THE POWER TO KNOW.

SAS® Warranty Analysis 4.1
Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2008. *SAS® Warranty Analysis 4.1: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Warranty Analysis 4.1: Administrator's Guide

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

1st electronic book, November 2008

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

Contents

Chapter 1

About SAS Warranty Analysis

Contents

Introduction

SAS Warranty Analysis is an integrated, Web-deployed, reporting and analysis tool. The application provides you with the flexibility and functionality to do the following:

- identify problems
- prioritize warranty-related activities
- determine the root causes for warranty issues

SAS Warranty Analysis uses established, familiar technologies such as the Eclipse Rich Client Platform, HTML, Common Gateway Interface (CGI), and JavaScript to render the application interface. The program logic is written in the Base SAS and SAS macro languages. The application is written to be executed on SAS 9.1.3.

SAS Warranty Analysis is designed specifically for warranty reporting and analysis for any organization that warrants its products. It employs the following:

- an industry-specific data model that was designed for advanced analysis and reporting
- specialized, industry-recognized algorithms to process and display warranty data

The rich client interface enables you to do the following:

- examine warranty data from diverse sources
- drill down on details
- distribute analytical results using report-sharing, exporting, or more traditional print and e-mail processes

SAS Warranty Analysis enables companies to overcome the data availability barrier. It provides superior methods for decision makers to gain advanced understanding of warranty issues and the actions required to resolve those issues.

Components of SAS Warranty Analysis

SAS Warranty Analysis includes four main functional components:

- data warehousing
- analysis and reporting
- advanced analytics
- the clients

These components are seamlessly integrated to offer a true end-to-end solution that accepts data from various systems and sources in differing formats, brings the data together, and applies the appropriate analytical methods to support making advanced business decisions.

SAS Enterprise Guide® enhances the capabilities of SAS Warranty Analysis by providing ad hoc access to the SAS Warranty Analysis data mart and to data sources that are external to SAS Warranty Analysis.

The SAS Warranty Analysis Add-In enables SAS Enterprise Guide 4.1 users to do the following:

- find the location of the data mart
- open a project and create a report of PARMSL.DATASOURCEIDS
- access their own data sets with descriptive data set labels
- access data sets from the complete SAS Warranty Analysis data mart
- select a data set for inclusion in a SAS Enterprise Guide 4.1 project
- perform further analyses on warranty data in a SAS Enterprise Guide project

NOTE: Users are allowed to see only their own data sets.

Purpose of SAS Warranty Analysis

SAS Warranty Analysis provides the following:

- functionality for a wide range of users with varying information needs
- accessibility to a centralized warranty data mart
- availability of warranty-focused analytics and reports that can rapidly discover and disseminate vital performance information

SAS Warranty Analysis provides the following base analyses:

- details table
- exposure
- forecasting
- geographic
- multivariate statistical drivers
- Pareto
- reliability
- statistical drivers
- text analysis
- time of claim
- trend/control
- trend by exposure

SAS Warranty Analysis also provides the following techniques for early detection of emerging issues:

- analytical automated process
- threshold watch list process
- analytical watch list process

Client Applications and Sessions

Overview

SAS Warranty Analysis 4.1 manages a session for each user who logs on. The behavior depends on which client is used for logon. The following sections describe each client application, its purpose, and how it manages sessions.

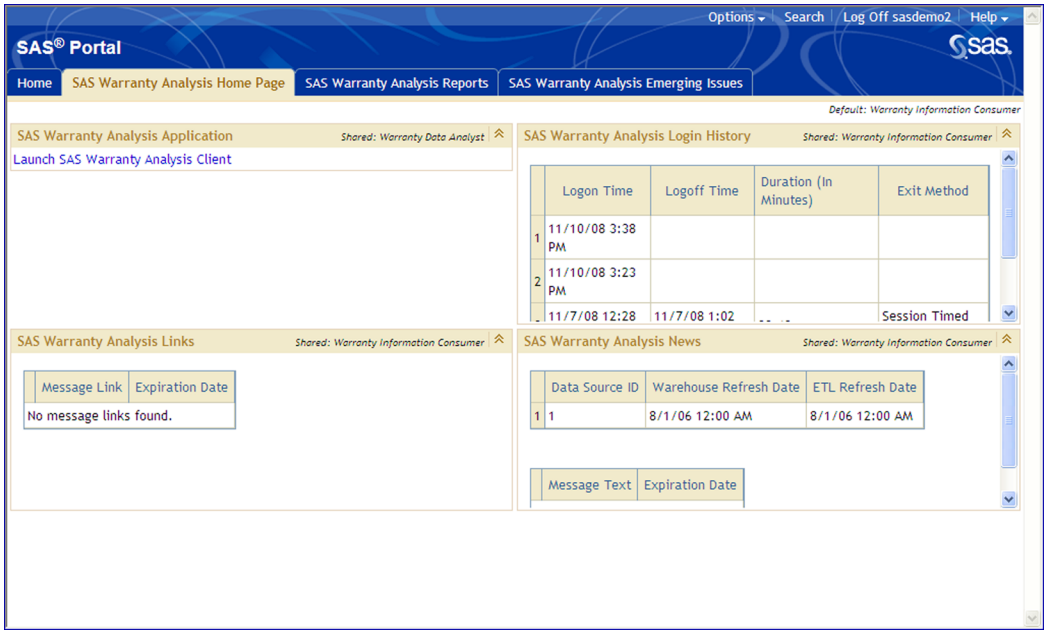
Access to the clients is controlled through security settings that you set up. See Chapter 5, "Security," for more information.

The SAS Warranty Analysis Thin Client

The most frequently used client is the Web-based thin client. See Figure 1.1. This client enables users to browse analytic results that are generated by other users and shared for public consumption. These results include analytic reports and emerging issues output. Your site might also show site-specific information that is created by the SAS Consulting® services personnel who set up your system.

The thin client is not a stand-alone Web application. It is displayed as a set of tabs in the SAS Information Delivery Portal, which comes with the SAS Business Intelligence platform.

Figure 1.1 The Thin Client



Users must log on to the SAS Portal to access the warranty-related tabs and the portlet-based information that they contain. User sessions are programmed to expire after a configurable amount of time.

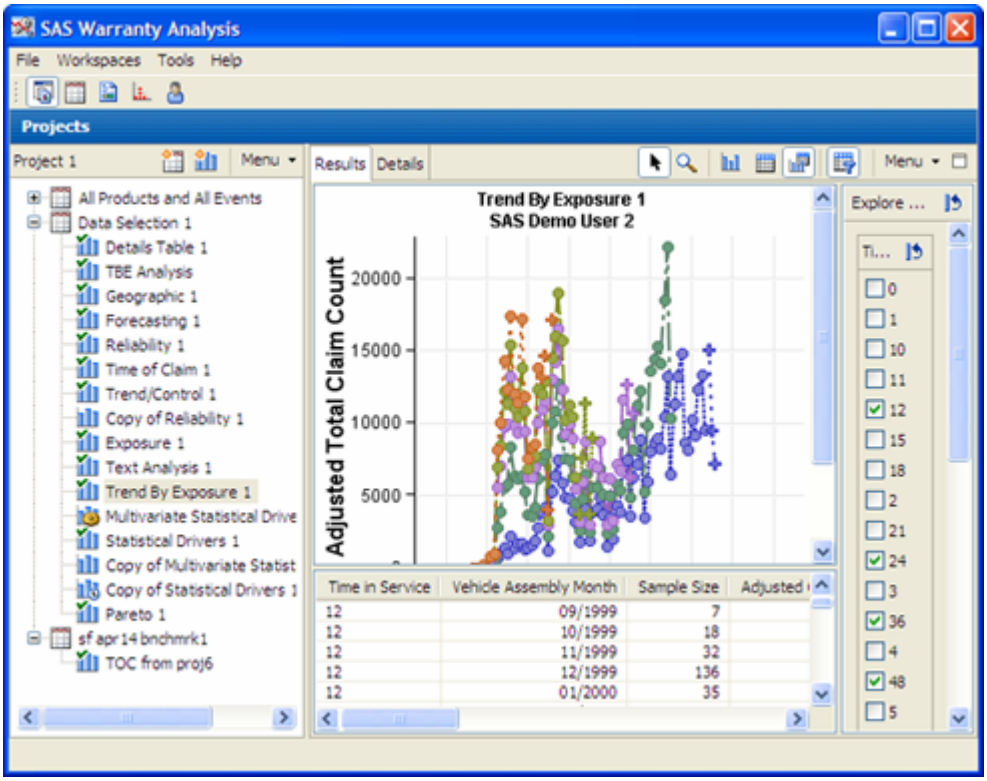
Access to the SAS Portal and access to warranty content in it are governed by separate security settings. A user can be granted access to the SAS Portal but denied access to the warranty-related portion of the portal content. When this happens, the warranty-related tabs are visible, but the related pages display only a message stating that the user is not authorized to view the warranty-related content.

Users can launch the rich client from a link in the SAS Portal. When the rich client is launched in this way, the user ID and server information are passed on to the logon dialog. After entering the password for the initial launch of the rich client, users can select an option to have the logon dialog "remember" the password for subsequent launches.

The SAS Warranty Analysis Rich Client

The rich client application (Figure 1.2) is meant for a smaller audience of analysts, data experts, and administrators. The rich client enables users to create projects, data selections, and analyses. Analyses can be saved as reports to be displayed in the thin client. Emerging issues analyses can also be created and can be displayed as thin client content.

Figure 1.2 The Rich Client



The rich client is installed on a client machine in one of two ways:

- by invoking it from the thin client, which runs a Web-enabled download that uses Java Web Start technology
- by running a client installation program

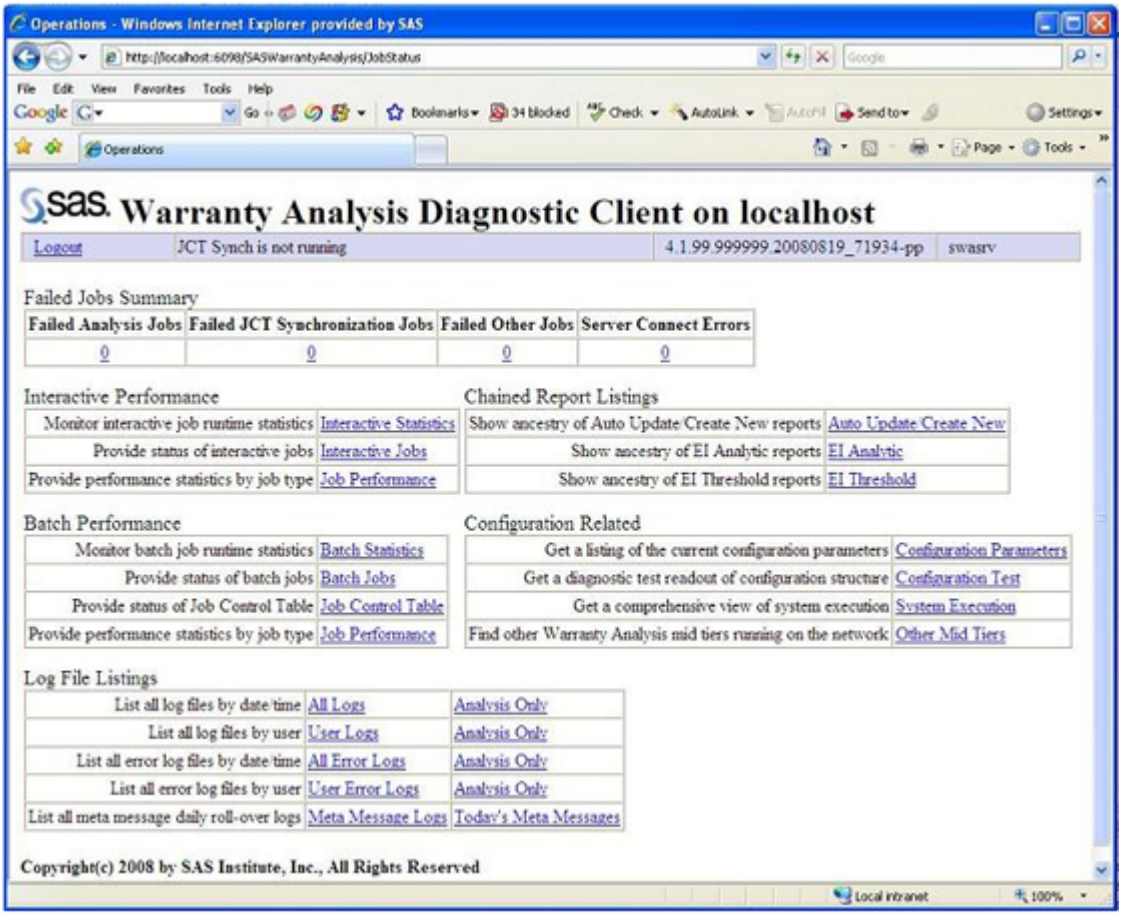
The rich client session never expires unless the middle tier or the SAS tier are shut down and restarted.

NOTE: The rich client starts a session of its own in all cases, even when it is launched from the thin client. Thus, the rich client can run indefinitely, even after the portal session times out.

The SAS Warranty Analysis Diagnostic Client (Experimental)

The diagnostic client application (Figure 1.3) is an experimental program that does not create or show any business information. It presents various views that reflect how SAS Warranty Analysis is running, enables you to retrieve log files more easily, and performs other useful operations.

Figure 1.3 The Diagnostic Client



The diagnostic client starts its own session that expires when the thin client times out. You must be a full administrator to log on to this client.

See Chapter 7, "Using the Diagnostic Client (Experimental)," for more information.

Chapter 2

System Architecture

Contents

Introduction

SAS Warranty Analysis uses a multi-tiered architectural style that incorporates these tiers:

- the client tier
- the middle tier
- the server tier
- the SAS tier
- the data tier

The Client Tier

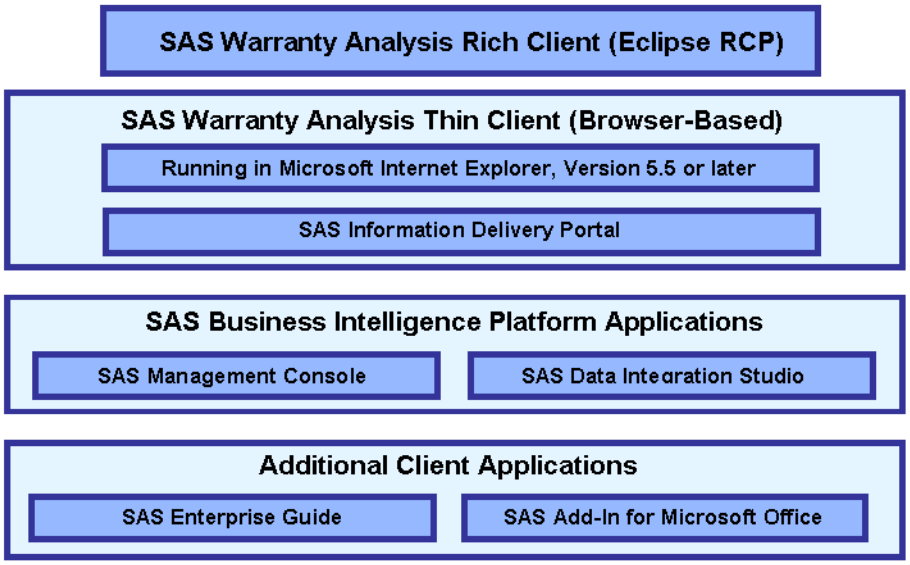
The SAS Warranty Analysis client tier (Figure 2.1) supports both a thin client and a rich client. The SAS Warranty Analysis rich client was developed by using the Eclipse Rich Client Platform (RCP) and Java Swing frameworks. The thin client is built on top of the SAS Information Delivery Portal. The thin client provides read-only access to public reports and emerging issues alerts that have been authored with the SAS Warranty Analysis rich client. The thin client is the primary interface through which users view their warranty data. Which functionality is available to each

user is controlled by that user's role in your organization. See Chapter 5, "Security," for additional discussion of roles.

Java Web Start is the primary vehicle for distributing the SAS Warranty Analysis rich client. Users select a link within the SAS Warranty Analysis thin client to download the rich client from the middle tier. Java Web Start also updates the rich client when new releases are available. If the machine on which the rich client is being installed does not have the appropriate Java Runtime Environment (JRE), then Java Web Start directs the user to the JRE download site. The SAS Warranty Analysis rich client runs in an "all-permissions trusted" environment. (See section 5.6, "Trusted Environments," in the *Java Network Launching Protocol and API Specification (JSR-56) Version 1.0.1* for additional information.)

The SAS Warranty Analysis rich client communicates with the middle tier by using Java Remote Method Invocation.

Figure 2.1 SAS Warranty Analysis Client Tier

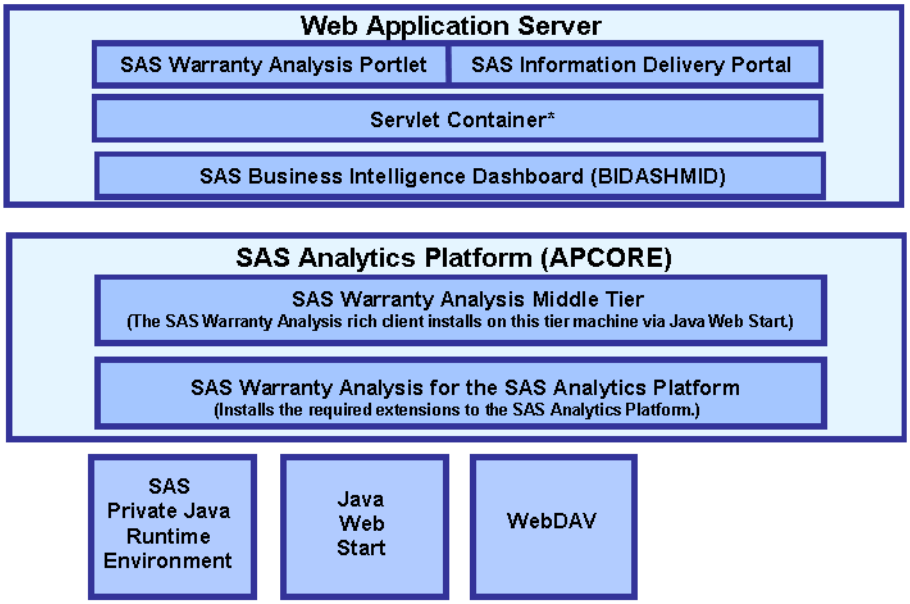


The Middle Tier

The SAS Warranty Analysis middle tier (Figure 2.2) runs within the SAS Analytics Platform, which provides access to SAS Business Intelligence technologies. The middle tier fulfills client requests by applying business rules and controlling access to shared resources (for example, SAS workspace server connections).

The SAS Warranty Analysis diagnostic client is served from a servlet container that is embedded in the SAS Analytics Platform.

Figure 2.2 SAS Warranty Analysis Middle Tier



* Apache Tomcat, IBM WebSphere Application Server, or BEA WebLogic

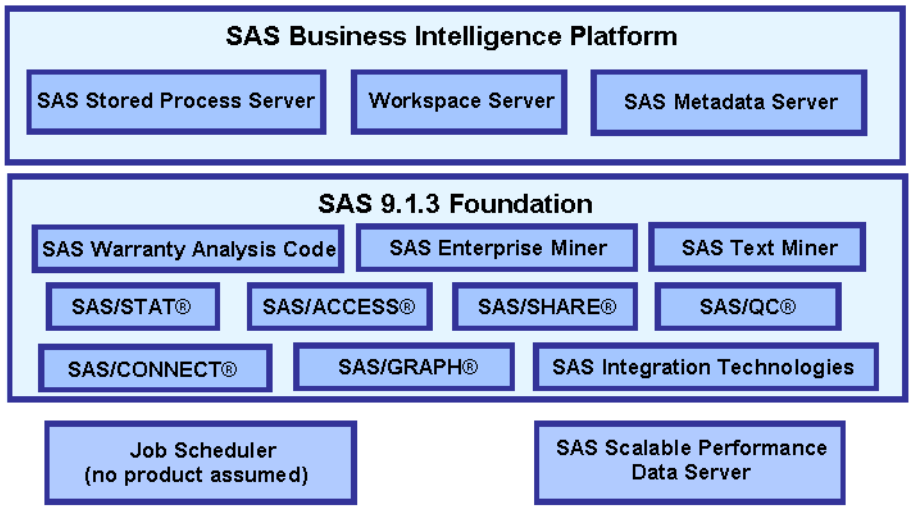
The Server Tier

The SAS Warranty Analysis server tier (Figure 2.3) uses the SAS Metadata Server, SAS Stored Processes Server, SAS workspace server, and the SAS information service to fulfill user requests to analyze warranty data. Most of the work is performed by the SAS workspace server. All access to the data tier from the middle tier is issued through the SAS workspace server, and SAS Analytics are also run from the SAS workspace server. The SAS Stored Process Server runs short, utility-type programs on behalf of the middle tier.

The SAS Management Console and the SAS Metadata Server are used to administer SAS Warranty Analysis users, groups, and roles. The SAS Metadata Server handles the following:

- user authentication
- access to the rich client
- access authorization to SAS Warranty Analysis objects, such as
  - data selections
  - projects
  - reports
- access to SAS services, such as the SAS workspace server

Figure 2.3 SAS Warranty Analysis Server Tier



SAS Tier

All actual analytics, and much of the data handling and data preparation, are handled within SAS. Small utility programs that accomplish tasks such as populating an analysis dialog are run as stored processes. Larger processes, such as running an actual analytic, use the SAS workspace server.

SAS also handles all data sets, which belong to the warranty data tier (see "Warranty Data Tier" on page 12).

Warranty Data Tier

SAS Warranty Analysis creates and manages many types of objects (for example, data selections, projects, and analyses). These objects are maintained in SAS data sets and catalogs that are stored in several different utility libraries known as USERDL, USERRL, EIOUT, and USERFDL.

The USERDL library contains information that defines users' data selections, projects, analyses, and reports. This information is used to display the objects in the user interface and to construct the SQL code that subsets the data mart and creates filtered data sets.

The USERRL library contains information that defines the results of analyses (reports). These data are used to display each report in the user interface, to maintain summarized data, and to track the filtered data that was used to create a report.

The EIOUT library contains information about individual batch runs of the emerging issues processes (including alerts, analytic watch list analyses, and threshold watch list analyses).

Unlike the previously mentioned libraries, which are shared by all users, there is a separate USERFDL library for each user. Each user's USERFDL library stores the filtered data that is created by that user.

The location of the USERFDL library for each user is defined in the PARMSL.USERATTRIBUTES data set. The contents of the library include the query SQL, the claims data set, the products data set, and the labor codes and replaced parts data sets (if they exist) for each filtered data subset that is created.

Another library, PARMSL, stores the metadata that tailors the application to a specific site. These data sets are configured by SAS Consulting services during the installation process.

In addition to the data sets and libraries, the SAS Metadata Server contains entries for each project, data selection, and report. Security settings can be applied to these entries to control access to these items based on user permissions.

Warranty data are stored in the SAS Warranty Analysis data mart. The data mart resides in a library that is usually referred to as SWADM. The name and location of the data mart are specified in the PARMSL.DATASOURCEIDS data set. This library represents the warranty data of each individual customer, and it is documented in the information that is supplied by SAS Consulting services during the installation of the software.

Batch Processing

SAS Warranty Analysis enables users to run ad hoc analyses, to produce reports, and to run emerging issues analyses either interactively or as batch processes that are scheduled to run at a specified time. To permit greater flexibility, scheduling is not handled by SAS Warranty Analysis. Instead, users specify when the work will run, and a job scheduler (such as Platform Computing's Job Scheduler/LSF product) triggers processing via a custom program that updates the SAS Warranty Analysis job control table.

Your SAS Consulting services representative provides details of the scheduling setup for your site.

See Chapter 9, "Working with Batch Processes," for additional information about batch processing.

Chapter 3

Finding Scripts, Files, and Other System Components

Contents

Overview

The physical locations that are described in this chapter are the default locations that are created during the standard installation and configuration processes. The SAS Consulting services staff working at your site are likely to change some or all of these locations when they customize your setup to meet your site's requirements. Your SAS Consulting services representatives provides documentation of the locations for your site.

Finding Installation and Configuration Files

This administrator's guide indicates where you can find key resources by referring to the following two standard file system locations where SAS places the software:

- The SAS installation directory (written as *<SASInstall>*): refers to the directory path where the SAS installation process initially places the software bundle resources prior to any site-specific configuration.
- The SAS configuration directory (written as *<SASConfig>*): refers to the directory path where the SAS configuration process writes the finalized software image.

Different files are written to these locations, depending on the server tier. Also, the SAS Consulting services representative who sets up your system might change these locations. Your SAS Consulting services representative provides documentation of the exact paths to these locations for your site.

In general, you can find these locations in the following locations in a Windows environment:

- *<SASInstall>* defaults to *C:\Program Files\SAS*
- *<SASConfig>* defaults to *C:\SAS\<ImageName>* where *<ImageName>* represents the site-specific name for your software bundle as defined in the SAS plan file that is part of your order.

For UNIX, these locations can vary more widely, and there is no standard default. Please check with your SAS Consulting services representative to determine the file locations for your site.

Finding SAS Server Tier Components

SAS Metadata Server Start Scripts

See the section "Metadata Server Start Scripts " on page 42 in Chapter 6, "Starting and Stopping the System," for details.

Warranty Object Spawner Scripts

See the section "Warranty Object Spawner Scripts " on page 42 in Chapter 6, "Starting and Stopping the System," for details.

Warranty Share Server Scripts

See the section "Warranty Share Server Scripts " on page 42 in Chapter 6, "Starting and Stopping the System," for details.

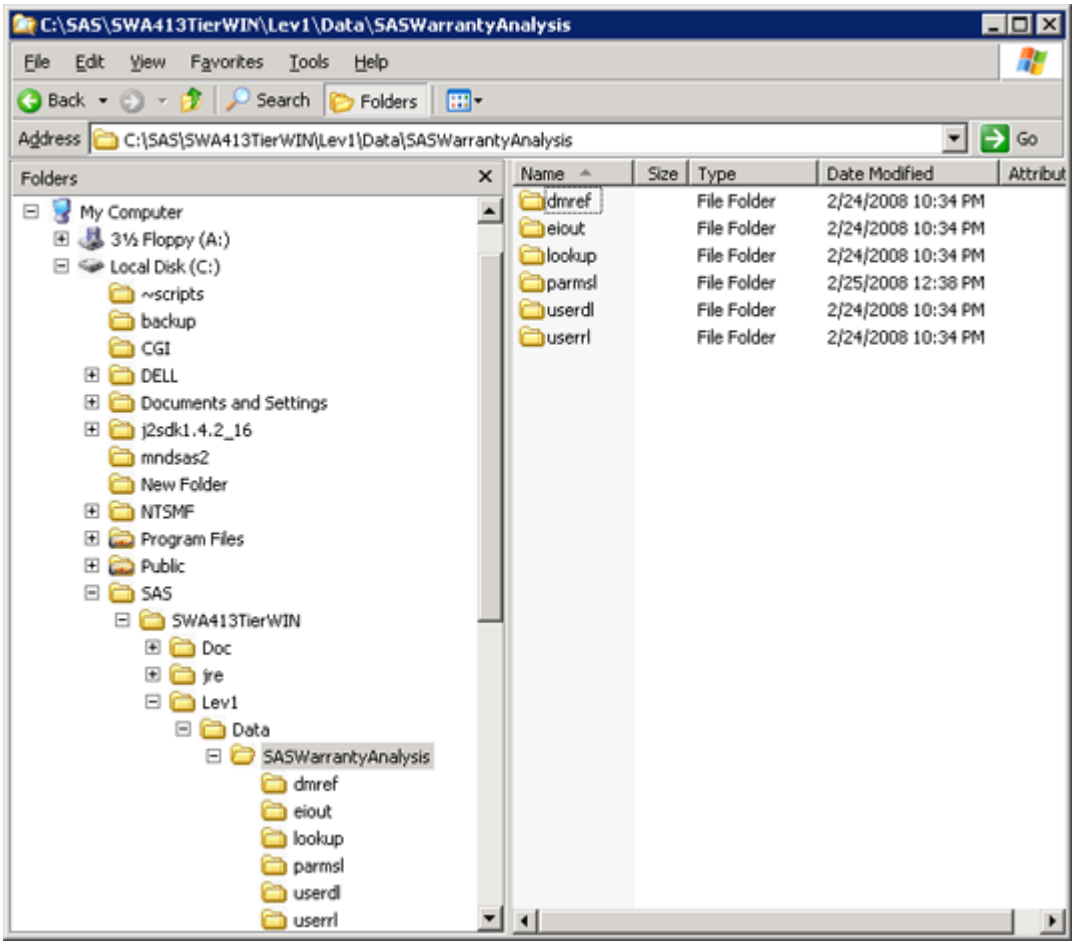
Application Server Scripts

See the section "Application Server Scripts" on page 43 in Chapter 6, "Starting and Stopping the System," for details.

Data Library Directories

By default, the SAS Warranty Analysis libraries are found in the following subdirectories in the *<SASConfig>\Lev1\Data\SASWarrantyAnalysis* folder in a Windows environment: *dmref*, *eiout*, *lookup*, *parmsl*, *userdl*, and *userrl*. (See Figure 3.1.)

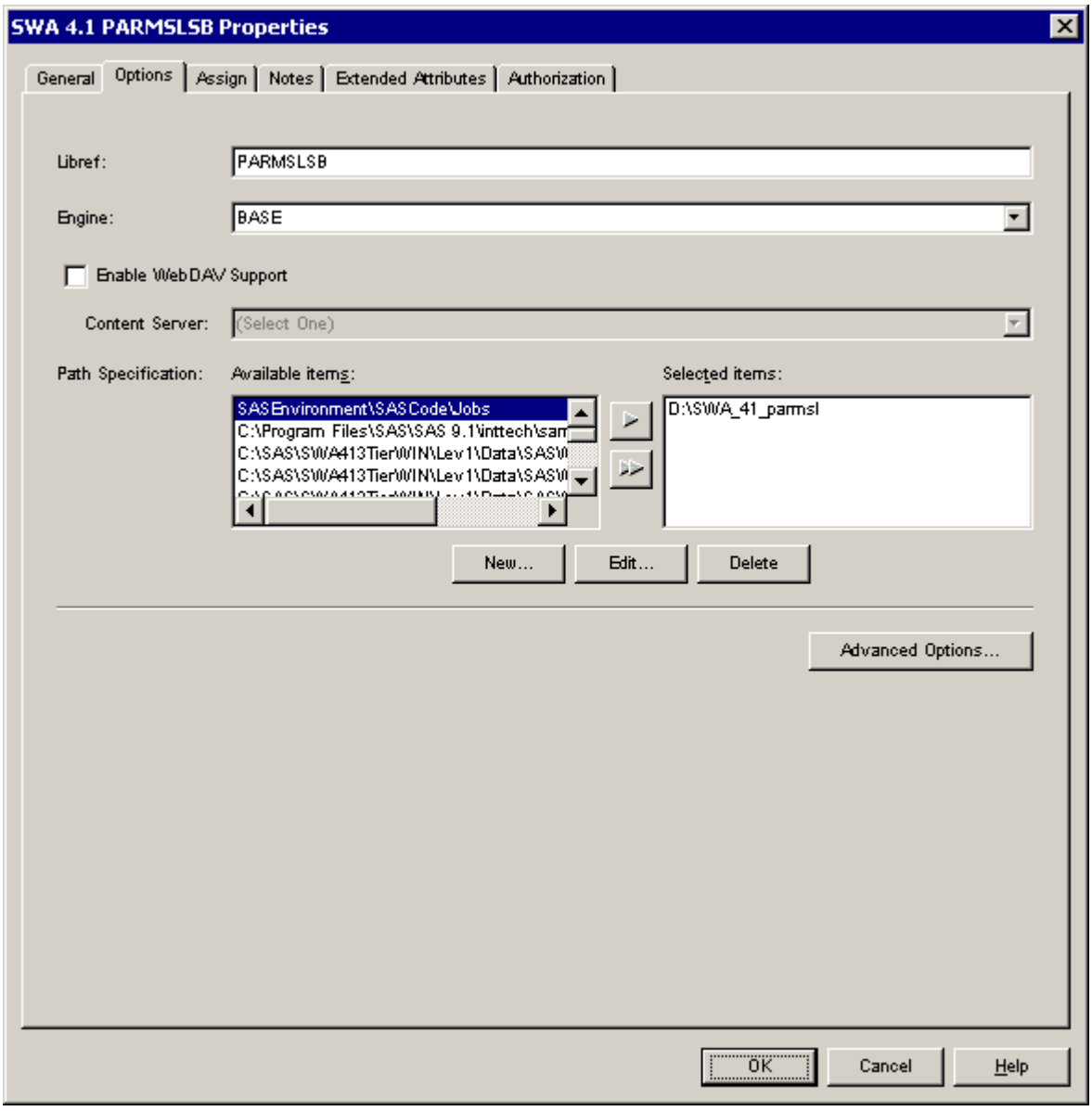
Figure 3.1 Viewing the Data Libraries in Windows Explorer



The SAS Consulting services personnel who set up your system might move some of these libraries (especially the data mart and user libraries) to site-specific locations. You can view the locations of the data library directories for PARMSL, USERDL, USERRL, and EIOUT in the SAS Management Console under **Data Library Manager ►SAS Data Libraries**. (See Figure 3.2.)

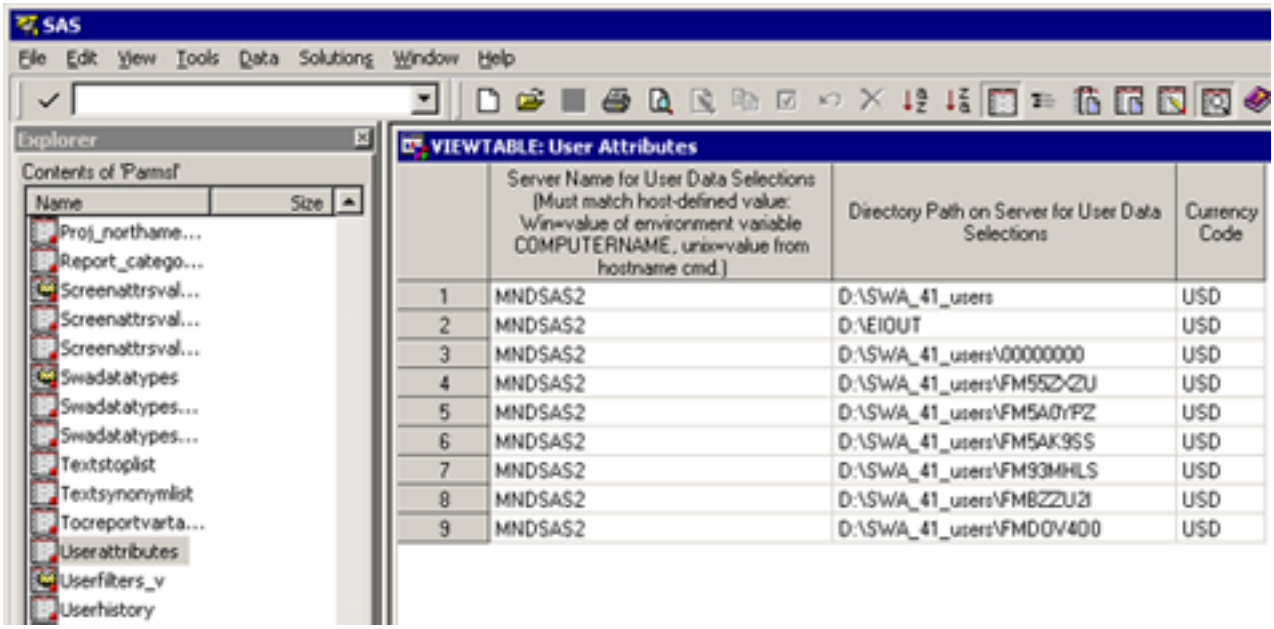
The entries **SWA 4.1 PARMSLSB**, **SWA 4.1 USERDLSB**, **SWA 4.1 USERRLSB**, and **SWA 4.1 EIOUTSB** contain the path specifications for each of these libraries.

Figure 3.2 Path Specification for PARMSLB



The directories for all USERFDL libraries, which are used to store the filtered data sets for each user, are defined in the SERVERPATH column of the PARMSL.USERATTRIBUTES data set. (See Figure 3.3.)

Figure 3.3 Directories for USERFDL Libraries



Finding Middle Tier Components

Diagnostic Client URL

By default, the diagnostic client runs off of port 6098 on the middle-tier machine. The URL to invoke it is as follows:

```
http://hostname:6098/SASWarrantyAnalysis/JobStatus
```

Analytics Platform Scripts

See the section "Analytics Platform Scripts" on page 43 in Chapter 6, "Starting and Stopping the System," for details.

Warranty app.config File

The SAS Warranty Analysis extensions for the Analytics Platform are configured largely by the *app.config* property file. The *app.config* file contains a set of named properties with associated values (in name/value pairs) similar to what is shown in the following example. This example shows just

a small part of what can be in the *app.config* file for the parameters that are used to connect to the SAS Metadata Server.

```
#Warranty Analysis application-level configuration values
metaserver.hostname=mndmyserver2.na.myserver.com
#metaserver.hostname=mndmyserverv.unx.myserver.com
metaserver.hostport=8561
metaserver.trusteduserid=defaulttrusteduserid
metaserver.trusteduserpw=defaulttrusteduserpw
metaserver.deploymentname=Analytics Platform – Foundation Services
metaserver.authdomainname=DefaultAuth
metaserver.repositoryname=Foundation
```

You can find the *app.config* file in a directory such as the following in a Windows environment:

<SASInstall>\SAS\SASAPCore\apps\SASWarrantyAnalysis\app.config

You can find the *app.config* file in a directory such as the following for UNIX:

<SASInstall>/SAS/SASAPCore/apps/SASWarrantyAnalysis/app.config

CAUTION: The *app.config* file contains system values. It should be altered only by someone with a comprehensive understanding of the SAS Warranty Analysis system. Your SAS Consulting services contact provides additional instructions for editing any sections of the *app.config* file that need to be altered during routine administration.

Chapter 4

Working with Domain Objects

Contents

Overview of Domain Objects

The domain objects that SAS Warranty Analysis creates are the "business objects" that store users' choices and other information that is related to business concepts—reports, for example. The SAS Warranty Analysis domain objects record what a rich client user does while working through the various analytical processes that are available in the system.

SAS Warranty Analysis manages most of these objects automatically throughout their life cycle; however, there are a few minor administrative tasks that you can perform manually. These administrative tasks are limited to special cases. With regard to domain objects, the most important roles for administrators include ensuring that the system runs smoothly and ensuring that sufficient disk space is always available.

NOTE: The SAS Warranty Analysis generates a large amount of data, most of which belongs to specific users. To ensure that the growth of users' data does not impede SAS Warranty Analysis

processes, you can move users' data onto a separate volume. You can do this in advance, or you can wait and move users as the need arises and more disk space becomes available. Instructions for moving users' data are provided in the section "Move Users to Different Volumes" on page 28.

The following sections explain how analytic definitions and results are stored and organized on your servers as domain objects and describe related administrative tasks.

Domain Object Types and Characteristics

This chapter covers the following types of domain objects:

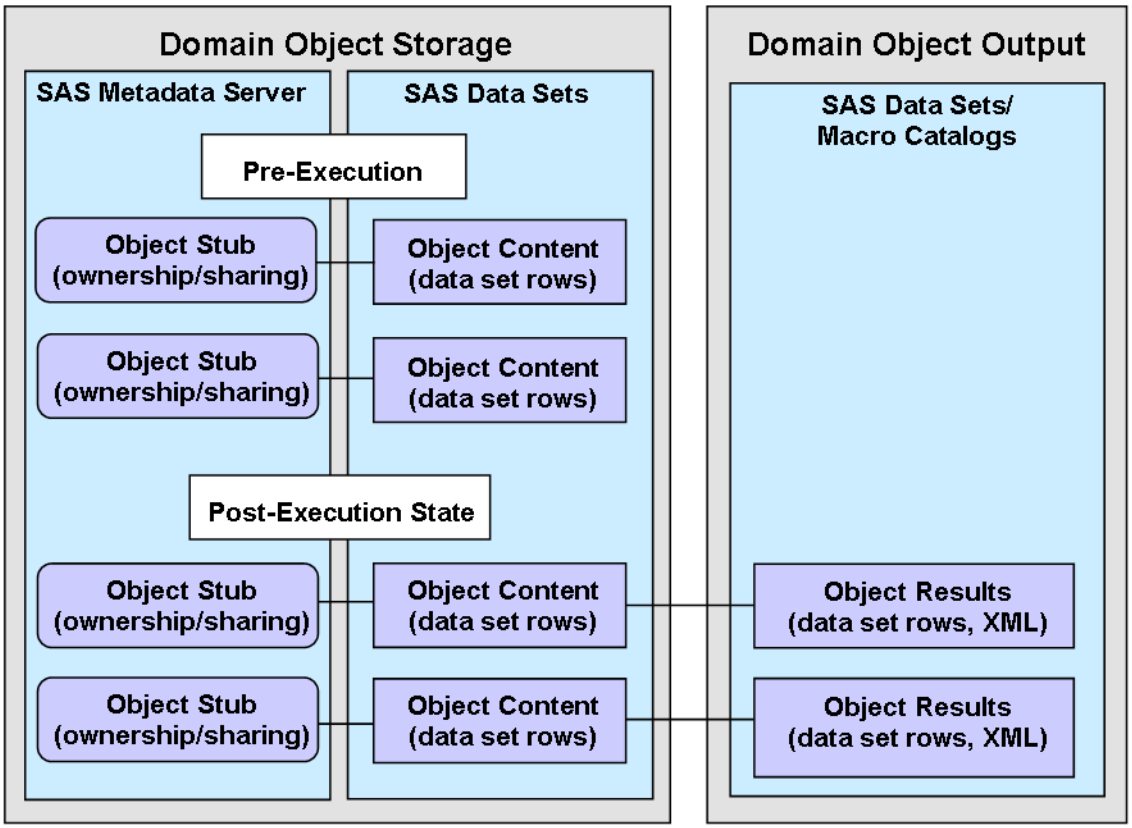
- Project definitions govern how the rich client organizes the work that a user performs. A project definition is just an organizing container for data selections and analyses and never has any execution results associated with it.
- Data selection definitions determine how the rich client stores the instructions for extracting from the data mart the data that is used in warranty analyses. A data selection definition produces a filtered subset, which is one or more SAS data sets that contain an extraction from the SAS Warranty Analysis data mart. This method enables multiple analyses to run against the same filtered subset, and it helps analyses to run more quickly than if they ran against the data mart directly.
- Report definitions determine how the rich client makes analytical results available to be shared with other users. A report definition produces summarized analytic results that consist of XML and SAS data sets that are later displayed in the SAS Warranty Analysis clients.

These domain objects share the following characteristics:

- They are retained in SAS data sets and SAS Metadata Server objects.
- They are used to drive back-end processes that produce results in the form of data artifacts.

Figure 4.1 indicates the basic layout for storage of persistent domain objects. Execution results are not attached to the domain objects. The results are created during execution and are then associated with domain objects.

Figure 4.1 Persistence of Domain Objects



The content and execution results of different domain objects are stored in different places. See "Warranty Data Tier" on page 12 in Chapter 2, "System Architecture," for the SAS libraries used to store different kinds of persistent data.

Domain Object and Results Ownership

In order for SAS Warranty Analysis to run well, a number of rules and dependencies must be honored. For example, "ownership" of domain objects is handled differently from "ownership" of the execution results that are associated with domain objects. The connections among the domain objects, their execution results, and the users who created the domain objects are complex. If you attempt to manually move data around in the system, it is extremely likely that you will damage some of those connections.

Different mechanisms are used for securing domain objects and results. Domain objects are secured by using metadata access privileges. Results data are secured with file system access control by placing them in special directories that are associated with a specific user. This distinction can help you to better understand the following sections.

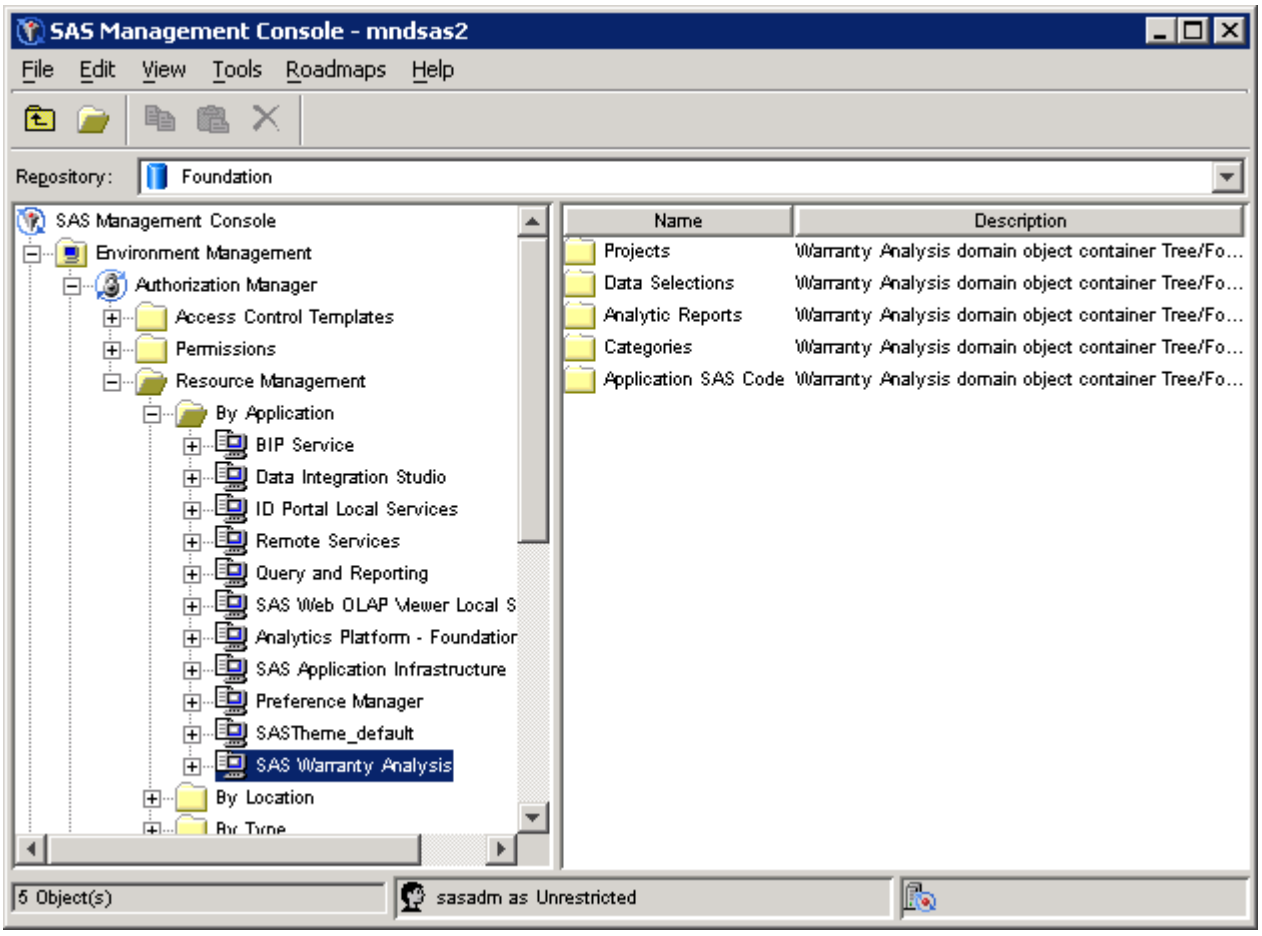
Domain Object Ownership

SAS Warranty Analysis uses the SAS Metadata Server to secure access to domain objects and to grant ownership privileges to users. The rules for access to domain objects are as follows:

- The user who creates an object "owns" it (that is, can see and modify it).
- Ownership of each object is also granted to members of the SAS Warranty Analysis Full Administrators group.
- An object's owner can grant read-only access to other users and groups.

Use the SAS Management Console to find the metadata components of warranty domain objects, as shown in Figure 4.2.

Figure 4.2 The Metadata Components of SAS Warranty Analysis Domain Objects



Use the SAS Management Console to see the access privileges that are assigned to specific users.

CAUTION: Do *not* manually alter any of the system-assigned privileges that are related to warranty domain objects that persist into the SAS Metadata Server. Do *not* manually alter the privileges that are related to any of the container objects, such as the folders or the SAS Warranty Analysis software component itself. The system relies on these settings to work properly.

Execution Results Ownership

Two of the SAS Warranty Analysis domain objects, data selections and analytic results produce execution results. These results are owned by the user who created the domain object that produces the results. (Analytic results can be shared as public reports.)

Data Selection Results

The data selection domain object produces filtered subsets. The following characteristics of filtered subsets are very important to note:

- Filtered subsets can take a long time for the system to create.
- Filtered subsets that result from executing a data selection can require a large amount of disk space.

For these reasons, filtered subsets are an expensive resource. Most of the disk space that SAS Warranty Analysis uses (other than for the data mart) is taken up by filtered subsets.

Ownership of filtered subsets is indicated within the system as follows. Each filtered subset is created in a specially named subdirectory. The name of the subdirectory associates the directory and its contents with the user who created the data selection that produced the filtered subset.

Analytic Results

Analytic results are owned by the user who created the analysis or report. But, unlike filtered subsets, analytic results are not stored in a subdirectory that is associated with the user. Instead, analytic results are stored in SAS catalogs that can be found in the SAS library that is associated with that type of analysis (USERRL or EIOUT). These catalogs are described in Chapter 2, "System Architecture."

Issues with Manual Deletion of Results

Do *not* manually delete the results of SAS Warranty Analysis domain objects.

CAUTION: All domain objects that have results share the following behavior. If the results of domain object are deleted in any way other than by the SAS Warranty Analysis system during normal operation, then the domain object's definition is no longer usable. You can view the various

settings in the client programs, but the objects are no longer functional. The only way to make these domain objects functional again is to re-create them with the same settings and then rerun them.

CAUTION: Do not rename or copy results from one user's directory to another. This circumvents the SAS Warranty Analysis system. Such changes are not picked up by the system and are not handled correctly.

Filtered Subset Sharing, Updating, and Copying

Because filtered subsets are such an expensive resource, SAS Warranty Analysis was designed to attempt to reduce their impact on the capacities of the servers that run the system. The application attempts to reuse filtered subsets whenever possible and tracks these filtered subsets to determine when they need to be updated and copied.

Filtered subsets are created or updated during the execution of an analysis, if the data has not been generated yet or if it is out of date. Each analysis depends on a data selection definition; therefore, it also depends on the filtered subset that is associated with the data selection. The system checks all of these dependencies (and other factors) to determine whether a filtered subset should be created. Once a filtered subset is created, it is reused by all analyses that require it.

For instance, SAS Warranty Analysis attempts to conserve disk space by allowing multiple projects and public reports to share data selection definitions. As long as two or more projects or reports are using the same data selection definition, then the filtered subset is not duplicated. However, when a user who has imported a public report performs an "Analyze in Project" action, the filtered data are copied to that user's *filteredData* folder.

NOTE: Obsolete reports should be deleted on a regular basis. Deleting obsolete reports also deletes the filtered subsets that are associated with them and frees system resources.

When Data Selection Definitions Are Copied

Multiple data selection definitions can point to the same filtered subset. This is one way in which SAS Warranty Analysis reuses a filtered subset. The following circumstances cause a data selection definition to be copied:

- saving a project-based analysis as a public report
- drilling into an emerging issues threshold report or public report to perform further analyses in a project

SAS Warranty Analysis does not attempt to reuse data selection definitions, but rather uses them to track reuse of filtered subsets. See "When a Previous Version of a Filtered Subset Is Kept" on page 27.

When a Previous Version of a Filtered Subset Is Kept

The following circumstances cause the system to keep an older version of a filtered subset and generate a newer version:

- A retired (static) public report or project-based analysis still uses the older filtered subset.
- The data selection definition has been updated.
- The data mart has been refreshed.

When Data Selection Definitions Are Copied

Filtered subsets are copied when a user imports a public report from another user and then performs an "Analyze in Project" action.

Automatic Deletion of Filtered Subsets

It is important for users to delete analyses and reports when they are no longer needed. This practice reduces the impact of filtered data sets on disk space because SAS Warranty Analysis automatically deletes any filtered subsets that are no longer associated with any analysis or report.

When Someone Leaves the Company

The user who creates a domain object owns that object. Administrators can see and delete the object, but not the filtered subsets that are owned by user and are stored in that user's system directory. SAS Warranty Analysis cannot transfer ownership of a domain object from one user to another.

To ensure efficient transfer of information and data access privileges when a user leaves the company, the administrator should do the following:

1. Make sure that the exiting employee does a knowledge transfer of the domain objects.
2. Have the employee who is taking over the exiting employee's work duplicate all of that employee's object definitions and rerun them.
3. Have the exiting employee delete the successfully duplicated objects and the related filtered subsets.

CAUTION: Do *not* manually move data and permissions. See "Issues with Manual Deletion of Results" on page 25.

When Available Disk Space Is Low

Overview

SAS Warranty Analysis requires that adequate disk space be available on all tiers in order to perform properly. Administrators must monitor disk space and take measures to clear disk space as needed. Use the Disk Usage utility on the **Tools** menu in the rich client to monitor disk space. See "The Disk Usage Utility" in the "Tools Overview" section of either the online Help or *SAS Warranty Analysis User's Guide* for more information.

CAUTION: If any domain object results are deleted, then the domain objects no longer execute properly, and you must re-create the domain objects with the same settings and execute them again.

Clear Existing Disk Space

Try the following actions when disk space gets full:

- Ask your user community to delete old reports and analyses.
- Monitor the decrease as old definitions are deleted.
- Monitor disk space usage over time to make sure your servers have enough space.

Move Users to Different Volumes

If the preceding measures do not alleviate the problem, then you can move specific users to another volume if your SAS server has multiple volumes. In SAS Warranty Analysis 4.1, the user-filtered data *must* remain on the application server. This data can be moved to a different volume on the application server, but it must never be moved to a separate computer.

To move a user's filtered data folder to a different volume:

1. In an interactive SAS session, edit the PARMSL.USERATTRIBUTES data set. For the specified user, change the value of the column SERVERPATH (the directory path on the server for user data selections) to point to the desired drive and base path. Leave intact the lowest-level folder, which is assigned by the application.
2. Create the folder where you want to store the user's data. The name of this folder is supplied by the SERVERPATH column, and the *\filteredData* subdirectory is under it. You **must** use this naming convention, or the application will fail to find the moved data.

3. Use the operating system to copy or move all of the filtered data sets from the original location to the new location defined by *\SERVERPATH\filteredData*.

4. When the user logs in to the application again, the new location for the user's filtered subsets is used. When the user creates new filtered data sets, the data sets are stored in the new location.

When Data Losses Occur

Overview

Various kinds of system failures can cause data to be lost. Data loss applies to single domain objects or to the full collection being managed by the system.

Filtered Subset Loss

It is best to back up filtered subsets; however, this backup is not required. SAS Warranty Analysis is still functional to some extent even if all the filtered subsets are lost, but there is some loss of functionality. For example, you cannot create a new analysis under a data node if the data for that node no longer exists.

If the filtered subsets are lost, but the data selection definitions (in the Data Selections workspace) remain intact, users must complete the following steps to re-create their filtered data sets:

1 In the Data Selections workspace, re-create the simple data selection or combined data selection definitions from their product and event components. Use new names for the new definitions.

2 In the Projects workspace, click **Replace data selection**, and select the newly re-created simple or combined data selection definitions.

3 Rerun the analyses using these definitions to re-create the lost data.

4 When prompted to replace the existing analyses, click **Edit**. (There is no need to first copy the existing analyses, because their data are lost.)

SAS Metadata Server Data Loss

Your SAS Metadata Server's persistent storage should always be backed up so that the data are never lost. However, one or more domain objects could possibly lose their metadata objects. Without the

metadata object, the domain object will never be seen again in any SAS Warranty Analysis client. All access to domain objects occurs through the SAS Metadata Server.

For more information about backing up the SAS Metadata Server's repository, see chapters 6 and 7 of the *SAS 9.1.3 Intelligence Platform System Administration Guide*:

http://support.sas.com/documentation/configuration/bisag.pdf

Domain Object Table Loss

The SAS Warranty Analysis application tables (or "warranty metadata," which is stored in SAS data sets and is not to be confused with the SAS Metadata Server data) should always be backed up so that the domain object tables are never lost. However, one or more domain objects could possibly lose their tabular data. If this happens, the SAS Metadata Server objects still exist. The system ignores any domain object IDs that come back from SAS Metadata Server queries and do not match up with any table-based information.

Chapter 5

Security

Contents

Overview

SAS Warranty Analysis administrators grant users access to functionality by making users members of the standard SAS Warranty Analysis groups.

The following chapters describe the group structure and explain how to use it to grant or deny users access to system functionality.

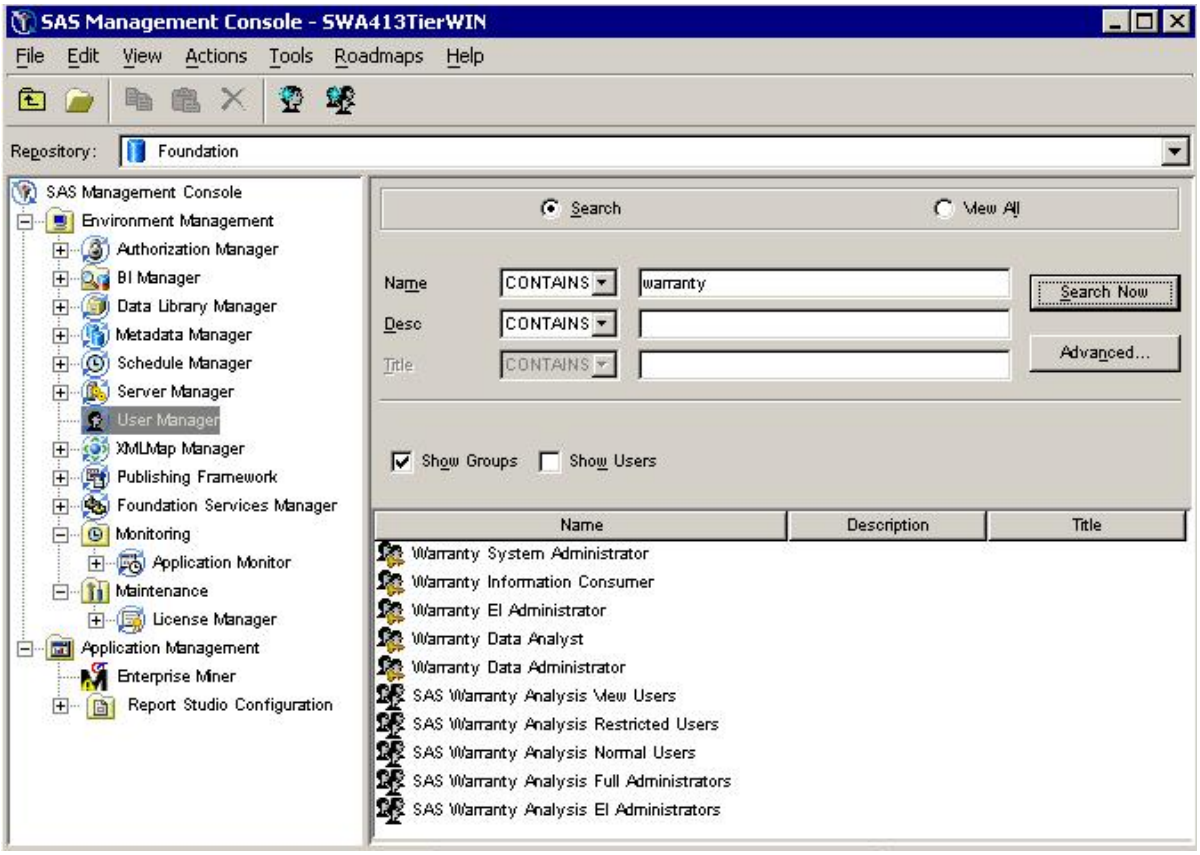
SAS Business Intelligence Platform Groups and Roles

The SAS Business Intelligence platform employs the "Group" and "Role" concepts regarding system security:

- **Group** protects access to system resources. System resources become available within the system when an administrator assigns a user to a group. Each group encompasses associated roles.
- **Role** provides access to specific software features. The software features that a user can access in the client user interfaces are determined by the roles that are associated with the group that the user is assigned to.

Use SAS Management Console to view the SAS Warranty Analysis roles and groups, as shown in Figure 5.1. (The icons for groups have three heads, and the icons for roles have two heads with a key.)

Figure 5.1 SAS Warranty Analysis Roles and Groups in SAS Management Console



SAS Warranty Analysis has a group for every role. Membership in a group enables a member to access any system resources that are controlled through the group's access control list (ACL). In

addition, membership in a group automatically assigns the user to any role that is associated with the group. This role membership enables the user to access all of the software functionality that is enabled by the role.

NOTE: It is important to use group membership to associate users with roles. If you assign a user to a role directly, the user does not have access to required system resources. In this case, the system would malfunction for that user.

Protected Resources

Overview

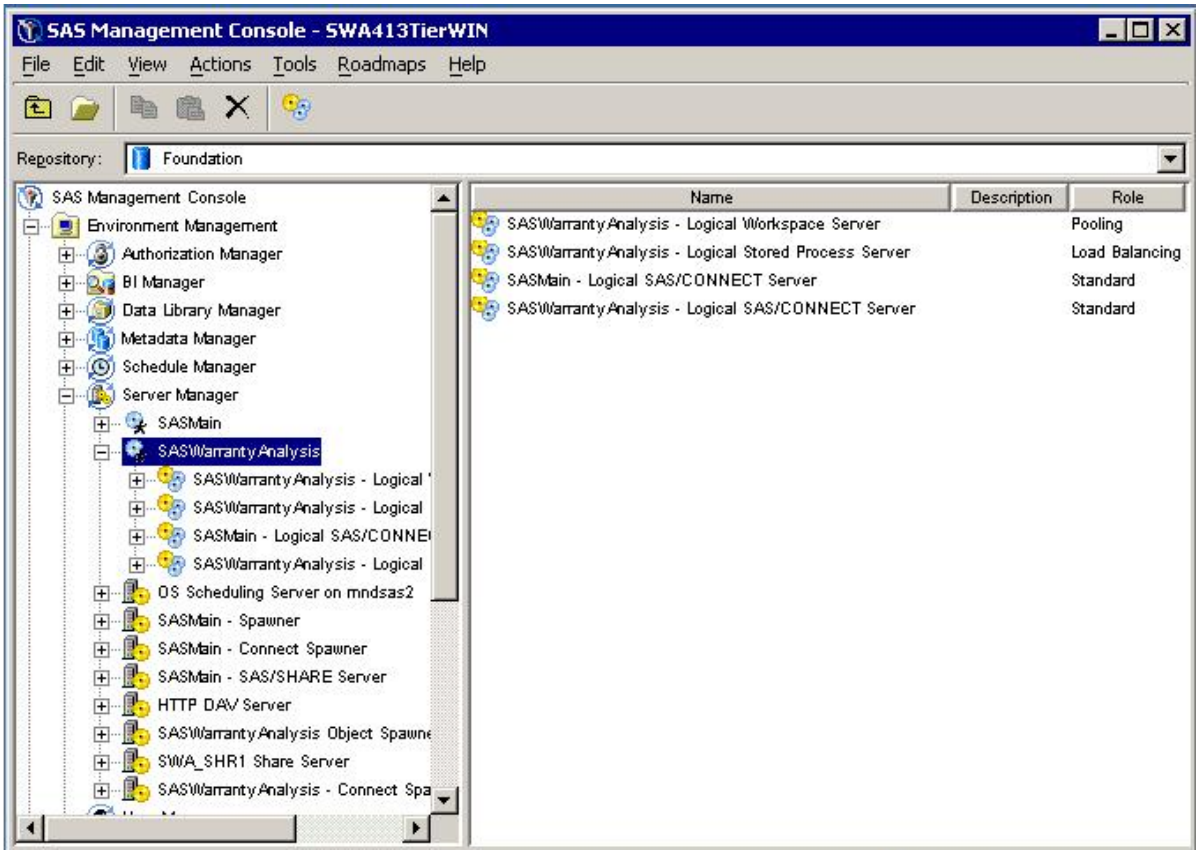
SAS Warranty Analysis uses a number of system resources. The application protects these resources by assigning access privileges to the various system groups and requiring that a user be a member of a group in order to acquire the access privileges of that group.

"Security Model" on page 35 provides additional information about the system resources.

Application Server

SAS Warranty Analysis provides its own SAS Business Intelligence platform application server with application-specific command-line options. This application server is called SWAWarranty-Analysis (see Figure 5.2). SAS Warranty Analysis cannot use the default SASMain application server. The Warranty system groups provide the access privileges that are necessary to run work on this application server.

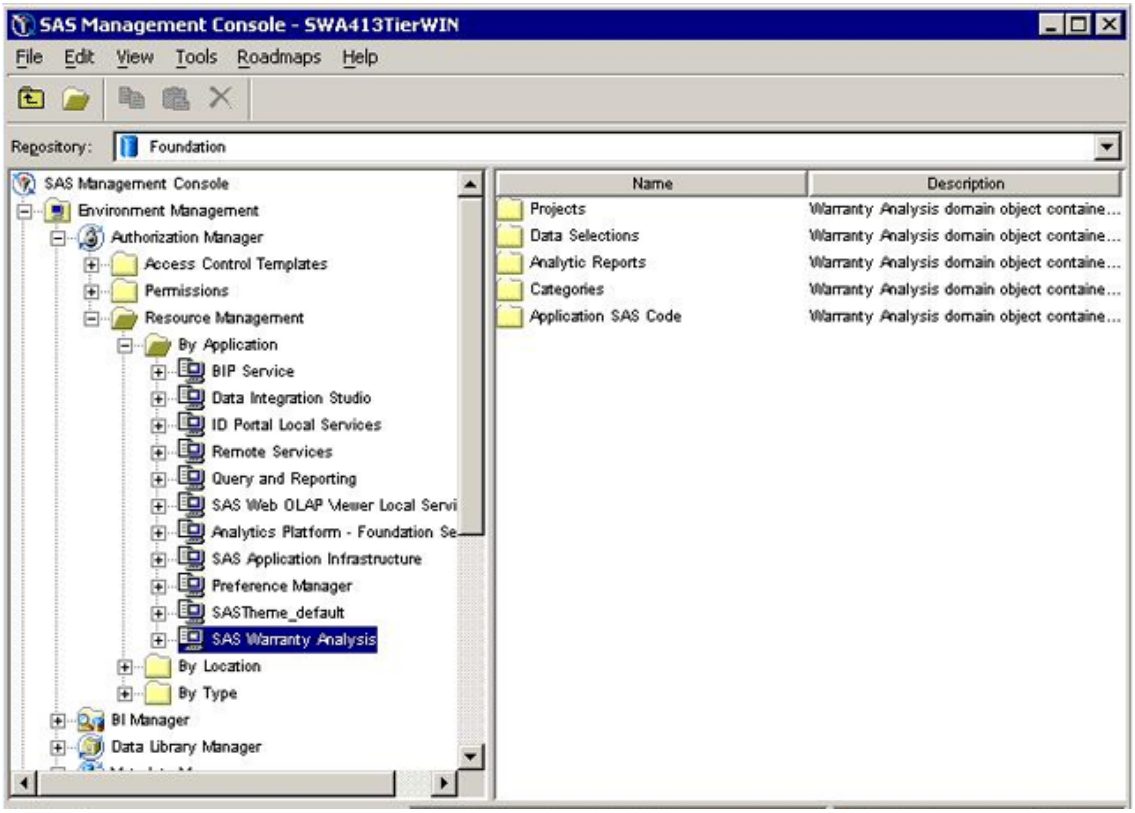
Figure 5.2 The SASWarrantyAnalysis Application Server As Seen in SAS Management Console



Application Metadata

As explained in Chapter 4, "Working with Domain Objects," SAS Warranty Analysis has a place to store the metadata portion of its domain objects The system is installed and configured to have access privileges to the SAS Warranty Analysis software component and to the folders beneath it, as shown in Figure 5.3. Additionally, access privileges are placed on the domain object metadata objects when they are created.

Figure 5.3 SAS Warranty Analysis Metadata Objects



Security Model

Overview

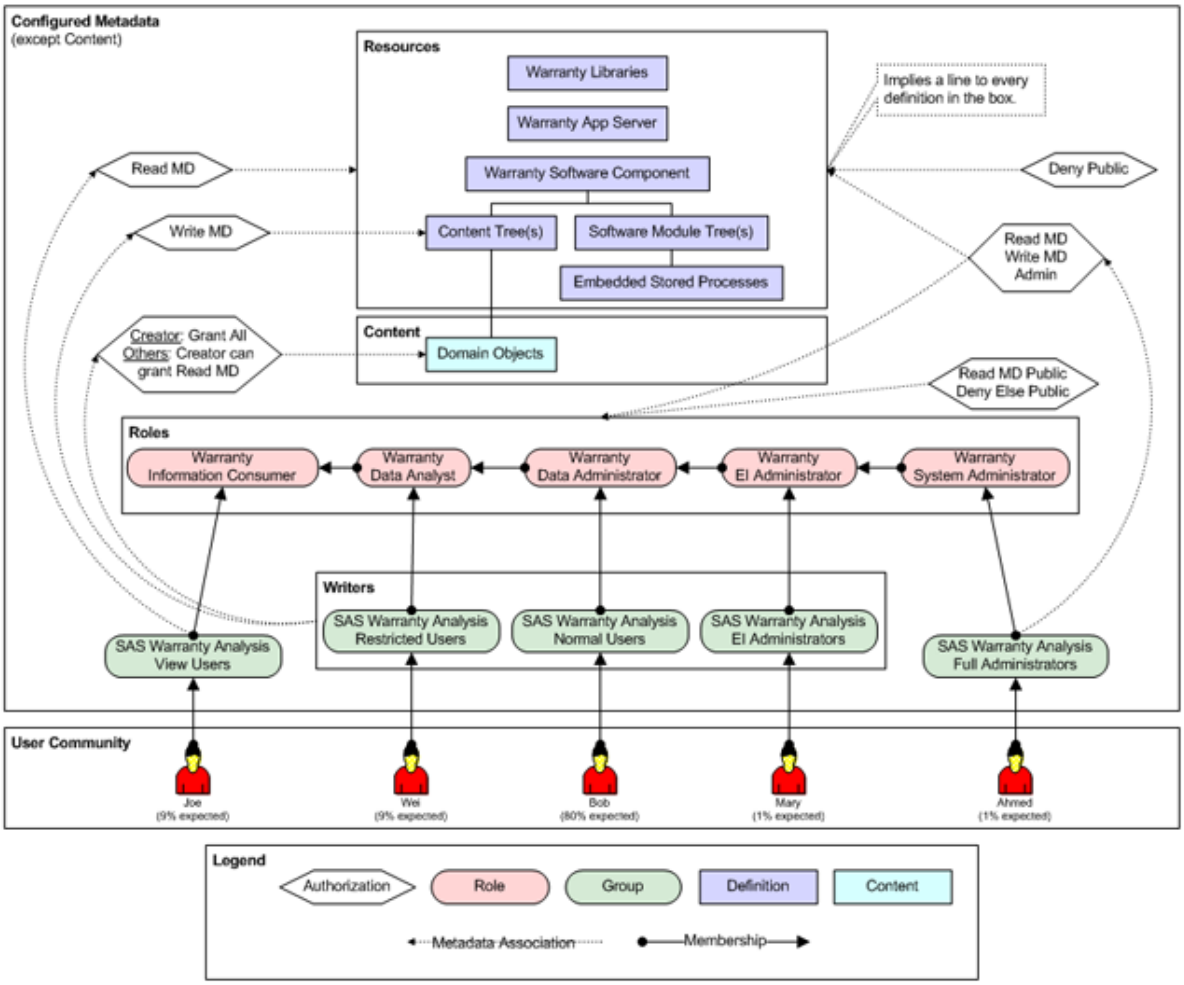
Figure 5.4 shows how data and computing resources are secured in the SAS Warranty Analysis system. The roles (in pink) show how user types are defined within the system, and the roles control which software features users can access. The roles are cumulative; if a user is assigned a role, he or she automatically receives all the privileges of all the roles to the left of the assigned role.

The groups (in green) define the access privileges to the various system resources (in blue) that the system uses.

NOTE: Always associate users with roles by assigning membership to the groups that the roles correspond to. SAS Management Console enables you to assign a user to a role, but this user would not receive the access privileges that are provided by group membership and therefore would be unable to run the software.

CAUTION: Do not change any permissions for any of the roles or groups. SAS Warranty Analysis depends on these settings remaining as they are.

Figure 5.4 Warranty Metadata Definitions and Security



NOTE: The abbreviation "MD" means "metadata."

Roles

The SAS Warranty Analysis system roles are as follows:

- **Warranty Information Consumer**: enables access to any content in the thin client. **NOTE:** All SAS Warranty Analysis users must be assigned to the SAS Warranty Analysis View Users group, which then applies this role.
- **Warranty Data Analyst**: enables access to the rich client, but not to data selection editing.
- **Warranty Data Administrator**: enables access to the rich client's data selection-building screens. **NOTE:** The user also needs to be associated with the Warranty Data Analyst role.

- **Warranty EI Administrator**: enables access to the emerging issues run-time setup functionality.
- **Warranty System Administrator**: enables access to all other screens that are related to system configuration and that are not included in this list.

Each group has a corresponding role. The groups are members of the role, so that if you make a user a member of a group, that user is automatically a member of the corresponding role.

Nested Groups for Cumulative Access Privileges and Roles

SAS Warranty Analysis groups have nested membership. This nesting means that users with membership in a given group inherit membership in all other groups that are subordinate to the initial group.

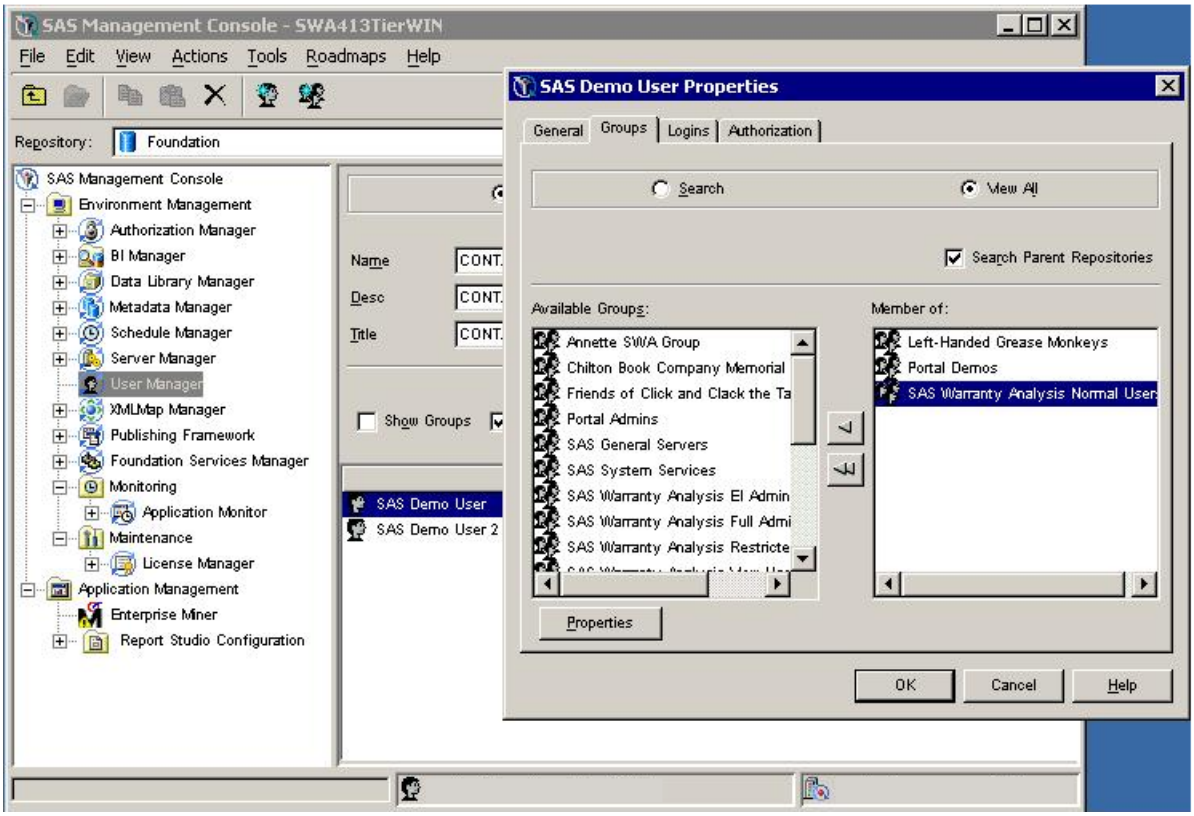
For example, if you make a user a member of the SAS Warranty Analysis Normal Users group, then that user also has access to all of the functionality that is provided by the SAS Warranty Analysis Restricted Users group and the SAS Warranty Analysis View Users group because these groups are subordinate to, or nested under, the Normal Users group.

Granting Access through Group Membership

You can grant a user access to the SAS Warranty Analysis system simply by going into the SAS Management Console and making that user a member of one of the five SAS Warranty Analysis groups. (See Figure 5.5.)

NOTE: You do not need to make a user a member of multiple groups. Group membership nesting makes this unnecessary.

Figure 5.5 Assigning Group Membership



Denying Access to Emerging Issues

The SAS Warranty Analysis system offers an optional role and group for occasions when you might want to enable a user to see public reports but not emerging issues alerts. For example, some SAS Warranty Analysis customers enable their suppliers to view some of these public reports, but they do not want the suppliers to see emerging issues alerts.

For this reason, the system supports these optional metadata definitions:

- **Role**: Warranty Deny EI Access
- **Group**: SAS Warranty Analysis Deny EI Access

If this role and group are set up, the system automatically prevents users who are members of any other groups from seeing emerging issues reports.

Creating New Groups for Domain Object Sharing

Use the SAS Management Console to add new groups. No warranty-specific privileges need to be applied to them.

Filtering Out Groups from Domain Object Sharing

Overview

SAS Warranty Analysis enables users to share data selection definitions and public reports with other users. Users can open a dialog box in the rich client where they can choose users and groups as targets for sharing.

A disadvantage of using the SAS Management Console and SAS 9.1.3 to administer Groups is that the platform makes no distinction between system-oriented groups and business-oriented groups. The system-oriented groups are created during the default installation and configuration process. The business-oriented groups are created later by SAS Warranty Analysis administrators. All groups that are displayed in the SAS Management Console (unless they are otherwise filtered out) show up as potential targets for information sharing in the rich client. Filtering out groups that are not appropriate for sharing keeps the display simpler and easier to use. For this reason, SAS Warranty Analysis automatically filters out the default groups. But if you add more groups later, you must filter these out manually.

The following sections describe what is automatically excluded from the user interfaces for domain object sharing and what you should exclude manually.

Roles Never Show Up for Sharing

SAS Warranty Analysis does not show any roles for the purpose of domain object sharing. Roles are not used in sharing domain objects; so they are excluded from display in the user interface.

Groups Never Show Up for Sharing

SAS Warranty Analysis automatically excludes the default groups from the user interfaces for sharing of domain objects. The system was created with special assumptions about how groups are used that make it preferable not to use groups for sharing.

Exclude Additional Groups Using app.config

More system groups can be added during additional site-specific configuration. None of these groups are appropriate for sharing. If groups other than the default groups are added during installation and configuration for your site, you should exclude these additional groups from displaying in the user interface for sharing of domain objects.

To add more group names to the list of groups to be excluded, open the *app.config* file and scroll down to a series of numbered properties like those in the following example. (See Chapter 3, "Finding Scripts, Files, and Other System Components," for the location of the *app.config* file.)

```
# Identities to filter out of the sharing user interfaces
sharing.identityFilter.name1=SASUsers
sharing.identityFilter.name2=Public
sharing.identityFilter.name3=SAS System Services
sharing.identityFilter.name4=SAS General Servers
sharing.identityFilter.name5=SAS Administrator
sharing.identityFilter.name6=SAS Demo User
sharing.identityFilter.name7=SAS Warranty Analysis Server User
sharing.identityFilter.name8=SAS Web Administrator
sharing.identityFilter.name9=SAS Trusted User
sharing.identityFilter.name10=Performance Testing
sharing.identityFilter.name11=Portal Admins
sharing.identityFilter.name12=Portal Demos
```

Add to this list the group names that you want to exclude from displaying in the user interface. You can have as many such properties as you need.

CAUTION: These properties are numbered (for example, `name1` to `name5`), and the numbers must be strictly sequential. As soon as SAS Warranty Analysis fails to find the next one in sequence, it stops applying the properties.

NOTE: You must restart the SAS Analytics Platform server to apply these changes.

Chapter 6

Starting and Stopping the System

Contents

Overview

This section explains how to start and stop the entire SAS Warranty Analysis system (which might consist of multiple machines). Chapter 2, "System Architecture," describes the SAS tier and the middle tier. Scripts in both tiers need to be addressed.

NOTE: The scripts described in this chapter are the detault scripts created by the standard installation and configuration process for the SAS Warranty Analysis bundle. The SAS Consulting services staff who set up your site's system might have customized the scripts to make the starting and stopping process easier and more streamlined. SAS Consulting services staff provide documentation that describes these site-specific customizations.

Tier Dependencies

The SAS Warranty Analysis middle tier depends on the resources available from the SAS Warranty Analysis server tier. Because of this dependency, start up the system as follows:

1. **SAS tier**: The SAS services must be started first, or the middle tier will not run.
2. **middle tier**: The middle tier comes up next to enable clients to run.

Shut down the system as follows:

1. **middle tier**: The middle tier must shut down and release all SAS tier resources (such as server connections) and terminate the ability of clients to invoke anything else on the SAS tier.
2. **SAS tier**: The SAS tier shuts down cleanly without any locked sessions.

Metadata Server Start Scripts

In a UNIX environment, use the following command:

```
<SASConfig>/Lev1/SASMain/MetadataServer/MetadataServer.sh start}
```

NOTE: *<SASConfig>* refers to your SAS Business Intelligence configuration folder.

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\SASMain\MetadataServer\MetadataServer.bat start
```

Warranty Object Spawner Scripts

In a UNIX environment, use the following command:

```
<SASConfig>\Lev1\SASMain\MetadataServer\MetadataServer.bat start
```

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\SASWarrantyAnalysis\ObjectSpawner\ObjectSpawner.bat start
```

Warranty Share Server Scripts

In a UNIX environment, use the following command:

```
<SASConfig>/Lev1/SASWarrantyAnalysis/ShareServer/ShareServer.sh start
```

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\SASWarrantyAnalysis\ShareServer\ShareServer.bat start
```

Application Server Scripts

Before you start your Web application server, start the remote services.

In a UNIX environment, use the following command:

```
<SASConfig>/Lev1/web/StartRemoteServices.sh
```

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\web\StartRemoteServices.bat
```

For Web application servers such as Oracle BEA Weblogic and IBM Websphere, instructions for starting are provided as part of the general application server instructions. The SAS configuration process provides a script to start Apache Tomcat.

In a UNIX environment, use the following command:

```
<SASConfig>/Lev1/web/startServletContainer.sh
```

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\web\startServletContainer.bat
```

Analytics Platform Scripts

In a UNIX environment, use the following commands to start the SAS Analytics Platform server:

```
   cd <SASInstall>/SAS/SASAPCore/bin
nohup apserver -headless start &
```

NOTE: *<SASInstall>* refers to the folder where SAS products are installed.

In a Windows environment, the scripts to start and stop the Analytics Platform are provided in the menu.

Select **Start ►Programs ►SAS**.

You can also start the SAS Analytics Platform by using the following command:

```
"C:\TestSetup\SAS Install\SAS\SASAPCore\bin\apserver.bat" start
```

Service Dependencies

The following SAS server tier services are started first in the following order:

1. metadata server
2. warranty object spawner
3. warranty SHARE server

The following middle tier services are started next in the following order:

1. remote services application
2. SAS Analytics Platform
3. Web server (Apache Tomcat, Oracle BEA Weblogic, or IBM Websphere)

The client tier does not have any services that need to be started. However, if the server tiers are not started, then the thin client fails to show web pages, and the rich client detects that the servers are not available, displays an error message, and terminates itself.

Chapter 7

Using the Diagnostic Client (Experimental)

Contents

Overview

The diagnostic client is an experimental client that is part of the SAS Warranty Analysis solution. It enables you to view some of the system's back-end processes. The diagnostic client can also help you to trace system errors that need to be worked through with SAS Technical Support.

Use the following URL to access the diagnostic client:
`http://hostname:6098/SASWarrantyAnalysis/JobStatus`

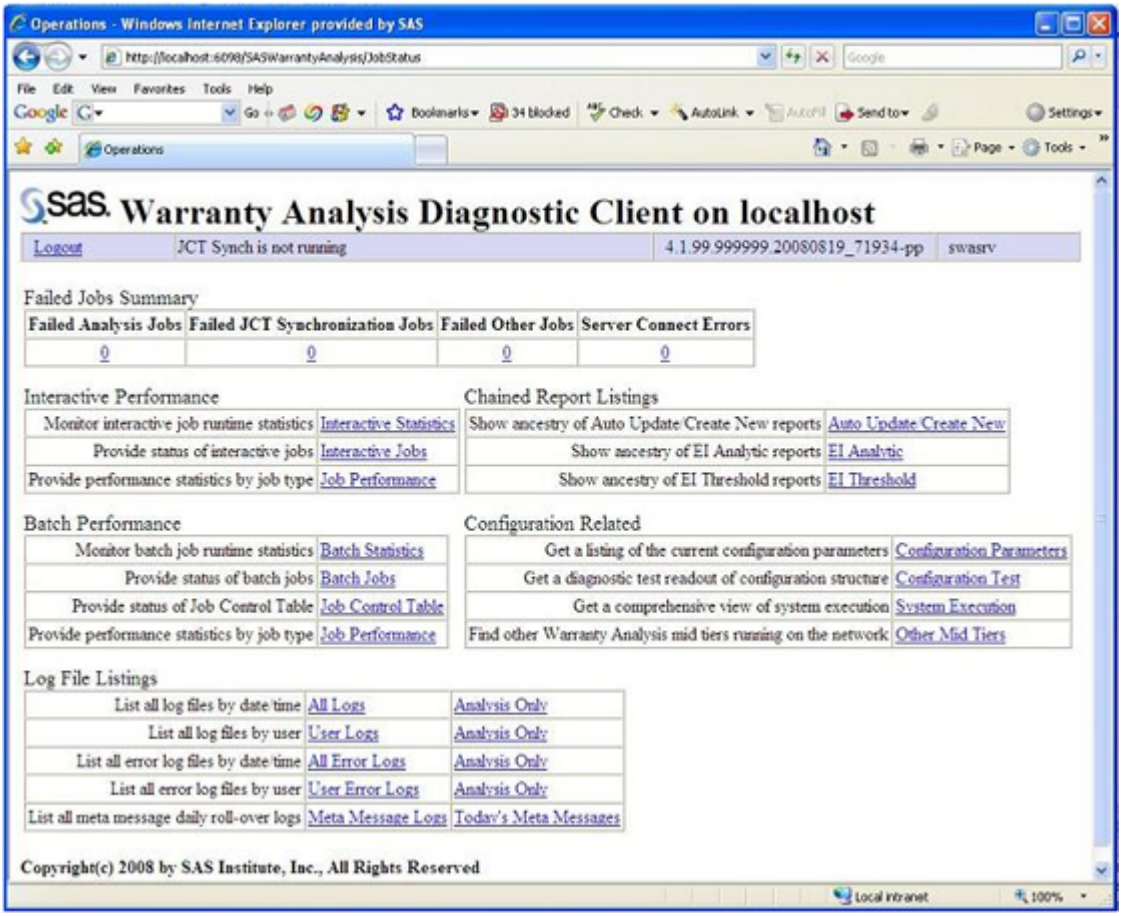
This chapter describes those parts of the diagnostic client not covered in other chapters. For example, the diagnostic client's job-related views are discussed in Chapter 8, "Working with Jobs, Tasks, and Queues."

NOTE: You must be a member of the SAS Warranty Analysis Full Administrators group to use the diagnostic client.

Main View

The main view of the diagnostic client (Figure 7.1) shows all available operations. Almost all of these views are read-only. Some of the views are useful for monitoring system health.

Figure 7.1 The SAS Warranty Analysis Diagnostic Client



Navigation Bar

Links on the navigation bar include a logoff link and a link back to the home (operation) page if you are not already on the home page.

The navigation bar also displays the user ID of the currently logged in user, the build number, how often the Job Control Table synchronization cycle runs, and when it last ran.

Failed Jobs Summary

The Failed Jobs Summary area of the diagnostic client shows how many critical errors appear in each job log. Error logs age out of the system, so this view generally shows only the most recent errors. Click the error count to drill to a table of job descriptions. Each job description has a link to a related job log. This is a quick way to see what critical errors have happened in the system.

The Failed Jobs Summary area displays the following summaries, as shown in Figure 7.1:

- **Failed Analysis Jobs**: Monitor the Failed Analysis Jobs log to determine what might be interfering with analysis runs. If runs have failed, scan the log file and note any errors. Your SAS Consulting service representative or SAS Technical Support can assist you with interpreting the errors.
- **Failed JCT Synchronization Jobs**: Job control table (JCT) synchronization is the process by which the SAS Warranty Analysis middle tier keeps synchronized with the PARMSL Job-Control table. These jobs are important because they keep batch processing moving. Errors should not occur in these jobs. If you do notice any such errors, investigate them immediately.
- **Failed Other Jobs**: Most jobs are either for analyses or for JCT synchronization; However, other types of jobs run, and this last count shows how many of them have errors.
- **Server Connect Errors**: Server Connect Errors is a dashboard counter that displays the number of server connections that have failed. These are serious errors indicating that the SAS back end is not working properly. Click the error count link to display a table that lists the number and nature of each problem. If you see many of these errors, then restart the SAS and middle-tier machines and have everyone log on again to continue their work. If problems continue after the restart, then call SAS Technical Support for assistance.

Interactive Performance

This set of screens is discussed in Chapter 8, "Working with Jobs, Tasks, and Queues."

Batch Performance

This set of screens is discussed in Chapter 8, "Working with Jobs, Tasks, and Queues."

Chained Report Listings

This set of views assists you with low-level debugging rather than routine administrative maintenance. See Figure 7.1. These views enable you to drill into public reports, emerging issues threshold reports, or emerging issues analytical reports to see how they have been cloned during the batch "auto-update/create new" process. These views are useful for tracing the contents of the USERDL.analysis table for more in-depth debugging.

Configuration Related

This set of views enables you to see or test aspects of the configuration. See Figure 7.1. The views are as follows:

- **Configuration Parameters**: This view provides a partial display of the *app.config* file configuration parameters that are currently in force.
- **Configuration Test**: This view performs a modest set of diagnostic tests to see how the system responds.
- **System Execution**: This view provides the most comprehensive view of how the queuing system is running. It shows the job queue, available tasks waiting to run, and the execution queues with work in progress. You can verify that work is running smoothly by monitoring this view.
- **Other Mid Tiers**: This view is more of a debugging tool. It causes the middle tier to use a SAS Analytics Platform facility to ping the network and see which other SAS Warranty Analysis middle tiers, if any, are running.

Log File Listings

The Log File Listings section of the diagnostic client window is especially helpful when you are working with SAS Technical Support because it provides several ways to find specific log files. See Figure 7.1. The SAS Warranty Analysis system records a large number of smaller log files, one per job, to help you to identify the specific circumstances that might have caused an error. The smaller log files prevent the need to weed through one large, all-inclusive log file. For each category of log file listings, you can select the first link, which displays all available logs, or the second link, which displays a more limited set of logs.

The links in this section include the following:

- **List all log files by date/time:** The first link displays all of the log files in chronological order. You probably do not want to use this link unless you are looking for the most recent log files. The second link filters for only the analysis job runs.
- **List all log files by user:** The first link displays a complete listing of all of the log files on the system, sorted by user. The second link shows only analysis job runs sorted by user.
- **List all error log files by date/time:** The first link displays a chronological listing of all error log files that contain errors. The second link shows only analysis job runs in chronological order.
- **List all error log files by user:** The first link provides a listing of all log files that contain errors, sorted by user. The second link shows analysis job runs that contain errors, sorted by user.

The last set of links do not lead to job log files, but to a journal file of major events that occurred in queuing.

- **Meta Message Logs:** Lists all available meta message log files.
- **Today's Meta Messages:** Takes you directly to the current day's meta message log file.

Anatomy of a Job Log File Listing

SAS Warranty Analysis creates a new log file for each job that runs. This log provides a comprehensive view of what happened during the job on both the SAS tier and the middle tiers.

The diagnostic client provides a way to navigate through the log files to view each error or warning that is generated. The following sections explain what to look for in a job log file listing.

Job Log Header

The first part of a job log contains header information, which includes the following:

- job file name
- optional table of errors with links to the errors and warnings in the log
- start date, time, and duration for the job

Following is an example of this content:

Retrieve Log File

Bottom

Error	Warning
Error 1 Error 2	Warning 1

```
Log for Job AnalysisLog_ANFKHB7GPZ_1220044868191

Started=Fri Aug 29 17:21:08 EDT 2008 Duration=1m 27.451s SubjectId=ANFKHB7GPZ SubjectType=MULTIVARIATE
```

Job Log Java Error

Job logs contain every kind of error and warning that can be generated during a job (for example, Java errors, SAS errors, and SAS warnings). A Java error shows the Java stack trace for an error that occurs during the Java portion of a job. An example of this follows:

```
Task com.sas.analytics.mfg.warranty.core.table.FetchTablePageTask

Error 1 Top Previous Next Bottom

Error during task (1):java.lang.IllegalArgumentException: Unsupported date value: .
        at com.sas.analytics.mfg.warranty.core.dview.client.DateInputFieldValidator.formatValue(DateInp
        at com.sas.analytics.mfg.warranty.core.persistence.sql.table.TableReader.loadPage(TableReader.j
        at com.sas.analytics.mfg.warranty.core.persistence.sql.table.TableReader.readPage(TableReader.j
        at com.sas.analytics.mfg.warranty.core.table.FetchTablePageTask.doExecute(FetchTablePageTask.ja
        at com.sas.analytics.mfg.warranty.core.persistence.cmd.BaseSqlTask.execute(BaseSqlTask.java:70)
        at com.sas.analytics.mfg.warranty.queue.task.Task.run(Task.java:507)
        at com.sas.analytics.mfg.warranty.queue.ExecutionQueue$TaskRunner.runTaskCode(ExecutionQueue.ja
        at com.sas.analytics.mfg.warranty.queue.ExecutionQueue$TaskRunner.run(ExecutionQueue.java:668)
        at java.lang.Thread.run(Thread.java:534)
```

Job Log SAS Error or Warning

The job log SAS errors and warnings show up in the content as follows:

```
Error 1 Top Previous Next Bottom

ERROR 22-322: Syntax error, expecting one of the following: a quoted string, !, !!, &, (, *,
              >, >=, ?, AND, AS, BETWEEN, CONTAINS, EQ, EQT, FORMAT, FROM, GE, GET, GT, GTT,
              LEN, LENGTH, LET, LIKE, LT, LTT, NE, NET, NOT, NOTIN, OR, TRANSCODE, ^, ^=, |,

NOTE: Line generated by the macro variable "L_PRODVAR2".
597                                     PRODUCT_ATTRIBUTE_2
                                        ___________________
                                        76

Error 2 Top Previous Next Bottom

ERROR 76-322: Syntax error, statement will be ignored.
```

Ignore Errors Section

Occasionally, sections of the log file contain particular blocks of SAS code that are deliberately surrounded with special log comments. The log comments prevent any errors that might happen from causing the job to fail. These blocks of code are used when the SAS code is expected to generate a benign error such as failure to achieve a table lock. For example, when the **`SKIP_SAS_ERRORS()`** code is used, the software anticipates file contention and tries again to achieve the lock if the first attempt fails. These SAS code blocks look like the following example:

```
SKIP_SAS_ERRORS(): START

WARNING: The SQL option UNDO_POLICY=REQUIRED is not in effect. If an error is detected when processing this UPDATE

         error will not cause the entire statement to fail.
NOTE: 1 row was updated in USERDL.ANALYSIS.

SKIP_SAS_ERRORS(): FINISH
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.06 seconds
      cpu time            0.01 seconds
```

Job Log Statistics Summary

Every job log ends with a summary of the performance statistics for the job run. This content resembles the following example:

```
NOTE: *** SWAMain Finished Execution ***
MLOGIC(SWAMAIN):  Ending execution.
598
599


Job Performance Statistics Top Last Error Last Warning

*** Runtime summary for job [warrantyAnalysis.AnalyticRunJob-JFKHB7P7J]
  Job Cumulative Statistics:
    Job count: 1
    Error count: 2
    Runtime: 00:01:27.467
    Submit-to-start: 33.234s 37%
    Job optimized: 78ms 0%
    Log post-process: 16ms 0%
    Finish-to-remove: 54.187s 61%
  Task Cumulative Statistics:
    Task count: 4
    Task sequence count: 1
    Wait-to-start: 0ms 0%
    Server Connections:
      Workspace(1): 46ms 0%
    Task execution: 54.187s 61%
      Log harvest: 12.562s 14%
      Task post-process: 16ms 0%
    Between tasks: 0ms 0%
```

Chapter 8

Working with Jobs, Tasks, and Queues

Contents

Overview

This chapter explains how SAS Warranty Analysis manages work through the software stack.

Most queuing tasks are done in the rich client. See "The Job Queue Utility" in the "Tools Overview" section of either the online Help or *SAS Warranty Analysis User's Guide* for more information about the job queue utility in the rich client.

NOTE: The rich client's administrative options for queuing affect only batch jobs.

The information in this chapter can help you to better understand Chapter 9, "Working with Batch Processes."

About Jobs, Tasks, and Queues

SAS Warranty Analysis performs all work, whether batch work or work in response to user interactions with a client, within the context of a "job." A job is a logical unit of work that includes one or

more tasks. Each task is a specific processing operation such as running a query or executing SAS code.

Every job is run in a queue, where its progress can be monitored. SAS Warranty Analysis runs all jobs on a first-come, first-served basis. Depending on the hardware capacity, a specified number of jobs can run simultaneously.

Managing Interactive and Batch jobs

Managing Interactive jobs

Interactive jobs are created and run in response to user actions within a SAS Warranty Analysis client. Interactive jobs, which are expected to run quickly (that is, in several seconds or less), are not tracked in the same way as batch jobs. After an interactive job starts, the client does not allow further actions until the results from that job are returned.

Managing Batch jobs

Batch jobs are created and run for any work that is expected to take a long time. This includes executing an analysis either by a user from the rich client or by means of the batch processing subsystem (see Chapter 9, "Working with Batch Processes"). Batch jobs are tracked more thoroughly than interactive jobs are tracked. When a user invokes a batch analysis in the rich client, the client does not prevent the user from doing other work. While the analysis runs, its status is updated; for example, the middle tier notifies the client to change the "gear" icon to a green check when the analysis finishes.

Managing Queues in the Rich Client

Users who are members of the SAS Warranty Analysis Full Administrators group have more options available to them in the rich client than other users. Among these options are the following:

- the ability to delete jobs that have not yet started running
- the ability to remove finished jobs that were submitted by any user

Using the Diagnostic Client to View Queues (Experimental)

Viewing Interactive Jobs

The diagnostic client provides several useful read-only views into the SAS Warranty Analysis queuing subsystem.

The URL to invoke the diagnostic client is `http://hostname:6098/SASWarrantyAnalysis/JobStatus`.

The Interactive Jobs view (Figure 8.1) shows a summary of the currently running interactive jobs. This view refreshes frequently so that you can see how the system is supporting its various interactive clients.

The **Submitted Jobs** section shows the following:

- the number of jobs that are currently running (**Running**)
- the highest number of jobs that have run simultaneously (**Max Running**)
- the number of jobs that are in the queue to run but have not yet started (**Waiting**)
- the highest number of jobs that have been in the queue waiting to run (**Max Waiting**)
- the number of jobs that are done (**Done**)

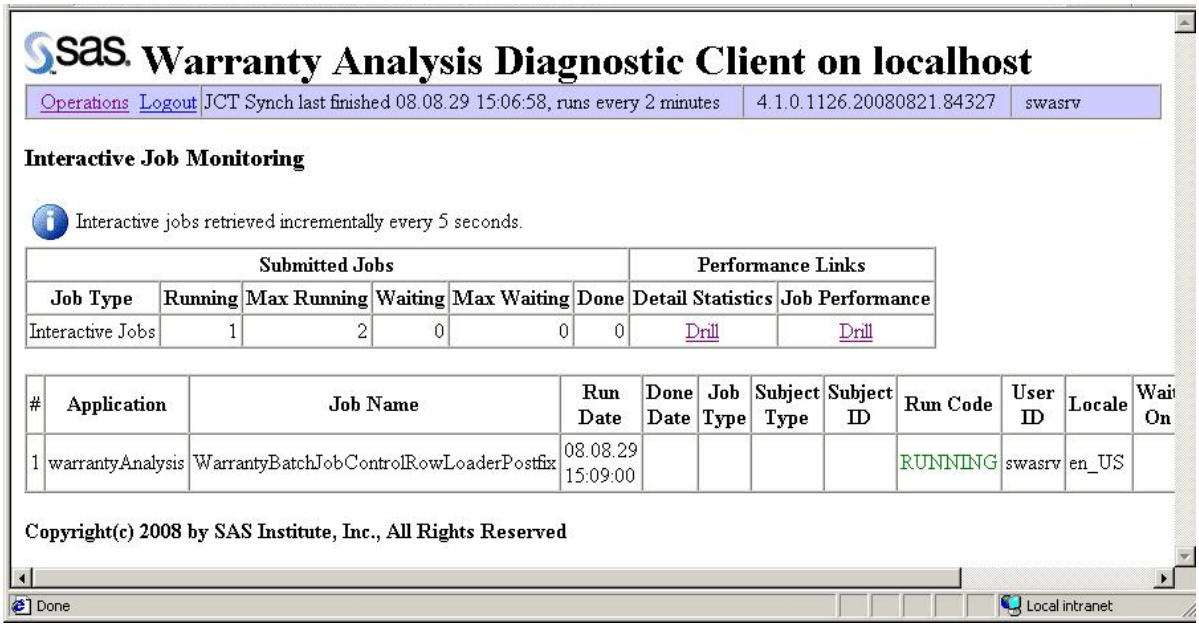
The **Performance Links** section contains links that connect to more detailed screens about system performance.

Other fields provide the following information for each job:

- **Job Name**: the internal names of currently running jobs
- **Run Date**: the date and time when the job starts
- **Done Date**: the date and time when the job completes
- **Run Code**: the status of the job (**Running**, **Resetting**, **Finished**, or **Starting**)
- **User ID**: the logon ID of the user who requests the job

Job Type, **Subject Type**, and **Subject ID** are internal to the system.

Figure 8.1 The Interactive Jobs View



Viewing Batch Jobs

The Batch Jobs view (Figure 8.2) shows a summary of the currently running batch jobs. This view refreshes frequently so that you can track changes in the batch environment.

The **Submitted Jobs** section shows the following:

- the number of jobs that are currently running (**Running**)
- the highest number of jobs that have run simultaneously (**Max Running**)
- the number of jobs that are in the queue to run but have not yet started (**Waiting**)
- the highest number of jobs that have been in the queue waiting to run (**Max Waiting**)
- the of jobs that are done (**Done**)

The **Performance Links** section contains links that connect to more detailed screens about system performance.

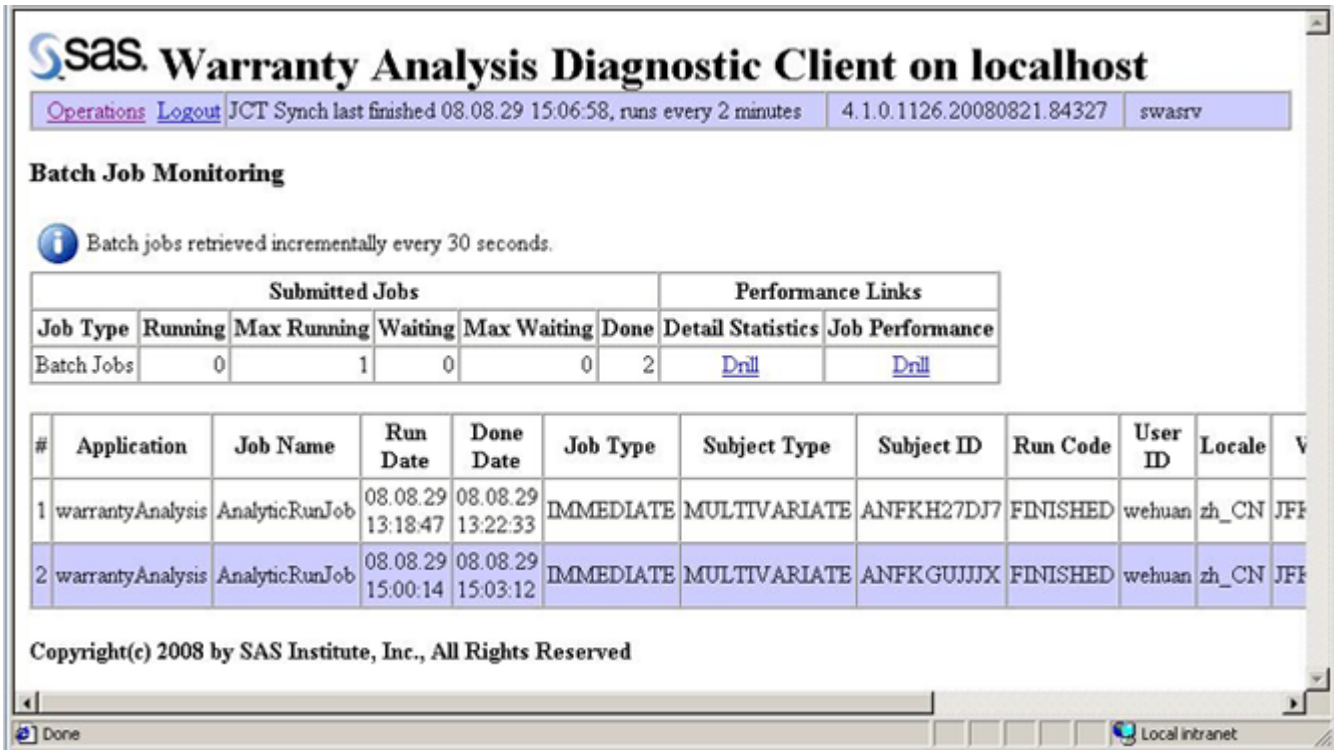
Other fields provide the following information for each job:

- **Job Name**: the internal names of currently running jobs
- **Run Date**: the date and time when the job starts
- **Done Date**: the date and time when the job completes
- **Run Code**: the status of the job (**Running**, **Resetting**, **Finished**, or **Starting**)

- **User ID**: the logon ID of the user who requests the job

Job Type, **Subject Type**, and **Subject ID** are internal to the system,

Figure 8.2 The Batch Jobs View

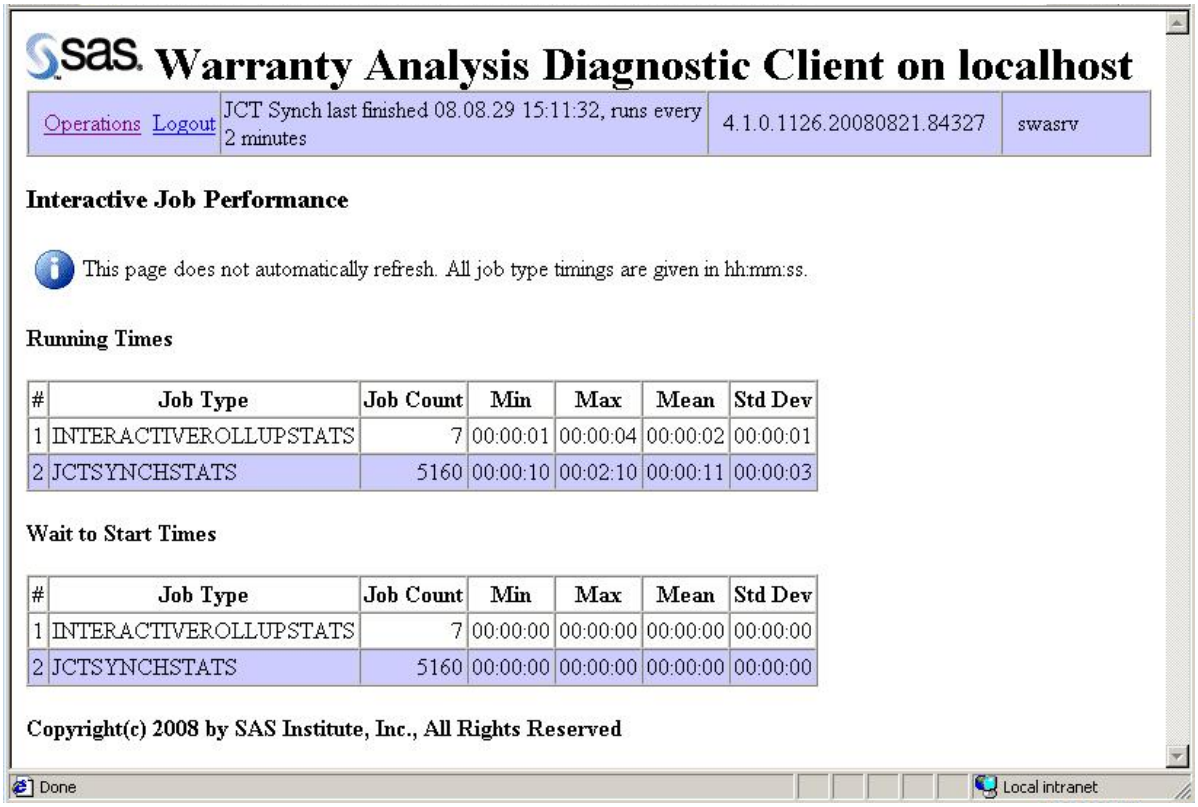


Using the Diagnostic Client to View Job Performance

Viewing Interactive Job Performance

The Interactive Job Performance view (Figure 8.3) summarizes statistics for all of the interactive jobs that have run so far. The **Running Times** section provides information about total running time, and the **Wait to Start Times** section provides information about how long jobs take to start.

Figure 8.3 The Interactive Job Performance View



sas Warranty Analysis Diagnostic Client on localhost

Operations Logout | JCT Synch last finished 08.08.29 15:11:32, runs every 2 minutes | 4.1.0.1126.20080821.84327 | swasrv

Interactive Job Performance

This page does not automatically refresh. All job type timings are given in hh:mm:ss.

Running Times

#	Job Type	Job Count	Min	Max	Mean	Std Dev
1	INTERACTIVEROLLUPSTATS	7	00:00:01	00:00:04	00:00:02	00:00:01
2	JCTSYNCHSTATS	5160	00:00:10	00:02:10	00:00:11	00:00:03

Wait to Start Times

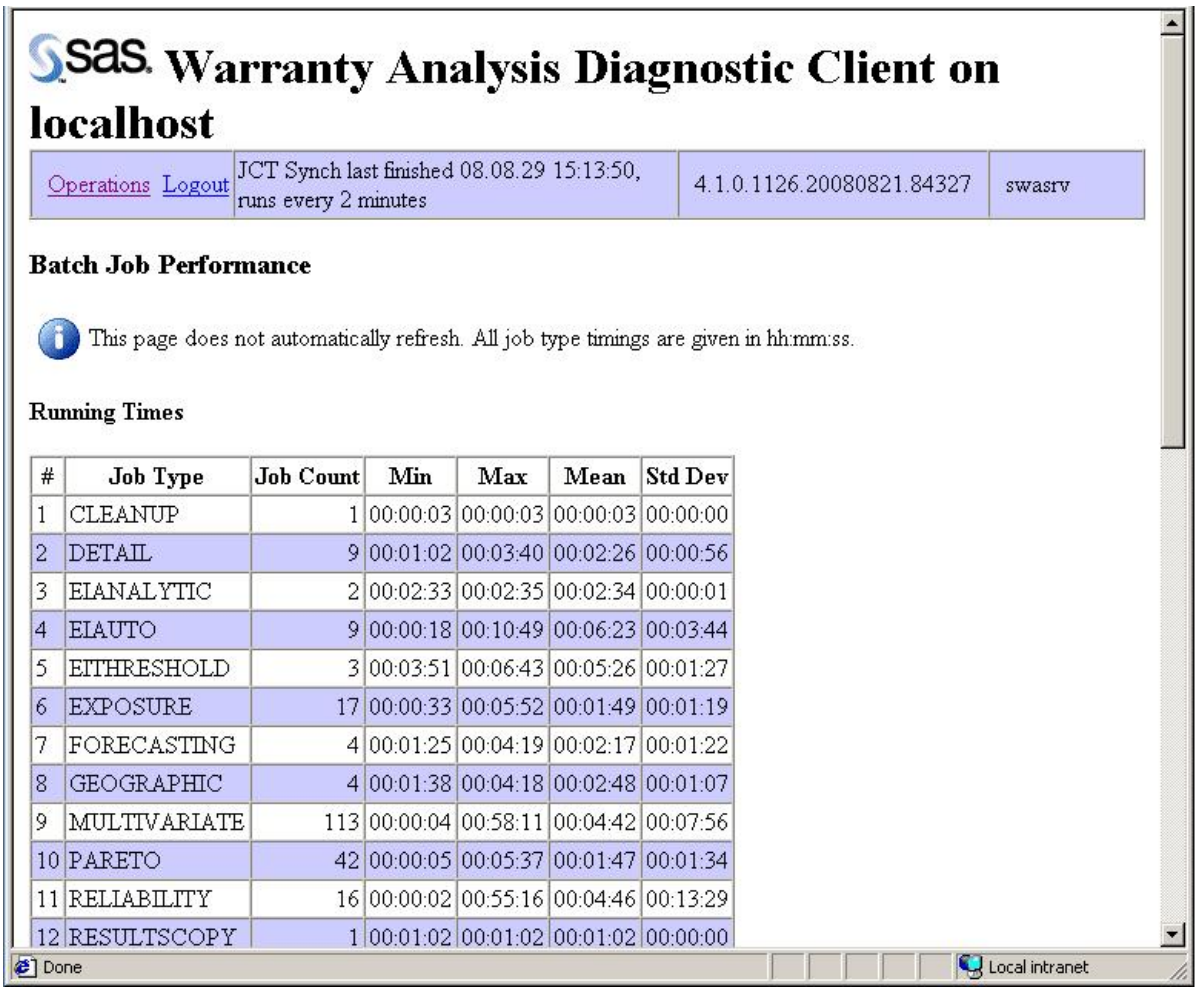
#	Job Type	Job Count	Min	Max	Mean	Std Dev
1	INTERACTIVEROLLUPSTATS	7	00:00:00	00:00:00	00:00:00	00:00:00
2	JCTSYNCHSTATS	5160	00:00:00	00:00:00	00:00:00	00:00:00

Done | Local intranet

Viewing Batch Job Performance

The Batch Job Performance view (Figure 8.4), shows a summary of all the batch jobs run so far. This view also shows a comparison view for how long it took for jobs to start up. The **Running Times** section provides information about total running time, and the **Wait to Start Times** section provides information about how long jobs take to start.

Figure 8.4 The Batch Job Performance View



sas Warranty Analysis Diagnostic Client on localhost

Operations Logout	JCT Synch last finished 08.08.29 15:13:50, runs every 2 minutes	4.1.0.1126.20080821.84327	swasrv

Batch Job Performance

This page does not automatically refresh. All job type timings are given in hh:mm:ss.

Running Times

#	Job Type	Job Count	Min	Max	Mean	Std Dev
1	CLEANUP	1	00:00:03	00:00:03	00:00:03	00:00:00
2	DETAIL	9	00:01:02	00:03:40	00:02:26	00:00:56
3	EIANALYTIC	2	00:02:33	00:02:35	00:02:34	00:00:01
4	EIAUTO	9	00:00:18	00:10:49	00:06:23	00:03:44
5	EITHRESHOLD	3	00:03:51	00:06:43	00:05:26	00:01:27
6	EXPOSURE	17	00:00:33	00:05:52	00:01:49	00:01:19
7	FORECASTING	4	00:01:25	00:04:19	00:02:17	00:01:22
8	GEOGRAPHIC	4	00:01:38	00:04:18	00:02:48	00:01:07
9	MULTIVARIATE	113	00:00:04	00:58:11	00:04:42	00:07:56
10	PARETO	42	00:00:05	00:05:37	00:01:47	00:01:34
11	RELIABILITY	16	00:00:02	00:55:16	00:04:46	00:13:29
12	RESULTSCOPY	1	00:01:02	00:01:02	00:01:02	00:00:00

Done Local intranet

Chapter 9

Working with Batch Processes

Contents

Overview of Batch Jobs

For SAS Warranty Analysis, a batch job is either of the following:

- any job that is started from the rich client but does not "freeze" the user interface while the job is running. For example, if a user selects the **Run now** option to update an analysis, the analysis starts immediately and runs in the background. The user does not have to wait for the analysis to end before he or she can perform other tasks.

  These jobs run in the background immediately when they are invoked, unlike jobs that are started using the **Run later** option, which must be scheduled for execution.

- all work that is run outside of any SAS Warranty Analysis client (for example, by using command lines or a scheduling package). Examples include emerging issues reports or public reports that are defined within the rich client. By default, these reports become batch jobs that do not run until a SAS Warranty Analysis command-line program releases them to run. This type of work usually consists of batch jobs that you administer during off-peak hours to reduce the impact on the servers that run your SAS Warranty Analysis system.

The SAS Warranty Analysis system includes the following types of batch jobs:

- **analysis, submit later**: When a user selects the **Submit Later** option to schedule an analysis for later execution, the analysis is queued for batch processing.
- **public report**: A user in the Projects workspace can save a given analysis as a report. These reports can be set to run once (**Auto Update**) or run repeatedly (**Auto Update Create New**). Both of these end up in batch processing.
- **emerging issues analytic and emerging issues threshold**: A user in the Emerging Issues area of the Administration workspace can define reports that are similar to public reports, but whose results are displayed in the Emerging Issues workspace instead of the Reports workspace. These emerging issues reports run as individual analyses just as analyses in the Projects workspace and reports in the Reports do.
- **emerging issues automated**: A user in the Administration workspace can also define automated reports. These reports differ from all other reports in that a more involved analytic process scans a potentially larger amount of data and generates potentially hundreds of reports. Like emerging issues analytic reports and emerging issues threshold reports, emerging issues automated reports can be viewed in the Emerging Issues workspace.

Each of these job types has its own requirements for running. Typically, it is best to run them separately from one another, because their run-time characteristics can vary. The following sections describe the scheduling and run-time characteristics of batch jobs in more detail.

Job Control Tables and Scheduling Integration

The SAS Warranty Analysis system was not designed with any particular scheduling package in mind. Rather, the system is designed to support any scheduling package, or none at all. If you prefer, you can drive batch processing from a command line. SAS Warranty Analysis enables you to run batch jobs in any environment by defining tables to store the batch job information and scripts to run the various types of jobs.

Job Control Tables

SAS Warranty Analysis has four tables in the PARMSL library that contain job-related information:

- **JobControl**: This table has one row for each batch job that is defined. This table stores rows for finished analysis jobs that were executed by using the **Run Now** option in the rich client. You can use the Job Queue utility in the rich client to clear out these rows before you invoke a batch run. Clearing out the rows makes it easier for you to see what remains to be run in the batch queue.
- **JobAttributes**: Each row in this table stores a single attribute of a batch job that has been defined in the JobControl table.

- **JobPerform**: This table stores measurements of how long each job runs.
- **JobSynchLock**: This table enables the SAS Warranty Analysis middle-tier processes and SAS tier processes to coordinate write access to the job control tables.

NOTE: You will not perform operations on these tables directly! This administration guide mentions them because you can see the contents of the JobControl table in the diagnostic client or by using the **Job Queue** utility in the rich client.

See Chapter 7, "Using the Diagnostic Client (Experimental)," for more information about monitoring batch processing. See "The Job Queue Utility" in the "Tools Overview" section of either the online Help or *SAS Warranty Analysis User's Guide* for more information about the job queue utility in the rich client.

Job Control Scripts

The JobControl table contains a number of columns that are involved in managing batch jobs. To make batch processing easier to manage, the system includes a few scripts that enable you to invoke batch job processing on each of the batch job types:

- **`start_OP`** (**`.bat`** or **`.sh`**) enables off-peak analyses (that is, analyses submitted by using the **Run Later** option in the rich client) to be run as a set.
- **`start_BR`** (**`.bat`** or **`.sh`**) enables batch (public) reports to be run as a set.
- **`start_EI`** (**`.bat`** or **`.sh`**) enables emerging issues (analytic, threshold, or automated) to be run as a set.

Each of these scripts is described in more detail, with examples, in the sections that follow.

Batch Processing Overview

The strategy for batch processing design varies with each site. There might be few enough jobs to run them all in one evening, or processing might need to be spread over a number of days. The SAS Consulting services personnel who install your system help you to determine what works best for you.

Regardless of how much time it takes, you should run each of the following types of batch processing separately:

- data mart refresh
- off-peak (analyses submitted by using the **Run Later** option)

- batch report (public reports)
- emerging issues analytic
- emerging issues threshold
- emerging issues automated

The job control scripts can be used to invoke each of these processes separately. The data mart refresh script, which is site-specific, is provided by your SAS Consulting services representative.

Refresh the data marts first, and then start processing of the various batch job types.

Running the Batch Scripts

The Emerging Issues Batch Script

The syntax of the **start_EI** script is as follows.

In a Windows environment:

```
<SasRootPath>\wrtyanlmva\sasmisc\batch\start_EI.bat <EIType>
<datasourceid> [<maxtimelimit> [<minutes2sleep>]]
```

In a UNIX environment:

```
<SasRootPath>/misc/wrtyanlmva/batch/start_EI.sh <EIType>
<datasourceid> [<maxtimelimit> [<minutes2sleep>]]
```

The script parameters are as follows:

- EIType (required): specifies which emerging issues analysis to be run. The value must be `'EIANALYTIC'`, `'EIAUTOMATED'`, or `'EITHRESHOLD'`.
- datasourceid (required): specifies a single existing data source ID number; typically `1` or `2`, and so on.

  NOTE: A data source ID is generally a positive integer (for example, 1, 2, and so on). The scripts **start_EI** and **start_BR** both require a data source ID specification; they are generally executed following a successful update of the data mart that corresponds to the specified data source IDs.

  - For **start_EI**, the datasourceid parameter can only be a single data source ID (for example, **start_EI EIAUTOMATED 1**).
  - For **start_BR**, the datasourceid parameter can be a single data source ID, a list of data source IDs separated by |, a combined data source ID, or a list of combined data source IDs separated by |. The specification of a data source ID list or a combined data source ID must be enclosed in double quotes. A combined data source ID is specified as a list of comma-separated data source IDs.

- maxtimelimit (optional): specifies the maximum time, in hours, during which the batch process can submit jobs to the execution queue. The default value is `999` (hours). The maxtimelimit option is not necessary for emerging issues automated runs.

  NOTE: Take care when specifying the maxtimelimit option if one of these scripts is used as a component of another master script. Once maxtimelimit is exceeded, control is given back to the master script even though analysis jobs initiated by **start_EI** or **start_BR** might still be running. If subsequent steps in the master script expect all activity initiated by **start_EI** or **start_BR** to be complete, then maxtimelimit should be set to an arbitrarily large number (for example, **maxtimelimit=999**). Since the units are hours, `999` should be ample time for completion.

- minutes2sleep (optional): specifies the time, in minutes, between job status checks on submitted jobs. The default value is `2`. If minutes2sleep is specified, then maxtimelimit must be specified. The minutes2sleep option is not necessary for emerging issues automated runs.

Examples Using the Emerging Issues Batch Script

Example 1: To run the emerging issues automated process for two data sources, include the following commands in the **start_EI** script:

```
start_EI.bat EIAUTOMATED 1
start_EI.bat EIAUTOMATED 2
```

Example 2: To run all emerging issues processes for one data source and specify maxtimelimit `7` and minutes2sleep `3`, include the following commands in the **start_EI** script:

```
start_EI.bat EIAUTOMATED 1
start_EI.bat EITHRESHOLD 1 7 3
start_EI.bat EIANALYTIC  1 7 3
```

Other Batch Scripts

The **start_BR** and **start_OP** scripts have the same syntax. The following syntax and examples are for **start_BR**, but they apply equally for **start_OP**.

The **start_BR** script executes any of the standard (that is, not emerging issues) batch analysis reports. It is invoked by providing the correct parameters to the **start_BR.sh** (or **.bat** in a Windows environment) shell script.

The syntax of the **start_BR** or **start_OP** script is as follows.

In a Windows environment:

```
<SasRootPath>\wrtyanlmva\sasmisc\batch\start_BR.bat <datasourceid>
[<logfilename> [<maxtimelimit> [<minutes2sleep>]]]
```

In a UNIX environment:

```
<SasRootPath>/misc/wrtyanlmva/batch/start_BR.sh <datasourceid>
[<logfilename> [<maxtimelimit> [<minutes2sleep>]]]
```

The script parameters are as follows:

- datasourceid (required): specifies existing data source ID number(s); typically `1` or `2`, and so on.

  NOTE: A data source ID is generally a positive integer (for example, 1, 2, and so on). The scripts **start_EI** and **start_BR** both require a data source ID specification; they are generally executed following a successful update of the data mart that corresponds to the specified data source IDs.

  - For **start_EI**, the datasourceid parameter can only be a single data source ID (for example, **start_EI EIAUTOMATED 1**).
  - For **start_BR**, the datasourceid parameter can be a single data source ID, a list of data source IDs separated by |, a combined data source ID, or a list of combined data source IDs separated by |. The specification of a data source ID list or a combined data source ID must be enclosed in double quotes. A combined data source ID is specified as a list of comma-separated data source IDs.

- logfilename (optional): specifies the file name of the log file. The path and extension (*.log*) are provided by the script. The default value is *batch*.

- maxtimelimit (optional): specifies the maximum time, in hours, during which the batch process can submit jobs to the execution queue. The default value is `6`. If maxtimelimit is specified, then logfilename must be specified.

NOTE: Take care when specifying the maxtimelimit option if one of these scripts is used as a component of another master script. Once maxtimelimit is exceeded, control is given back to the master script even though analysis jobs initiated by **start_EI** or **start_BR** might still be running. If subsequent steps in the master script expect all activity initiated by **start_EI** or **start_BR** to be complete, then maxtimelimit should be set to an arbitrarily large number (for example, **maxtimelimit=999**). Since the units are hours, 999 should be ample time for completion.

- minutes2sleep (optional): specifies the time, in minutes, between job status checks on submitted jobs. The default value is 2. If minutes2sleep is specified, then maxtimelimit must be specified.

Examples Using Other Batch Scripts

Example 1: To run batch reports for two data sources (1 and 2), include either of the following sets of commands in the **start_BR** script:

```
start_BR.bat 1}
start_BR.bat 2}
```

or

```
start_BR.bat "1|2"
```

Example 2: To run all batch reports for combined filters based on data source ID 1 and data source ID 2 and specify logfilename, maxtimelimit 7, and minutes2sleep 3, include the following command in the **start_BR** script:

```
start_BR.bat "1,2|2,1" batch_1C2 7 3
```

Example 3: To run all batch reports from either data source ID 1 or data source ID 2, include the following command in the **start_BR** script:

```
start_BR "1|2"
```

Example 3: To run all batch reports that use combined filters based on data source IDs 1 and 2, include the following command in the **start_BR** script:

```
start_BR "1,2|2,1"
```

The combined datasource ID “1,2” is not equivalent to “2,1”, so the specification “1,2|2,1” is required in order to run both.

SAS Log Files

SAS log files are created when the **start_EI** or **start_BR** scripts are executed. In a Windows environment, the logs are in the following location:

```
<SASRootPath>\wrtyanlmva\sasmisc\batch\logs
```

In a UNIX environment, the logs are in the following location:

```
<SASRootPath>/misc/wrtyanlmva/batch/logs
```

For Emerging Issues runs, the logs are named *<EIType>_<datasourceid>.log*.

For batch report runs, the logs are named *<logfilename>.log*; the default for logfilename is *batch*. If problems exist in report output, examine one of these logs first.

Chapter 10

Understanding Emerging Issues Processes

Contents

About Emerging Issues

Emerging issues processes are highly parameter-driven; therefore the implementation is site-specific. Your SAS Consulting services personnel provides site-specific documentation so that you know how to set up and administer your emerging issues functionality. See Chapter 9, "Working with Batch Processes," for instructions for invoking emerging issues batch processing.

The emerging issues functionality within SAS Warranty Analysis includes these processes:

- threshold watch list process
- automated analytical process
- analytical watch list process

These processes monitor warranty data and identify emerging issues by determining when the following have occurred in claims activity:

- upward trends that are statistically significant
- sudden increases that are above a value that you have specified as acceptable

The processes require input from users with an administrator role. Users with this profile must be members of the SAS Warranty Analysis EI Administrators group. See Chapter 5, "Security," for more information about SAS Warranty Analysis groups and roles.

Threshold Watch List

The threshold watch list process monitors values of a variable (called the "reporting variable") on a defined subset of data (specified by a data selection) in order to determine whether values of a calculated quantitative value (called the "analysis variable") surpass a specified threshold value.

Emerging issues administrators can use the rich client to define a new data selection, or they can import a data selection that another user has already defined.

A threshold watch list report definition consists of a data selection and a collection of analysis options, such as the reporting variable and a threshold value. A report definition can be defined only by users in one of these groups:

- SAS Warranty Analysis Full Administrators
- SAS Warranty Analysis Emerging Issues Administrators

Members of either of these groups are referred to as "emerging issues administrators."

These report definitions are processed in batch whenever a refresh of the data mart is completed. A report is generated (that is, an issue is flagged) from a report definition when the calculated analysis variable exceeds the specified threshold value for at least one value of the reporting variable in the subset of data that is being analyzed. The threshold output displays in the form of a Pareto chart the values of the reporting variable for which the analysis variable exceeds the threshold value.

When creating a report definition, the emerging issues administrator can specify which users are to receive e-mail notification whenever any values of the reporting variables are flagged as issues. However, any SAS Warranty Analysis user who is not specifically denied permission to view the emerging issues watch list reports can access these reports from the Emerging Issues workspace.

Automated Analytical Process and Analytical Watch List Process

The automated analytical process and the analytical watch list process are both used to monitor claims activity on a defined subset of units in order to identify significant upward shifts in claims activity. Instead of using a specified threshold value, these processes use analytical methods to compare current claim activity levels with expected future levels (based on history) and to flag those items for which the actual level is significantly greater than the expected level. The automated analytical process and the analytical watch list process are similar in their use of the same analytic methods. The two processes differ in how the defined subset of units that are being monitored is derived and how parameters that affect the processing (such as the sensitivity of the statistical tests performed) are defined. Additional differences are noted in subsequent sections of this document.

For these processes, the variable and the subset of product information that is to be monitored are defined by individual emerging issues administrators.

For the automated analytical process, the claim variable that is to be monitored is called the "defined entity," and the rule (or variable hierarchy) that is used to group product units is called the

"defined hierarchy." The defined hierarchy defines the product information subsets that are to be monitored. The automated process monitors the values of the defined entity by the values of the defined hierarchy. A defined entity, defined hierarchy, and additional processing parameters make up an automated analytical run group. An automated analytical run group is the set of analysis options that are used to create the emerging issues reports.

For the analytical watch list process, the subset of product information that is to be monitored is defined by a data selection. Emerging issues administrators can define a data selection by using the data selection function in the application, or they can import a data selection that another user of the application has already defined.

The emerging issues administrator also selects the variable that is to be monitored (called the "reporting variable") and additional processing options such as the sensitivity of hypothesis tests and number of periods to monitor. Additional parameters that affect how the process runs are also defined in the application metadata. The reporting variable is similar to the defined entity in the automated process, but the reporting variable can include product attributes in addition to claim attributes, whereas the defined entity can only be a claim attribute. Each report definition consists of a data selection, a reporting variable, and other analysis options. The report definition for the analytical watch list process is similar to the run group for the automated analytical process.

Report definitions for the analytic watch list process are processed in batch on a scheduled basis whenever a refresh of the data mart is completed. A report is generated (that is, an issue is flagged) from a report definition when the process detects a significant upward shift in claims activity for any value of the reporting variable that is being analyzed.

When creating a report definition for the analytical watch list, the emerging issues administrator can specify users to receive e-mail notification whenever any values of the reporting variables are flagged as issues. However, any SAS Warranty Analysis user who is not specifically denied permission to view the emerging issues watch list reports can access these reports from the Emerging Issues workspace.

In addition to the analysis options settings, the automated analytical and analytical watch list processes also incorporate business rules that are designed to exclude from monitoring those items that do not have adequate sample sizes and data to support this type of analysis. The business rules are created in the application metadata and are the same for both processes.

Two statistical analysis methods are applied in the analytical watch list and automated analytical processes to identify upward shifts in claims activities:

- production period analysis, a method proposed by Wu and Meeker (2002). The analysis monitors claims activity and the sample size at risk relative to particular production periods for different time-in-service periods (for example, 1 month in service, 2 months in service, and so on). Given a particular production period and in-service period combination, the analysis compares the actual claim rate to the expected claim rate and flags an issue when the actual claim level is significantly greater than the expected level. The expected level of claims activity for a calendar period is based on the population at risk within that calendar period, historic claim rates, seasonality factors (automated analytical process only), and the specified false alarm rate. The output includes a matrix chart that identifies the flagged periods in red.

- claim period analysis, also called the Model A method. The claim period analysis monitors

claim counts across calendar periods and flags an issue when the actual claim count that occurs in a calendar period significantly exceeds what is expected to occur for that calendar period. The expected level of claims activity for a calendar period is based on the population at risk within that calendar period, historic claim rates, seasonality factors (analytic automated process only), and the specified false alarm rate. The output from the claim period analysis is a plot of claim count (y-axis) versus calendar periods (x-axis). An issue is flagged if the actual claim count is greater than or equal to the critical value for any of the calendar periods that are being investigated. The chart shows three lines: actual claim count, expected claim count, and calculated critical value.

Additional Reading

Please refer to the site-specific documentation that your SAS Consulting services representative provides for more specific information about the parameter-driven configuration that is required for SAS Warranty Analysis emerging issues.

Reference

Wu, H. and Meeker, W. (2002), "Early Detection of Reliability Problems Using Information from Warranty Databases," *Technometrics,* 79, 120–133.

Chapter 11

Configuring System Options

Contents

Overview of Configuring System Options

This chapter explains how to change the configuration of various system options that are read in by the SAS Warranty Analysis middle tier and used throughout the system. All of these configuration options are specified in the *app.config* file, which is the SAS Warranty Analysis configuration file within the SAS Analytics Platform extensions area.

In a Windows environment, you can find the *app.config* file in a directory such as the following:

<SASInstall>\SAS\SASAPCore\apps\SASWarrantyAnalysis\app.config

In a UNIX environment, you can find the *app.config* file in a directory such as the following:

<SASInstall>/SAS/SASAPCore/apps/SASWarrantyAnalysis/app.config.

CAUTION: The *app.config* file is a Java properties file. Therefore, it must always be an ASCII file. If you need to enter property values in languages other than English, you must use the Unicode representation of those values. The *app.config* file cannot be saved in any other file format except ASCII. If the format of this file is not ASCII, then the system will not function normally.

NOTE: The default values listed in the following sections refer to the values that the system provides if the property is not included in the *app.config* file.

Report Generation Configuration Parameters

The report generation configuration parameters control how the SAS Warranty Analysis system generates and formats reports.

report.autoUpdateCounter

Each report that a user creates uses processing time in batch. The SAS Warranty Analysis system is designed to minimize batch overhead; so a given report stops refreshing after a number of iterations unless a user takes action to keep the report in play. The system assigns this configuration parameter value as the report's refresh counter. The system decrements the counter each time the report runs, and it does not run any more when the counter reaches zero. The user must edit the report to reset the counter.

NOTE: The counter is decremented either by invoking the report interactively from the rich client or by batch processing when the data mart is refreshed.

The default value of report.autoUpdateCounter is `5`.

lookupTablePageSize

The SAS Warranty Analysis system does not show large amounts of data in one scrollable screen. This parameter controls the maximum number of rows that a lookup table displays per page.

The default value of lookupTablePageSize is `300`.

maxPagedTablePagesForPdfExport

The SAS Warranty Analysis system does not print out all the rows of a large table into a PDF document. If a user wants to see all the rows, then the client program's paging table views can be used. This parameter controls the number of pages that a table outputs before the PDF generation process truncates the table.

The default value of maxPagedTablePagesForPdfExport is `2`.

pdfgen.default.fontSize

SAS Warranty Analysis provides control over the point size of all text in a PDF report with this parameter.

The default value of pdfgen.default.fontSize is `12`.

pdfgen.default.fontName

SAS Warranty Analysis provides control over the font to be used in a PDF report with this parameter.

NOTE: SAS Warranty Analysis generates PDF content on the client machine when running the rich client, but it generates PDF content on the middle tier machine when running the web client. This parameter sets the font name for both machines unless the middle tier configuration parameter is set differently by the pdfgen.default.midTier.fontName parameter.

The default value of pdfgen.default.fontSize is `'Arial Unicode MS'`.

pdfgen.default.midTier.fontName

SAS Warranty Analysis enables the PDF content that is invoked by the Web client to use a different font than PDF content that is invoked by the rich client. This parameter must be set when the middle tier machine either has a different set of fonts installed or is a different platform entirely from the client machines (for example, the middle tier is running a UNIX variant and the clients are running on Windows platforms). In these situations, PDF generation from the Web fails unless this parameter is set properly.

The value of this parameter defaults to the value specified for the pdfgen.default.fontName parameter.

Sharing Groups

SAS Warranty Analysis uses SAS Metadata Server Persons and Groups as the means to share warranty content with other users. By default, the creating user and all members of the SAS Warranty Analysis Full Administrators group can read and modify a given analytical definition (for example, **Public Report** or **Data Selection**). Owners can then grant read access to others by opening the rich client UI, invoking the share dialog box, and selecting **Persons and Groups**.

However, you most likely do not want the SAS Warranty Analysis content-sharing UI to show every person and group that is available in the SAS Metadata Server. The intention is for the system administrator to create a set of warranty business groups and show only those. The *app.config* mechanism for this is to use the sequential property base name sharing.identityFilter.name and create a list of person and group names to subtract from all those available.

An example of this content follows. The numbers need to start at 1 and increase sequentially.

CAUTION: Any gap in the sequence causes the rest of the entries to be ignored.

```
# Identities to filter out of the sharing user interfaces
sharing.identityFilter.name1=SASUsers
sharing.identityFilter.name2=Public
sharing.identityFilter.name3=SAS System Services
sharing.identityFilter.name4=SAS General Servers
sharing.identityFilter.name5=SAS Administrator
sharing.identityFilter.name6=SAS Demo User
sharing.identityFilter.name7=SAS Warranty Analysis Server User
sharing.identityFilter.name8=SAS Web Administrator
sharing.identityFilter.name9=SAS Trusted User
sharing.identityFilter.name10=Performance Testing
sharing.identityFilter.name11=Portal Admins
sharing.identityFilter.name12=Portal Demos
```

NOTE: The five SAS Warranty Analysis system groups might appear in *app.config* by default. These groups can be removed, because the system automatically excludes them from showing up regardless of the content of *app.config*. It does not hurt to have them in the list, but they are not necessary. Just make sure that the group names defined in the Group Names section match the names in the SAS Metadata Server.

Initialization Parameters

The SAS Analytics Platform and SAS Warranty Analysis application initialization parameters are set during the installation and configuration process, and most do not need to be changed. They are documented in the following sections in case credentials or passwords are updated or your site installation requires customization.

NOTE: When you see the value configured, it means the code does not supply a default value and that the proper values are inserted in the *app.config* file when the SAS configuration wizard runs.

metaserver.hostname

This parameter specifies the host name for the machine that hosts the SAS Metadata Server.

There is no default value. The proper values are automatically inserted in the *app.config* file when the SAS configuration wizard runs.

metaserver.hostport

This parameter specifies the host port for the machine that hosts the SAS Metadata Server.

There is no default value. The proper values are automatically inserted in the *app.config* file when the SAS configuration wizard runs.

metaserver.trusteduserid

This parameter specifies the user logon ID for the credential that is used to initialize the SAS Warranty Analysis extensions within the SAS Analytics Platform application server.

There is no default value. The proper values are automatically inserted in the *app.config* file when the SAS configuration wizard runs.

metaserver.trusteduserpw

This parameter specifies the user logon password for the credential that is used to initialize the SAS Warranty Analysis extensions within the SAS Analytics Platform application server.

There is no default value. The proper values are automatically inserted in the *app.config* file when the SAS configuration wizard runs.

metaserver.deploymentname

This parameter specifies the name of the SAS Business Intelligence Platform services deployment that the SAS Analytics Platform and SAS Warranty Analysis use to run all SAS server work.

There is no default value. The proper values are automatically inserted in the *app.config* file when the SAS configuration wizard runs.

metaserver.authdomainname

This parameter specifies the name of the SAS Metadata Server authentication domain to use for all metadata access.

The default value of metaserver.authdomainname is `'DefaultAuth'`.

metaserver.repositoryname

This parameter specifies the name of the SAS Metadata Server repository to use for all metadata access.

The default value of metaserver.repositoryname is `'Foundation'`.

apserver.host

This parameter specifies the name of the server machine where the SAS Analytics Platform application server is running.

There is no default value. The proper values are automatically inserted in the *app.config* file when the SAS configuration wizard runs.

System Parameters

The system parameters control a set of miscellaneous (and largely unrelated) behaviors in the SAS Warranty Analysis system.

security.loginIdsCaseSensitive

By default, SAS Warranty Analysis treats logon IDs as case-insensitive. The default works fine in a Windows environment; however, in a UNIX environment, you should change this value to `'true'` because UNIX authentication respects the case of the login ID.

The default value of security.loginIdsCaseSensitive is `'false'`.

metaserver.trusteduserlocale

This parameter specifies the locale value that is used for batch work if no other locale is specified for any jobs that are run as the SAS Warranty Analysis initialization user credential (see "metaserver.trusteduserid" on page 78). This parameter is seldom used for batch reports, because they take on the locale of the analysis.

The default value of metaserver.trusteduserlocale is `'en_US'`.

sas.wsps.driver.program

This parameter specifies the path in the SAS Metadata Server to the SAS Warranty Analysis stored process that is used to kick off all SAS tier work that uses the workspace server.

The default value is set during installation and configuration. The only reason the default value should be changed is if site customization requires a different version of the stored process to be run. In this case, a SAS Consulting services representative changes this value.

sas.stps.driver.program

This parameter specifies the path in the SAS Metadata Server to the SAS Warranty Analysis stored process that is used to kick off all SAS tier work that uses the SAS Stored Process Server.

The default value is set during installation and configuration. The only reason this should be changed is if site customization requires a different version of the stored process to be run. In this case, a SAS Consulting services representative changes this value.

user.login.history.threshold.limit

SAS Warranty Analysis keeps track of logon attempts. This parameter specifies the number of records that need to be maintained in the logon history for each user.

The default value of user.login.history.threshold.limit is `5`.

ETL.messageCounter

This parameter specifes the number of refreshes that show in messages in the clients.

The default value of ETL.messageCounter is `3`.

The SAS Warranty Analysis middle tier monitors the system to detect when a data mart refresh event happens.

Job Control Table Synchronization Parameters

The job control table (JCT) synchronization is a process that the SAS Warranty Analysis middle tier runs every few minutes (see "queue.jobControlTableSynchIntervalMinutes" on page 82). The configuration parameters described in this section show how to control the many tasks performed by the JCT synchronization.

analysis.rescueOrphanedMinutes

If the SAS Warranty Analysis middle tier machine (or services) is stopped and restarted while SAS jobs is in the queue to be run, then it is possible for a SAS job to become "orphaned." This configuration option affects analysis jobs that never actually started executing.

A job becomes orphaned if the following sequence of events occurs:

1. The job is released to run.
2. The analysis is set to **Updating**.
3. The middle tier is stopped and restarted.

When a job becomes orphaned, the analysis (and its JCT record) stay in the **Updating** state because nothing happens to change the state. However, it is impossible to discriminate between an orphaned job and normal analysis execution, where there can be a delay between release and starting execution. After the delay defined by this parameter, measured against the job start time, the JCT synchronization process resets the jobs back to the **Ready** state.

The default value of analysis.rescueOrphanedMinutes is `5`.

queue.jobControlTableSynchIntervalMinutes

This parameter dictates how many minutes pass between runs of the JCT synchronization process.

The default value of queue.jobControlTableSynchIntervalMinutes is `5`.

queue.housekeepingIntervalMinutes

This parameter dictates how many minutes pass between runs of the general housekeeping process. The main purpose of this process is to delete old log files.

The default value of queue.housekeepingIntervalMinutes is `60`.

queue.finishedJobExpirationIntervalMinutes

This parameter dictates how many minutes a job can be left in the **Finished** state before being deleted.

The default value of queue.finishedJobExpirationIntervalMinutes is `1440`.

queue.statisticsSnapshotCountMax

This parameter dictates how many SAS Warranty Analysis queuing snapshots can be cached in memory. You can see the queuing snapshots in the diagnostic client.

The default value of queue.statisticsSnapshotCountMax is `200`.

Logging Parameters

queuing.stats

This parameter causes all statistics related to SAS Warranty Analysis queue management and batch processing to be printed to the SAS Analytics Platform console output. The Java stream used is System.err.

The default value of queuing.stats is `'false'`.

logging.sql

This parameter causes every SQL statement run by the system to be written to the SAS Analytics Platform console output. These SQL statements are not currently written to the job log files.

The default value of logging.sql is `'false'`.

logging.xml

This parameter causes the analysis options and analysis output XML file to be written to the job log. This XML information is not currently written to the SAS Analytics Platform console output.

The default value of logging.xml is `'false'`.

logging.jobLogExpirationDaysOld

This parameter controls how many days old a normal log file must be to be automatically deleted.

The default value of logging.jobLogExpirationDaysOld is `5`.

logging.jobLogSystemExpirationHoursOld

This parameter controls how many hours old a system log file must be to be automatically deleted.

The default value of logging.jobLogSystemExpirationHoursOld is `4`.

logging.stripHtml

This parameter controls whether the HTML syntax is stripped from the job log files. Keeping the HTML intact enhances the log content with navigation links and highlighting of errors in red, warnings in orange, and other important information in blue.

The default value of logging.stripHtml is `'N'`.

Numeric Formatting Options

This set of configuration parameters controls the formatting and rounding of numbers in the SAS Warranty Analysis system.

default.decimal.fractiondigits

This parameter specifies the default fraction digits that are used in rounding off the decimal values that are stored in SAS datasets. This configuration is used mainly for precision searching of formatted values (in the SAS ROUNDE function) against the raw data that is stored in SAS data sets.

The default value of default.decimal.fractiondigits is `0.001`.

default.currency.fractiondigits

This parameter specifies how many decimal fraction digits are used to round off currency values stored in SAS data sets. This configuration is used mainly for precision searching of formatted values (in the SAS ROUNDE function) against the raw data that is stored in SAS data sets.

The default value of default.currency.fractiondigits is `0.01`.

default.percent.fractiondigits

This parameter specifies how many decimal fraction digits are used to round off percentage values stored in SAS data sets. This configuration is used mainly for precision searching of formatted values (in the SAS ROUNDE function) against the raw data that is stored in SAS data sets.

The default value of default.percent.fractiondigits is `'(String) null'`.

default.integer.fractiondigits

This parameter specifies how many decimal fraction digits are used to round off integer values stored as decimals in SAS data sets with integer formats. This configuration is used mainly for precision searching of formatted values (in the SAS ROUNDE function) against the raw data that is stored in SAS data sets.

The default value of default.integer.fractiondigits is `'(String) null'`.

Role Names

The SAS Warranty Analysis system uses SAS Metadata Server role names in its code. The names to be used are defined by the following configuration properties.

CAUTION: Keep these names synchronized with the actual names used in the SAS Management Console, or the system will not work properly.

metaPersist.infoConsumerRoleName

This parameter specifies a role name to match the corresponding role in the SAS Management Console for the Warranty Information Consumer role.

The default value of metaPersist.infoConsumerRoleName is `'Warranty Information Consumer'`.

metaPersist.dataAnalystRoleName

This parameter specifies a role name to match the corresponding role in the SAS Management Console for the Warranty Data Analyst role.

The default value of metaPersist.dataAnalystRoleName is `'Warranty Data Analyst'`.

metaPersist.dataAdministratorRoleName

This parameter specifies a role name to match the corresponding role in the SAS Management Console for the Warranty Data Administrator role.

The default value of metaPersist.dataAdministratorRoleName is `'Warranty Data Administrator'`.

metaPersist.batchAdministratorRoleName

This parameter specifies a role name to match the corresponding role in the SAS Management Console for the Warranty EI Administrator role.

The default value of metaPersist.batchAdministratorRoleName is `'Warranty EI Administrator'`.

metaPersist.systemAdministratorRoleName

This parameter specifies a role name to match the corresponding role in the SAS Management Console for the Warranty System Administrator role.

The default value of metaPersist.systemAdministratorRoleName is `'Warranty System Administrator'`.

metaPersist.denyEIAccessRoleName

This parameter specifies a role name to match the corresponding role in the SAS Management Console for the Warranty Deny EI Access role.

The default value of metaPersist.denyEIAccessRoleName is `'Warranty Deny EI Access'`.

Group Names

The SAS Warranty Analysis system uses SAS Metadata Server group names in its code. The names to be used are defined by the following configuration properties.

CAUTION: Keep these names synchronized with the actual names used in the SAS Management Console, or the system will not work properly.

See Chapter 5, "Security," for more information.

metaPersist.viewUserGroupName

This parameter specifies a group name to match the corresponding group in the SAS Management Console for the SAS Warranty Analysis View Users group.

The default value of metaPersist.viewUserGroupName is `'SAS Warranty Analysis View Users'`.

metaPersist.restrictedUsersGroupName

This parameter specifies a group name to match the corresponding group in the SAS Management Console for the SAS Warranty Analysis Restricted Users group.

The default value of metaPersist.restrictedUsersGroupName is `'SAS Warranty Analysis Restricted Users'`.

metaPersist.normalUsersGroupName

This parameter specifies a group name to match the corresponding group in the SAS Management Console for the SAS Warranty Analysis Normal Users group.

The default value of metaPersist.normalUsersGroupName is `'SAS Warranty Analysis Normal Users'`.

metaPersist.eiAdminGroupName

This parameter specifies a group name to match the corresponding group in the SAS Management Console for the SAS Warranty Analysis EI Administrators group.

The default value of metaPersist.eiAdminGroupName is `'SAS Warranty Analysis EI Administrators'`.

metaPersist.adminGroupName

This parameter specifies a group name to match the corresponding group in the SAS Management Console for the SAS Warranty Analysis Full Administrators group.

The default value of metaPersist.adminGroupName is `'SAS Warranty Analysis Full Administrators'`.

metaPersist.denyEIAccessGroupName

This parameter specifies a group name to match the corresponding group in the SAS Management Console for the SAS Warranty Analysis Deny EI Access group.

The default value of metaPersist.denyEIAccessGroupName is `'SAS Warranty Analysis Deny EI Access'`.

Storage Location Names

The SAS Warranty Analysis system makes use of SAS Metadata Server software component and folder names in its code. The names to be used are defined by the following configuration properties.

CAUTION: Keep these names synchronized with the actual names used in the SAS Management Console, or the system will not work properly.

See Chapter 5, "Security," for more information.

warranty.softwareComponentName

This parameter specifies a software component name to match the corresponding software component in the SAS Management Console.

The default value of warranty.softwareComponentName is `'SAS Warranty Analysis'`.

warranty.projectTreeName

This parameter specifies a folder name to match the corresponding folder in the SAS Management Console for the folder where SAS Warranty Analysis project definition metadata objects are stored.

The default value of warranty.projectTreeName is `'Projects'`.

warranty.dataSelectionTreeName

This parameter specifies the folder name to match the corresponding folder in the SAS Management Console for the folder where SAS Warranty Analysis data selection definition metadata objects are stored.

The default value of warranty.dataSelectionTreeName is `'Data Selections'`.

warranty.analyticReportTreeName

This parameter specifies the folder name to match the corresponding folder in the SAS Management Console for the folder where SAS Warranty Analysis public report definition metadata objects are stored.

The default value of warranty.analyticReportTreeName is `'Analytic Reports'`.

warranty.categoryTreeName

This parameter specifies a folder name to match the corresponding folder in the SAS Management Console for the folder where SAS Warranty Analysis category definition metadata objects are stored.

The default value of warranty.categoryTreeName is `'Categories'`.

warranty.applicationSASCodeTreeName

This parameter is not used in SAS Warranty Analysis 4.1. In future releases, it is intended to house all stored procedure definitions.

The default value of warranty.applicationSASCodeTreeName is `'Application SAS Code'`.

Chapter 12

Working With Technical Support

Contents

Overview

SAS Technical Support can be reached in any of the following ways:

Phone: 919-677-8008
E-mail: support@sas.com
Internet: **http://support.sas.com**

When you contact SAS Technical Support, the help desk representative asks for the following information:

- your SAS site number
- the operating system on the machine where the problem occurred
- the version of the SAS software that you are using

The following sections help you to collect additional information to assist your SAS Technical Support representative in resolving your problem.

Record the Problem Context

First, create an accurate description of the problem based on information from the SAS Warranty Analysis user who is experiencing a problem. The following questions are a good starting point:

- What workspace was the user working in?

- What role was the user a member of?
- If applicable, what task was the user attempting when the problem occurred (for example, what analysis was submitted or what report was requested)?

Provide instructions for duplicating the problem scenario in as much detail as possible. For example:

1. Log on to SAS Warranty Analysis as <userid>.
2. Click **Resources**.
3. Click **My Reports**.
4. View the report named <name>.
5. Include a description of the data selection, if applicable, and its name.
6. Provide screen shots that illustrate the problem.
7. Provide a description of any recent changes or disruptions in the environment, such as:
   - software upgrades
   - database modifications
   - power interruption
   - Internet service interruptions

General System Questions

Here are some specific questions that your SAS Technical Support contact might ask you. Be prepared to answer them as part of a general diagnostic process.

- Is this issue related to a specific user?
- Is the SAS Warranty Analysis system administrator able to log on to the application?
- Is the error related to the user interface or to batch reports?
- Are the SAS Warranty Analysis services running in the middle tier via the SAS Analytics Platform?
- Is the SAS Information Delivery Portal running in the middle tier?
- Is the remote services application running in the middle tier?
- Are the SAS Business Intelligence services running in the SAS tier?
- What are the versions of the operating system and SAS Warranty Analysis?

- What is the error message?
- Can the user reproduce the reported issue?
- Has this error occurred previously?
- Is there any other pertinent information that might be helpful?

Gathering Information from the Log Files

Most of the time, a job log file is available to help diagnose problems. The following kinds of job log files are available in SAS Warranty Analysis:

- Analysis execution logs are associated with the analysis ID assigned to them.
- Non-analysis execution logs are identified by the log file name, which suggests the operation that was in progress when an error occurred.

See "Anatomy of a Job Log File Listing" on page 49 for instructions for looking up log files. Use the user name, analysis ID, and the date and time of the problem to narrow down which log file to collect if you are using the diagnostic client. The diagnostic client can subset logs by user and error. For example, if a user reports an error for an analysis ID, click the **User Error Logs** link in the diagnostic client to display a list of the logs with errors, grouped by user. Click the link for the user in the **Error Log Files by User ID** section and locate the log by using the analysis ID and the time when the problem occurred.

You can use the browser's **View Source** option to get the text of the log and save it to a file. Then you can e-mail it to SAS Technical Support, where it will be placed as an attachment to the trouble ticket.

Alternatively, if you get the log file from the file system, you need to go to the *<analytics platform location>/apps/SASWarrantyAnalysis/logs* directory and find the file in the user's directory.

Appendix A

Troubleshooting

Contents

The Analytics Platform Server Fails to Start Properly

Problem: The SAS Analytics Platform console output shows that exceptions occurred during start-up.

Solution: Make sure that the user ID and password that are specified in the *app.config* file to initialize the SAS Warranty Analysis queuing subsystem are correct. Look at the errors that are displayed, and contact SAS Technical Support for assistance.

An Analysis Fails to Execute

An analysis might fail to execute for several possible reasons. Locate the logs for the analysis run that fails, and provide information from the logs to SAS Technical Support for assistance.

The Rich Client Fails to Open from the SAS Warranty Analysis Landing Page

Problem: Pop-up blockers prevent the rich client from starting from the SAS Warranty Analysis landing page. If you click **Launch SAS Warranty Analysis Client** on the SAS Warranty Analysis portal landing page and a new window opens, but the rich client does not open, it is likely that a pop-up blocker in the browser is preventing the rich client from opening.

Solution: To turn off the pop-up blocker in Internet Explorer, select **Tools►Pop-up Blocker►Turn Off Pop-up Blocker**.

If third-party applications such as the Google Toolbar or the Yahoo Toolbar are installed, ensure that their pop-up blockers are turned off as well. All pop-up blockers must be turned off!

Deletion of a Data Selection Fails

Problem: A user attempts to delete a data selection definition, and the operation fails. The log indicates that a permissions error at the operating system level prevented the filtered subset from being deleted.

Solution: Make sure to explicitly give full rights to the SASSRV operating system credential for the root directories used to store user-filtered data sets (the USERFDL library locations). The paths are defined in the SERVERPATH column of the PARMSL.USERATTRIBUTES table.

Those full rights are required so that any filtered subsets can be deleted by SASSRV during the analysis results deletion process.

A User Is Granted Access to SAS Warranty Analysis, but the System Malfunctions

Problem: You grant access to the system for a user, but the system malfunctions frequently and does not allow progress for that user.

Solution: Make sure the user was granted access to a group and not directly to a role. See Chapter 5, "Security," for more information.

Any User Logging into System Receives the "System Experienced Fatal Error" Message

Problem: If the "System Experienced Fatal Error" message displays, then it is likely that the SAS Analytics Platform failed to start properly, especially the SAS Warranty Analysis extensions to the SAS Analytics Platform. The most common cause is that the user credential specified in the SAS Warranty Analysis *app.config* file could not log on successfully because it was not set up correctly during installtion.

Solution: Look in the Analytics Platform log, and find the first error listed. The SAS Analytics Platform log location is site-specific. Your SAS Consulting services contact will tell you where to find it. The first error is the best information related to the problem. There can be other causes as well. Work with SAS Technical Support to extract all required log files to diagnose the problem.

The User Interface Is Unresponsive or an Interactive Job Does Not Progress

Problem: Interactive jobs are meant to run fairly quickly; therefore, it is unusual for an interactive job (as seen in the diagnostic client's Interactive Jobs view) to last more than thirty seconds or so.

Solution: If you have a user whose client appears to be unresponsive and the diagnostic client indicates that one of this user's jobs is not progressing, then you must restart the middle tier.

A Batch Job Does Not Progress

Problem: A batch job appears not to be progressing.

Solution: First, make sure that the job is actually not progressing, because it could be an analytical job file that is processing a large amount of data. Depending on site-specific factors, analysis jobs can take from a few minutes up to half an hour, and a few can take even longer.

If a job does not finish after half an hour, consult the batch job performance statistics in the diagnostic client and note the average and maximum job times for the particular analytic being used. If the statistics indicate that a long run time is common for this analysis, then the job is probably running as expected. Emerging issues automated runs can also take a while to complete.

If all else fails and it appears that the batch job really is failing to progress, you might need to restart the servers. Restart the servers only if you have no other recourse.

JobSynchLock Is Locked

Problem: An error message in the job logs indicates that the JobSynchLock is locked. If the JobSynchLock table is locked, the system cannot synchronize the job control tables.

Solution: Complete the following steps:

1. Stop the middle tier.
2. Stop the SAS tier.
3. Restart the middle tier.
4. Restart the SAS tier.

The lock is automatically cleared by the restart.

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.