata permission do not appear as projects in the Projects dialog box. The user cannot select the project or its content. This environment is not visible in any SAS Forecast Server client as well.

*Note:* Each file system location on a SAS 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's content and the ability to properly coordinate client access.

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

### Configure Environment Permissions

To configure permissions for an environment:

1. In the file system, grant full control to all users and user groups who are allowed to access the environment. A user group for all users of the environment is

recommended. For more information about user groups, see [“Set Permissions for Forecasting Environments in UNIX Operating Environments”](#) on page 22.

2. Start SAS Management Console. Connect as a SAS administrator (for example, `sasadm@saspw`).
3. Click the **Folders** tab.
4. Expand the **System**, **Applications**, **SAS Forecast Server**, **Forecast Server 12.1**, and **Environments** folders.
5. Right-click on an environment folder, and then select **Properties**. The environment Properties dialog box appears.
6. Click the **Authorization** tab.
7. Select a user or user group in **Users and Groups**.
8. Grant the **ReadMetadata** permission to the user or user group, and then click **OK**.

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

## Project Permissions

### About Project Permissions

A project is similar to an environment in physical structure. The content of a project is saved to an environment subdirectory. The security for the file system was specified when you created the environment. Because a project's content is saved in an environment subdirectory, the project files inherit the permissions from the environment. New project metadata is saved in this environment. It inherits the permissions from the environment. As a result, no additional security is required when a project is 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 permissions for a project:

1. Start SAS Management Console. Connect as a SAS administrator (for example, `sasadm@saspw`).
2. Click the **Folders** tab.
3. Expand the **System**, **Applications**, **SAS Forecast Server**, **Forecast Server 12.1**, and **Environments** folders.
4. Select an 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 a user or user group in **Users and Groups**.
8. Grant the **ReadMetadata** permission to the user or user 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 project. By enabling project sharing, any user who can see a project can access the project. You can enable project sharing by selecting the **Allow other users to view and edit this project** check box in the New Project wizard or the Project Properties dialog box.

### Enable Project Sharing

To enable project sharing:

1. Start SAS Management Console. To log on, your user account must be assigned to the Forecast Server: Administrator group. You must have a valid operating system account.
2. Click the **Plug-ins** tab. Expand the **Application Management** and **Forecast Server** folders, and then expand the environment folder that contains the project that you want to share.
3. Right-click on a project, and then select **Properties**. The Project 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 use of environments does divide users into implied work groups. For example, all users who can access a particular environment can be considered a work group. Therefore, when you enable sharing of a project, you are essentially sharing the project with other members of the environment work group. A user who does not have access to the environment in which the project is stored cannot access the project. Use caution when copying projects that are shared with another environment because the work group changes based on the environment.



## Chapter 6

# Administering Reports to Use in SAS Forecast Studio

---

<b>Overview of Reports</b> .....	<b>41</b>
What Is a Report? .....	41
Types of Reports .....	42
Manage Reports .....	43
Access the Reports in SAS Forecast Studio .....	44
<b>Use the Sample Reports</b> .....	<b>45</b>
Deploy the Sample Reports .....	45
Remove the Sample Reports .....	45
<b>Register Reports</b> .....	<b>46</b>
Preregistration Tasks .....	46
Register a Report .....	47
<b>Create Reports</b> .....	<b>48</b>
Types of Report Parameters .....	48

---

## 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 the *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 generate 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.

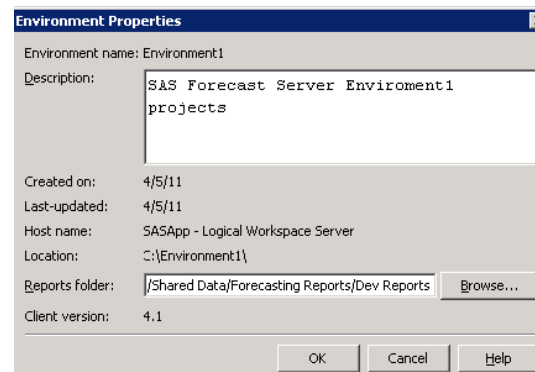
When you register a report object in SAS Management Console for use with the SAS Forecast Server, you must use the FS\_REPORT keyword to distinguish the report object from other stored processes. The sample reports provided with the SAS Forecast Server use the FS\_SAMPLE keyword to distinguish them from 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.

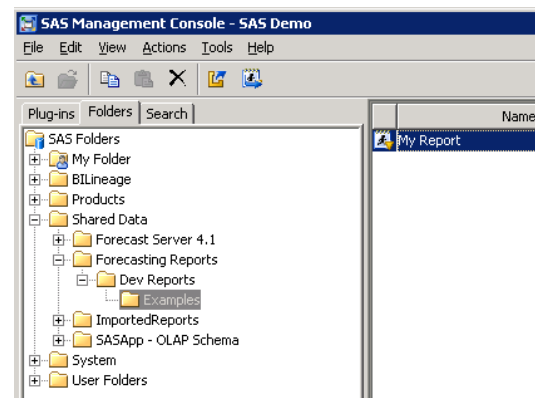
For example, suppose that you configure your environment to use the **/Shared Data/Forecasting Reports/Dev Reports** report folder as shown in the following display:

**Display 6.1** Configuring the Environment's Root Report Folder



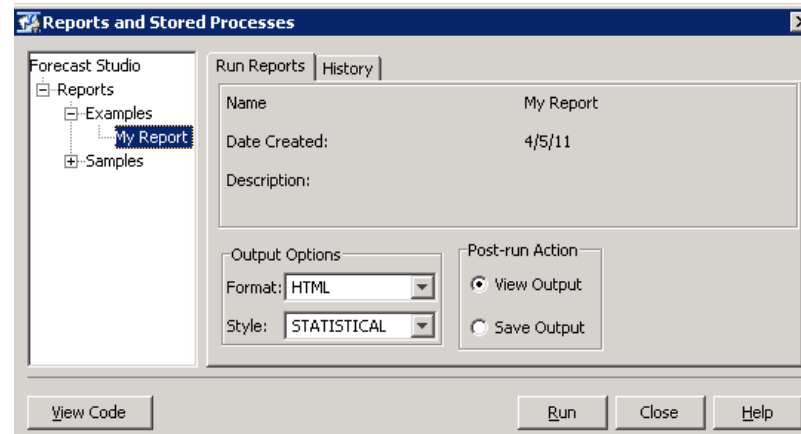
In addition, suppose that you create the **My Report** stored process metadata object at the folder location **/Shared Data/Forecasting Reports/Dev Reports/Examples**, as shown in the following display:

**Display 6.2** 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.

**Display 6.3** 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.

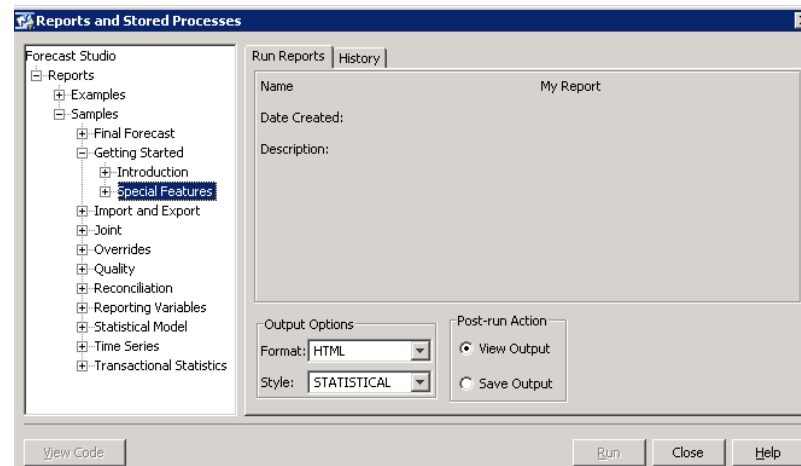
Before you can create a report for use in SAS Forecast Studio, you must let the environment know where your report folder is located. You can configure the location for the report folder in the Environment Properties dialog box.

## 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:

**Display 6.4** 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.
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**. 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 **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.
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.

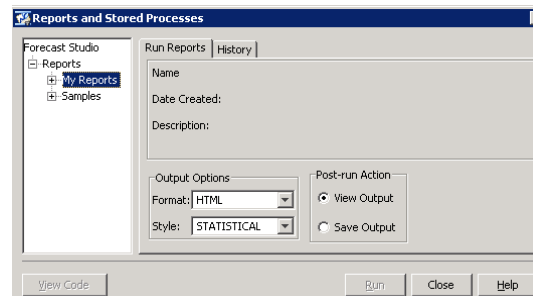
For example, the example.sas file is saved in the **C:\MyStoredProcesses** directory. The source repository location is **C:\MyStoredProcesses**, and the source filename is example.sas.

*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:

**Display 6.5** Example Report Hierarchy



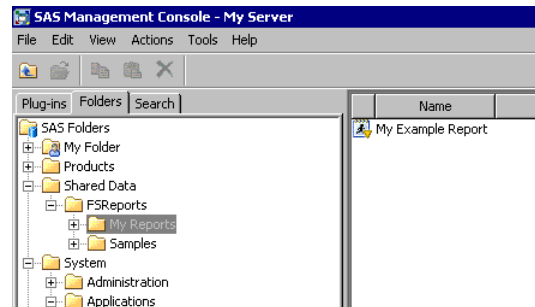
For more information about creating report folders, see the documentation for SAS Management Console at <http://support.sas.com/documentation/onlinedoc/sasmc>.

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**:

**Display 6.6** 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 the *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 the *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 6.1 on page 49](#).)

#### **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 6.1** *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
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\_INDVAR\_S 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 6.2** 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:  <code>%include "&amp;HPF_INCLUDE";</code>	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 6.3** 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
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 6.4** 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 6.5** Basic Project Macro Variables

Name	Description	Format
HPF_PROJECT	Project name.	SAS name
HPF_DESC	Project description.	SAS label
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	<p>File system path to the project include file.</p> <p>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.</p> <p>By default, all SAS libraries and catalogs are assigned with Read-Only access (ACCESS=READONLY).</p> <p>For example, the following SAS code assigns project library names with Read-Only access:</p> <pre>%include "&amp;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 6.6** 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 $n^{\text{th}}$ 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 $n^{\text{th}}$ 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
HPF_NUM_INDEPVARS	Number of independent variables. If there are no independent variables, HPF_NUM_INDEPVARS is set to zero.	Positive integer

Name	Description	Format
HPF_INDEPVAR#	<p>Space-delimited list of all independent variable names.</p> <p>The order of the independent variable names is the same as specified in the project.</p> <p>The macro variable is always defined. If there are no independent variables, HPF_INDEPVAR# is set to NULL.</p>	List of SAS names separated by a single space
HPF_INDEPVAR#	<p>Individual independent variable names listed in the <math>n^{\text{th}}</math> position of the ordered list of independent variables (HPF_INDEPVAR#).</p> <p>The first independent variable name is stored in HPF_INDEPVAR1, the second in HPF_INDEPVAR2, and the last is stored in HPF_INDEPVAR&amp;HPF_NUM_INDEPVAR#.</p> <p>If there are no independent variables (&amp;HPF_NUM_INDEPVAR# is zero), these macro variables are not defined.</p>	SAS name
HPF_NUM_REPORTVARS	<p>Number of reporting variables.</p> <p>If there are no reporting variables, then HPF_NUM_REPORTVARS is set to zero.</p>	Positive integer
HPF_REPORTVARS	<p>Space-delimited list of all reporting variable names.</p> <p>The order of the reporting variable names is the same as specified in the project.</p> <p>The macro variable is always defined. If there are no reporting variables, HPF_REPORTVARS is set to NULL.</p>	List of SAS names separated by a single space
HPF_REPORTVAR#	<p>Individual reporting variable names listed in the <math>n^{\text{th}}</math> position of the ordered list of reporting variables (HPF_REPORTVARS).</p> <p>The first reporting variable name is stored in HPF_REPORTVAR1, the second in HPF_REPORTVAR2, and the last is stored in HPF_REPORTVAR&amp;HPF_NUM_REPORTVARS.</p> <p>If there are no reporting variables (&amp;HPF_NUM_REPORTVARS is zero), these macro variables are not defined.</p>	SAS name
HPF_TIMEID	Time ID variable name.	SAS name
HPF_TIMEID_FORMAT	SAS format of the time ID variable.	SAS format name

Name	Description	Format
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_LIBNAME.&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_LIBNAME.&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
HPF_LEVEL_BYVARS#	Space-delimited list of BY variable names associated with the $n^{\text{th}}$ 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

Name	Description	Format
HPF_LEVEL_DATAWHERE#	<p>Input data filtering WHERE clause for the <math>n^{\text{th}}</math> level, where <math>n</math> ranges from 1 to &amp;HPF_NUM_LEVELS.</p> <p>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.</p> <p>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&amp;HPF_NUM_LEVELS.</p> <p><i>Note:</i> You must unquote this macro variable. For example,</p> <pre>%unquote (&amp;&amp;HPF_CURRENT_DATAWHERE&amp;n) ;</pre>	SAS WHERE clause
HPF_LEVEL_OUTWHERE#	<p>Output data filtering WHERE clause for the <math>n^{\text{th}}</math> level, where <math>n</math> ranges from 1 to &amp;HPF_NUM_LEVELS.</p> <p>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.</p> <p>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&amp;HPF_NUM_LEVELS.</p> <p><i>Note:</i> You must unquote this macro variable. For example,</p> <pre>%unquote (&amp;&amp;HPF_LEVEL_OUTWHERE&amp;n) ;</pre>	SAS WHERE clause
HPF_LEVEL_LIBNAME#	<p>SAS library reference for the <math>n^{\text{th}}</math> level, where <math>n</math> ranges from 1 to &amp;HPF_NUM_LEVELS.</p> <p>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&amp;HPF_NUM_LEVELS.</p>	SAS LIBNAME

Name	Description	Format
HPF_LEVEL_NSERIES#	<p>Number of series associated with the <math>n^{th}</math> level, where <math>n</math> ranges from 1 to &amp;HPF_NUM_LEVELS.</p> <p>The number of series at level 1 (the top) is stored in HPF_LEVEL_NSERIES1.</p> <p>The number of series at the lowest level (the leaves) is stored in HPF_LEVEL_NSERIES&amp;HPF_NUM_LEVELS.</p>	Positive integer

**Diagnose Macro Variables**

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

**Table 6.7** 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.	<i>p</i> -value

**Model Selection Macro Variables**

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

**Table 6.8** 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_HOLDOUTPERCENT	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 6.9** 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.	<i>p</i> -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 6.10** 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 6.11** 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	<p>Input data WHERE clause for the currently selected node.</p> <p>This WHERE clause can be used to subset the input data set to obtain information about the currently selected node of the hierarchy.</p> <p><i>Note:</i> You must unquote this macro variable. For example,</p> <pre>%unquote (&amp;HPF_CURRENT_DATAWHERE) ;</pre>	SAS WHERE clause
HPF_CURRENT_OUTWHERE	<p>Output data WHERE clause for the currently selected node.</p> <p>This WHERE clause can be used to subset the output data set to obtain information about the currently selected node of the hierarchy.</p> <p><i>Note:</i> You must unquote this macro variable. For example,</p> <pre>%unquote (&amp;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	<p>Space-delimited list of BY variable names for the currently selected level of the hierarchy.</p> <p>The macro variable is always defined. If there are no BY variables, HPF_CURRENT_BYVARS is set to NULL.</p>	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	<p>Horizon date, datetime, or time value of the current node.</p> <p>The time ID value of the start of the multi-step ahead forecast for the currently selected node of the hierarchy.</p>	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 45](#).

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();

%stpbeg;
```

## Chapter 7

# Creating and Configuring Libraries

---

<b>Libraries</b> .....	<b>63</b>
Overview of Libraries .....	63
External Library Assignments .....	63
Libraries Assigned by the SAS Forecast Server .....	64
Authorization Checks .....	65
<b>Enable Expanded Support for Libraries</b> .....	<b>65</b>
<b>How to Define a Pre-assigned Metadata Library</b> .....	<b>66</b>
<b>How to Define a Metadata Library That Is Assigned by the SAS Forecast Server</b> .....	<b>67</b>
<b>How to Define an Automatic Environment Library</b> .....	<b>67</b>
<b>How to Define a Configured Environment Library</b> .....	<b>68</b>

---

## 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 Statements: Reference*.

*Note:* Library names cannot begin with an underscore character. A library name that begins with an underscore is the naming convention reserved for an internal SAS library.

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

### External Library Assignments

Libraries can be assigned when a SAS server session is created or in the start-up code for a specific forecasting environment. Because these libraries are not assigned by the SAS Forecast Server, they are external library assignments.

Here are the four main sources of external library assignments:

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

### ***Libraries Assigned by the SAS Forecast Server***

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

Because of security, on-demand libraries are not available in the SAS Forecast Server by default. SAS Forecast Server has a configuration setting that enables the use of each type of on-demand library. Types include manually assigned metadata libraries, configured environment libraries, and automatic environment libraries. For more information, see the [“Enable Expanded Support for Libraries” on page 65](#).

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

- Libraries that are defined in the metadata, but they are not pre-assigned. When the 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. This assignment method ensures that metadata permissions are followed when accessing the library.

*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 it 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 they are located in a metadata-defined library that is not pre-assigned.

- Configured environment libraries that are defined in the `<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 full path to the library directory
  - whether access to the library should be Read-Only

When the 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 `<environment-directory>\Libraries` directory. The

name of the subdirectory implies the LIBNAME for the library assignment and the access settings for the library.

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

Configured and automatic environment libraries enable you to perform the following tasks:

- Manage libraries for users who have access to the file system.
- Define local libraries for individual environments.

## Authorization Checks

A key to good library management and use is performing authorization checks on the data accesses that are attempted by users. How these authorization checks are performed depends on the type of library that the user is trying to access.

Two strategies are used.

- The SAS Intelligence Platform and the SAS Metadata Server require that a user has ReadMetadata permission on a library object for the library to be available. For libraries that are defined in the metadata, the SAS Forecast Server requires that the user has ReadMetadata permission for the user to have Read access to the contents of the library. It requires that the user has WriteMetadata permission for the user to have Write access to the contents of the library. Permissions can be set for elements within the library, such as a data set (table) or a data set variable (column). For example, when a client application tries to read a variable in a data set, the SAS Forecast Server searches for an associated Column, Table, or Library object, in that order. An authorization check is performed on the first object that is found.

**TIP** You can create Table objects for any type of metadata library using the Data Library Manager plug-in for SAS Management Console. For more information, see the online Help for the Data Library Manager plug-in for SAS Management Console.

- All other libraries (for example, the configured and automatic environment libraries) are assigned using the BASE engine. Or, they are assigned in a way that cannot be determined by the SAS Forecast Server (for example, when a LIBNAME statement is included in a SAS Foundation configuration file or SAS server configuration file). For these libraries, the SAS Forecast Server checks the dictionary tables for the SAS session. Permissions for a library are specified by the Read-Only access setting in the dictionary table.

Any additional security settings are not visible to the SAS Forecast Server. As a result, errors might occur because SAS Forecast Security cannot see security settings before an action is attempted. These errors often occur when working with SAS/ACCESS libraries.

---

## Enable Expanded Support for Libraries

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

To enable library access:

1. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw). Connect to a metadata repository.
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Server**.
6. In **Forecast Server > Library Access**, select whether users can use libraries using the **Manual-assign metadata libraries**, **Automatic environment libraries**, and **Configured environment libraries** menus.
7. Click **OK**.

---

## How to Define a Pre-assigned Metadata Library

To define a pre-assigned metadata library:

1. Create a metadata definition for the library.
  - a. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw). Connect to a metadata repository.
  - b. Expand the **Data Library Manager** folder, and select **Libraries**.
  - c. Right-click the library that you want to pre-assign, and select **Properties**.
  - d. Select the **Options** tab.
  - e. Click **Advanced Options**. The Advanced Options dialog box appears.
  - f. Select the **Library is Pre-Assigned** check box. The selected library is assigned whenever a SAS session is started for one of the assigned SAS servers.
  - g. Verify that the library is assigned to the SAS servers.
  - h. Click **OK**.
2. Set the metadata permissions on the new library for the SAS Forecast Server product administrator and users.
  - a. In SAS Management Console, expand the **Data Library Manager** folder, and select **Libraries**.
  - b. Right-click the library, and select **Properties**.
  - c. Select the **Authorization** tab.
  - d. In **Users and Groups**, select the product administrator, and grant permissions by selecting the **Grant** check boxes. Select each SAS Forecast Server user, and grant permissions by selecting the **Grant** check boxes.
 

Verify that the fsmeta account has ReadMetadata permission on the library. If the fsmeta account does not have ReadMetadata permission, the 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 the SAS Forecast Server

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

1. Verify that the SAS Forecast Server is configured to support automatic environment libraries. For more information, see the [“Enable Expanded Support for Libraries” on page 65](#).
2. Create a metadata definition for the library.
  - a. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw). Connect to a metadata repository.
  - b. Expand the **Data Library Manager** folder, and select **Libraries**.
  - c. Right-click the library that you want to pre-assign, and select **Properties**.
  - d. Select the **Options** tab.
  - e. Click **Advanced Options**. The Advanced Options dialog box appears.
  - f. Verify that the library is assigned to the SAS servers.
  - g. Click **OK**.
3. Set the metadata permissions on the new library for the SAS Forecast Server product administrator and users.
  - a. In SAS Management Console, expand the **Data Library Manager** folder, and select **Libraries**.
  - b. Right-click the library, and select **Properties**.
  - c. Select the **Authorization** tab.
  - d. In **Users and Groups**, select the product administrator, and grant permissions by selecting the **Grant** check boxes. Select each SAS Forecast Server user, and grant permissions by selecting the **Grant** check boxes.

Verify that the fsmeta account has ReadMetadata permission on the library. If the fsmeta account does not have ReadMetadata permission, the 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.

---

## How to Define an Automatic Environment Library

*Note:* You can create an environment using the SAS Forecast Project Manager. For more information, see the *SAS Forecast Studio: User's Guide* or the online Help for the SAS Forecast Project Manager.

To define an automatic environment library:

1. Verify that the SAS Forecast Server is configured to support automatic environment libraries. For more information, see the [“Enable Expanded Support for Libraries” on page 65](#).

2. In the `<environment-directory>\Libraries` directory, create a subdirectory for the automatic environment library. The name of the subdirectory implies the LIBNAME for the library assignment. For example, if the LIBNAME is Hpfuser, then the name of the subdirectory 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, a library is assigned Read and Write access. If the library should have Read-Only access, append an **.r** to the directory name. For example, the **hpfuser.r** directory creates the Hpfuser library with Read-Only access.

When a user opens this forecasting environment in SAS Forecast Studio, the new library is available.

---

## How to Define a Configured Environment Library

*Note:* You can create an environment using the SAS Forecast Project Manager. For more information, see the *SAS Forecast Studio: User's Guide* or the online Help for the SAS Forecast Project Manager.

To define a configured environment library:

1. Verify that the SAS Forecast Server is configured to support configured environment libraries. For more information, see the [“Enable Expanded Support for Libraries” on page 65](#).
2. In the `<environment-directory>\Config` directory, create a `libs.sas7bdat` file. This file (data set) should contain three character variables: LIBNAME, PATH, and READONLY.

*Note:* If the `libs.sas7bdat` data set is not created, the SAS Forecast Server tries to create it when a user opens this forecasting environment in SAS Forecast Studio.

3. Edit the `libs.sas7bdat` 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 READONLY variable, specify **yes**.

When a user opens this forecasting environment in SAS Forecast Studio, the new library is available.



## Chapter 8

# Customizing Project Code, Time Intervals, Formats, and Events

---

<b>Add Start-Up and Shutdown Code to a Project</b> .....	<b>69</b>
Start-Up or Shutdown Code .....	69
Configure Code to Run .....	70
Add Start-Up and Shutdown Code to Your Project .....	70
<b>Create Custom Time Intervals</b> .....	<b>71</b>
Create a Custom Time Interval .....	71
Example 1: Create a StoreHours Custom Interval .....	72
Example 2: Create a Custom Interval Definition Interactively .....	73
<b>Create a Custom Format</b> .....	<b>74</b>
<b>Create a Customized List of Events</b> .....	<b>75</b>
Create a Customized Lists of Events .....	75
Example: Create Super Bowl Events .....	76

---

## Add Start-Up and Shutdown Code to a Project

### *Start-Up or Shutdown Code*

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

*Note:* Do not use start-up code to assign a library that contains the input data set for a project. The input data set must be available before the project is created or opened. The library cannot be assigned before the input data.

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

Although start-up and shutdown code is associated with a project, the scope of the code applies to an entire SAS Forecast Server session. It is important that the shutdown code closes any processes that were started by the start-up code. For example, suppose you specified the MPRINT system option to run when the project was opened. If you do not turn off the MPRINT system option in the shutdown code, when you close the project and open another project in the same environment, SAS Forecast Studio will still be tracing any SAS statements that are generated for macro variables. The MPRINT option

is still specified in the log. If you turn off the MPRINT system option in the shutdown code, when you close the project, tracing is turned off when you open another project.

Start-up code is saved in the startup.sas file, and shutdown code is saved in the shutdown.sas file. These files are in the `<project-directory>\Config` directory. The code can be modified only when the project is open in SAS Forecast Studio.

## Configure Code to Run

Site administrators can specify whether the code should run for each deployment of the SAS Forecast Server. If the code should not run, a message appears, stating that code execution is not permitted. This message appears at the top of the SAS Start-up and Shutdown Code dialog box. Because of security, the code is not run by default.

To specify that the code should run for a deployment of SAS Forecast Server:

1. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
2. On the **Plug-ins** tab, expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Server**.
6. In **Forecast Server**, select whether start-up and shutdown code will be run using the **Project scripts** menu.
7. Click **OK**.
8. Restart the SAS Forecast Server for these changes to take effect.

## Add Start-Up and Shutdown Code to Your Project

When you create a project, you can add start-up and shutdown code using the New Project wizard. After the project is created, you can add or edit this code 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-ins for SAS Management Console. For more information, see [“Manage Roles and Capabilities” on page 23](#).

To add start-up or shutdown code:

1. Open SAS Forecast Studio. Select **File** ⇒ **Projects**. In the Projects dialog box, open the project that you want to modify.
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. On the **Start-up Code** and **Shutdown Code** tabs, enter the SAS code that you want to include in the project.

*Note:* If the code should not run, a message appears at the top of the SAS Start-up and Shutdown Code dialog box. To enable the code to run, see [“Configure Code to Run” on page 70](#).

5. 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** check box. To view the code that is saved in memory, click **View the stored shutdown code**.

*Note:* When a project is opened, the shutdown code (that will run when you close the project) is saved 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 the changes to be reflected in the saved copy of the shutdown code.

6. Click **OK** to save your changes and close the SAS Start-up and Shutdown Code dialog box. Click **OK** again to close the Project Properties dialog box.

---

## Create Custom Time Intervals

### Create a Custom Time Interval

SAS Forecast Server includes numerous time intervals. However, your site might need a custom time interval.

To create a custom interval:

1. Using a DATA step program or a text 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 the raw data contains date values (the number of days since January 1, 1960), use a date value for the BEGIN variable and assign a date format. If the raw data contains datetime values (the number of seconds since midnight January 1, 1960), use a datetime value for the BEGIN variable and assign a datetime format.

You can define the end of each period (using an END variable) and the seasonal cycle (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 the END variable for each observation is one less than the value of the BEGIN variable for 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. Calculations include forecasting and operations that might extend beyond the time series (such as filters).

#### **CAUTION:**

**An error will occur if a date or datetime value (in the actual data, in a forecast, or in a plot) is outside the range of the custom interval definition.**

For your custom interval, the value for the BEGIN variable must start with the earliest historical date 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 value 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 the SAS Forecast Server, the syntax for the INTERVALDS= system option is more restrictive than the syntax 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 to 32 characters. The name cannot contain an underscore or an embedded numeric character (with the exception that the name can end with one or more digits). For example, StoreHours12 is a valid name.

Store12Hours is an invalid name. When you specify multiple custom 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 library reference 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 intervals=(StoreHours=CustIntLib.StoreHoursDS);
```

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

### Example 1: Create a StoreHours Custom Interval

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

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:

```
-intervals (StoreHours=CustIntLib.StoreHoursDS);
```

2. Start a SAS session on the SAS Workspace Server. In this SAS session, complete these steps:

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

```
data CustIntLib.StoreHoursDS (keep=BEGIN END); 1
  start = '01JAN2009'D;
  stop  = '31DEC2009'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;

```

- a** 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 dates of the first and last observations in the data set.

- b** The DO loop analyzes each observation from 01JAN2009 to 31DEC2009 (inclusive).
- Use the WEEKDAY function to determine the day of the week for a specific date. By default, the day of the week is 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 datetime value.
- c** The FORMAT statement specifies a DATETIME. format for the values of the BEGIN and END variables.

### Example 2: Create a Custom Interval Definition Interactively

This example shows how to create a custom interval 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. For **Name**, enter the LIBNAME that you specified in the INTERVALDS= system option.
4. For **Path**, click **Browse**, and select the path where the custom interval will be stored. Click **OK** to close the Select dialog box. Click **OK** again to close the New Library dialog box.
5. In the SAS Explorer, right-click the library that you 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 top of column A, and select **Column Attributes**.
8. Specify these properties:
  - Change **Name** from **A** to **BEGIN**.

- Select **Numeric** for **Type**.
- Enter **DATE9 .** as the format and informat.

Click **Close**.

- Click the first cell in the BEGIN column to edit that cell. Type **01JAN2011**, and press Enter.
- In the second cell, enter the date of the next period. Continue entering dates in cells as needed. When you are finished, close **VIEWTABLE**. Click **Yes** to save your changes. In the Save As dialog box, select the library that you entered in step 3.
- For **Member Name**, enter the name of the data set that you specified in the INTERVALS= system option.
- Click **Save** to save the data set.

---

## Create a Custom Format

SAS Forecast Server includes numerous formats. However, you might have formats at your site that are not available in SAS Forecast Studio. Using the FORMAT procedure, you can create custom formats and make them available. For more information about this procedure, see the *Base SAS Procedures Guide*.

To create a custom format:

- In PROC FORMAT, specify where to store the custom format using the LIBRARY= option.
  - To store the format in a library in the default location, write the following SAS code:

```
libname library
<SAS-configuration-directory>/Levn/SASApp/SASEnvironment/SASFormats;
proc format library=library;
...
```

- To store the format in the **C:\myfmts** location on the SAS Workspace 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).

- 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—is 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 the SAS Forecast Server.

- If the format is stored in the default library location (**<SAS-configuration-directory>/Lev1/SASApp/SASEnvironment/SASFormats**), then no further modifications are required.
- If the format is stored in a different location, you must configure SAS to search in that library in addition to the default library.

To search in 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. In the following code, the SET system option associates the FSFMTS catalog with the Myfmts library on a UNIX system. Set the FMTSEARCH system option to FSFMTS so that the SAS Forecast Server searches this library.

```
-set FSFMTS ("/myfmts")
-fmtsearch (FSFMTS)
```

When you restart the SAS Workspace Server, the SAS Forecast Server resolves any references to custom formats that are stored in **/myfmts**.

---

## Create a Customized List of Events

### Create a Customized Lists of Events

SAS Forecast Studio provides a list of predefined events. However, you might need to add to this list or create a customized list. To create an event, you must use the HPFEVENTS procedure first, and then use the DATEKEYS procedure to associate a date key, label, and locale with this event. The procedure results are saved in 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 these event data sets so that the events appear in SAS Forecast Studio.

To create a customized list:

1. Using a DATA step program or a text 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 SAS configuration file (**sasv9\_usermods.cfg** or **sasv9.cfg**) that is used by the SAS Workspace Server.

The customized list of events should appear in the list of events the next time you start SAS Forecast Studio.

### Example: Create Super Bowl Events

1. Start a SAS session on the SAS 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 SAS 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 events data set.

```

EVENTSDS=("c:\eventslist")

```



## Chapter 9

# Miscellaneous Administration Tasks

---

<b>Change Configuration Properties in SAS Management Console</b> . . . . .	<b>79</b>
<b>Specify a Time-Out</b> . . . . .	<b>79</b>
Specify a Time-Out for SAS Forecast Studio . . . . .	79
Specify a Time-Out for the SOAP Bridge . . . . .	80
<b>Configure a SAS Environment URL</b> . . . . .	<b>80</b>
<b>Set the Preferred SAS Environment in the Logon Dialog Box for SAS Forecast Studio</b> . . . . .	<b>81</b>
<b>Configure the Archive Functionality</b> . . . . .	<b>81</b>
<b>Remove a Product Configuration with the SAS Deployment Manager</b> . . . . .	<b>82</b>
<b>Archive a Project</b> . . . . .	<b>83</b>

---

## Change Configuration Properties in SAS Management Console

The SAS Forecast Server middle tier has several configuration properties. 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. If you change the value of a configuration property, you must restart the SAS Forecast Server. For more information about the Configuration Manager, see the SAS Management Console online Help.

*Note:* If your site allows JMX access, then you can use a JMX call to indicate when the middle tier should reload the configuration properties. A JMX call can eliminate the need to restart the SAS Forecast Server for the new configuration properties to take effect. For more information, see the [“JMX MBeans” on page 188](#).

---

## Specify 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 five minutes to see whether the SAS Forecast Studio

session has timed out. You can change the values of the time-out and how frequently the SAS Forecast Server checks the session.

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

1. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Server**.
6. In **Forecast Server > Session Timeout**, specify new values for the **Idle time limit** option and the **Timeout check interval** option.

Click **Reset**.

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

### ***Specify a Time-Out for the SOAP Bridge***

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

---

## **Configure a SAS Environment URL**

For some deployments of SAS Forecast Studio, the list of SAS environments that are available to users at logon might be fixed and universal. For these deployments, you can add a SAS environment URL as part of the configuration for the Java Web Start client. By using a central SAS environment URL, you do not need to manually specify the URL on every client.

If you use a central URL, a user cannot change the logon choices on his or her machine. If the logon choices need to vary from machine to machine, you should not use a central URL.

To configure the central URL:

1. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Studio (Java Web Start)**.

6. In **SAS environment URL**, specify the URL for the sas-environment.xml file.  
Click **Reset**.
7. Restart the Web application server for these changes to take effect.

---

## Set the Preferred SAS Environment in the Logon Dialog Box for SAS Forecast Studio

When a user logs on to SAS Forecast Studio, he or she must specify a SAS environment to use. You can set a preferred SAS environment that is selected by default. Or, you can force users to use a specific SAS environment when logging on to SAS Forecast Studio. If you force users to use a specific environment, you should configure the SAS environment URL in the middle tier to use this same environment. For more information, see the [“Configure a SAS Environment URL” on page 80](#).

*Note:* If the preferred SAS environment does not exist at run time, SAS Forecast Studio assumes that 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. Connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Studio (Java Web Start)**.
6. In **Preferred SAS environment**, specify the name of the SAS environment that should be selected by default in the logon dialog box.

*Note:* This name is case sensitive.

Click **Reset**.

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

---

## Configure the Archive Functionality

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

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

To modify archiving properties:

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

2. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Server**.
6. In **Forecast Server > Archiving**, you can set the following archiving properties:

Property	Description	Default Value
Compression level	Specifies how much to compress a new archive.	The same as the default for Java
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 an archive. A comma-separated list of values is expected.	sas7bdat
Catalog files to CPORT (by extension)	Specifies catalog files that you want to include in an archive. A comma-separated list of values is expected.	sas7bcatalog
Additional CPORT option	Enables you to include additional options to insert into the PROC CPORT statement that is used to create an archive.	(blank)

Click **Reset**.

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

---

## Remove a Product Configuration with the SAS Deployment Manager

To remove a product configuration:

1. Navigate to the SASHome directory. For example, on a Windows system, navigate to **C:\Program Files\SASHome\SASDeploymentManager\9.3**.  
Double-click **sasdm.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 connection information to the SAS Metadata Server, and then click **Next**.

5. Select the product that you want to remove, and then click **Next**. For example, to remove the SAS Forecast Server, select **Forecast Server 12.1**.
6. SAS Forecast Server has metadata that describes user-defined environments and projects. This metadata is stored in the SAS Metadata Server. If you are removing the SAS Forecast Server, you can select to unregister the user content. This option removes the SAS Forecast Server metadata when you remove the SAS Forecast Server. However, the metadata in the file system is not removed. Therefore, you can use it to re-create the metadata if needed.

*Note:* You can use the SAS Forecast Server Plug-ins for SAS Management Console and the SAS Forecast Project Manager to register and unregister user content. For more information about registering and unregistering user content, see the online Help for the SAS Forecast Server Plug-ins for SAS Management Console and the online Help for SAS Forecast Project Manager. In addition, you can use the macros in the SAS Forecast Batch Interface. For more information, see the [“About the SAS Forecast Batch Interface” on page 94](#).

7. In **Summary**, click **Start**.
8. Restart the Web application server for these changes to take effect.

---

## Archive a Project

You can archive a project by using the SAS Forecast Server Plug-ins for SAS Management Console or the SAS Forecast Project Manager. For more information, see the online Help for the SAS Forecast Server Plug-ins for SAS Management Console or the online Help for the SAS Forecast Project Manager. In addition, you can archive projects by using the %FSEXPORT macro.

To archive a project by using the SAS Forecast Server Plug-ins for SAS Management Console:

1. Start SAS Management Console. Select a connection profile.
2. Click the **Plug-ins** tab. Expand the **Application Management** and **Forecast Server** folders. Expand the environment folder that contains the project that you want to archive.
3. Expand the **Projects** folder, and right-click on a project. Select **Archive** from the menu. The Archive dialog box appears.
4. Enter a name for the archived project. By default, the SAS Forecast Server uses the project name. The name must be a valid SAS name. It cannot be the name of an existing archived project.
5. Enter a description for the archived project.
6. Enter a location for the archived project.
7. Click **OK**.





## **Part 4**

---

# Using the SAS Forecast Server Clients

<i>Chapter 10</i>	
<b>SAS Forecast Studio</b> .....	87
<i>Chapter 11</i>	
<b>SAS Forecast Batch Interface</b> .....	93
<i>Chapter 12</i>	
<b>SAS Forecast Studio Tasks in Other SAS Software</b> .....	169



## Chapter 10

# SAS Forecast Studio

---

<b>Requirements for Starting SAS Forecast Studio</b> . . . . .	<b>87</b>
<b>Run SAS Forecast Studio from a Local Installation</b> . . . . .	<b>87</b>
<b>Use Java Web Start</b> . . . . .	<b>88</b>
How to Launch SAS Forecast Studio . . . . .	88
Configure the Logon Behavior for Java Web Start . . . . .	88
Customize the Default URL for Java Web Start . . . . .	90
<b>Start SAS Forecast Studio with Options</b> . . . . .	<b>90</b>
Open a Specific Forecasting Environment or Project in SAS Forecast Studio . . . .	90
Specify the Number of Observations to Use to Detect the Time Interval . . . . .	90
Specify the Sample Size for Validating BY Variable Values and Formats . . . . .	91

---

## Requirements for Starting SAS Forecast Studio

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

---

## Run SAS Forecast Studio from a Local Installation

SAS Forecast Studio runs only on the Windows operating system. To start SAS Forecast Studio on the system where it is installed, select **Start** ⇒ **All Programs** ⇒ **SAS** ⇒ **SAS Forecast Studio** ⇒ **SAS Forecast Studio 12.1**.

## Use Java Web Start

### How to Launch SAS Forecast Studio

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

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

- Open the SAS Forecast Server Current Status Web page. The default URL for this page is the location from which the SAS Forecast Server is deployed. An example 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 Java Web Start. You can launch SAS Forecast Studio from `http://<your-server-name>:8080/SASForecastServer/main.jnlp`.
- Use the `LaunchForecastStudio` directive.

### Configure the Logon Behavior for Java Web Start

#### Default Logon Behavior for the Local Installation of SAS Forecast Studio and Java Web Start

The behavior of SAS Forecast Studio is the same whether you launch SAS Forecast Studio from a local installation or by using Java Web Start. The main difference between these two choices is visible in the logon dialog box.

- When you launch SAS Forecast Studio from a local installation, the default value in the **SAS environment** list is the last SAS environment that you used on that machine. If you have not previously logged on to SAS Forecast Studio from the machine, the **SAS environment** list displays the default SAS environment.
- When you launch SAS Forecast Studio using Java Web Start, the **SAS environment** list includes **(host deployment)**, as well as any available SAS environments. The **(host deployment)** value represents the SAS deployment from which SAS Forecast Studio was downloaded. When you launch SAS Forecast Studio using Java Web Start, the default value in the **SAS environment** list is the last SAS environment that you used. If you have not previously logged on to SAS Forecast Studio, the **SAS environment** list defaults to **(host deployment)**.

#### Configure the Logon Behavior on the Server

Using the Configuration Manager in the SAS Forecast Server Plug-ins for SAS Management Console, you can control the logon behavior for Java Web Start.

You can set the following options:

##### SAS environment URL

specifies the URL for the `sas-environment.xml` file. For more information, see the [“Configure a SAS Environment URL” on page 80](#).

**Preferred SAS environment**

specifies the SAS environment that should be selected by default in the logon dialog box. For more information, see the [“Set the Preferred SAS Environment in the Logon Dialog Box for SAS Forecast Studio”](#) on page 81.

**Lock SAS environment selection**

specifies whether users can select a different SAS environment in the logon dialog box. When you specify both the **Preferred SAS environment** and **Lock SAS environment selection** options, you force users to use a specific SAS environment.

If you change any of these options, you must restart the SAS Forecast Server or use a JMX call to apply the new values.

**Configure the Logon Behavior on the Client**

On the client, you can configure the logon behavior of Java Web Start in either of these ways:

- You can specify the URL for the sas-environment.xml file in the SAS\_ENV\_DEFINITION\_LOCATION environment variable.
- If you run the SAS Deployment Wizard on a client, and the URL for the sas-environment.xml file was specified during the SAS installation in the local sassw.config file, you can use the SASHOME environment variable to specify where the client deployment is installed. When the SASHOME environment variable is defined, the client automatically checks the configuration files of the current deployment for a configured sas-environment.xml file to use.

If you configure the logon behavior by using both ways, the value from the SAS\_ENV\_DEFINITION\_LOCATION environment variable is used.

To create either of these environment variables on a client running Windows XP, perform the following steps. If you are running another version of the Windows operating system, then these steps might be slightly different.

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	Variable Name	Variable Value
SAS_ENV_DEFINITION_LOCATION	SAS_ENV_DEFINITION_LOCATION	The location of the sas-environment.xml file. An example is <code>http://&lt;Web-server-name&gt;/&lt;server-path&gt;/sas-environment.xml</code> .
SASHOME	SASHOME	The location where SAS was installed. An example is <code>C:\Program Files\SASHome</code> .

Click **OK**. The SAS\_ENV\_DEFINITION\_LOCATION and SASHOME variables now appear in the list of system variables.

4. Click **OK**.

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

### Customize the Default URL for Java Web Start

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

To open a specific forecasting environment or project in SAS Forecast Studio, add a parameter to this URL.

- `forecasting.launch.environment` specifies a forecasting environment.
- `forecasting.launch.project` specifies the name of a project.

**TIP** You are not required to specify the project parameter when you specify the environment parameter.

For example, if your URL is `http://localhost:8080/SASForecastServer/main.jnlp?forecasting.launch.environment=Default&forecasting.launch.project=Project1`, then SAS Forecast Studio attempts to connect to the environment named Default. If the connection is successful, SAS Forecast Studio attempts to open the project named Project1.

---

## Start SAS Forecast Studio with Options

### Open a Specific Forecasting Environment or Project in SAS Forecast Studio

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

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

When you run SAS Forecast Studio using `launchFile="C:\Europe\Sales\Products2012.fs"`, SAS Forecast Studio attempts to open the Default environment. If the connection is successful, SAS Forecast Studio attempts to open the Products2012 project.

### 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. Usually, a few distinct time ID values are needed to detect the time interval. However, a data set with many repeated values of the time ID variable

sometimes requires a larger sample of observations to get enough distinct values. Increasing the number of observations might improve time-interval detection when you are using data sets that have many repeated values. However, values of 100,000 observations or larger could cause noticeably slower performance when you are creating a new project.

To configure SAS Forecast Studio to use a specified number of observations:

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\12.1** 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, **JavaArgs\_14=-Dcom.sas.analytics.forecasting.interval\_sample=50000**.

### ***Specify the Sample Size for Validating BY Variable Values and Formats***

When you start SAS Forecast Studio, you can specify a default value for the sample size. The sample is used to check the validity of BY variable values and formats. The default value is applied 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 the sample size, the first *n* observations of the data set are used to answer the following questions:

- 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 of these questions is “yes,” then an appropriate error dialog box is displayed. 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\12.1** 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, **JavaArgs\_14=-Dcom.sas.analytics.forecasting.by\_validation\_sample=75000**.





## Chapter 11

# SAS Forecast Batch Interface

<b>About the SAS Forecast Batch Interface</b>	<b>94</b>
Overview	94
Installation and Availability	94
General Usage	94
Work with the SAS Forecast Server	95
Work with SAS Forecasting for Desktop	96
Summary of Macros	97
<b>Dictionary</b>	<b>99</b>
FSADDEVT Macro	99
FSCLEAR Macro	102
FSCOPY Macro	102
FSCREATE Macro	104
FSDELARC Macro	117
FSDELENV Macro	118
FSDELEVT Macro	119
FSDELPRJ Macro	121
FSEVTREQ Macro	122
FSEXPALL Macro	123
FSEXPOR Macro	125
FSEXPSET Macro	126
FSGETENV Macro	128
FSGETPRJ Macro	129
FSGETURP Macro	130
FSIMPALL Macro	131
FSIMPORT Macro	132
FSLOAD Macro	134
FSLOGIN Macro	135
FSLOGOUT Macro	136
FSLOOP Macro	136
FSMIGALL Macro	138
FSMIGPRJ Macro	139
FSMOVE Macro	140
FSNEWENV Macro	142
FSREGENV Macro	143
FSREGPRJ Macro	144
FSREN Macro	145
FSRUNPRJ Macro	146
FSRUNRPT Macro	148
FSSETDAT Macro	150
FSSETDEF Macro	151
FSSETOWN Macro	152

FSSETPUB Macro .....	153
FSUNREG Macro .....	154
FSUNRENV Macro .....	155
FSUPDATE Macro .....	156
FSVER Macro .....	168

---

## About the SAS Forecast Batch Interface

### Overview

The SAS Forecast Batch Interface is available for the SAS Forecast Server and SAS Forecasting for Desktop. Macros in the SAS Forecast 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 are used to manage environments and projects.

### Installation and Availability

When your site licenses the SAS Forecast Server or SAS Forecasting for Desktop, the SAS Forecast Batch Interface is automatically included in all SAS Foundation installations. The SAS Forecast Batch Interface component is installed in the extension directory **forecastbat**. On Windows, the macros in the SAS Forecast Batch Interface reside in an autocall library in **!SASROOT/forecastbat/sasmacro**. On UNIX, they reside in the **SASFoundation/9.3/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 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();
```

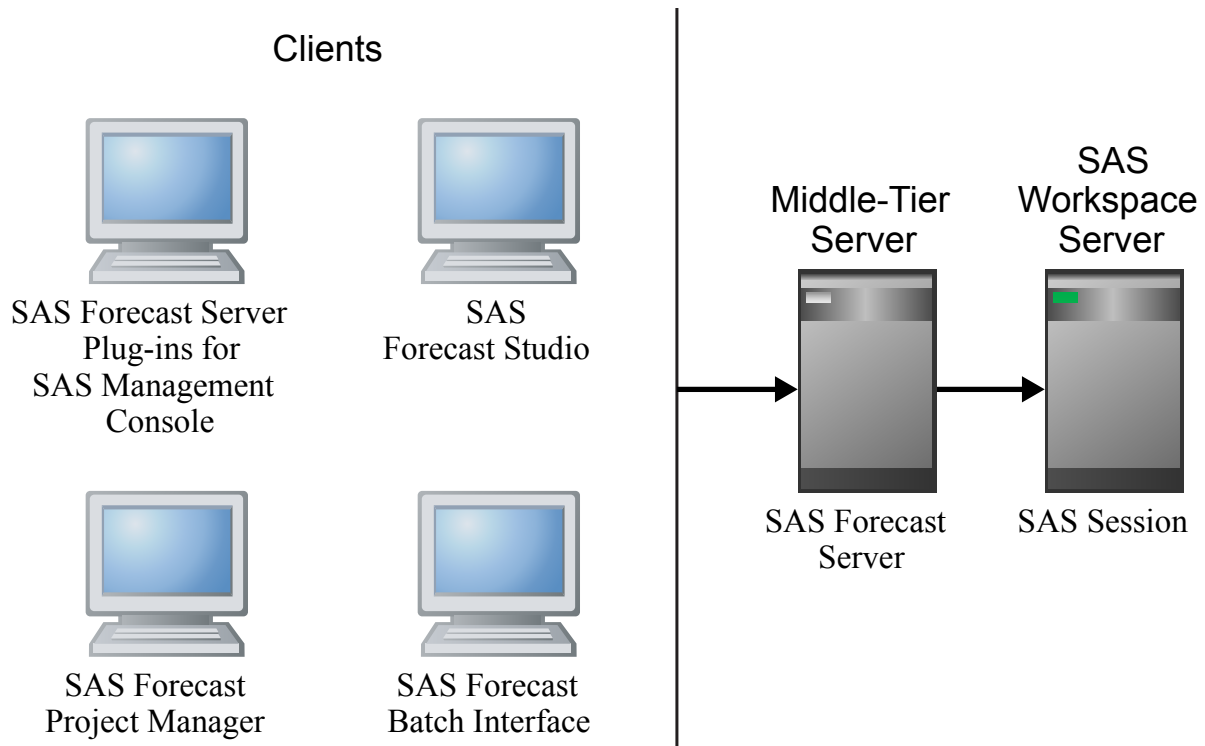
In this example, the arguments for FSLOGIN are intentionally omitted because they depend on the type of deployment. There is more information about these arguments in the subsections that follow.

*Note:* Prior to release 12.1, each SAS Forecast 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.

## Work with the SAS Forecast Server

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



As with the other SAS Forecast Server clients, the SAS Forecast 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 Batch Interface. SAS Forecast 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 menu.)

*Note:* FSLOGIN is the only macro that refers to SAS environments. All other macros in the SAS Forecast Batch Interface refer to forecasting environments.

SAS Forecast Server deployments require credential-based authentication when creating a session. The arguments `USER` and `PASSWORD` are used to provide authentication information. A plain text password in macro code is not recommended. Instead, use `PROC PWENCODE` to create an encrypted password.

For example, to encrypt the password `SASmeta1`, submit the following code in the SAS Program Editor:

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

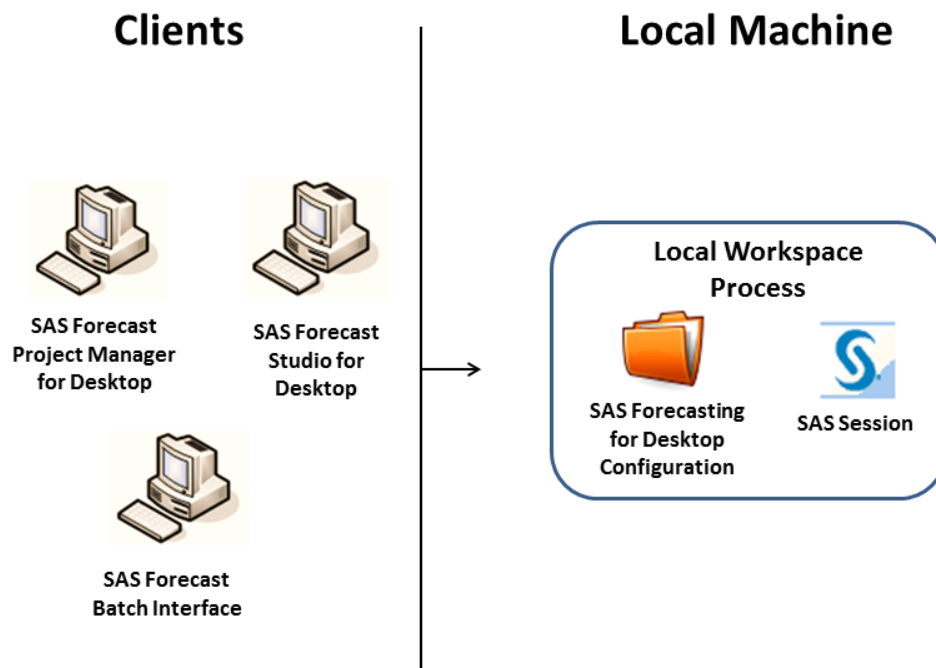
Copy the encrypted password from the SAS log, and paste the password into your macro code. For more information about the PWENCODE procedure, see the *Base SAS 9.3 Procedures Guide*.

For example, suppose your site has only one SAS environment named DEFAULT and the SAS Forecast Server is included. Running the previous example with the SASDEMO identity and an encrypted password {sas002}123 produces the following code:

```
%FSLOGIN(sasEnvironment=default, user=sasdemo, password={sas002}123);
%FSGETENV(print=YES);
%FSLOGOUT();
```

### Work with SAS Forecasting for Desktop

The following figure shows how the SAS Forecast 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 Boolean argument. Here is the complete code for the previous example for SAS Forecasting for Desktop:

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

## Summary of 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 macros in the SAS Forecast Batch Interface:

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 project.
FSCREATE	Creates a new project in batch mode.  <i>Note:</i> If you license SAS Forecast Studio for Desktop, the FSCREATE macro is not available. It is available only if you license the 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 project.
FSDELENV	Deletes an existing forecasting environment.
FSDELEVT	Deletes specified events from the event repository.
FSDELPRJ	Deletes an existing project.
FSEVTREQ	Sets the required attributes for events in the project's event repository.
FSEXPALL	Archives all projects.
FSEXPORT	Archives a single project.
FSEXPSET	Exports the project settings to a local file.
FSGETENV	Shows all of the details about a forecasting environment.
FSGETPRJ	Shows all of the details about a project.
FSGETURP	Lists the names of unregistered projects in a specified forecasting environment.

Macro Name	Description
FSIMPALL	Imports a collection of projects from the project archives.
FSIMPORT	Imports a single project from the project archives.
FSLOAD	Opens an existing SAS Forecast Server project and loads global macro variables that describe the project.
FSLOGIN	Creates a new session for a specified instance of the middle tier.
FSLOGOUT	Closes a session of a specified instance of the middle tier.
FSLOOP	Iterates through a 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 the SAS Forecast Server.
FSMIGPRJ	Migrates an existing SAS Forecast Server project to the current version of the SAS Forecast Server.
FSMOVE	Moves a SAS Forecast Server project to a different forecasting environment.
FSNEWENV	Creates a new SAS Forecast Server forecasting environment.
FSREGENV	Registers an existing directory structure as a forecasting environment. You can also use this macro to register projects in the environment.
FSREGPRJ	Registers a project.
FSREN	Renames a project.
FSRUNPRJ	Runs an existing project at a specified stage.
FSRUNRPT	Runs a 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 a macro argument.
FSSETOWN	Assigns the owner of a project.
FSSETPUB	Changes the public access (sharing) status of a project.
FSUNREG	Unregisters an existing project.

Macro Name	Description
FSUNRENV	Unregisters a forecasting environment.
FSUPDATE	Updates the settings of an existing SAS Forecast Studio project.
FSVER	Returns the version of the SAS Forecast Batch Interface.

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

---

## 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 were previously 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 arguments and are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151.

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 separated by spaces.

EVENTCOMBINATIONRULE = ADD | MAX | MIN | MINNZ | MINMAG | MULT

specifies which combination rule to use when the event has several values that overlap in the same time period. Here are 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 date and time that 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 value 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.

EVENTPERIODICITYOCCURRENCES = *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 the SAS Forecast Server automatically fits a model. The default value is NO. Here are the valid values:

NO	specifies that the event should 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 event should be included in the model as long as the event parameters are significant.
YES	specifies that the event should be included in the model as long as the model can be diagnosed.
UNDEF	specifies that the event should 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 are separated by spaces. Here are the valid values:

- Dates in the ddmmmyyyy format (for example, 01JAN2012).
- Datetime 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 [“FSSETDEF Macro” on page 151](#).

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:

&FSADDEVT = SUCCESS|ERROR

### Example

```
%fsaddevt (projectname=Project1,
           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 project.

---

## Syntax

```
%FSCOPY (SOURCEENVIRONMENT=, DESTINATIONENVIRONMENT=,
SOURCEPROJECT=, DESTINATIONPROJECT=[ options ] ) ;
```

## Details

### Required Arguments

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.

**SOURCEENVIRONMENT** = *environment-name*

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

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

If this option is omitted, the current stored default value is used. The default value is Default. A global default value for SOURCEENVIRONMENT can be specified with the FSSETDEF macro. For more information, see [“FSSETDEF Macro” on page 151](#).

**DESTINATIONENVIRONMENT** = *environment-name*

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

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

If this option is omitted, the current stored default value is used. The default value is Default.

A global default value for DESTINATIONENVIRONMENT can be specified with the FSSETDEF macro.

For more information, see [“FSSETDEF Macro” on page 151](#).

**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 this option is omitted, then the current stored default value is used.

A global default value for the SOURCEPROJECTNAME can be specified with the FSSETDEF macro. For more information, see [“FSSETDEF Macro” on page 151](#).

**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 this option is omitted, then the current stored default value is used.

A global default value for the DESTINATIONPROJECTNAME can be specified with the FSSETDEF macro.

For more information, see [“FSSETDEF Macro” on page 151](#).

### Options

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

**ARCHIVEFOLDER** = *directory-name*

specifies the directory in which 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 directory for the archive folder and the remote archive folder is 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= option or the REMOTEARCHIVEFOLDER= option, both options are set to the Work directory in 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 and destination environments 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 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 in which 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 archived project.

## Results

The FSCOPY global macro variable indicates whether the FSCOPY macro terminated successfully or encountered errors:

&FSCOPY= SUCCESS | ERROR

## Example

```
%fscopy(sourceprojectname=prdat1,
destinationprojectname=prdat2,
sourceEnvironment=Default,
destinationEnvironment=Default,
);
```

---

## FSCREATE Macro

The FSCREATE macro creates a new project in batch mode.

---

## Syntax

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

## Details

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

## 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 were previously 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 option, the current stored default value for PROJECTNAME is used. The value of this default cannot be specified with the FSSETDEF macro.

DATA = *SAS-data-set-name*

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 variables can be specified only if HIERARCHY= NO | FALSE. Multiple variable names are separated by spaces.

## Options

The following options can be used with the FSCREATE macro. Options must follow 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. 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 of all variables, specify **ACCUMULATE=AVG**.
- To use a total accumulation of all variables except for VAR1, which will use an average accumulation, specify **ACCUMULATE=TOTAL AVG (var1)**.
- To use a total accumulation of all variables except VAR1 and VAR2, which will use an average accumulation, and VAR3, which will accumulate the observations based on standard deviation, specify **ACCUMULATE=TOTAL AVG (var1 var2) STD (var3)**.

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

specifies the adjustment options for the dependent variables. If this option is omitted, the current stored 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.

**BY = *variable***

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

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

specifies whether the 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 model. 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 valid values for the selection criterion:

- AADJRSQ—Amemiya's adjusted R-square
- ADJRSQ—adjusted R-square
- 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 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-square
- RWRSQ—random walk R-square
- SBC— Schwarz Bayesian information criterion
- SMAPE—mean absolute symmetric percent error
- SSE—sum of squares error
- 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 ARIMAX option, the UCM option, or both the ARIMAX 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. 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 ARIMAX option, the UCM option, or both the ARIMAX 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 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 residuals 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, equally restricted least squares to compute the combination weights. The results minimize the  $l_2$  norm of the combined forecast residuals subject to the constraint 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 constraint that the weights be nonnegative.

- RANKWGT** assigns weights using the rank of the candidate forecasts at the time the 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 time series across the combination, one or more combination contributors can have a missing value. This option determines the treatment of missing values in the final combination for time series 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 1.
- 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 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 HORMISSPCT 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 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 ARIMAX option, the UCM option, or both the ARIMAX 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 ARIMAX option, the UCM option, or both the ARIMAX 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 SAS Forecast Server forecasting 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 151.

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 value must be the name of a BY variable. By default, no level is specified for fitting exponential smoothing models.

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.

**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 are from 1 to 100.

This option is different from the COMBINEMISSPCT option, which is for the estimation region. The COMBINEMISSPCT option and the HORMISSPCT 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.

**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 the 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 enables project settings to be adjusted individually after the initial import of the project settings from the file.

**INPUT** = *variable*

specifies the input (or independent) variables. Variable names are separated by spaces.

**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 are the 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 a date interval to create a 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 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 ARIMA models. The default value is 2.

MAXPCTOUTLIERS = *n*

specifies the maximum number of outliers to include in ARIMA models specified 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 the SAS Forecast Server uses the smaller value. For example, suppose 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-name*

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 the SAS Forecast Server reconciles the hierarchy. By default, the 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. Variable names are separated by spaces.

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

specifies whether to produce forecasts when the project is created. The default value is YES. The NO value is equivalent to "Save SAS code to produce forecasts later" in SAS Forecast Studio. This means that the SAS code (.sas file in the project folder) for the task (for example, CREATE\_PROJECT\_IMPORT\_DATA.sas) should be submitted when forecasts are needed. If the FSRUN macro is submitted without first submitting this code, it will automatically select the task and submit 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, the 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 specify 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 seasonality 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—Each missing value is set to 0.
- MISSING—Each missing value is set to missing. This is the default value.
- AVG—Each missing value is set to the accumulated average value.
- MIN—Each missing value is set to the accumulated minimum value.
- MED—Each missing value is set to the accumulated median value.

- MAX—Each missing value is set to the accumulated maximum value.
- FIRST—Each missing value is set to the accumulated first nonmissing value.
- LAST—Each missing value is set to the accumulated last nonmissing value.
- PREV—Each missing value is set to the previous accumulated nonmissing value. Missing values at the beginning of the accumulated series remain missing.
- NEXT—Each missing value is set to the next accumulated nonmissing value. 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 in the INTERVAL= option. You can specify the multiplier in the INTERVAL option. For example, if you specify 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 can specify the offset in the INTERVAL option. For example, if you specify INTERVAL=WEEK.2, then the time interval is WEEK, and the offset 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 are the valid values:

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 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 are the valid values:

- |       |                                                           |
|-------|-----------------------------------------------------------|
| 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 the 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. This option is 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.

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 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 are the valid values:

- |       |                                                         |
|-------|---------------------------------------------------------|
| 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 FSCREATE global macro variable indicates whether the FSCREATE macro terminated successfully or encountered 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 can be used across multiple projects. For example, you can use SAS code to explore and use data in your SAS Forecast Server projects. You can also use these macro variables in SAS stored processes.

## Example

```
%fscreate(projectname=pd1,
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,
setmissing=last,
zeromiss=left,
trimmiss=right,
back=1,
start=01Feb1998,
interval=MONTH,
seasonality=12,
idformat=MMYY.,
timemultiplier=1,
timeshift=1,
weekenddays=45,
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 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 value was previously 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 archived project 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 arguments and are separated by commas.

ARCHIVEFOLDER = *archive-folder-name*

specifies the directory containing the archived project to delete. If this option is not specified, the default location is assumed.

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151.

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 archived project. You must specify this option only if you do not specify the ARCHIVEFOLDER= option. The value of this default cannot be specified with the FSSETDEF macro.

### Results

The FSDELARC global macro variable indicates whether the FSDELARC macro terminated successfully or encountered errors:

&FSDELARC = SUCCESS | ERROR

### Example

```
%fsdelarc(projectname=pd1,
archivename=ArchPD1.far,
);
```

---

## FSDELENV Macro

The FSDELENV macro deletes an existing forecasting environment. When you delete an environment, the registration and file system content are deleted.

---

## Syntax

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

## Details

### Required Arguments

The following argument is required with the FSDELENV macro.

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting environment. This option has no default value. The value of this default cannot be specified with the FSSETDEF macro.

### Options

The following options can be used with the FSDELENV macro. Options must follow 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 terminated successfully or encountered errors:

```
&FSDELENV = SUCCESS | ERROR
```

## Example

```
%fsdelenv(enviroment=Default
);
```

---

## FSDELEVT Macro

The FSDELEVT macro deletes specified 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 value was previously stored..

EVENTNAMES = *name1 name2*

specifies the events that you want to delete. Only one event name is required. Event names are separated by spaces. Event names are case sensitive.

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

EVENTINTEGRITYRULE = NONE | CHECK | FORCE

determines how events are deleted when those events are in combination events 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.

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 151](#).

### Results

The FSDELEVT global macro variable indicates whether the FSDELEVT macro terminated successfully or encountered errors:

&FSDELEVT= SUCCESS | ERROR

### Example

```
%fsdelevt(projectname=Project1,
          eventnames=evt2 evt4,
```

```
environment=Default
);
```

---

## FSDELPRJ Macro

The FSDELPRJ macro deletes an existing project. You can also delete any related archived projects in the default archive location.

---

### 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 is required with the FSDELPRJ macro if no default value was previously 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 PROJECTNAME is used. The value of this default cannot be specified with the FSSETDEF macro.

#### Options

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

**DELETEARCHIVES** = YES | TRUE | NO | FALSE | 1 | 0  
 specifies whether the 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 SAS Forecast Server forecasting 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 151](#).

**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 terminated successfully or encountered errors:

&FSDELPRJ = SUCCESS | ERROR

**Example**

```
%fsdelprj (projectname=prdat4,
deleteArchives=YES,
);
```

---

**FSEVTREQ Macro**

The FSEVTREQ macro sets the required attributes for 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 with the FSEVTREQ macro if no default values were previously 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 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 FSEVTREQ macro. Options must follow 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 SAS Forecast Server forecasting 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 151](#).

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 151](#).

### Results

The FSEVTREQ global macro variable indicates whether the FSEVTREQ macro terminated successfully or encountered errors:

&FSEVTREQ = SUCCESS | ERROR

### Example

```
%fsevtreq(projectname=prdata1,
eventname=_ALL_,
eventrequired=undef
);
```

---

## FSEXPALL Macro

The FSEXPALL macro archives all projects. 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 arguments and are separated by commas.

ARCHIVEFOLDER = *directory-name*

specifies the directory in which to save the archives. If this option is not specified, the archives are saved to project subfolders in the default location.

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

exports data sets and catalogs using CPORT. You must specify this option if you unarchive the project on a different platform. The default is NO.

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151.

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-name*

specifies the fully qualified name of a SAS 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 this option is set to 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 terminated successfully or encountered 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 was created.
MODIFIED	specifies the date and time when the project was last modified.
CREATEDBY	specifies the user ID that created the project.
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 while archiving the project.

In addition, archives for all projects are created in the default archive location. The name of the archived project 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 archives a single project.

---

## 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 were previously 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. The file extension is automatically assumed to be .far. Use the ARCHIVEFOLDER= option to specify the directory of the archive file.

**DESCRIPTION** = *text*

specifies a description for the archive. The description is recorded in the Manifest.fs file in the archive. If a comma is in the description string, the description must be enclosed in quotation marks.

### Options

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

ARCHIVEFOLDER = *directory-name*

specifies the directory in which to save the archive. If this option is not specified, the archive is saved in the default location.

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

exports data sets and catalogs using CPORT. You must specify this option if you unarchive the project on a different platform. The default is NO.

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 151](#).

### Results

The FSEXPORT global macro variable indicates whether the FSEXPORT macro terminated successfully or encountered errors:

&FSEXPORT = SUCCESS | ERROR

An archived project with the ARCHIVENAME= filename is created in the default location.

### Example

```
%fsexport( projectname=pd1,
archivename=ArchPD1.far,
description=Project pd1
);
```

---

## 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 is required with the FSEXPSET macro if no default values were previously stored.

**OUTPATH** = *full-path-of-output-file*  
specifies the path and filename of the local file. This file has an .fps extension.

### Options

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

**ENVIRONMENT** = *environment-name*  
specifies the name of the SAS Forecast Server forecasting 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 151](#).

**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 151](#).

### Results

The FSEXPSET global macro variable indicates whether the FSEXPSET macro terminated successfully or encountered errors:

&FSEXPSET = SUCCESS | ERROR

## Example

```
%fsexpset (outpath=C:\ForecastServer\Projects\Settings.fps,
environment=Default,
projectname=Project3,
);
```

---

## FSGETENV Macro

The FSGETENV macro shows all of the details about a forecasting environment.

---

### Syntax

```
%FSGETENV([options]) ;
```

### Details

#### Options

The following options can be used with the FSGETENV macro. Options must follow 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-name*

specifies the fully qualified name of a SAS 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 variable indicates whether the FSGETENV macro terminated successfully or encountered 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 SAS 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 the SAS Forecast Server that the environment uses.

## Example

```
%fsgetenv(out=work.getprojects,
print=YES
);
```

---

## FSGETPRJ Macro

The FSGETPRJ macro shows all of the details about a project.

---

### Syntax

```
%FSGETPRJ([options]);
```

### Details

#### Options

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

**ENVIRONMENT** = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

**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-name*

specifies the fully qualified name of a SAS 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 variable indicates whether the FSGETPRJ macro terminated successfully or encountered 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 the 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 lists the names of any unregistered projects in a specified forecasting 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

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

**ENVIRONMENT** = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

**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-name*

specifies the fully qualified name of a SAS 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 terminated successfully or encountered errors:

&FSGETURP = SUCCESS | ERROR

### Example

```
%fsgeturp(environment=Default,
out=sales.unregistered,
print=YES
);
```

---

## FSIMPALL Macro

The FSIMPALL macro imports a collection of projects from the project archives. This macro should be used only by a user who has 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 were previously stored. The required arguments are separated by commas.

PROJECTDS = *SAS-data-set-name*

specifies the fully qualified name of a SAS data set that contains the list of the SAS Forecast Server projects to import. The format is *library.dataset*. The PROJECTDS= data set is created when you specify the OUT= option in 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 equals 1 if an error was detected during archiving. If the ARCHIVEFAIL variable is included, only the archives where the value of the ARCHIVEFAIL variable is 0 are imported.

ARCHIVEFOLDER = *directory-name*  
specifies the directory that contains the archived projects (for example, **C:\SAS\ForecastStudio\Archives**).

### Options

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

ENVIRONMENT = *environment-name*  
specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 in the archive folder location. If this option is set to NO, all archives are assumed to be stored in the default archive folder location. The default value is NO.

### Results

The FSIMPALL global macro variable indicates whether the FSIMPALL macro terminated successfully or encountered errors:

&FSIIMPALL = 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 more information, see [“FSMIGALL Macro” on page 138](#).

### Example

```
%fsimpall(projectds=work.projects,
archivefolder=\\sourceserver\SAS\ForecastStudio\Archives
);
```

---

## FSIMPORT Macro

The FSIMPORT macro imports a single project from the project archives.

---

### 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 value was previously stored.

ARCHIVEPATH = *directory-name*  
specifies the location of the archived project.

### Options

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

ENVIRONMENT = *environment-name*  
specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 151](#).

### Results

The FSIMPORT global macro variable indicates whether the FSIMPORT macro terminated successfully or encountered errors:

&FSIMPORT = SUCCESS | ERROR

## Example

```
%fsimport (projectname=prdat2,
archivepath=C:\SAS\ForecastStudio\Archives\pd1\ArchPD1.far
);
```

## FSLOAD Macro

The FSLOAD macro opens an existing SAS Forecast Server project and loads global macro variables that describe the project.

### Syntax

```
%FSLOAD([options]) ;
```

### Details

#### Options

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

**ENVIRONMENT** = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

**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 151](#).

#### Results

The FSLOAD global macro variable indicates whether the FSLOAD macro terminated successfully or encountered 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 can be used across multiple projects. For example, you can use SAS code to explore and use data in your SAS Forecast Server projects.

### Example

```
%fsload(projectname=pd1,
environment=default
);
```

---

## FSLOGIN Macro

The FSLOGIN macro creates 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. This allows the middle tier to be used to identify the session.

**Note:** FSLOGIN stores information about the most recent session in a global variable for use by other macros. You can specify the middle tier explicitly by using the MIDTIER= argument.

---

### Syntax

```
%FSLOGIN(DESKTOP=,USER=, PASSWORD=[,options]) ;
```

### Details

#### Required Arguments

The following arguments are required with the FSLOGIN macro if no default values were previously stored. The required arguments are separated by commas.

DESKTOP = YES | NO

specifies whether the specific instance of the middle tier is being run in the SAS Forecast Server or in SAS Forecasting for Desktop. This argument is used only the first time a session is created for that instance of the middle tier. By default, this argument is set to NO, and the new session is created for the SAS Forecast Server.

USER = *user-name*

specifies the user name that you use to log on to SAS Forecast Studio. This argument 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 argument is required when DESKTOP=NO, which is the default setting.

#### Options

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

SASENVIRONMENT = *environment-name*

specifies the symbolic name of the SAS environment. 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. When you specify this option, you must give each middle tier a unique label (name) so that a later macro call can be directed to the correct session. A name should never be reused for a different middle tier during the same SAS session, even after previous SAS sessions are closed.

This option is typically omitted and the default value is used. For the SAS Forecast Server, the default label is fsmain.

**Results**

The FSLOGIN global macro variable indicates whether the FSLOGIN macro terminated successfully or encountered errors:

&FSLOGIN=SUCCESS | ERROR

**Example**

```
%fslogin(user=sasdemo,
password=Password1,
sasEnvironment=default,
desktop=NO
);
```

---

**FSLOGOUT Macro**

The FSLOGOUT macro closes a session of a specific instance of the middle tier.

---

**Syntax**

%FSLOGOUT([options ]);

**Details****Options**

The following option can be used with the FSLOGOUT macro. Options must follow arguments.

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 variable indicates whether the FSLOGOUT macro terminated successfully or encountered errors:

&FSLOGOUT=SUCCESS | ERROR

---

**FSLOOP Macro**

The FSLOOP macro iterates through a data set and invokes a callback macro with the values from each row.

---

**Syntax**

%FSLOOP(DATA=,CALLBACK=,VARLIST=[,options]);

## Details

### Required Arguments

The following arguments are required with the FSLOOP macro if no default values were previously stored. The required arguments are separated by commas.

**DATA** = *data-set-name*

specifies the data set that contains the arguments that you want to use.

**CALLBACK** = *name-of-handler-macro*

specifies the handler macro to invoke for each row of arguments. The macro must implement the expected signature.

**VARLIST** = *var-name1 var-name2 . . . var-name<sub>n</sub>*

specifies the variables in the data set to pass to the calls to the handler macro.

Variable names must be separated by spaces. The handler macro 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 variable names to other names of macro arguments.

### Options

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

**ARGDATA** = *data-set-name*

specifies the data set that contains additional fixed argument values. These arguments are passed to each generated macro call with the arguments from the input data set. The ARGDATA= data set must be formatted for two character columns named Name and Value. By default, no fixed arguments are used.

**VARARGS** = *macro\_varname1 macro\_varname2...macro\_varname<sub>n</sub>*

specifies the variable names for the macro input that corresponds to the variable names in the VARLIST= argument. If you do not specify the VARARGS=

argument, the argument names for the handler macro are assumed to match the variable names in the input data set. For example, if VARLIST=NAME

DESCRIPTION, but the macro expects the arguments to be named X= and Y=, then you should specify VARARGS= X Y for the FSLOOP macro.

### Results

The FSLOOP global macro variable indicates whether the FSLOOP macro terminated successfully or encountered errors:

&FSLOOP=SUCCESS | ERROR

## Example

```
%FSLOGIN(sasenvironment=default,
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 migrates all existing SAS Forecast Server projects to the current version of the SAS Forecast Server.

---

### Syntax

```
%FSMIGALL(OUT= [,options]) ;
```

### Details

#### Options

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

**ENVIRONMENT** = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

**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-name*

specifies the fully qualified name of a SAS 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 variable indicates whether the FSMIGALL macro terminated successfully or encountered 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 was created.
MODIFIED	specifies the date and time when the project was 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 migrates an existing SAS Forecast Server project to the current version of the SAS Forecast Server.

---

### Syntax

```
%FSMIGPRJ([options]) ;
```

## Details

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 151](#).

### Results

The FSMIGPRJ global macro variable indicates whether the FSMIGPRJ macro terminated successfully or encountered errors:

&FSMIGPRJ=SUCCESS | ERROR

## Example

```
%fsmigprj (projectname=pd1,
environment=default
);
```

---

## FSMOVE Macro

The FSMOVE macro moves a project to a new destination.

---

### Syntax

```
%FSMOVE([options]);
```



## Details

### Options

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

ARCHIVEFOLDER = *directory-name*

specifies the directory in which 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. This option is 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 the destination environment. This option is required only if there are multiple environments available to the user. If this option is omitted, the current stored default value is used. The default value is Default. A global default value for DESTINATIONENVIRONMENT can be specified with the FSSETDEF macro. For more information, see [“FSSETDEF Macro” on page 151](#).

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 DESTINATIONPROJECTNAME can be specified with the FSSETDEF macro. For more information, see [“FSSETDEF Macro” on page 151](#).

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 the source environment. This option is required only if there are multiple environments available to the user. If this option is omitted, the current stored default value is used. The default value is Default. A global default value for SOURCEENVIRONMENT can be specified with the FSSETDEF macro. For more information, see [“FSSETDEF Macro” on page 151](#).

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 SOURCEPROJECTNAME can be specified with the FSSETDEF macro. For more information, see [“FSSETDEF Macro” on page 151](#).

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 variable indicates whether the FSMOVE macro terminated successfully or encountered errors:

&FSMOVE=SUCCESS | ERROR

**Example**

```
%fsmove(sourceprojectname=prdat1,
destinationprojectname=prdat2,
remoteArchiveFolder=C:\SAS\ForecastStudio\Archives,
sourceEnvironment=Default,
destinationEnvironment=Default,
);
```

---

**FSNEWENV Macro**

The FSNEWENV macro creates a new SAS Forecast Server forecasting environment.

---

**Syntax**

%FSNEWENV(ENVIRONMENT=,PATH= [,options]) ;

**Details****Required Arguments**

The following arguments are required with the FSNEWENV macro if no default values were previously stored. The required arguments are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the forecasting 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**

The following options can be used with the FSNEWENV macro. Options must follow 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-style absolute path to the Base SAS Folder in the metadata (for example, /MyContent/Reports). Searches for reports start in this folder. This

option is used only when you are creating an environment for a networked middle tier.

WSSERVER = *workspace-server-name*

specifies the name of the logical SAS Workspace Server. The default value is SASAPP - Logical Workspace Server. This argument is ignored by SAS Forecasting for Desktop because it is assumed that the local machine is the host.

### Results

The FSNEWENV global macro variable indicates whether the FSNEWENV macro terminated successfully or encountered errors:

&FSNEWENV=SUCCESS | ERROR

### Example

```
%fsnewenv(environment=Default,
desc=A new environment,
path=C:\FSEnvironment,
reportspath=/MyContent/Reports,
wsserver=SASApp - Logical Workspace Server
);
```

---

## FSREGENV Macro

The FSREGENV macro registers an existing directory structure as a forecasting environment. You can also use this macro to register all projects in the environment.

---

### Syntax

```
%FSREGENV(ENVIRONMENT=, PATH=[,options]);
```

### Details

#### Required Arguments

The following arguments are required with the FSREGENV macro if no default values were previously stored. The required arguments are separated by commas.

ENVIRONMENT = *environment-name*

specifies the name of the forecasting 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 on the specified SAS Workspace Server.

#### Options

The following options can be used with the FSREGENV macro. Options must follow 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 the metadata any existing projects in the environment. The default value is YES.

REPORTSPATH = *UNIX-style-path*

specifies a UNIX-style absolute path to the Base SAS Folder in the metadata (for example, **/MyContent/Reports**). Searches for reports start in this directory.

WSSERVER = *workspace-server-name*

specifies the name of the logical SAS Workspace Server. The default value is SASAPP - Logical Workspace Server. This argument is ignored by SAS Forecasting for Desktop because it is assumed that the local machine is the host.

### Results

The FSREGENV global macro variable indicates whether the FSREGENV macro terminated successfully or encountered 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
);
```

---

## FSREGPRJ Macro

The FSREGPRJ macro registers a project.

---

### 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 were previously 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 PROJECTNAME is used. The value of this default cannot be customized with the FSSETDEF macro.

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 variable indicates whether the FSREGPRJ macro terminated successfully or encountered errors:

&FSREGPRJ=SUCCESS | ERROR

### Example

```
%fsregprj (projectname=ProjectA,
environment=Default
);
```

---

## FSREN Macro

The FSREN macro renames a 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 value was previously stored.

NEWPROJECTNAME = *project-name*

specifies the new name of the SAS Forecast Server project. The name must be a valid SAS name.

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 151](#).

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 variable indicates whether the FSREN macro terminated successfully or encountered errors:

&FSREN=SUCCESS | ERROR

### Example

```
%fsren(projectname=prdat3,
newprojectname=prdat4
);
```

---

## FSRUNPRJ Macro

The FSRUNPRJ macro runs an existing project at a specified 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 were previously 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. Any changes to the input data set are not applied to the project.

### Options

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

ENVIRONMENT = *environment-name*  
specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 [“FSSETDEF Macro” on page 151](#).

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.

**Results**

The FSRUNPRJ global macro variable indicates whether the FSRUNPRJ macro terminated successfully or encountered errors:

&FSRUNPRJ=SUCCESS|WARNING|ERROR

*Note:* A warning message displays 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 can be used across multiple projects. For example, you can use SAS code to explore and use data in your SAS Forecast Server projects.

**Example**

```
%fsrunprj (projectname=pd1,
method=FORECAST,
importdata=NO,
retainchoose=YES
);
```

---

**FSRUNRPT Macro**

The FSRUNRPT macro runs a forecasting report and saves the results to a local directory.

---

**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 were previously stored. The required arguments are separated by commas.

OUTPATH = *directory-path*

specifies the local directory in which to save the output. An example is **C:\temp**.

REPORTPATH = *directory-path*

specifies the path in SAS Folders for the report definition. An example is **/MyContent/Reports/MyReport**.



## Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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. You can specify any valid ODS destination (for example, HTML, RTF, PDF, and so on). The default format is HTML.

ODSSTYLE = *ODS-style*

specifies the ODS style to apply to the output. The default style is Journal.

OUTFILE = *name-of-output-file*

specifies the name of the file that contains the output. By default, the name is assigned by the SAS Forecast 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 [“FSSETDEF Macro” on page 151](#).

PROMPTDATA = *data-set-name*

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. The Name column should contain the prompt name to identify the prompt (rather than the display label that is used for the prompt). To determine the prompt name, see the report definition in SAS Management Console. By default, no prompt values are provided, so if you run a report that requires prompt values, the report will fail.

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.

UNIT = *directory-path*

specifies the path to the focus unit in the project. 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 variable indicates whether the FSRUNRPT macro terminated successfully or encountered errors:

```
&FSRUNRPT=SUCCESS|ERROR
```

## Example

```
%fsrunrpt(reportpath=Samples/Getting Started/Introduction
/Introduction to Stored Processes,
outpath=C:\temp,
environment=Default,
projectname=pdsale4,
odsdest=RTF
);
```

---

## FSSETDAT Macro

The FSSETDAT macro changes the input data source for a project.

---

### 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 value was previously stored.

**DATA** = *input-data-set*

specifies the new input data set. The value must be in the *libname.memname* format.

#### **Options**

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

**ENVIRONMENT** = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

**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 151](#).

### Results

The FSSETDAT global macro variable indicates whether the FSSETDAT macro terminated successfully or encountered errors:

&FSSETDAT= SUCCESS | ERROR

### Example

```
%fssetdat (data=fslib.fsprj,
projectname=Project3,
midtier=server1,
environment=Default
);
```

---

## FSSETDEF Macro

The FSSETDEF macro stores a default value for a macro argument.

---

### Syntax

```
%FSSETDEF(MIDTIER=,KEY=,VALUE=);
```

### Details

#### Required Arguments

The following arguments are required with the FSSETDEF macro if no default values were previously stored. These required arguments are 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 for which the default is being set. Here are the valid values:

ENVIRONMENT	specifies that the value is for an environment variable (for example, ENVIRONMENT, SOURCEENVIRONMENT, and DESTINATIONENVIRONMENT).
PROJECT	specifies that the value is for a project variable (for example, PROJECTNAME, SOURCEPROJECTNAME, and DESTINATIONPROJECTNAME).

MIDTIER = *label*

identifies which middle tier 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 [“MIDTIER = label” on page 135](#).

VALUE = *default-argument-value*

specifies the value to store with the key. This value is used as the default argument value as appropriate by macros that support that configurable default.

### Options

There are no options for the FSSETDEF macro.

### Results

The FSSETDEF global macro variable indicates whether the FSSETDEF macro terminated successfully or encountered errors:

&FSSETDEF=SUCCESS | ERROR | WARNING

### Example

```
%fssetdef(midtier=fsmain,
key=projectname,
value=sales
);
```

---

## FSSETOWN Macro

The FSSETOWN macro assigns the owner of a 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 argument is required with the FSSETOWN macro if no default value was previously stored.

OWNER = *user-name*

specifies the new owner of the project.

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 151](#).

### Results

The FSSETOWN global macro variable indicates whether the FSSETOWN macro terminated successfully or encountered errors:

&FSSETOWN=SUCCESS| ERROR

### Example

```
%fssetown(projectname=Project3,
owner=sastrust,
environment=Default
);
```

---

## FSSETPUB Macro

The FSSETPUB macro changes the public access (sharing) status of a 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 value was previously stored.

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

specifies whether the project should be available to all users.

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 151](#).

### Results

The FSSETPUB global macro variable indicates whether the FSSETPUB macro terminated successfully or encountered errors:

&FSSETPUB=SUCCESS| ERROR

### Example

```
%fssetpub(project=Project3,
ispublic=YES,
environment=Default
);
```

---

## FSUNREG Macro

The FSUNREG macro unregisters an existing project. The project files are not deleted. You can register the project again by using the project files and the FSREGPRJ macro.

---

### 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 value was previously 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 PROJECTNAME is used. The value of this default cannot be customized with the FSSETDEF macro.

### Options

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

ENVIRONMENT = *environment-name*

specifies the name of the SAS Forecast Server forecasting 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 151](#).

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 variable indicates whether the FSUNREG macro terminated successfully or encountered errors:

&FSUNREG=SUCCESS | ERROR

### Example

```
%fsunreg(projectname=prdat2,
environment=Default
);
```

---

## FSUNRENV Macro

The FSUNRENV macro unregisters a forecasting environment. Unregistering an environment removes the registration for the environment and all projects in that environment without deleting the content in the file system. You can use the FSREGENV macro variable to re-register an environment and its projects.

---

### Syntax

```
%FSUNRENV(ENVIRONMENT=[,options]);
```

## Details

### Required Arguments

The following argument is required with the FSUNRENV macro if no default value was previously stored.

ENVIRONMENT = *environment-name*

specifies the name of the forecasting environment. This option has no default value. The value of this default cannot be specified with the FSSETDEF macro.

### Options

The following options can be used with the FSUNRENV macro. Options must follow 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 variable indicates whether the FSUNRENV macro terminated successfully or encountered errors:

&FSUNRENV=SUCCESS | ERROR

## Example

```
%fsunrenv(environment=Default,
midtier=server1
);
```

---

## FSUPDATE Macro

The FSUPDATE macro updates the settings of an existing SAS Forecast Studio project.

**Note:** Most of the arguments in the FSCREATE macro are also in the FSUPDATE macro. 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 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 Server 12.1, **Set accumulation to the value used for aggregation** is checked by default. If the `ACCUMULATE=` or `AGGREGATE=` arguments are used, **Set accumulation to the value used for aggregation** is not selected. This setting persists, so the next time the project is opened in SAS Forecast Server 12.1, **Set accumulation to the value used for aggregation** will still not be 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 of all variables, specify `ACCUMULATE=AVG`.
- To use a total accumulation of all variables except for VAR1, which will use an average accumulation, specify `ACCUMULATE=TOTAL AVG(var1)`.
- To use a total accumulation of all variables except VAR1 and VAR2, which will use an average accumulation, and VAR3, which will accumulate the observations based on standard deviation, specify `ACCUMULATE=TOTAL AVG(var1 var2) STD(var3)`.

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

specifies the adjustment options for the dependent variables. If this option is omitted, the current stored 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 Server 12.1, **Set accumulation to the value used for aggregation** is checked by default. If the ACCUMULATE= or AGGREGATE= arguments are used, **Set accumulation to the value used for aggregation** is not selected. This setting persists, so the next time the project is opened in SAS Forecast Server 12.1, **Set accumulation to the value used for aggregation** will still not be 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 the 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 valid values for the selection criterion:

- AADJRSQ—Amemiya's adjusted R-square
- ADJRSQ—adjusted R-square
- 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 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-square
- RWRSQ —random walk R-square
- SBC—Schwarz Bayesian information criterion
- SMAPE—mean absolute symmetric percent error
- SSE—sum of squares error
- 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 ARIMAX option, the UCM option, or both the ARIMAX 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. 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 ARIMAX option, the UCM option, or both the ARIMAX 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 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 residuals 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, equally restricted least squares to compute the combination weights. The results minimize the $l_2$ norm of the combined forecast residuals subject to the constraint 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 constraint that the weights be nonnegative.
RANKWGT	assigns weights using the rank of the candidate forecasts at the time the 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 time series across the combination, one or more combination contributors can have a missing value. This option determines the treatment of missing values in the final combination for time series 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 1.

**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 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 HORMISSPCT 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 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 ARIMAX option, the UCM option, or both the ARIMAX 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 ARIMAX option, the UCM option, or both the ARIMAX 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 SAS Forecast Server forecasting 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 151](#).

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 value must be the name of a BY variable. By default, no level is specified for fitting exponential smoothing 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 are 1 to 100.

This option is different from the COMBINEMISSPCT option, which is for the estimation region. The COMBINEMISSPCT option and the HORMISSPCT 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.

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. Variable names are separated by spaces.

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 are the 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 a date interval to create a 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 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 ARIMA models. The default value is 2.

MAXPCTOUTLIERS = *n*

specifies the maximum number of outliers to include in ARIMA models specified 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 the SAS Forecast Server uses the smaller value. For example, suppose 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 = *data-set-name*

specifies a model selection list that contains models that you can use. The default value is SASHELP.HPFDFTL.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 [“FSSETDEF Macro” on page 151](#).

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 the SAS Forecast Server reconciles the hierarchy. By default, the 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. Variable names are separated by spaces.

SEASONALITY = *n*

specifies the length of a season. When possible, the 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 specify SEASONALITY=13.

SEASONTTEST = *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 seasonality 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 value is set to 0.
- MISSING—Each missing value is set to missing. This is the default value.
- AVG—Each missing value is set to the accumulated average value.
- MIN—Each missing value is set to the accumulated minimum value.
- MED—Each missing value is set to the accumulated median value.
- MAX—Each missing value is set to the accumulated maximum value.
- FIRST—Each missing value is set to the accumulated first nonmissing value.
- LAST—Each missing value is set to the accumulated last nonmissing value.
- PREV—Each missing value is set to the previous accumulated nonmissing value. Missing values at the beginning of the accumulated series remain missing.
- NEXT—Each missing value is 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 in the INTERVAL= option. You can specify the multiplier in the INTERVAL option. For example, if you specify 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 can specify the offset in the INTERVAL option. For example, if you specify INTERVAL=WEEK.2, then the time interval is WEEK, and the offset 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 are the valid values:

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 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 are the valid values:

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 the 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. This option is 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.

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 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 are the valid values:

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 terminated successfully or encountered 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 can be used across multiple projects. For example, you can use SAS code to explore and use data in your SAS Forecast Server projects.

## Example

```
%fsupdate (projectname=pd1,  
reconciliation=YES,  
publicaccess=NO  
);
```

---

## FSVER Macro

The FSVER macro returns the version of the SAS Forecast Batch Interface.

---

### Syntax

```
%FSVER();
```

### Details

#### ***Required Arguments***

There are no arguments for the FSVER macro.

#### ***Results***

The output from this macro is FSVERSION, which lists the version number.

## Chapter 12

# SAS Forecast Studio Tasks in Other SAS Software

---

<b>About the SAS Forecast Studio Tasks</b> .....	<b>169</b>
<b>Prerequisites for Using the SAS Forecast Studio Tasks</b> .....	<b>169</b>
<b>Grant Access to the SAS Forecast Studio Tasks</b> .....	<b>170</b>

---

## About the SAS Forecast Studio Tasks

The Forecast Studio Create Project task, the Forecast Studio Open Project task, and the Forecast Studio Override Project task enable you to create and work with SAS Forecast Studio projects in SAS Enterprise Guide and in the SAS Add-In for Microsoft Office. 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. SAS Add-In for Microsoft Office 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 for Microsoft Office, you must complete the following tasks.

- Install the SAS Forecast Server 12.1.
- Install SAS Enterprise Guide 5.1 or the SAS Add-In 5.1 for Microsoft Office on each client machine.
- Configure the SAS Forecast Server to use the SAS Add-In for Microsoft Office.

**TIP** To avoid having to reconfigure SAS Forecast Studio, make sure that you enable the use of the SOAP bridge for the SAS Forecast Server during deployment. For more information, see the [“How to Install and Configure the SAS Forecast Server” on page 15](#).

---

## Grant Access to the SAS Forecast Studio Tasks

*Note:* When you initially configured the SAS Forecast Server, you might have selected **Grant access to SAS Forecast Studio tasks** in the SAS Forecast Server: SOAP Configuration step of the SAS Deployment Wizard. This access is required to enable the use of SAS Forecast Studio tasks in SAS Enterprise Guide and in the SAS Add-In for Microsoft Office. If you selected this option, then you can skip the following reconfiguration steps. For more information, see the [“How to Install and Configure the SAS Forecast Server”](#) on page 15.

If you did not grant access to SAS Forecast Studio tasks when you initially configured the SAS Forecast Server, you must reconfigure the SAS Forecast Server Mid-Tier before you can use the SAS Forecast Studio tasks in SAS Enterprise Guide or in the SAS Add-In for Microsoft Office. You need to reconfigure only the machine that is running the SAS Forecast Server Mid-Tier. You do not need to unconfigure the SAS Forecast Server first.

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

1. Stop the Web application server. You will need to reconfigure the Web application server after reconfiguring the SAS Forecast Server.
2. Start the SAS Deployment Wizard from your SAS Software Depot. For example, on a Windows system, double-click the setup.exe file located in your SAS Software Depot folder.
3. In the SAS Deployment Wizard, complete the following 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**. Select the check box for your Web application server.
  - e. In the SAS Forecast Server: SOAP Configuration step, select **Grant access to SAS Forecast Studio tasks**.
  - f. Complete the remaining steps in the SAS Deployment Wizard. Use the same values in these steps that you specified during your initial deployment. Click **Finish**.

## **Part 5**

---

# SAS Forecast Server Utility Macros

### *Chapter 13*

**Using the SAS Forecast Server Utility Macros** ..... 173





## Chapter 13

# Using the SAS Forecast Server Utility Macros

---

<b>About the SAS Forecast Server Utility Macros</b> .....	<b>173</b>
<b>Dictionary</b> .....	<b>173</b>
HPF_HIER_ROLLUP .....	173
HPF_ImportEventsFromExcel .....	175
HPF_PART_EQUAL .....	177
HPF_PART_SPLIT .....	178

---

## About the SAS Forecast Server Utility Macros

Utility macros can be used to manage your SAS Forecast Server projects.

---

## Dictionary

---

### HPF\_HIER\_ROLLUP

The HPF\_HIER\_ROLLUP macro performs hierarchical aggregation of forecasting data sets.

---

#### 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;
%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.

**Note:** SAS/ACCESS Interface to PC Files must be licensed to run this 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

```
%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.

---

### Syntax

```
%HPF_PART_EQUAL(DATASET=,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.

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 will contain 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**

```
%HPF_PART_EQUAL(dataset=sashelp.pricedata,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.

---

**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

```
%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

### *Appendix 1*

**Troubleshooting the SAS Forecast Server** ..... 183



## Appendix 1

# Troubleshooting the SAS Forecast Server

---

<b>Gather Information</b>	<b>184</b>
Overview	184
Operating System Environment and Configuration Information	184
Problem Description	185
Sample Data	185
Log Files	185
<b>Additional Resources</b>	<b>186</b>
The Status Page for the SAS Forecast Server	186
The Status Page for the SAS Forecast Server: SOAP Service	187
JMX MBeans	188
<b>Troubleshoot the SAS Forecast Server</b>	<b>189</b>
SAS License Is Expiring	189
Project Owner Cannot Access Project	189
<b>Troubleshoot the Java Web Start Clients</b>	<b>190</b>
Java Version Missing for Java Web Start	190
JNLP File Is Not Signed	190
Enable the Java Web Start Cache	190
<b>Troubleshoot the Logon Process</b>	<b>191</b>
Cannot Locate a SAS Environment URL during the Logon Process	191
Logon Process Cannot Locate Any Compatible SAS Environments	191
<b>Troubleshoot the SAS Forecast Server Plug-ins for SAS Management Console</b>	<b>192</b>
Cannot Access Application Management Plug-ins in SAS Management Console	192
Cannot Access the SAS Forecast Server Plug-ins for SAS Management Console	192
SAS Forecast Server Is Not Available in the Current Deployment	192
User Is Unrestricted	193
Environment Action Fails	193
Default Environment Is Missing and Java Exceptions When Adding a New Environment	193
<b>Troubleshoot the SAS Forecast Batch Interface</b>	<b>193</b>
CLASSPATH Variable Is Not Set	193
LOG4J System Property Is Not Initialized	194
<b>Performance Tuning for the SAS Forecast Server</b>	<b>194</b>

## Gather Information

### Overview

When you are troubleshooting unexpected application behavior, it is important to isolate the problem and describe it and the context in which it occurs. There are general classes of information that can expedite resolving a technical problem:

- operating system environment 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

Information to Gather	Done
Details of your operating environment	<input type="checkbox"/>
Detailed description of the problem (including what it takes to reproduce the problem)	<input type="checkbox"/>
Sample data that would help reproduce the problem	<input type="checkbox"/>
Log files	<input type="checkbox"/>
Stack traces from any dialog boxes or consoles	<input type="checkbox"/>

### Operating System Environment and Configuration Information

If you request help from SAS Technical Support, then providing the following information about your installation can result in resolving the problem faster:

- 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 the 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 this information only once, unless it has changed from previous reports of this information.

### **Problem Description**

Provide a problem description that includes as much information as possible. Include a description of the general task that you are trying to accomplish, your user ID, your roles and permissions, and what has happened during the SAS session. Provide answers to questions, 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 specific to a locale? If so, which locales are having problems?

### **Sample Data**

If possible, capture the data that caused the problem. SAS Technical Support might request your input data set so that it can better replicate your operating environment.

### **Log Files**

#### **SAS Forecast Studio Log**

To view the log in SAS Forecast Studio, select **Tools** ⇒ **SAS Log**.

To include additional information in the log:

1. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
2. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
3. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
4. Click the **Settings** tab.
5. In the left pane, select **Forecast Server**.
6. In **Forecast Server > Logging**, set these options:
  - Disable filtering. By default, log output produced by internal activities is omitted from the log.
  - Enable the inclusion of JDBC messages. By default, JDBC messages are not included in the log.

Click **Reset**.

#### **SAS Forecast Server Log**

Log files for the SAS Forecast Server are in the form **SASForecastServer\*.log**.

If possible, provide SAS Technical Support with the log for your Web application server.

---

## Additional Resources

### *The Status Page for the SAS Forecast Server*

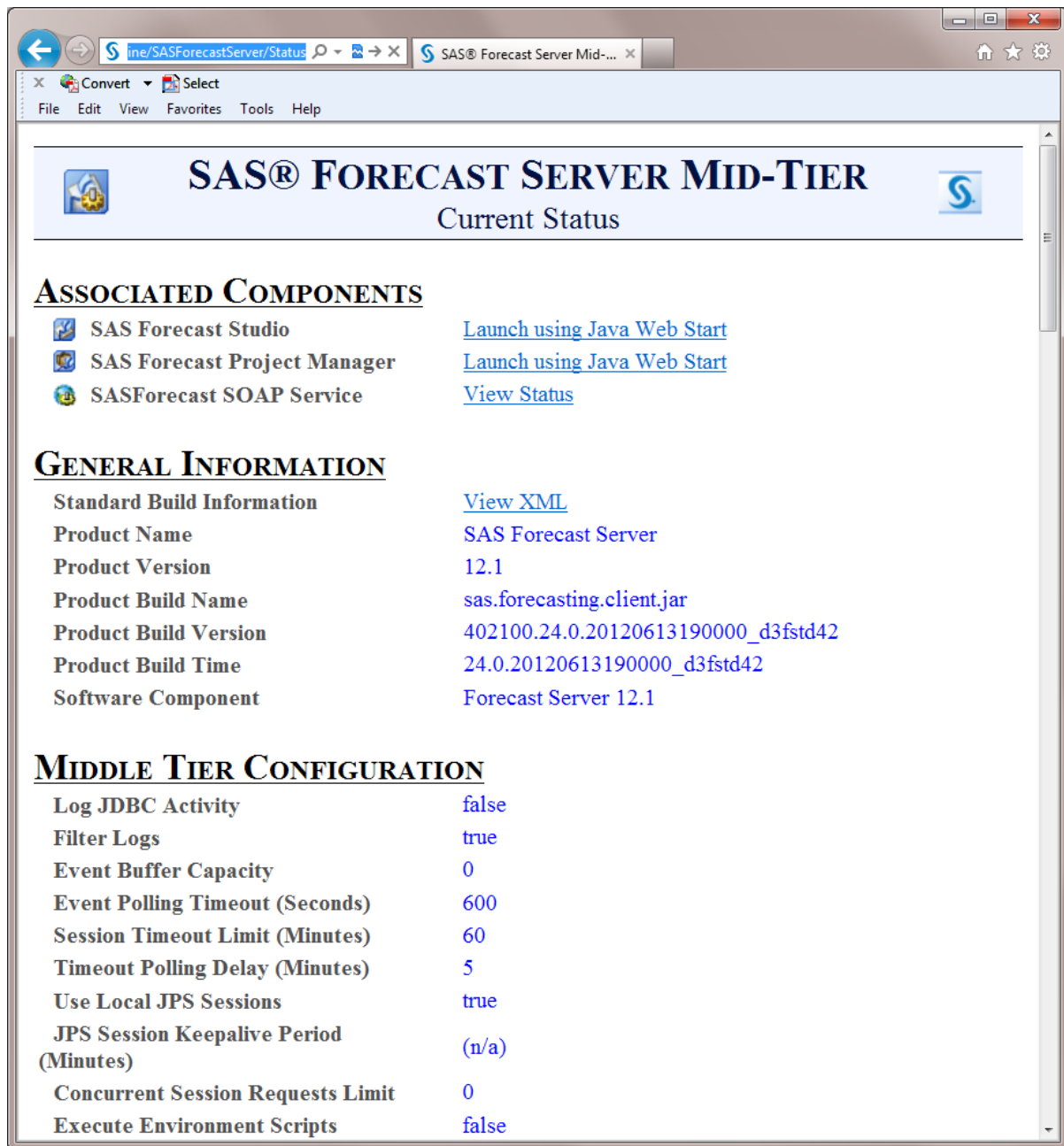
From the SAS Forecast Server Mid-Tier Current Status page, you can see the current settings of the configuration options and you can monitor the activity of the current session. From this page, you can see the release numbers of the SAS Forecast Server clients. This release information will help you determine whether the clients are compatible with the middle tier. The default URL for the status page is the location where the SAS Forecast Server services are deployed. An example of this URL is **`http://<your-server-name>:<port-number>/SASForecastServer/Status`**.

*Note:* You can access this page by using the ViewForecastServerStatus directive.

The SAS Forecast Server Mid-Tier Current Status page provides the following information:




- Links from which you can launch SAS Forecast Studio and SAS Forecast Project Manager as Java Web Start clients.
- A link to the status of the SAS Forecast SOAP Service.
- General information, such as what version of the SAS Forecast Server you are running. This information is used by the SAS Forecast Server clients to determine their compatibility with the middle tier.
- The configuration properties of the middle tier.
- The configuration properties of the Java Web Start client.
- The status of the SAS Workspace Server services, such as the SASApp - Logical Workspace Server, that are being used by the SAS Forecast Server.
- For each of the current sessions, a summary of the actions (or capabilities) that are available to each user. For more information, see [“Manage Roles and Capabilities” on page 23](#).

Here is an example of a status page for the SAS Forecast Server:



The screenshot shows a web browser window with the address bar displaying `line/SASForecastServer/Status`. The page title is **SAS® FORECAST SERVER MID-TIER** with the subtitle **Current Status**. The page is divided into three main sections:

### ASSOCIATED COMPONENTS

 SAS Forecast Studio	<a href="#">Launch using Java Web Start</a>
 SAS Forecast Project Manager	<a href="#">Launch using Java Web Start</a>
 SASForecast SOAP Service	<a href="#">View Status</a>

### GENERAL INFORMATION

Standard Build Information	<a href="#">View XML</a>
Product Name	SAS Forecast Server
Product Version	12.1
Product Build Name	<code>sas.forecasting.client.jar</code>
Product Build Version	<code>402100.24.0.20120613190000_d3fstd42</code>
Product Build Time	<code>24.0.20120613190000_d3fstd42</code>
Software Component	Forecast Server 12.1

### MIDDLE TIER CONFIGURATION

Log JDBC Activity	false
Filter Logs	true
Event Buffer Capacity	0
Event Polling Timeout (Seconds)	600
Session Timeout Limit (Minutes)	60
Timeout Polling Delay (Minutes)	5
Use Local JPS Sessions	true
JPS Session Keepalive Period (Minutes)	(n/a)
Concurrent Session Requests Limit	0
Execute Environment Scripts	false

### ***The Status Page for the SAS Forecast Server: SOAP Service***

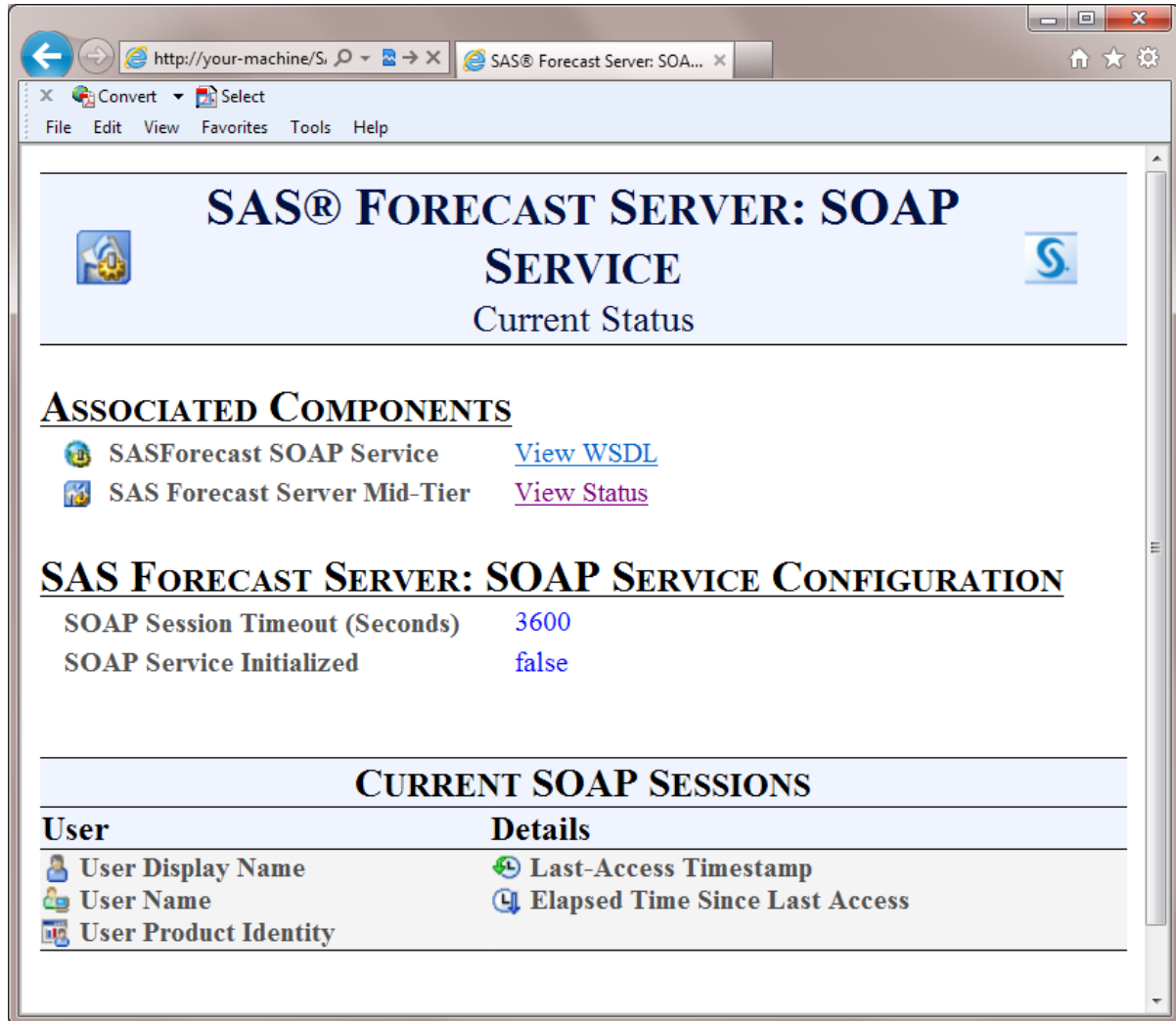
The SOAP service (bridge) is required to run SAS Forecast Studio tasks in the SAS Add-In for Microsoft Office or SAS Enterprise Guide. You can access the status page for this service from the status page for the SAS Forecast Server. For more information, see the “[The Status Page for the SAS Forecast Server](#)” on page 186.

The SAS Forecast Server: SOAP Service Current Status page provides the following information:

- A link to the WSDL for the SOAP service.

- A link to the status of the SAS Forecast Server.
- The configuration properties for the SOAP service.
- Details about each of the current SOAP sessions.

Here is an example of a status page for the SOAP service:



## JMX MBeans

Although the status page for the SAS Forecast Server provides a summary of the run-time state of the middle tier, you cannot modify the state from this page. SAS Forecast Server provides a simple JMX MBean that you can use to modify the state of the SAS Forecast Server. For some modifications, you can use JMX calls to apply changes to the SAS Forecast Server instead of restarting the middle tier. The JMX MBean provides details about the run-time state, but these details are limited to information directly related to actions that are supported by the JMX MBean.

Here are the actions supported by the JMX MBean:

- General Operation
  - `isInitialized()` reports whether the middle tier has completed the initialization that is performed at start-up.



- **reloadConfiguration()** reloads the SAS Forecast Server configuration from the metadata and attempts to apply any changes to it.

*Note:* Configuration information is also cached in the configuration service. The configuration service must be reloaded before the **reloadConfiguration()** call to avoid getting the configuration information from an old cache. The configuration service provides a JMX call that can be used to reload the cached information without restarting the middle tier.

- **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 SAS Workspace Server being retested when it is accessed again. You perform this action when a server has been classified as unusable, but you have corrected the problem.
  - **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:* Only use this action to close sessions that no longer have clients associated with them. For example, you might use this action when a client process must be manually terminated. Forcing a session on the middle tier to close causes any active clients to fail. This action could result in data corruption, depending on what task the active client was performing when the middle-tier session was closed.

**TIP** After executing a JMX call, you can confirm your modification by using the status page for the SAS Forecast Server.

---

## Troubleshoot the SAS Forecast Server

### ***SAS License Is Expiring***

If you install the SAS Forecast Server as an add-on product to an existing SAS 9.3 deployment, you might see error messages about your SAS license expiring. If you are adding the SAS Forecast Server to an existing SAS 9.3 deployment or if you are renewing the software license for the SAS Forecast Server, you must update the SAS installation data (SID) file in the metadata.

To resolve these error messages, see “Updating the SID File in Metadata for SAS Solutions” in the *SAS Intelligence Platform: Installation and Configuration Guide*.

### ***Project Owner Cannot Access Project***

If you cannot access a SAS Forecast Server project that you own, you should 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 a project, access the following in SAS Forecast Studio:

- the Project Properties dialog box
- the **Owner** column of the table in the Projects dialog box

You can determine the owner of the project by using the SAS Forecast Server Plug-ins for SAS Management Console, the SAS Forecast Project Manager, or the FSGETPRJ macro.

To view the token value for the user who is currently logged on, access the following:

- the current sessions table on the SAS Forecast Server Mid-Tier Current Status page
- the **Show my (<identity-token>) projects only** check box in the Projects dialog box in SAS Forecast Studio

*Note:* In some cases, a token mismatch can occur when migrating projects from a previous release. Prior to SAS Forecast Server 4.1, a user ID at the logon prompt was used to generate the token value. Starting with 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 value of the project owner must be updated during the migration process.

---

## Troubleshoot the Java Web Start Clients

### ***Java Version Missing for Java Web Start***

Java Web Start does not work because the required version of the JRE is not installed on the client tier. For SAS 9.3 products, the required Java version is 1.6 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 get a warning that the JNLP file is not signed. For the SAS Forecast Server, a JNLP file is dynamically generated. It cannot be signed. You can ignore this warning.

### ***Enable the Java Web Start 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 the cache is disabled for any reason, complete these steps to enable it:

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.

## Troubleshoot the Logon Process

### **Cannot Locate a SAS Environment URL during the Logon Process**

During the logon process, the SAS Forecast Server clients look for a URL that references a `sas-environment.xml` file. This file contains the list of SAS environments that a user can select during the logon process. The logon process searches the following 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 local deployment of any SAS Forecast Server client, you are prompted for the URL value to store in the `sassw.config` file. If the SAS Forecast Server client cannot locate this URL, you should verify that the information in the `sassw.config` file is correct.

Launching a SAS Forecast Server client 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, the SAS Forecast Server client might generate an error if the configured URL cannot be used. To resolve this error, verify that a valid URL is specified. You can choose to remove the URL because it is optional for the Java Web Start clients.

### **Logon Process Cannot Locate Any Compatible SAS Environments**

During the logon process, the SAS Forecast Server clients run a compatibility test on each SAS environment. This test is primarily used to ensure that each SAS environment that a user can select in the logon dialog box is associated with a compatible deployment of the SAS Forecast Server. Only SAS environments that pass the compatibility test appear in the logon dialog box.

If no SAS environments pass the test, an error message states that no SAS environments could be found. If you get this error 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 compatibility test uses the build version of `sas.forecasting.client`. Each SAS Forecast Server deployment lists its version number on its status page. The corresponding client version can be determined from the client picklist.
- The servers associated with the deployment are running and responsive.

---

## Troubleshoot the SAS Forecast Server Plug-ins for SAS Management Console

### ***Cannot Access Application Management Plug-ins in SAS Management Console***

When you log on to SAS Management Console, the **Forecast Server** plug-in does not display in the **Application Management** folder. To resolve this issue, assign your user ID to the **Management Console: Advanced** role. This role is required so that the SAS Forecast Server Plug-ins for SAS Management Console and other solution-specific plug-ins are available to you.

To assign a user ID to this role:

1. Open SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
2. On the **Plug-ins** tab, select **User Manager**. In the right pane, a list of users, groups, and roles appears.
3. Right-click the **SAS Administrator** user, and select **Properties**. The SAS Administrator Properties dialog box appears.
4. Select the **Groups and Roles** tab.
5. From the list of available groups and roles, select **Management Console: Advanced**. Add it to the **Member of** list.
6. Click **OK**.

### ***Cannot Access the SAS Forecast Server Plug-ins for SAS Management Console***

When you try to access the SAS Forecast Server Plug-ins for SAS Management Console, access is refused. The plug-ins use your logon credentials for SAS Management Console. Therefore, the logon identity must be configured as a product user and must have the appropriate capabilities assigned to it.

*Note:* Do not configure internal accounts (such as sasadm@saspw) for use with the SAS Forecast Server to solve this problem. You should log on to SAS Management Console with a product user account.

### ***SAS Forecast Server Is Not Available in the Current Deployment***

You might get an error message that there is no usable SAS Forecast Server instance in the current deployment. This error is often generated because the compatibility test failed. This is often because hot fixes were only partially applied. You might get this error if you are connecting to a deployment that does not include the SAS Forecast Server.

### ***User Is Unrestricted***

When using the **Forecast Server** plug-in, you might get a warning message that the current user is unrestricted. The unrestricted designation is used for special accounts, such as sasadm@saspw. This unrestricted designation effectively disables the checking of all metadata permissions and capabilities. So, in a sense, an unrestricted user account is assigned all of the metadata permissions and capabilities that are available for the product. Having an unrestricted user is generally not recommended because the unrestricted designation applies only to the SAS Metadata Server and not to other resources, such as the SAS Workspace Server. Unless the user has been configured to use the product, then any actions performed by the unrestricted user fail, and data corruption could occur. Therefore, the unrestricted designation should be used only with special accounts (such as sasadm@saspw) that are not used with the product.

### ***Environment Action Fails***

When you perform maintenance actions on environments (copy, move, delete, and so on), exclusive access to the environment is typically required for the duration of the action. During this time, the environment and all of its projects are inaccessible to users. Similarly, if the environment or one of its projects is already in use when the action is attempted, then the action fails. Therefore, it is recommended that you coordinate with users and establish times during which you can perform environment maintenance.

#### **CAUTION:**

**You should not use the session kill support to force users off the system when maintenance is performed. Forcibly ending an active user session could result in corrupted data.**

### ***Default Environment Is Missing and Java Exceptions When Adding a New Environment***

When you install the SAS Forecast Server as an add-on product to an existing SAS deployment, you have the option of creating a default forecasting environment. This forecasting environment should appear in the SAS Forecast Server Plug-ins for SAS Management Console. If the picklists were not merged when you added the SAS Forecast Server to your deployment, then the default forecasting environment might not appear. In addition, trying to add a new forecasting environment under these conditions results in several Java exceptions.

For information about how to resolve this issue, see SAS Note 44095 at <http://support.sas.com/kb/44/095.html>.

---

## **Troubleshoot the SAS Forecast Batch Interface**

### ***CLASSPATH Variable Is Not Set***

You do not have to set the CLASSPATH variable before using the macros. If you get the following message while executing the macros, you can 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 get the following warning message while executing the macros, you can 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 the 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

*Note:* In general, the other SAS Forecast Server clients do not require tuning because they handle management tasks. However, the HEAP setting applies to all Java Web Start clients (both SAS Forecast Studio and SAS Forecast Project Manager), so any changes to the maximum heap size affect both clients.

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.
- Set the **Maximum heap size** property in SAS Management Console.
  1. Start SAS Management Console. Connect as a SAS administrator (for example, sasadm@saspw).
  2. Click the **Plug-ins** tab.
  3. Expand the **Configuration Manager** and **SAS Application Infrastructure** folders.
  4. Right-click the **Forecast Server 12.1** folder, and select **Properties**. The Forecast Server 12.1 Properties dialog box appears.
  5. Click the **Settings** tab.
  6. In the left pane, click **Virtual Machine Options**.
  7. Specify a value for the maximum heap size.
  8. Click **Reset**.
- 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\12.1\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>.





# 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

---

## A

- architecture
  - diagram [4](#)
  - overview [4](#)
- archives
  - creating [83](#)
  - deleting [117](#), [121](#)
  - importing [131](#), [132](#)
- archiving projects [81](#)
- authorization for libraries [65](#)
- automatic environment libraries [64](#)
- automatic environment library
  - creating [67](#)

## B

- batch interface [5](#), [94](#)
- BY variables
  - validating [91](#)

## C

- capabilities [23](#)
- Classpath variable [193](#)
- components
  - client tier [5](#)
  - middle tier [5](#), [6](#)
  - SAS Forecast Project Manager [5](#)
  - SAS Time Series Studio [6](#)
- configuration properties
  - archiving projects [81](#)
  - SAS environment URL [80](#)
  - session time out [79](#)
  - setting [79](#)
  - URL for Java Web Start [90](#)
- configurations
  - removing [82](#)
- configured environment libraries [64](#)
- configured environment library
  - creating [68](#)
- custom formats [74](#)

- custom time intervals [71](#)
  - examples [72](#), [73](#)

## D

- data sets
  - changing [150](#)

## E

- environments
  - attributes [128](#)
  - configuring [13](#)
  - creating [13](#), [142](#)
  - deleting [118](#)
  - failures [193](#)
  - overview [13](#)
  - permissions [37](#)
  - registering [143](#)
  - unregistering [155](#)
- events
  - adding [99](#)
  - creating [122](#)
  - deleting [119](#)
- external library assignments [63](#)

## F

- formats
  - creating a custom format [74](#)
  - validating [91](#)
- FSADDEVT macro [99](#)
- FSCLEAR macro [102](#)
- FSCOPY macro [102](#)
- FSCREATE macro [104](#)
- FSDELARC macro [117](#)
- FSDELENV macro [118](#)
- FSDELEVT macro [119](#)
- FSDELPRJ macro [121](#)
- FSEVTREQ macro [122](#)
- FSEXPALL macro [123](#)

FSEXPORT macro 125  
 FSEXPSET macro 126  
 FSGETENV macro 128  
 FSGETPRJ macro 129  
 FSGETURP macro 130  
 FSIMPALL macro 131  
 FSIMPORT macro 132  
 FSLOAD macro 134  
 FSLOGIN macro 135  
 FSLOGOUT macro 136  
 FSLOOP macro 136  
 FSMIGALL macro 138  
 FSMIGPRJ macro 139  
 FSMOVE macro 140  
 FSNEWENV macro 142  
 FSREGENV macro 143  
 FSREGPRJ macro 144  
 FSREN macro 145  
 FSRUNPRJ macro 146  
 FSRUNRPT macro 148  
 FSSETDAT macro 150  
 FSSETDEF macro 151  
 FSSETOWN macro 152  
 FSSETPUB macro 153  
 FSUNREG macro 154  
 FSUNRENV macro 155  
 FSUPDATE macro 156  
 FSVER macro 168

**I**

installation  
     how to 15  
 INTERVALDS= system option 71

**J**

Java stack traces 185  
 Java Web Start  
     configuring 80  
     configuring login 88  
     JRE 190  
     JWS cache 190  
     starting 88  
     URL 90  
 JMX access 79  
 JMX MBeans 188  
 JNLP file 190  
 JWS cache 190

**L**

libraries  
     assigned by SAS Forecast Server 64  
     authorization 65  
     automatic environment 64

    configured environment 64  
     configuring 63  
     creating 63  
     enabling support 65  
     external library assignments 63  
     not pre-assigned 67  
     overview 63  
     preassigned 66  
 log4j system property 194  
 logging out 136  
 logging server for Java 194  
 logs  
     SAS Forecast Server 185

**M**

macro variables  
     clearing 102  
 macros 94  
 metadata  
     registering 144  
     unregistering 154, 155  
 middle tier 5

**N**

new sessions  
     creating 135

**O**

operating system accounts  
     UNIX environments 22  
     users 11  
 owners  
     projects 152

**P**

permissions  
     environments 37  
     file system 36  
     metadata 36  
     SAS Forecast Server Metadata User 35  
     unrestricted 193  
     user access 33  
     user groups 22  
     WriteMetadata 36  
 post-installation tasks  
     middle tier 79  
 pre-installation tasks  
     administrative user account  
         recommendations 12  
 project owners 189  
     changing 35  
     overview 35

- project scripts 69
- projects
  - accessing 189
  - archiving 123, 125
  - attributes 129
  - copying 102
  - creating 104
  - deleting 117, 121
  - exporting 123, 125
  - exporting settings 126
  - importing 131, 132
  - loading 134
  - moving 140
  - opening 146
  - owners 152
  - permissions 38
  - registering 130, 138, 143, 144
  - renaming 145
  - running 146
  - sharing 39, 153
  - start-up and shutdown code 69
  - unregistered project names 130
  - unregistering 154
  - updating settings 156
  - upgrading 139
- properties
  - configuration 79

## R

- Remote Services 4
- reports
  - about reports 41
  - about stored processes 41
  - accessing in SAS Forecast Studio 44
  - active series macro variables 59
  - basic project macro variables 52
  - control macro variables 50
  - custom reports 48
  - data hierarchy and settings macro variables 52
  - diagnose macro variables 58
  - dynamic report parameters 48
  - forecast macro variables 58
  - internal report parameters 49
  - keywords 42
  - managing 43
  - metadata macro variables 51
  - model selection macro variables 58
  - normal report parameters 48
  - parameter 48
  - preregistration tasks 46
  - reconciliation macro variables 59
  - registering 46, 47
  - required macro calls 62
  - running 148

- sample reports 45
- special report features 62
- roles 23
  - configuring users 28

## S

- SAS Add-In for Microsoft Office 169
- SAS Deployment Manager 82
- SAS Enterprise Guide 169
- SAS environment
  - Log On dialog box 81
  - URL 80
- SAS environments
  - compatibility 191
  - Log On dialog box 191
- SAS Forecast Batch Interface 94
  - version number 168
- SAS Forecast Project Manager
  - about 5
- SAS Forecast Server
  - architecture 4
  - capabilities 23
  - components 5
  - initial users 33
  - logs 185
  - middle tier 5
  - overview 3
  - performance tuning 194
  - roles 23
  - security layers 36
  - Status page 186
- SAS Forecast Server macros 5
  - Classpath variable 193
- SAS Forecast Server Metadata User
  - overview 34
  - permissions 35
- SAS Forecast Server Plug-in for SAS Management Console 5
- SAS Forecast Studio 5
  - reports 44
  - SAS environment 81
  - session time out 79
  - starting 87
- SAS Forecast Studio tasks
  - accessing 170
  - overview 169
  - prerequisites 169
  - SOAP bridge 170
- SAS Intelligence Platform
  - components 4
  - integration 6
- SAS Management Console
  - configuration properties 79
- SAS Metadata Server 4, 87
- SAS Object Spawner 4

- SAS Remote Services [87](#)
- SAS Services Application [4](#)
- SAS Time Series Studio
  - about [6](#)
- SAS Web Infrastructure Platform [4](#)
- SAS Workspace Server [4](#)
- security
  - file system permissions [36](#)
  - layers [36](#)
  - WriteMetadata [36](#)
- servers
  - required [87](#)
- session time out [79](#)
- sessions
  - closing [136](#)
- shutdown code
  - projects [69](#)
- SOAP bridge [80](#), [170](#)
  - connection time out [80](#)
- start-up code
  - projects [69](#)
- Status page [186](#)

## T

- time intervals
  - custom [71](#)

- detecting [90](#)
- troubleshooting
  - environment access [193](#)
  - gathering information [184](#)
  - JRE [190](#)
  - SAS Forecast Server Plug-ins for SAS
    - Management Console is not
      - available [192](#)
    - system performance [194](#)

## U

- UNIX environments
  - operating system accounts [22](#)
- user accounts
  - product administrator [11](#)
  - users of SAS Forecast Studio [11](#)
- user groups [22](#)
  - permissions [22](#)
- users
  - configuring [28](#)
  - unrestricted [193](#)

## W

- Web application server [87](#)
- WriteMetadata permissions [36](#)