



THE  
POWER  
TO KNOW.

# **SAS<sup>®</sup> Warranty Analysis 4.3**

## **Administrator's Guide**



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *SAS® Warranty Analysis 4.3: User's Guide*. Cary, NC: SAS Institute Inc.

**SAS® Warranty Analysis 4.3: User's Guide**

Copyright © 2011, SAS Institute Inc., Cary, NC, USA

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

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

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

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

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

1st electronic book, February 2011

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

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

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

---

# Contents

<b>Chapter 1 • About SAS Warranty Analysis</b>	<b>1</b>
Introduction	1
Components of SAS Warranty Analysis	2
Client Applications and Sessions	2
<b>Chapter 2 • System Architecture</b>	<b>7</b>
Introduction	7
The Client Tier	7
The Middle Tier	8
The Server Tier	8
SAS Tier	9
Warranty Data Tier	9
Batch Processing	10
<b>Chapter 3 • Finding Scripts, Files, and Other System Components</b>	<b>11</b>
Overview	11
Finding Installation and Configuration Files	11
Finding SAS Server Tier Components	12
<b>Chapter 4 • Working with Domain Objects</b>	<b>15</b>
Overview of Domain Objects	15
Domain Object Types and Characteristics	16
Domain Object and Results Ownership	17
When Someone Leaves the Company	20
When Available Disk Space Is Low	21
When Data Losses Occur	22
<b>Chapter 5 • Security</b>	<b>23</b>
Overview	23
SAS Business Intelligence Platform Groups and Roles	24
Protected Resources	25
Security Model	26
Creating a User Account	28
Granting Access through Group Membership	28
Denying Access to Emerging Issues	29
Creating New Groups for Domain Object Sharing	29
Filtering Out Groups from Domain Object Sharing	29
Restricting Access to Data By Using Column-Level Security	30
<b>Chapter 6 • Starting and Stopping the System</b>	<b>35</b>
Overview	35
Tier Dependencies	35
Metadata Server Start Scripts	36
Object Spawner Scripts	36
Share Server Scripts	36
Application Server Scripts	36
Service Dependencies	37
<b>Chapter 7 • The Diagnostic Client (Experimental)</b>	<b>39</b>
Overview	39
Using the Diagnostic Client	40

Start the Diagnostic Client . . . . .	40
Main Page . . . . .	40
Navigation Bar . . . . .	41
Diagnostic Summary Area . . . . .	42
Interactive Performance Area . . . . .	42
Batch Performance Area . . . . .	43
General Performance Area . . . . .	43
Metadata Management Area . . . . .	43
Chained Report Listings Area . . . . .	44
Log File Listings Area . . . . .	44
Anatomy of a Job Log File Listing . . . . .	45
User Management/Miscellaneous Tools Area . . . . .	47
Configuration Related Area . . . . .	48
<b>Chapter 8 • Working with Jobs, Tasks, and Queues . . . . .</b>	<b>49</b>
Overview . . . . .	49
About Jobs, Tasks, and Queues . . . . .	49
Managing Interactive and Batch Jobs . . . . .	50
Managing Queues in the Rich Client . . . . .	50
Using the Diagnostic Client to View Queues (Experimental) . . . . .	50
Using the Diagnostic Client to View Job Performance . . . . .	53
<b>Chapter 9 • Working with Batch Processes . . . . .</b>	<b>55</b>
Overview of Batch Jobs . . . . .	55
Job Control Tables and Scheduling Integration . . . . .	56
Batch Processing Overview . . . . .	57
Running the Batch Scripts . . . . .	58
<b>Chapter 10 • Understanding Emerging Issues Processes . . . . .</b>	<b>63</b>
About Emerging Issues . . . . .	63
Additional Reading . . . . .	65
Reference . . . . .	66
<b>Chapter 11 • Using SAS Warranty Analysis Business Rules with SAS Enterprise Guide . . . . .</b>	<b>67</b>
Overview . . . . .	67
Setting User Access Permissions . . . . .	67
<b>Chapter 12 • Configuring System Options in SAS Management Console . . . . .</b>	<b>69</b>
Overview . . . . .	71
Sharing Groups . . . . .	71
Analysis . . . . .	72
Filter Subset Management . . . . .	72
Identities > Groups . . . . .	72
Identities > Roles . . . . .	73
Information Map Security . . . . .	74
Logging . . . . .	75
Mid Tier Caching . . . . .	76
Queuing > Application Server Settings > Interactive Work . . . . .	76
Queuing > Application Server Settings > Batch Work . . . . .	77
Queuing > Application Server Settings > Generic Task Queue . . . . .	78
Queuing > Application Server Settings > Information Service Task Queue . . . . .	79
Queuing > Logging . . . . .	79
Queuing > Housekeeping . . . . .	79
Queuing > Job Control Table Synchronization . . . . .	80
Queuing > SAS Share Server . . . . .	80
Queuing > Teradata Server . . . . .	81

Table Display . . . . .	81
Other Property Values That You Can Override . . . . .	82
<b>Chapter 13 • Working With Technical Support . . . . .</b>	<b>103</b>
Overview . . . . .	103
Record the Problem Context . . . . .	103
General System Questions . . . . .	104
Gathering Information from the Log Files . . . . .	104
<b>Appendix 1 • Troubleshooting . . . . .</b>	<b>107</b>
An Analysis Fails to Execute . . . . .	107
The Rich Client Fails to Open from the SAS Warranty Analysis Landing Page . . . . .	107
Deletion of a Data Selection Fails . . . . .	108
A User Is Granted Access to SAS Warranty Analysis, but the System Malfunctions . . . . .	108
Any User Logging into the System Receives the “System Experienced Fatal Error” Message . . . . .	108
The User Interface Is Unresponsive or an Interactive Job Does Not Progress . . . . .	108
A Batch Job Does Not Progress . . . . .	109
JobSynchLock Is Locked . . . . .	109



## Chapter 1

# About SAS Warranty Analysis

---

<b>Introduction</b> .....	<b>1</b>
<b>Components of SAS Warranty Analysis</b> .....	<b>2</b>
<b>Client Applications and Sessions</b> .....	<b>2</b>
Overview .....	2
The SAS Warranty Analysis Thin Client .....	2
The SAS Warranty Analysis Rich Client .....	3
The SAS Warranty Analysis Diagnostic Client (Experimental) .....	4

---

## 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.

SAS Warranty Analysis is designed specifically for warranty reporting and analysis for any organization that warrants its products. It uses 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 different 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 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 project
- perform further analyses on warranty data in a SAS Enterprise Guide project

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

---

## Client Applications and Sessions

### Overview

SAS Warranty Analysis 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. For more information, see [Chapter 5, “Security,” on page 23](#).

### ***The SAS Warranty Analysis Thin Client***

The most frequently used client is the Web-based thin client. 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 on-site SAS support personnel who set up your system.



The screenshot displays the SAS Warranty Analysis Client interface within the SAS Portal. The interface is divided into several sections:

- SAS Warranty Analysis Client:** Contains a link to "Launch SAS Warranty Analysis Client".
- SAS Warranty Analysis Login History:** A table showing login details for three users.
- SAS Warranty Analysis News:** A section for news items, currently showing "No messages found".
- SAS Warranty Analysis Links:** A section for links, currently showing "No message links found".

	Logon Time	Logoff Time	Duration (In Minutes)	Exit Method
1	1/21/11 11:29:34 AM EST			
2	1/20/11 5:05:06 AM EST	1/20/11 5:09:03 AM EST	3.95	User logged off.
3	1/20/11 1:44:40 AM EST	1/20/11 2:25:10 AM EST	40.50	Session timed out.

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.

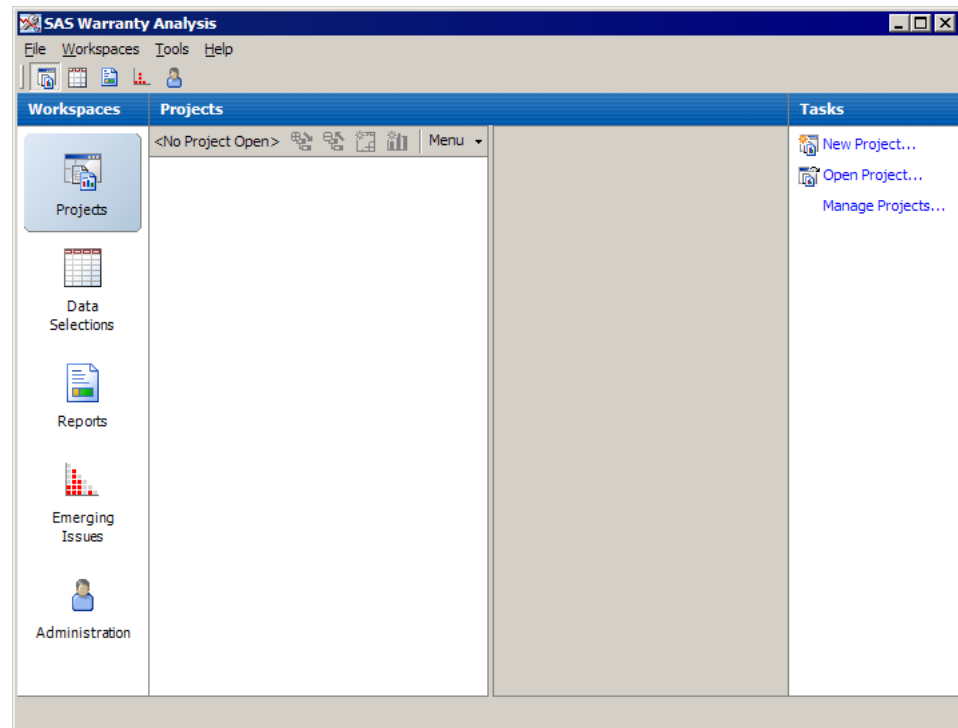
Users must log on to the SAS Portal to access the warranty-related tabs and the portlet-based information that they contain. User sessions are programmed to expire after a configurable amount of time.

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 box.

### ***The SAS Warranty Analysis Rich Client***

The rich client application is intended 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.



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 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. For more information, see [Chapter 7, “The Diagnostic Client \(Experimental\),” on page 39](#).

# SAS. Warranty Analysis Diagnostic Client on [www.sas.com](http://www.sas.com)

<a href="#">Operations</a>	JCT Synch last finished 2011.01.24 10:00:24, runs every 2 minutes	4.3.0.0.20110112.2057	nananana	<a href="#">Log Out</a>
----------------------------	---	-----------------------	----------	-------------------------

## Operations

### Diagnostic Summary

Failed Analysis Jobs	Failed JCT Synchronization Jobs	Failed Other Jobs	Server Connect Errors	Memory Cache Size (K)	File Cache Size (K)	Log File Size (K)
<a href="#">2</a>	<a href="#">0</a>	<a href="#">0</a>	<a href="#">0</a>	<a href="#">10,822</a>	<a href="#">2,328</a>	<a href="#">41,716</a>

### Interactive Performance

Monitor interactive job runtime statistics	<a href="#">Interactive Statistics</a>
Provide status of interactive jobs	<a href="#">Interactive Jobs</a>
Provide performance statistics by job type	<a href="#">Job Performance</a>

### Chained Report Listings

Show ancestry of Auto Update/Create New reports	<a href="#">Auto Update/Create New</a>
Show ancestry of EI Analytic reports	<a href="#">EI Analytic</a>
Show ancestry of EI Threshold reports	<a href="#">EI Threshold</a>

### Batch Performance

Monitor batch job runtime statistics	<a href="#">Batch Statistics</a>
Provide status of batch jobs	<a href="#">Batch Jobs</a>
Provide status of Job Control Table	<a href="#">Job Control Table</a>
Provide performance statistics by job type	<a href="#">Job Performance</a>

### Log File Listings

List all log files by date/time	<a href="#">All Logs</a>	<a href="#">Analysis Only</a>
List all log files by user	<a href="#">User Logs</a>	<a href="#">Analysis Only</a>
List error log files by date/time	<a href="#">All Error Logs</a>	<a href="#">Analysis Only</a>
List error log files by user	<a href="#">User Error Logs</a>	<a href="#">Analysis Only</a>
List Enterprise Analytic child logs	<a href="#">All Child Logs</a>	<a href="#">Error Logs Only</a>
List meta message logs	<a href="#">Meta Message Logs</a>	<a href="#">Today's Meta Messages</a>

### General Performance

Get a comprehensive view of system execution	<a href="#">System Execution</a>
Get a reduced view of recent job timings	(disabled)
Show status of mid tier caching	<a href="#">Cache View</a>
Show batch job density/performance per day	<a href="#">Batch Density View</a>

### User Management/Miscellaneous Tools

Domain object ownership across users	<a href="#">Object Ownership</a>
System Activity Snapshot By User	<a href="#">System Activity</a>
Type in SAS code and/or SQL queries and run them to get a log	<a href="#">Diagnostic Trial</a>
Type in property/value pairs to pass to the Warranty driver program	<a href="#">Run Driver Program</a>

### Metadata Management

Upgrade migrated analysis results to Warranty 4.3	<a href="#">Run Migrated Analyses</a>
Manage orphaned metadata server objects	<a href="#">Orphaned Metadata</a>
Manage orphaned domain object table-based content	<a href="#">Orphaned Table Data</a>

### Configuration Related

Get a listing of the current configuration parameters	<a href="#">Configuration Parameters</a>
Get a diagnostic test readout of configuration structure	<a href="#">Configuration Test</a>
Retrieve the most useful PROC OPTIONS values from the SAS tier	<a href="#">SAS Options Listing</a>



## Chapter 2

# System Architecture

---

<b>Introduction</b> .....	<b>7</b>
<b>The Client Tier</b> .....	<b>7</b>
<b>The Middle Tier</b> .....	<b>8</b>
<b>The Server Tier</b> .....	<b>8</b>
<b>SAS Tier</b> .....	<b>9</b>
<b>Warranty Data Tier</b> .....	<b>9</b>
<b>Batch Processing</b> .....	<b>10</b>

---

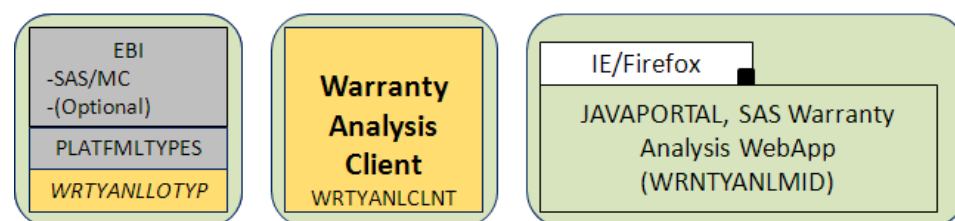
## Introduction

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 supports both a thin client and a rich client.

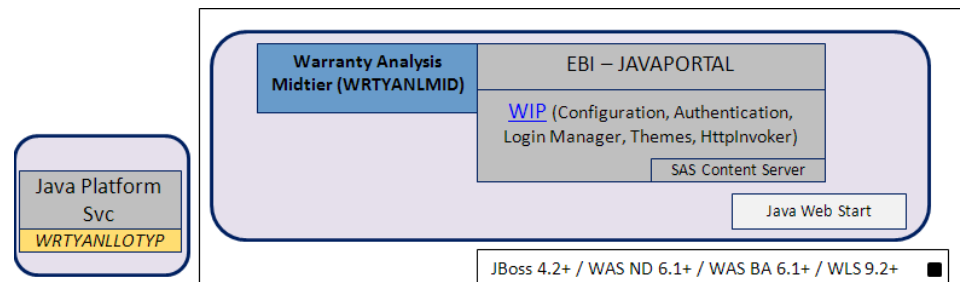


The SAS Warranty Analysis rich client was developed by using the Eclipse Rich Client Platform (RCP) and Java Swing frameworks. The thin client is built on top of the SAS Information Delivery Portal. The thin client provides read-only access to public reports and emerging issues alerts that have been 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. For additional discussion of roles, see [Chapter 5, "Security,"](#) on page 23.

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 Middle Tier

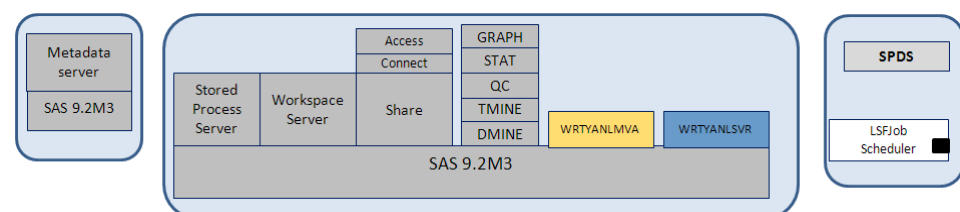
The SAS Warranty Analysis middle tier fulfills client requests by applying business rules and controlling access to shared resources (for example, SAS Workspace Server connections).



The SAS Warranty Analysis diagnostic client is served from the same servlet container as the SAS Warranty Analysis WAR file.

## The Server Tier

The SAS Warranty Analysis server tier uses the SAS Metadata Server, SAS Stored Process Server, SAS Pooled 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.

SAS Management Console and 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

---

## 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 box 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 9.](#))

---

## 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 utility libraries known as USERDL, USERRL, EIOUT, USERTL, 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).

The USERTL library contains the compiled statgraph templates used to render graphics in analysis results.

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 on-site SAS support personnel 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 is 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 on-site SAS support personnel 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 on-site SAS support personnel provide details of the scheduling setup for your site.

For additional information about batch processing, see [Chapter 9, "Working with Batch Processes,"](#) on page 55.



## Chapter 3

# Finding Scripts, Files, and Other System Components

---

<b>Overview</b> .....	<b>11</b>
<b>Finding Installation and Configuration Files</b> .....	<b>11</b>
<b>Finding SAS Server Tier Components</b> .....	<b>12</b>
SAS Metadata Server Start Scripts .....	12
Warranty Object Spawner Scripts .....	12
Warranty Share Server Scripts .....	12
Application Server Scripts .....	12
Data Library Directories .....	12

---

## 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 on-site SAS support personnel 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 on-site SAS support personnel provide 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 before 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 on-site SAS support personnel who sets up your system might change these locations. Your on-site SAS support personnel 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.

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

- `<SASInstall>` defaults to `/usr/local/SAS`.
- `<SASConfig>` defaults to `/usr/local/<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.

---

## Finding SAS Server Tier Components

### **SAS Metadata Server Start Scripts**

For details, see [“Metadata Server Start Scripts” on page 36](#).

### **Warranty Object Spawner Scripts**

For details, see [“Object Spawner Scripts” on page 36](#).

### **Warranty Share Server Scripts**

For details, see [“Share Server Scripts” on page 36](#).

### **Application Server Scripts**

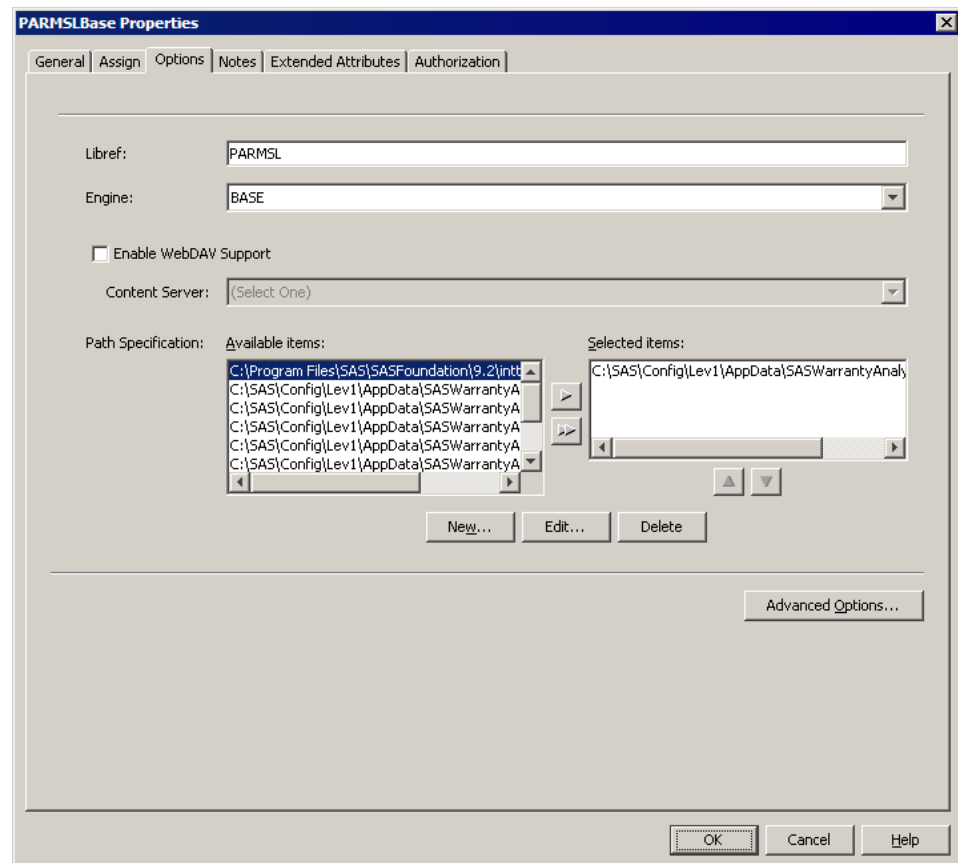
For details, see [“Application Server Scripts” on page 36](#).

### **Data Library Directories**

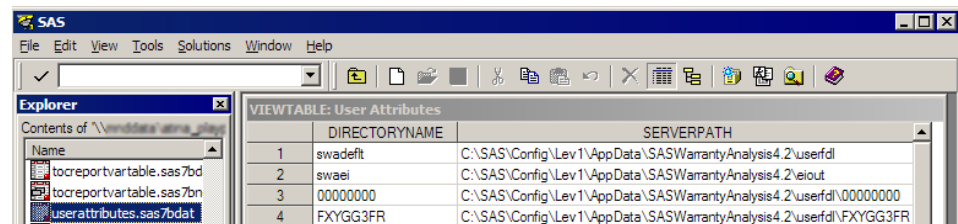
By default, the SAS Warranty Analysis libraries are found in the following subdirectories in the `<SASConfig>\Lev1\AppData\SASWarrantyAnalysis4.3` folder in a Windows environment: `eiout`, `lookup`, `parms1`, `userdl`, `userrl`, `usertl`, and `userfdl`.

The on-site SAS support 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 SAS Management Console under **Data Library Manager** ⇒ **Libraries**.

The entries **PARMSLBASE**, **USERDLBASE**, **USERRLBASE**, **USERTL**, and **EIOUTBASE** contain the path specifications for each of these libraries.



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.





## Chapter 4

# Working with Domain Objects

---

<b>Overview of Domain Objects</b> . . . . .	<b>15</b>
<b>Domain Object Types and Characteristics</b> . . . . .	<b>16</b>
<b>Domain Object and Results Ownership</b> . . . . .	<b>17</b>
Overview . . . . .	17
Domain Object Ownership . . . . .	17
Execution Results Ownership . . . . .	18
Filtered Subset Sharing, Updating, and Copying . . . . .	19
Automatic Deletion of Filtered Subsets . . . . .	20
<b>When Someone Leaves the Company</b> . . . . .	<b>20</b>
<b>When Available Disk Space Is Low</b> . . . . .	<b>21</b>
Overview . . . . .	21
Clear Existing Disk Space . . . . .	21
Move Users to Different Volumes . . . . .	21
<b>When Data Losses Occur</b> . . . . .	<b>22</b>
Overview . . . . .	22
Filtered Subset Loss . . . . .	22
SAS Metadata Server Data Loss . . . . .	22
Domain Object Table Loss . . . . .	22

---

## 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:* 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. For instructions to move users' data, see [“Move Users to Different Volumes” on page 21](#).

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

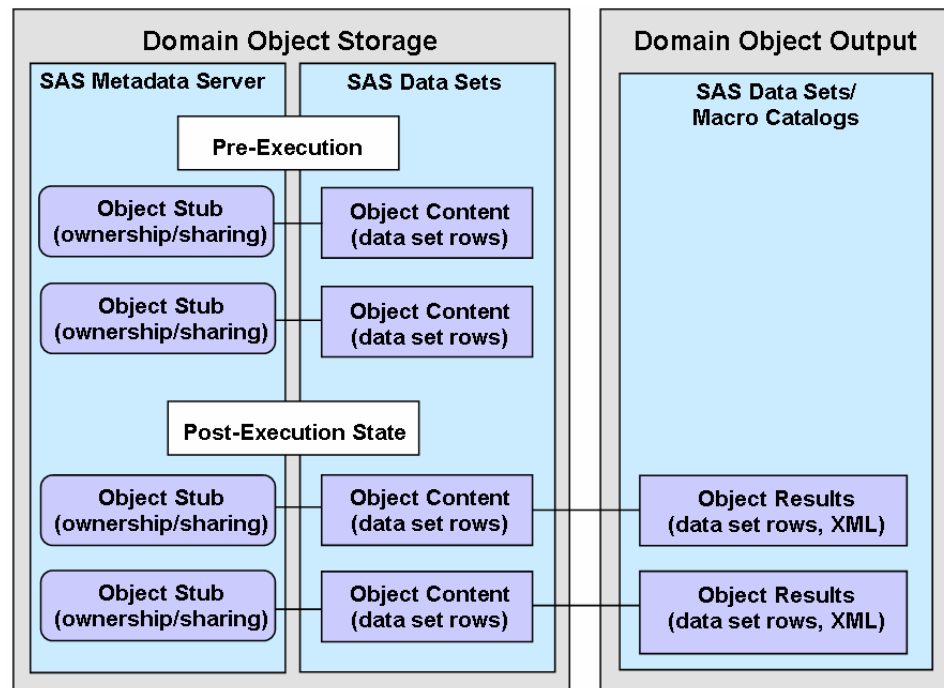
This chapter covers the following types of domain objects:

- 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.

Here is an illustration that depicts 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.



The content and execution results of different domain objects are stored in different places. For the SAS libraries used to store different types of persistent data, see [“Warranty Data Tier” on page 9](#).

## Domain Object and Results Ownership

### Overview

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 is 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.

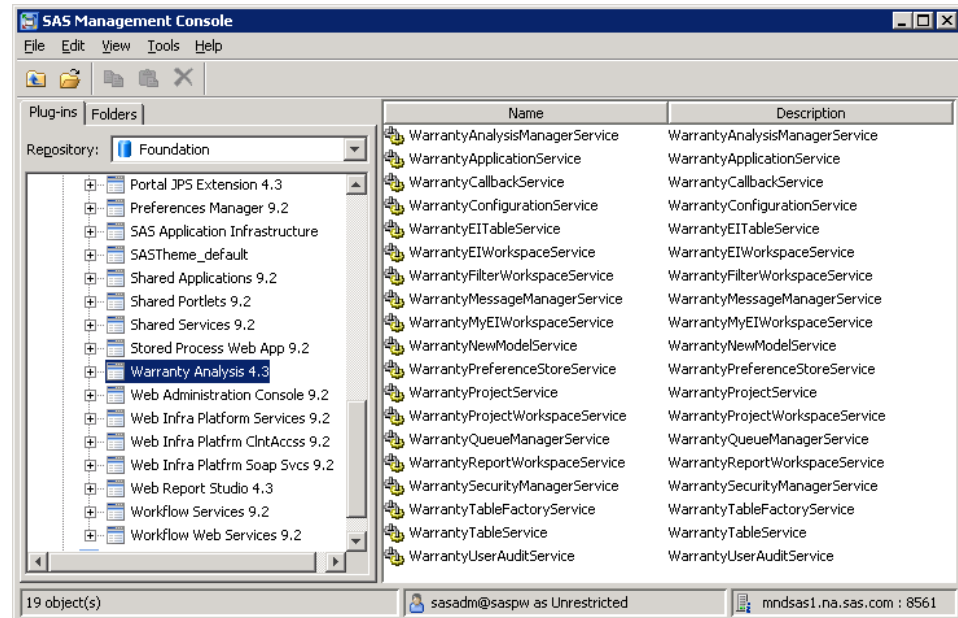
### Domain Object Ownership

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 SAS Management Console to find the metadata components of warranty domain objects, as shown here:



Use 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 SAS Warranty Analysis 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

### Overview

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 EIOU). These catalogs are described in [Chapter 2, “System Architecture,” on page 7](#).

### **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 detected by the system and are not handled correctly.

## **Filtered Subset Sharing, Updating, and Copying**

### **Overview**

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. After a filtered subset is created, it is reused by all analyses that require it.

For example, 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. When a user who has imported a public report performs an “Analyze in Project” action, the filtered data is reused from the public report owner’s `filteredData` folder.

### **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. For information, see [“When a Previous Version of a Filtered Subset Is Kept” on page 20](#).

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

### ***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.

### ***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 the 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. Ensure that the exiting employee trains another employee about 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.** For information, see [“Issues with Manual Deletion of Results” on page 19](#).

---

## 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 either the online Help or the *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.

*Note:* In SAS Warranty Analysis, 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.

*Note:* 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 types 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

If the filter subset for a report, project analysis, or Emerging Issues alert is deleted, a click-to or drill-to operation creates a new filter subset that is equivalent to the original filter subset. The new filter subset is created regardless of how the original filter subset was deleted: intentionally or unintentionally deleted by a user, automatically deleted by SAS Warranty Analysis because of the age of the filter subset, or deleted outside of SAS Warranty Analysis (such as at the file-system level).

### SAS Metadata Server Data Loss

Your SAS Metadata Server's persistent storage should always be backed up so that the data is 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 the *SAS 9.2 Intelligence Platform: System Administration Guide*:

<http://support.sas.com/documentation/cdl/en/bisag/60945/PDF/default/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

---

<b>Overview</b>	<b>23</b>
<b>SAS Business Intelligence Platform Groups and Roles</b>	<b>24</b>
<b>Protected Resources</b>	<b>25</b>
Overview	25
Application Server	25
Application Metadata	25
<b>Security Model</b>	<b>26</b>
Overview	26
Roles	27
Nested Groups for Cumulative Access Privileges and Roles	28
<b>Creating a User Account</b>	<b>28</b>
<b>Granting Access through Group Membership</b>	<b>28</b>
<b>Denying Access to Emerging Issues</b>	<b>29</b>
<b>Creating New Groups for Domain Object Sharing</b>	<b>29</b>
<b>Filtering Out Groups from Domain Object Sharing</b>	<b>29</b>
Overview	29
Roles Never Show Up for Sharing	30
Default Groups Never Show Up for Sharing	30
Exclude Additional Groups Using SAS Management Console	30
<b>Restricting Access to Data By Using Column-Level Security</b>	<b>30</b>
Overview	30
Register the Data Tables	31
Import the Information Map	31
Assign Authorization	31
Enable Column-Level Security	32

---

## Overview



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

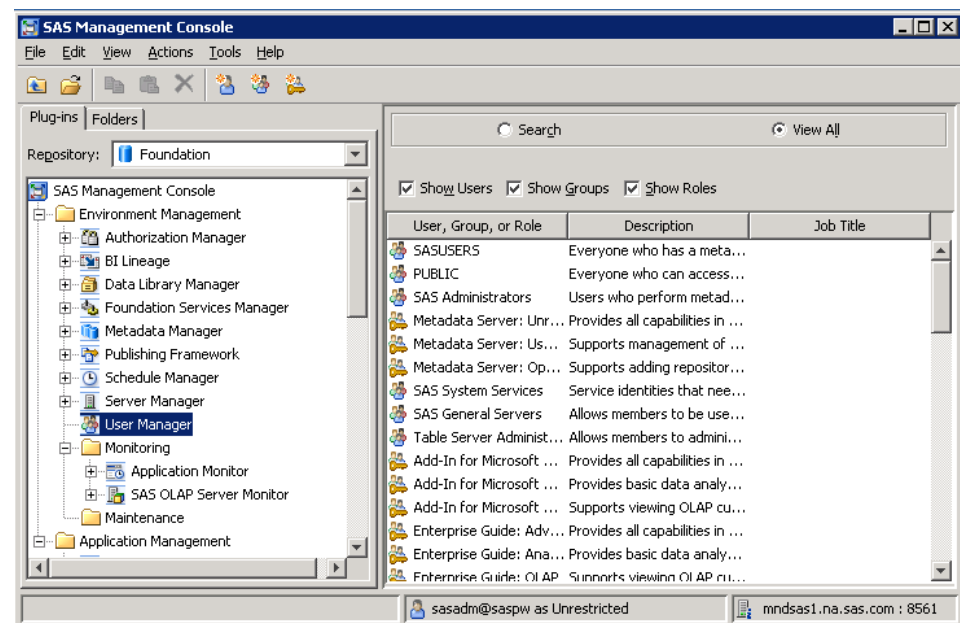
The following chapters describe the group structure and explain how to use it to grant or deny users access to system functionality.

## SAS Business Intelligence Platform Groups and Roles

The SAS Business Intelligence platform uses the “Group” and “Role” concepts regarding system security:

- Groups protect 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.
- Roles provide 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. (The icon for a group is , and the icon for a role is .)



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

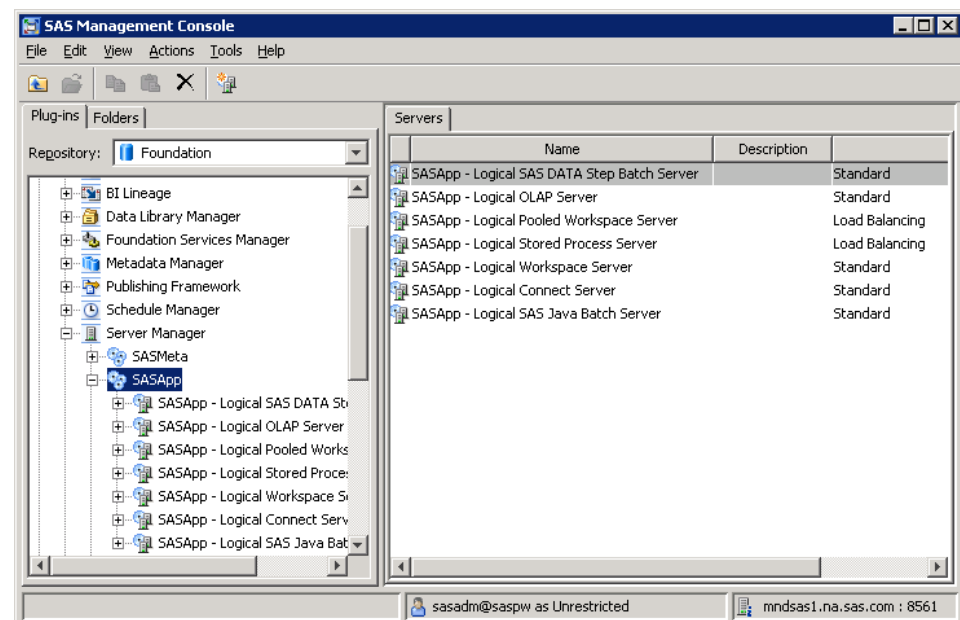
### Overview

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.

For additional information about the system resources, see [“Security Model” on page 26](#).

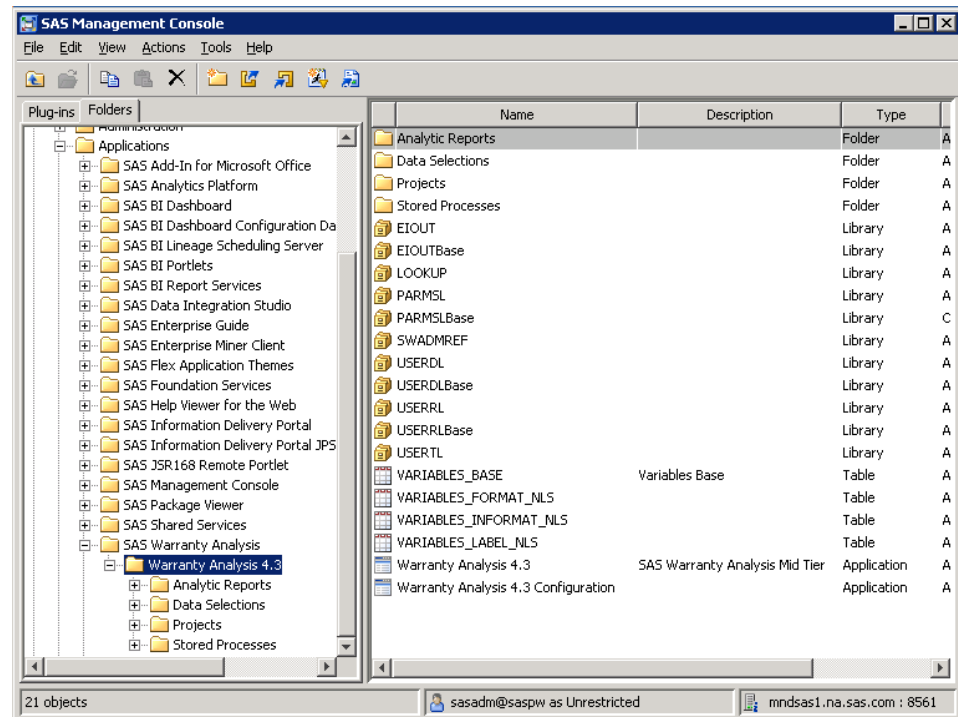
### Application Server

SAS Warranty Analysis uses the default SAS Business Intelligence platform application server. The Warranty system groups provide the access privileges that are necessary to work on this SAS application server.



### Application Metadata

As explained in [Chapter 4, “Working with Domain Objects,” on page 15](#), 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. Additionally, access privileges are placed on the domain object metadata objects when they are created.

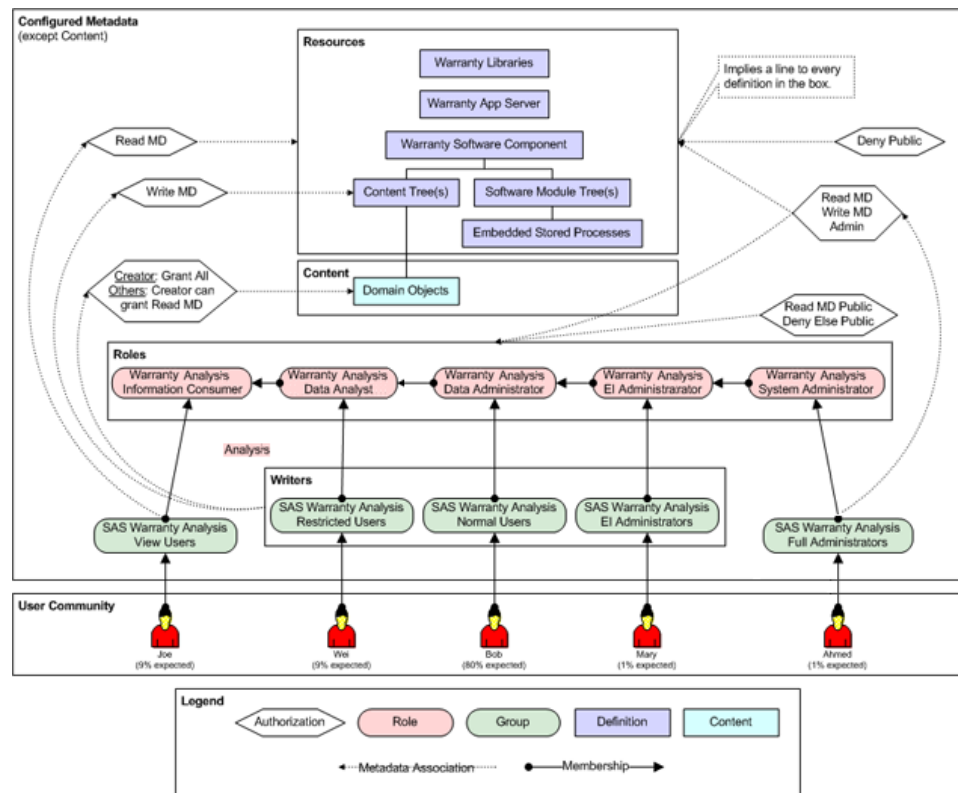


## Security Model

### Overview

The following figure 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 Warrant Analysis depends on these settings remaining as they are.

*Note:* The abbreviation “MD” means “metadata.”

## **Roles**

The SAS Warrant Analysis system roles are as follows:

**Warranty Analysis: Information Consumption**  
enables access to any content in the thin client.

*Note:* All SAS Warrant Analysis users must be assigned to the SAS Warrant Analysis View Users group, which then applies this role.

**Warranty Analysis: Data Analysis**  
enables access to the rich client, but not to data selection editing.

**Warranty Analysis: Data Administration**  
enables access to the rich client’s data selection-building screens.

*Note:* The user also needs to be associated with the Warranty Data Analysis role.

Warranty Analysis: EI Administration

enables access to the emerging issues run-time setup functionality.

Warranty Analysis: System Administration

enables access to all other screens that are related to system configuration and that are not included in this list.

### ***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 SAS Warranty Analysis Normal Users group.

---

## **Creating a User Account**

Here are the general steps to create a user account:

1. Do one of the following steps:
  - a. (UNIX) Create a UNIX user account with a valid home directory and a primary user group that is the same as the sasdemo user. The command to create a new user account is similar to the following:
 

```
useradd -d /users/swauser1 -g swagroup -m -s /bin/ksh swauser1
```

*Note:* To create a user account, you must be logged on as a user with root permissions. There are no additional local policy settings required on UNIX.
  - b. (Microsoft Windows) Add credentials for the user account in Microsoft Windows, and then set the local policy to allow the user account to log on as batch.
2. In SAS Management Console, do the following steps:
  - a. Add the user account.
  - b. In the properties for the user account, set the login properties to the credentials specified in step #1.
  - c. Add the user account to one of the SAS Warranty Analysis user groups.

---

## **Granting Access through Group Membership**

You can grant a user access to the SAS Warranty Analysis system by using SAS Management Console to make the user a member of one of the five SAS Warranty Analysis groups.

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

---

## Denying Access to Emerging Issues

Some SAS Warranty Analysis customers want to enable their suppliers to view some public reports, but the customers do not want the suppliers to see emerging issues alerts. SAS Warranty Analysis offers an optional role and group that enable you to deny a user the ability to see emerging issues. Although the user cannot see emerging issues, the user can see public reports.

When you deny access to emerging issues, this denial overrides any other emerging issues user access privileges granted by membership in the five standard SAS Warranty Analysis groups or any other group created at your site.

To deny access to emerging issues, SAS Warranty Analysis uses a specific group name (SAS Warranty Analysis Deny EI Access) in a specific role name (Warranty Analysis: Deny EI Access). After you add a user to the group, the user is denied access to emerging issues.

*Note:* The role name and the group name are not created by the standard installation and configuration process. You must create them manually, with these exact names, for the denial behavior to work.

To deny access to emerging issues:

1. Assign users and groups to the SAS Warranty Analysis Deny EI Access group.
2. Instruct the users that you added to the SAS Warranty Analysis Deny EI Access group to log off of SAS Warranty Analysis, and then to log back on.

*Note:* The access denial takes effect only on logon.

---

## Creating New Groups for Domain Object Sharing

Use 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 SAS Management Console and SAS software 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 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.

### ***Default 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 SAS Management Console***

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.

For more information, see [“Sharing Groups” on page 71](#).

---

## **Restricting Access to Data By Using Column-Level Security**

### ***Overview***

Often, an organization wants to enable certain users to access specific data columns in a data mart while also restricting those users from viewing other data columns in a data mart. For example, suppliers might be allowed to see certain data but be restricted from viewing cost-related information.

You can restrict certain users from accessing specific data columns by using row-level security in SAS Warranty Analysis metadata. Row-level security is a long-narrow representation of a data mart and is specified in SAS Information Map Studio.

Here are the general steps to implement column-level security:

1. Register the data tables.
2. Import the information map in SAS Management Console.
3. Assign authorization in SAS Information Map Studio.
4. Enable column-level security in SAS Management Console.

## Register the Data Tables

To register the data tables in SAS Management Console:

1. Start SAS Management Console, and then log on as the administrator.
2. On the **Folders** tab, navigate to **System** ⇒ **Applications** ⇒ **SAS Warranty Analysis** ⇒ **Warranty Analysis 4.3**.
3. Ensure that the data tables **VARIABLES\_\*** are listed in the right pane.  
If the data tables are listed, they are registered.
4. If the data tables **VARIABLES\_\*** are not listed in the right pane:

- a. Register the data table in the branch **Environment Management** ⇒ **Data Library Manager** ⇒ **Libraries** ⇒ **PARMSL**.

For information about how to do this, see the SAS Management Console Help about registering data tables.

- b. As you move through the wizard, register the tables **VARIABLES\_\***, and then specify the location as **System** ⇒ **Applications** ⇒ **SAS Warranty Analysis** ⇒ **Warranty Analysis 4.3**.

## Import the Information Map

To import the information map in SAS Management Console:

1. On the **Folders** tab, select **System** ⇒ **Applications** ⇒ **SAS Warranty Analysis** ⇒ **Warranty Analysis 4.3**.
2. Import the SAS package Variables.spk located in <SAS\_Install>\SAS\SASWarrantyAnalysisServerConfiguration\4.3\SASPackages.

For information about how to do this, see the SAS Management Console Help about importing objects.

3. As you move through the wizard, ensure that you do the following;
  - Select the **Include access controls** check box.
  - Select **SASApp** as the target application server.
  - Select **Warranty Analysis 4.3 (Shared Access to PARMSL)** as the target library.
  - For the target tables, select the source tables.
4. Check the SAS Management Console error logs, correct any errors that occurred, and then again import the information map.

## Assign Authorization

To assign authorization in SAS Information Studio:

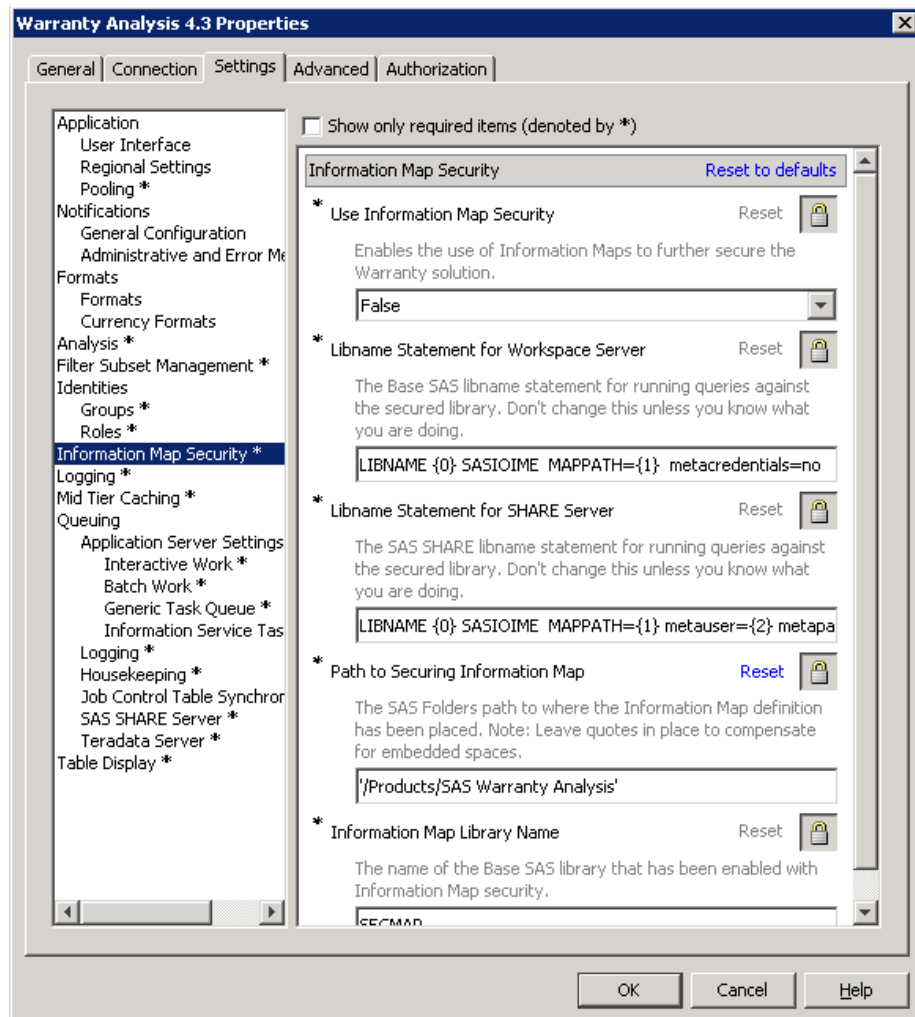
1. Start SAS Information Studio, and then open the information map Variables located in **/System/Applications/SAS Warranty Analysis/Warranty Analysis 4.3**.
2. Assign ReadMetadata and Read permissions to the following groups:

- SAS Warranty Analysis Full Administrators
  - SAS Warranty Analysis Normal Users
  - SAS Warranty Analysis Restricted Users
  - SAS Warranty Analysis View Users
  - SAS Warranty Analysis EI Administrators
3. To restrict access to the users in the SAS Warranty Analysis Normal users group, add or edit a condition by creating a filter, and then select the filter.  
  
For information about how to do this, see the SAS Management Console Help about the Row-Level Permission Condition dialog box.
  4. Save the information map, and then close SAS Information Map Studio.

### ***Enable Column-Level Security***

To enable column-level security in SAS Management Console:

1. In SAS Management Console, view the properties for **Warranty Analysis 4.3**.  
  
For information about how to do this, see the SAS Management Console Help about viewing and updating repository properties.
2. On the **Settings** tab, select **Information Map Security**.



3. Set the value of **Use Information Map Security** to **True**.

In SAS, perform the following tasks:

1. In the PARMSL library, open the data set **security\_options**.  
For information about how to do this, see the SAS Help about opening an existing table from the VIEWTABLE window.
2. If the default row does not exist, create it.
3. If necessary, change the default row's **Infomap Path** value to **' /System/Applications/SAS Warranty Analysis/Warranty Analysis 4.3'** and **Is Security Enabled** value to **Y**.

In SAS Warranty Analysis, log in using the restricted user account, and then create a new data selection and analysis to verify that the data columns are not shown.





## Chapter 6

# Starting and Stopping the System

---

<b>Overview</b> .....	<b>35</b>
<b>Tier Dependencies</b> .....	<b>35</b>
<b>Metadata Server Start Scripts</b> .....	<b>36</b>
<b>Object Spawner Scripts</b> .....	<b>36</b>
<b>Share Server Scripts</b> .....	<b>36</b>
<b>Application Server Scripts</b> .....	<b>36</b>
<b>Service Dependencies</b> .....	<b>37</b>

---

## 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,” on page 7](#), 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 default scripts created by the standard installation and configuration process for the SAS Warranty Analysis bundle. The on-site SAS support personnel who set up your site’s system might have customized the scripts to make the starting and stopping process easier and more streamlined. On-site SAS support personnel 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/SASMeta/MetadataServer/MetadataServer.sh  
start
```

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

In a Windows environment, use the following command:

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

---

## Object Spawner Scripts

In a UNIX environment, use the following command:

```
<SASConfig>/Lev1/ObjectSpawner/ObjectSpawner.sh start
```

In a Windows environment, use the following command:

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

---

## Share Server Scripts

In a UNIX environment, use the following command:

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

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\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/Applications/RemoteServices/  
RemoteServices.sh start
```

In a Windows environment, use the following command:

```
<SASConfig>\Lev1\Web\Applications\RemoteServices  
\RemoteServices.bat start
```

For Web application servers such as Oracle WebLogic and IBM WebSphere, instructions for starting are provided in the file `instructions.html`, which is generated after the SAS Deployment Wizard configures the application server. The file `Instructions.html` is located in `<SASConfig>\Lev1\Documents`.

The SAS configuration process provides a batch script to start JBoss.

```
<JBossInstallLocation>/bin/SASServer1.sh start
```

In a Windows environment, use the following command:

```
<JBossInstallLocation>\bin\SASServer1.bat start
```

---

## Service Dependencies

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

1. metadata server
2. object spawner
3. SAS/SHARE server

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

1. remote services application
2. Web server (JBoss, Oracle 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

# The Diagnostic Client (Experimental)

---

<b>Overview</b> .....	<b>39</b>
<b>Using the Diagnostic Client</b> .....	<b>40</b>
<b>Start the Diagnostic Client</b> .....	<b>40</b>
<b>Main Page</b> .....	<b>40</b>
<b>Navigation Bar</b> .....	<b>41</b>
<b>Diagnostic Summary Area</b> .....	<b>42</b>
<b>Interactive Performance Area</b> .....	<b>42</b>
<b>Batch Performance Area</b> .....	<b>43</b>
<b>General Performance Area</b> .....	<b>43</b>
<b>Metadata Management Area</b> .....	<b>43</b>
<b>Chained Report Listings Area</b> .....	<b>44</b>
<b>Log File Listings Area</b> .....	<b>44</b>
<b>Anatomy of a Job Log File Listing</b> .....	<b>45</b>
Overview .....	45
Job Log Header .....	45
Job Log Java Error .....	45
Job Log SAS Error or Warning .....	46
Ignore Errors Section .....	46
Job Log Statistics Summary .....	46
<b>User Management/Miscellaneous Tools Area</b> .....	<b>47</b>
<b>Configuration Related Area</b> .....	<b>48</b>

---

## 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.

This chapter describes those parts of the diagnostic client not addressed in other chapters. For example, the diagnostic client's job-related views are discussed in [Chapter 8, "Working with Jobs, Tasks, and Queues,"](#) on page 49.

---

## Using the Diagnostic Client

To use the diagnostic client, the following conditions must be satisfied:

- You must be a member of the SAS Warranty Analysis Full Administrators user group or the SAS Warranty Analysis EI Administrators user group.
- If you are logged in to the SAS Information Delivery Portal, you must log out before starting the diagnostic client.
- The diagnostic client must be started in a separate Web browser window and session that are dedicated solely to the diagnostic client.

*Note:* Do not use this Web browser window and session to access the SAS Information Delivery Portal or any other SAS Web-based client application.

The diagnostic client starts its own session that expires when Web server times out the browser session.

---

## Start the Diagnostic Client

To start the diagnostic client, type the following URL into a Web browser: **http://hostname:8080/SASWarrantyAnalysis**.

*Note:* The default port number is 8080. Your organization's port number might be different.

---

## Main Page

The main page of the diagnostic client shows all available operations. Almost all of this information is read-only. Some of the information is useful for monitoring system health.

# SAS. Warranty Analysis Diagnostic Client on [sas.com](http://sas.com)

<a href="#">Operations</a>	JCT Synch last finished 2011.01.24 10:00:24, runs every 2 minutes	4.3.0.0.20110112.2057	nananana	<a href="#">Log Out</a>
----------------------------	---	-----------------------	----------	-------------------------

## Operations

### Diagnostic Summary

Failed Analysis Jobs	Failed JCT Synchronization Jobs	Failed Other Jobs	Server Connect Errors	Memory Cache Size (K)	File Cache Size (K)	Log File Size (K)
<a href="#">2</a>	<a href="#">0</a>	<a href="#">0</a>	<a href="#">0</a>	<a href="#">10,822</a>	<a href="#">2,328</a>	<a href="#">41,716</a>

### Interactive Performance

Monitor interactive job runtime statistics	<a href="#">Interactive Statistics</a>
Provide status of interactive jobs	<a href="#">Interactive Jobs</a>
Provide performance statistics by job type	<a href="#">Job Performance</a>

### Chained Report Listings

Show ancestry of Auto Update/Create New reports	<a href="#">Auto Update/Create New</a>
Show ancestry of EI Analytic reports	<a href="#">EI Analytic</a>
Show ancestry of EI Threshold reports	<a href="#">EI Threshold</a>

### Batch Performance

Monitor batch job runtime statistics	<a href="#">Batch Statistics</a>
Provide status of batch jobs	<a href="#">Batch Jobs</a>
Provide status of Job Control Table	<a href="#">Job Control Table</a>
Provide performance statistics by job type	<a href="#">Job Performance</a>

### Log File Listings

List all log files by date/time	<a href="#">All Logs</a>	<a href="#">Analysis Only</a>
List all log files by user	<a href="#">User Logs</a>	<a href="#">Analysis Only</a>
List error log files by date/time	<a href="#">All Error Logs</a>	<a href="#">Analysis Only</a>
List error log files by user	<a href="#">User Error Logs</a>	<a href="#">Analysis Only</a>
List Enterprise Analytic child logs	<a href="#">All Child Logs</a>	<a href="#">Error Logs Only</a>
List meta message logs	<a href="#">Meta Message Logs</a>	<a href="#">Today's Meta Messages</a>

### General Performance

Get a comprehensive view of system execution	<a href="#">System Execution</a>
Get a reduced view of recent job timings	(disabled)
Show status of mid tier caching	<a href="#">Cache View</a>
Show batch job density/performance per day	<a href="#">Batch Density View</a>

### User Management/Miscellaneous Tools

Domain object ownership across users	<a href="#">Object Ownership</a>
System Activity Snapshot By User	<a href="#">System Activity</a>
Type in SAS code and/or SQL queries and run them to get a log	<a href="#">Diagnostic Trial</a>
Type in property/value pairs to pass to the Warranty driver program	<a href="#">Run Driver Program</a>

### Metadata Management

Upgrade migrated analysis results to Warranty 4.3	<a href="#">Run Migrated Analyses</a>
Manage orphaned metadata server objects	<a href="#">Orphaned Metadata</a>
Manage orphaned domain object table-based content	<a href="#">Orphaned Table Data</a>

### Configuration Related

Get a listing of the current configuration parameters	<a href="#">Configuration Parameters</a>
Get a diagnostic test readout of configuration structure	<a href="#">Configuration Test</a>
Retrieve the most useful PROC OPTIONS values from the SAS tier	<a href="#">SAS Options Listing</a>

Copyright(c) 2009 - 2011 by SAS Institute, Inc., All Rights Reserved

## Navigation Bar

Links on the navigation bar include a logout link and a link back to the home (operations) 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.

---

## Diagnostic Summary Area

The **Diagnostic Summary** area shows how many critical errors appear in each job log. Error logs age out of the system, so this area 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 **Diagnostic Summary** area displays the following summaries:

- **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 on-site SAS support personnel 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 JobControl 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.

- **Memory Cache Size (K)**

This area shows the size of the memory cache in kilobytes.

- **File Cache Size (K)**

This area shows the size of the file cache in kilobytes.

- **Log File Size (K)**

This area shows the size of the log file in kilobytes.

---

## Interactive Performance Area

For information, see [Chapter 8, “Working with Jobs, Tasks, and Queues,”](#) on page 49.



---

## Batch Performance Area

For information, see [Chapter 8, “Working with Jobs, Tasks, and Queues,”](#) on page 49.

---

## General Performance Area

The **General Performance** area shows information about system performance.

The **General Performance** area displays the following summaries:

- **System Execution**

This area 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 area.

- **Get a reduced view of recent job timings**

This listing summarizes and aggregates various timings for each job. It also shows the SQL queries that occurred during each job and task.

This link is disabled by default. You can enable this link by clicking the **Configuration Parameters** link in the **Configuration Related** area, and then clicking the **Enable** link adjacent to **Turn job timing capture on/off**.

- **Batch Density View**

This link shows batch-job density and performance for each day.

- **Cache View**

This area provides information about middle-tier caching.

---

## Metadata Management Area

The **Metadata Management** area shows information about metadata. This area provides links for use during installation and configuration when SAS Warranty Analysis is migrated to a newer version.

**CAUTION:**

**Use this area only under the guidance of SAS Technical Support.** These links are used only once.

The **Metadata Management** area displays the following summaries:

- **Run Migrated Analyses**
- **Orphaned Metadata**
- **Orphaned Table Data**

---

## Chained Report Listings Area

This set of views assists you with low-level debugging rather than routine administrative maintenance. 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.

---

## Log File Listings Area

The **Log File Listings** area 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. 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 area 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 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 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.

- **List Enterprise Analytic child logs**

The first link provides a listing of all child log files. The second link provides a listing of all log file that contain errors.

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**

This link lists all available meta message log files.

- **Today's Meta Messages**

This link takes you directly to the current day's meta message log file.

## Anatomy of a Job Log File Listing

### Overview

SAS Warranty Analysis creates a new log file for each job that runs. This log provides comprehensive information about 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 filename
- optional table of errors with links to the errors and warnings in the log
- start date, time, and duration for the job

Here is an example of this content:

#### Retrieve Log File

[Bottom](#)

Error	Warning
<a href="#">Error 1</a>	<a href="#">Warning 1</a>
<a href="#">Error 2</a>	

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 type 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(DateInputFieldValidator.java:50)
    at com.sas.analytics.mfg.warranty.core.persistence.sql.table.TableReader.loadPage(TableReader.java:100)
    at com.sas.analytics.mfg.warranty.core.persistence.sql.table.TableReader.readPage(TableReader.java:110)
    at com.sas.analytics.mfg.warranty.core.table.FetchTablePageTask.doExecute(FetchTablePageTask.java:100)
    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$TaskRunner.runTaskCode(ExecutionQueue$TaskRunner.java:668)
    at com.sas.analytics.mfg.warranty.queue.ExecutionQueue$TaskRunner.run(ExecutionQueue$TaskRunner.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%
```

Copyright(c) 2008 by SAS Institute, Inc., All Rights Reserved

---

## User Management/Miscellaneous Tools Area

The User Management/Miscellaneous Tools area enables you to run various tools.

### CAUTION:

**Use these tools only under the guidance of SAS Technical Support.**

The information includes the following:

- **Object Ownership**

This area lists the total number of objects and who owns them.

- **System Activity**

This area displays graphs of interactive jobs and batch jobs, as well as user activity statistics.

- **Diagnostic Trial**

This area enables an administrator to try SQL queries or SAS code fragments while debugging on-site problems.

- **Run Driver Program**

This area enables an administrator to run (apart from the clients) a SAS stored procedure (the driver program) that executes most of the SAS Warranty Analysis SAS code. For the code to work, the proper parameters must be supplied.

---

## Configuration Related Area

The **Configuration Related** area enables you to see or test aspects of the configuration. The information includes the following:

- **Configuration Parameters**

This area provides a partial display of the configuration parameters that are currently in force.

- **Configuration Test**

This area displays a set of diagnostic tests to determine how the system responds.

- **SAS Options Listing**

This area runs PROC OPTIONS to display a listing of the SAS engine settings that govern the running of SAS code. At certain times, this information can be useful to know. For example, the MEMSIZE system option value can help you ensure that SAS is configured properly for the underlying hardware.

## Chapter 8

# Working with Jobs, Tasks, and Queues

---

<b>Overview</b> .....	<b>49</b>
<b>About Jobs, Tasks, and Queues</b> .....	<b>49</b>
<b>Managing Interactive and Batch Jobs</b> .....	<b>50</b>
Managing Interactive Jobs .....	50
Managing Batch Jobs .....	50
<b>Managing Queues in the Rich Client</b> .....	<b>50</b>
<b>Using the Diagnostic Client to View Queues (Experimental)</b> .....	<b>50</b>
Viewing Interactive Jobs .....	50
Viewing Batch Jobs .....	52
<b>Using the Diagnostic Client to View Job Performance</b> .....	<b>53</b>
Viewing Interactive Job Performance .....	53
Viewing Batch Job Performance .....	53

---

## 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 either the online Help or the *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,”](#) on page 55.

---

## 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,” on page 55](#)). 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, including the ability to remove finished jobs that were submitted by any user.

---

## Using the Diagnostic Client to View Queues (Experimental)

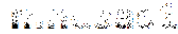
### *Viewing Interactive Jobs*

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

The **Interactive Job Monitoring** area shows a summary of the currently running interactive jobs. This area refreshes frequently so that you can see how the system is supporting its various interactive clients.




## **Warranty Analysis Diagnostic Client on**



<a href="#">Operations</a>	JCT Synch last finished 2011.01.24 11:28:41, runs every 1 minutes	4.3.99.999999.CoreAPI.20110113_78004-pp	<a href="#">Group Function</a>	<a href="#">Log Out</a>
----------------------------	---	---	--------------------------------	-------------------------

### Interactive Job Monitoring

 Interactive jobs retrieved incrementally every 5 seconds.

Submitted Jobs						Performance Links	
Job Type	Running	Max Running	Waiting	Max Waiting	Done	Detail Statistics	Job Performance
Interactive Jobs	0	1	0	0	0	<a href="#">Drill</a>	<a href="#">Drill</a>

#	Job Name	Run Date	Done Date	Job Type	Subject Type	Subject ID	Run Code	User ID	Locale	Wait On	Wait Type	Priority	Job ID
No data is available													

Copyright(c) 2009 - 2011 by SAS Institute, Inc., All Rights Reserved

The **Submitted Jobs** area 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
- **Locale:** the locale in which the job is run. The locale can differ from the locale that the client is running at the time that the analysis is created.
- **Wait On:** the ID of a job that must run before the selected job.
- **Wait Type:** the value NOWAIT (The job does not wait for any other job.), WAITSPECIFIC (The job waits until the **WaitOn** job is finished.), or WAITALLDEPENDENT (The job waits all other jobs waiting on the **WaitOn** job are finished.).
- **Priority:** is a number between 1 and 1000 that specifies the order of job execution for a job that is not running. This value is usually 500.

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

## Viewing Batch Jobs


The **Batch Job Monitoring** area shows a summary of the currently running batch jobs. This area refreshes frequently so that you can track changes in the batch environment.

### **Warranty Analysis Diagnostic Client on**

11:32:58 AM

<a href="#">Operations</a>	JCT Synch last finished 2011.01.24 11:32:58, runs every 1 minutes	4.3.99.999999.CoreAPI.20110113_78004-pp	Disconnected	<a href="#">Log Out</a>
----------------------------	---	---	--------------	-------------------------

#### Batch Job Monitoring

 Batch jobs retrieved incrementally every 30 seconds.

Submitted Jobs						Performance Links	
Job Type	Running	Max Running	Waiting	Max Waiting	Done	Detail Statistics	Job Performance
Batch Jobs	0	0	0	0	0	<a href="#">Drill</a>	<a href="#">Drill</a>

#	Job Name	Run Date	Done Date	Job Type	Subject Type	Subject ID	Run Code	User ID	Locale	Wait On	Wait Type	Priority	Job ID
No data is available													

Copyright(c) 2009 - 2011 by SAS Institute, Inc., All Rights Reserved

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**)
- **Locale**: the locale in which the job is run. The locale can differ from the locale that the client is running at the time that the analysis is created.
- **Wait On**: the ID of a job that must run before the selected job.
- **Wait Type**: the value NOWAIT (The job does not wait for any other job.), WAITSPECIFIC (The job waits until the **WaitOn** job is finished.), or WAITALLDEPENDENT (The job waits all other jobs waiting on the **WaitOn** job are finished.).
- **Priority**: is a number between 1 and 1000 that specifies the order of job execution for a job that is not running. This value is usually 500.

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.

## Using the Diagnostic Client to View Job Performance


### Viewing Interactive Job Performance

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

#### **Warranty Analysis Diagnostic Client on Windows**

<a href="#">Operations</a>	JCT Synch last finished 2011.01.24 11:47:59, runs every 1 minutes	4.3.99.999999.CoreAPI.20110113_78004-pp	00:00:00	<a href="#">Log Out</a>
----------------------------	---	---	----------	-------------------------

#### 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	Median	Std Dev
1	INTERACTIVEROLLUPSTATS	16423	00:00:00.0	00:05:00	00:00:00.1000	00:00:00.0	00:00:10
2	JCTSYNCHSTATS	52222	00:00:00.1000	00:23:44	00:00:03	00:00:03	00:00:23

#### Wait to Start Times

#	Job Type	Job Count	Min	Max	Mean	Median	Std Dev
1	INTERACTIVEROLLUPSTATS	16423	00:00:00.0	00:00:39	00:00:00.0	00:00:00.0	00:00:00.0
2	JCTSYNCHSTATS	52222	00:00:00.0	00:03:00	00:00:00.0	00:00:00.0	00:00:02

Copyright(c) 2009 - 2011 by SAS Institute, Inc., All Rights Reserved

### Viewing Batch Job Performance


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

## SAS Warranty Analysis Diagnostic Client on

11:47:59

<a href="#">Operations</a>	JCT Synch last finished 2011.01.24 11:47:59, runs every 1 minutes	4.3.99.999999.CoreAPI.20110113_78004-pp	<a href="#">Data Base</a>	<a href="#">Log Out</a>
----------------------------	---	---	---------------------------	-------------------------

### 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	Median	Std Dev
1	INTERACTIVEROLLUPSTATS	16423	00:00:00.0	00:05:00	00:00:00.1000	00:00:00.0	00:00:10
2	JCTSYNCHSTATS	52222	00:00:00.1000	00:23:44	00:00:03	00:00:03	00:00:23

#### Wait to Start Times

#	Job Type	Job Count	Min	Max	Mean	Median	Std Dev
1	INTERACTIVEROLLUPSTATS	16423	00:00:00.0	00:00:39	00:00:00.0	00:00:00.0	00:00:00.0
2	JCTSYNCHSTATS	52222	00:00:00.0	00:03:00	00:00:00.0	00:00:00.0	00:00:02

Copyright(c) 2009 - 2011 by SAS Institute, Inc., All Rights Reserved

## Chapter 9

# Working with Batch Processes

---

<b>Overview of Batch Jobs</b> .....	<b>55</b>
<b>Job Control Tables and Scheduling Integration</b> .....	<b>56</b>
Overview .....	56
Job Control Tables .....	56
Job Control Scripts .....	57
<b>Batch Processing Overview</b> .....	<b>57</b>
<b>Running the Batch Scripts</b> .....	<b>58</b>
The Emerging Issues Batch Script .....	58
Examples Using the Emerging Issues Batch Script .....	59
The Restore Batch Script .....	59
Other Batch Scripts .....	59
Examples Using Other Batch Scripts .....	60
SAS Log Files .....	61

---

## 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 **Submit 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 **Submit 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 workspace 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

### Overview

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 **Submit 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.

For more information about monitoring batch processing, see [Chapter 7, “The Diagnostic Client \(Experimental\),” on page 39](#). See “The Job Queue Utility” in either the online Help or the *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 **Submit 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 on-site SAS support 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 **Submit 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 batch script, which is site-specific, is provided by your on-site SAS support personnel.

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 batch script is as follows.

In a Windows environment:

```
<SASConfigPath>\Lev1\Applications\SASWarrantyAnalysis4.3\batch\start_EI.bat  
<EIType><datasourceid> [<maxtimelimit> [<minutes2sleep>]]
```

In a UNIX environment:

```
<SASConfigPath>/Lev1/Applications/SASWarrantyAnalysis4.3/batch/start_EI.sh  
<EIType><datasourceid> [<maxtimelimit> [<minutes2sleep>]]
```

The batch script parameters are as follows:

<EIType> (required)

specifies which emerging issues analysis to 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 batch 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 be only 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 batch scripts is used as a component of another master batch script. Once <maxtimelimit> is exceeded, control is given back to the master batch script even though analysis jobs initiated by start\_EI or start\_BR might still be running. If subsequent steps in the master batch 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 batch 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 batch script:

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

## The Restore Batch Script

The `exec_eirestore` batch script restores and re-initializes the Enterprise-Analytic (EIAUTO) issues by deleting Enterprise-Analytic Analysis runs based on an EI run group ID.

*Note:* You can display the Help and usage options for the `exec_eirestore` batch script by using the command-line switch `-H`.

The syntax of the `exec_eirestore` batch script is as follows.

In a Microsoft Windows environment:

```
<SASConfig>\Lev1\Applications\SASWarrantyAnalysis4.3\batch\exec_eirestore.ps1
<ModeCommandOptions>
```

In a UNIX environment:

```
<SASConfig>/Lev1/Applications/SASWarrantyAnalysis4.3/batch/exec_eirestore.sh
<ModeCommandOptions>
```

*Note:* The `exec_eirestore` batch script requires the Execute permission.

*Note:* Before running the `exec_eirestore` batch script located in `<SASConfig> \Lev1\Applications\SASWarrantyAnalysis4.3\batch` to remove Enterprise-Analytic results, create a directory named `logs` inside `<SASConfig> \Lev1\Applications\SASWarrantyAnalysis4.3\batch`. The `logs` directory is needed to write log files generated by the `exec_eirestore` batch script.

## Other Batch Scripts

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

The `start_BR` batch 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 batch script.

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

In a Windows environment:

```
<SASConfigPath>\Lev1\Applications\SASWarrantyAnalysis4.3\batch\start_BR.bat
<datasourceid>[<logfilename> [<maxtimelimit> [<minutes2sleep>]]]
```

In a UNIX environment:

```
<SASConfigPath>/Lev1/Applications/SASWarrantyAnalysis4.3/batch/start_BR.sh
<datasourceid>[<logfilename> [<maxtimelimit> [<minutes2sleep>]]]
```

The batch script parameters are as follows:

**<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 batch 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 be only 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 filename of the log file. The path and extension (.log) are provided by the batch 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 batch scripts is used as a component of another master batch script. Once **<maxtimelimit>** is exceeded, control is given back to the master batch script even though analysis jobs initiated by start\_EI or start\_BR might still be running. If subsequent steps in the master batch 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 batch 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 batch 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 batch script:

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

Example 4: 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 batch script:

```
start_BR.bat "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 batch scripts are executed.

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

```
<SASConfigPath>\Lev1\Applications\SASWarrantyAnalysis4.3\batch  
\logs
```

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

```
<SASConfigPath> /Lev1/Applications/SASWarrantyAnalysis4.3/  
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

---

<b>About Emerging Issues</b> .....	<b>63</b>
Overview .....	63
Threshold Watch List .....	63
Automated Analytical Process and Analytical Watch List Process .....	64
<b>Additional Reading</b> .....	<b>65</b>
<b>Reference</b> .....	<b>66</b>

---

## About Emerging Issues

### Overview

Emerging issues processes are highly parameter-driven; therefore the implementation is site-specific. Your on-site SAS support personnel provide site-specific documentation so that you know how to set up and administer your emerging issues functionality. For instructions for invoking emerging issues batch processing, see [Chapter 9, “Working with Batch Processes,”](#) on page 55.

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. For more information about SAS Warranty Analysis groups and roles, see [Chapter 5, “Security,”](#) on page 23.

### 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 be only 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 on-site SAS support personnel provide for more specific information about the parameter-driven configuration that is required for SAS Warranty Analysis emerging issues.

## Reference

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



## Chapter 11

# Using SAS Warranty Analysis Business Rules with SAS Enterprise Guide

---

Overview .....	67
Setting User Access Permissions .....	67

---

## Overview

You can use SAS Enterprise Guide or other SAS tools to calculate warranty metrics outside of SAS Warranty Analysis but retain the SAS Warranty Analysis methodology (business rules). You can calculate metrics by time in service and one (or many) categorical variables. The output is a SAS data set.

*Note:* This feature is intended for consultants and advanced users so that they can create custom reports.

To use SAS Warranty Analysis business rules with other SAS tools, you must set certain user access permissions.

---

## Setting User Access Permissions

To set the access permissions for a user to access the SAS Warranty Analysis Application Server context using other SAS tools:

1. Log in as an administration user on the SAS tier.
2. Set the user access permissions for the sasv9.cfg file located in **SASConfig\Lev1\SASApp** to Read & Execute and Read.

*Note:* The name for the SAS Application Server context can be different from the one shown in the previous example (WarrantyApp). The name is specified during the installation and configuration process at your site.

3. Set the user access permissions for the swabrmmain.sas file located in **SASConfig\Lev1\Applications\SASWarrantyAnalysis4.3\install** to Read & Execute and Read.



## Chapter 12

# Configuring System Options in SAS Management Console

---

<b>Overview</b>	<b>71</b>
<b>Sharing Groups</b>	<b>71</b>
<b>Analysis</b>	<b>72</b>
Maximum Individual Points to Analyze as Subset	72
<b>Filter Subset Management</b>	<b>72</b>
Number of Days Before Inactive Filter Subset is Deleted	72
<b>Identities &gt; Groups</b>	<b>72</b>
Overview	72
View Users	72
Restricted Users	73
Normal User	73
Emerging Issues Administrative Users	73
Full Administrative Users	73
Deny Emerging Issues Access Users	73
<b>Identities &gt; Roles</b>	<b>73</b>
Overview	73
Information Consumer	73
Data Analyst	74
Data Administrator	74
Emerging Issues Administrator	74
Full Administrator	74
Deny Emerging Issues Access	74
<b>Information Map Security</b>	<b>74</b>
Use Information Map Security	74
Libname Statement for Workspace Server	75
Libname Statement for SHARE Server	75
Path to Securing Information Map	75
Information Map Library Name	75
<b>Logging</b>	<b>75</b>
SQL Logging	75
XML Logging	75
<b>Mid Tier Caching</b>	<b>76</b>
Cache Root Directory	76
Days to Wait Before Deleting Unused Cache Files	76
<b>Queuing &gt; Application Server Settings &gt; Interactive Work</b>	<b>76</b>
Configuration Name	76
Server Context Name	76

Run Interactive Jobs . . . . .	76
Minimum Workspace Server Threads . . . . .	77
Maximum Workspace Server Threads . . . . .	77
Minimum Stored Process Server Threads . . . . .	77
Maximum Stored Process Server Threads . . . . .	77
<b>Queuing &gt; Application Server Settings &gt; Batch Work . . . . .</b>	<b>77</b>
Configuration Name . . . . .	77
Server Context Name . . . . .	77
Run Interactive Jobs . . . . .	78
Minimum Workspace Server Threads . . . . .	78
Maximum Workspace Server Threads . . . . .	78
Minimum Stored Process Server Threads . . . . .	78
Maximum Stored Process Server Threads . . . . .	78
<b>Queuing &gt; Application Server Settings &gt; Generic Task Queue . . . . .</b>	<b>78</b>
Minimum Threads . . . . .	78
Maximum Threads . . . . .	79
<b>Queuing &gt; Application Server Settings &gt; Information Service Task Queue . . . . .</b>	<b>79</b>
Minimum Threads . . . . .	79
Maximum Threads . . . . .	79
<b>Queuing &gt; Logging . . . . .</b>	<b>79</b>
Job Log Root Directory . . . . .	79
<b>Queuing &gt; Housekeeping . . . . .</b>	<b>79</b>
Days Before Normal Jobs Expire . . . . .	79
Hours Old Before System Job Logs Expire . . . . .	80
Minutes Between Job Control Synch Runs . . . . .	80
<b>Queuing &gt; Job Control Table Synchronization . . . . .</b>	<b>80</b>
Job Control Table Library Name . . . . .	80
Path to Job Synchronization Lock Stored Process . . . . .	80
<b>Queuing &gt; SAS Share Server . . . . .</b>	<b>80</b>
Use SHARE Server for Mid Tier . . . . .	80
<b>Queuing &gt; Teradata Server . . . . .</b>	<b>81</b>
Use Teradata Server from Mid Tier . . . . .	81
Connection URL . . . . .	81
Teradata JDBC Driver Class Name . . . . .	81
Database User ID . . . . .	81
Database User Password . . . . .	81
<b>Table Display . . . . .</b>	<b>81</b>
Lookup Table Page Size . . . . .	81
Maximum Table Pages for PDF Export . . . . .	81
<b>Other Property Values That You Can Override . . . . .</b>	<b>82</b>
Overview . . . . .	82
General Initialization . . . . .	82
SAS Warranty Analysis Thin Client Initialization . . . . .	83
Security . . . . .	94
Application Server . . . . .	95
Storage Location Names . . . . .	96
Job Control Table Synchronization . . . . .	96
Numeric Formatting Options . . . . .	98
Report Generation . . . . .	99
Miscellaneous . . . . .	100

---

## Overview

This chapter explains how to change the configuration of various system properties that are used by the SAS Warranty Analysis middle tier and used throughout the system.

*Note:* The default values listed in the following sections refer to the values that the system provides if you do not override the property values in SAS Management Console.

Here are the general steps to specify SAS Warranty Analysis properties in SAS Management Console:

1. In SAS Management Console, view the properties for **Warranty Analysis 4.3**.

For information about how to do this, see the SAS Management Console Help about viewing and updating repository properties.

2. On the **Settings** tab, select a branch.

The branches and properties in this chapter are presented in the order in which they appear in SAS Management Console.

---

## 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 probably do not want the SAS Warranty Analysis content-sharing user interface 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. In SAS Management Console, add the property `sharing.identityFilter.name` repeatedly to create a sequential list of user accounts and user group names to subtract from all those available.

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

### **CAUTION:**

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

```
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=SAS Warranty Analysis View Users
sharing.identityFilter.name11=SAS Warranty Analysis Restricted Users
sharing.identityFilter.name12=SAS Warranty Analysis Normal Users
```

```

sharing.identityFilter.name13=SAS Warranty Analysis EI Administrators
sharing.identityFilter.name14=SAS Warranty Analysis Full Administrators
sharing.identityFilter.name15=Performance Testing
sharing.identityFilter.name16=Portal Admins
sharing.identityFilter.name17=Portal Demo

```

*Note:* The default SAS Warranty Analysis system groups are automatically excluded.

---

## Analysis

### ***Maximum Individual Points to Analyze as Subset***

This property specifies the maximum number of individual points allowed to analyze as a subset.

The default value of **Maximum Individual Points to Analyze as Subset** is 10.

---

## Filter Subset Management

### ***Number of Days Before Inactive Filter Subset is Deleted***

This property specifies the number of days after which an unused filter subset is automatically deleted.

To disable automatic deletion, specify the value zero (0). You can specify a single value, or you can specify a comma-delimited list to offer a choice to users in the SAS Warranty Analysis rich client. To enable users to disable automatic deletions, include zero (0) in the list.

The default value of **Number of Days Before Inactive Filter Subset is Deleted** is 0.

---

## Identities > Groups

### ***Overview***

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

For more information, see [Chapter 5, “Security,” on page 23](#).

### ***View Users***

This property specifies the group name for the SAS Warranty Analysis View Users user group.

The default value of **View Users** is **SAS Warranty Analysis View Users**.

***Restricted Users***

This property specifies the group name for the SAS Warranty Analysis Restricted Users user group.

The default value of **Restricted Users** is **SAS Warranty Analysis Restricted Users**.

***Normal User***

This property specifies the group name for the SAS Warranty Analysis Normal Users user group.

The default value of **Normal User** is **SAS Warranty Analysis Normal Users**.

***Emerging Issues Administrative Users***

This property specifies the group name for the SAS Warranty Analysis Merging Issues Administrators user group.

The default value of **Emerging Issues Administrative Users** is **SAS Warranty Analysis EI Administrators**.

***Full Administrative Users***

This property specifies the group name for the SAS Warranty Analysis Full Administrators user group.

The default value of **Full Administrative Users** is **SAS Warranty Analysis Full Administrators**.

***Deny Emerging Issues Access Users***

This property specifies the group name for the SAS Warranty Analysis Deny EI Access user group.

The default value of **Deny Emerging Issues Access Users** is **SAS Warranty Analysis Deny EI Access**.

---

## **Identities > Roles**

***Overview***

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

***Information Consumer***

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

The default value of **Information Consumer** is **Warranty Analysis: Information Consumption**.

### ***Data Analyst***

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

The default value of **Data Analyst** is **Warranty Analysis: Data Analysis**.

### ***Data Administrator***

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

The default value of **Data Administrator** is **Warranty Analysis: Data Administration**.

### ***Emerging Issues Administrator***

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

The default value of **Emerging Issues Administrator** is **Warranty Analysis: EI Administration**.

### ***Full Administrator***

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

The default value of **Full Administrator** is **Warranty Analysis: System Administration**.

### ***Deny Emerging Issues Access***

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

The default value of **Deny Emerging Issues Access** is **Warranty Analysis: Deny EI Access**.

---

## **Information Map Security**

### ***Use Information Map Security***

This property specifies whether the SAS Warranty Analysis middle tier applies information map-based column-level security to analysis variables.

The default value of **Use Information Map Security** is **False**.



**Libname Statement for Workspace Server**

This property specifies the Base SAS LIBNAME statement used to run queries for the secured library.

The default value of **Libname Statement for Workspace Server** is **LIBNAME {0} SASIOIME MAPPATH={1} metacredentials=no**.

**Libname Statement for SHARE Server**

This property specifies the SAS/Share LIBNAME statement used to run queries for the secured library.

The default value of **Libname Statement for SHARE Server** is **LIBNAME {0} SASIOIME MAPPATH={1} metauser={2} metapass={3}**.

**Path to Securing Information Map**

This property specifies the location of the information map definition.

The default value of **Path to Securing Information Map** is **' /System/Applications/SAS Warranty Analysis/Warranty Analysis 4.3'**.

**Information Map Library Name**

This property specifies the Base SAS library that contains information map security.

The default value of **Information Map Library Name** is **SECMAP**.

## Logging

**SQL Logging**

This property specifies whether every SQL statement run by the system is written to the SAS Analytics Platform console output. These SQL statements are not currently written to the job log files.

The default value of **SQL Logging** is **False**.

**XML Logging**

This property specifies whether the analysis options and analysis output XML file is written to the job log. This XML information is not currently written to the SAS Analytics Platform console output.

The default value of **XML Logging** is **False**.

---

## Mid Tier Caching

### **Cache Root Directory**

This property specifies the fully-qualified path in which the SAS Warranty Analysis middle tier builds its directory structure to store a file cache to increase overall system performance.

The default value of **Cache Root Directory** for Microsoft Windows is `<SASConfig>\Lev1\Web\Logs\SASWarrantyAnalysis4.3\cache`.

The default value of **Cache Root Directory** for UNIX is `<SASConfig>/Lev1/Web/Logs/SASWarrantyAnalysis4.3/cache`.

### **Days to Wait Before Deleting Unused Cache Files**

This property specifies the number of days that must elapse after the last access of a SAS Warranty Analysis middle-tier cache file before it can be automatically deleted.

The default value of **Days to Wait Before Deleting Unused Cache Files** is 5.

---

## Queuing > Application Server Settings > Interactive Work

### **Configuration Name**

This property specifies the name of the interactive configuration.

The default (and required) value of **Configuration Name** is `SASApp_Interactive`.

### **Server Context Name**

This property specifies the name of the SAS Application Server context to use for SAS Warranty Analysis thread pools. This value must match the name specified in SAS Management Console (**Environment Management** ⇌ **Server Manager**).

The default value of **Server Context Name** is `SASApp`.

#### **CAUTION:**

**Do not change this value unless instructed by on-site SAS Support Personnel.**

### **Run Interactive Jobs**

This property specifies whether the value specified for the property **Configuration Name** runs interactive jobs.

The default value of **Run Interactive Jobs** is `True`.

**Minimum Workspace Server Threads**

This property specifies the minimum number of server threads to run for workspaces.

The default value of **Minimum Workspace Server Threads** is 10.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

**Maximum Workspace Server Threads**

This property specifies the maximum number of server threads to run for workspaces.

The default value of **Maximum Workspace Server Threads** is 10.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

**Minimum Stored Process Server Threads**

This property specifies the minimum number of server threads to run for stored processes.

The default value of **Minimum Stored Process Server Threads** is 6.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

**Maximum Stored Process Server Threads**

This property specifies the maximum number of server threads to run for stored processes.

The default value of **Maximum Stored Process Server Threads** is 6.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

---

## Queuing > Application Server Settings > Batch Work

**Configuration Name**

This property specifies the name of the batch configuration.

The default (and required) value of **Configuration Name** is `SASApp_Batch`.

**Server Context Name**

This property specifies the name of the SAS Application Server context to use. This value must match the name specified in SAS Management Console (**Environment Management** ⇒ **Server Manager**).

The default value of **Server Context Name** is `SASApp`.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

### ***Run Interactive Jobs***

This property specifies whether the value specified for the property Configuration Name runs interactive jobs.

The default value of **Run Interactive Jobs** is **True**.

### ***Minimum Workspace Server Threads***

This property specifies the minimum number of server threads to run for workspaces.

The default value of **Minimum Workspace Server Threads** is **10**.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

### ***Maximum Workspace Server Threads***

This property specifies the maximum number of server threads to run for workspaces.

The default value of **Maximum Workspace Server Threads** is **10**.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

### ***Minimum Stored Process Server Threads***

This property specifies the minimum number of server threads to run for stored processes.

The default value of **Minimum Stored Process Server Threads** is **6**.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

### ***Maximum Stored Process Server Threads***

This property specifies the maximum number of server threads to run for stored processes.

The default value of **Maximum Stored Process Server Threads** is **6**.

**CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

---

## **Queuing > Application Server Settings > Generic Task Queue**

### ***Minimum Threads***

This property specifies the minimum number of server threads to run for generic tasks.

The default value of **Minimum Threads** is 10.

### **Maximum Threads**

This property specifies the maximum number of server threads to run for generic tasks.

The default value of **Maximum Threads** is 10.

---

## **Queuing > Application Server Settings > Information Service Task Queue**

### **Minimum Threads**

This property specifies the minimum number of server threads to run for information service tasks.

The default value of **Minimum Threads** is 10.

### **Maximum Threads**

This property specifies the maximum number of server threads to run for information service tasks.

The default value of **Maximum Threads** is 10.

---

## **Queuing > Logging**

### **Job Log Root Directory**

This property specifies the root folder under which all job logs are written.

The default value of **Job Log Root Directory** is (Microsoft Windows) *SASConfig\Lev1\Web\Logs\SASWarrantyAnalysis4.3\logs* and (UNIX) *SASConfig/Lev1/Web/Logs/SASWarrantyAnalysis4.3/logs*.

---

## **Queuing > Housekeeping**

### **Days Before Normal Jobs Expire**

This property specifies the number of days that a normal log file must exist before it can be automatically deleted.

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

**Hours Old Before System Job Logs Expire**

This property specifies the number of hours that a system log file must exist before it can be automatically deleted.

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

**Minutes Between Job Control Synch Runs**

This property specifies the number of minutes that must pass between runs of the JCT synchronization process.

The default value of `queue.jobControlTableSynchIntervalMinutes` is **2**.

---

## Queuing > Job Control Table Synchronization

**Job Control Table Library Name**

This property specifies the SAS library that contains the queuing job control tables.

The default value of **Job Control Table Library Name** is **PARMSL**.

**Path to Job Synchronization Lock Stored Process**

This property specifies the path in the SAS folders (within SAS Management Console) in which the store process definition is located. Queuing calls this location in order to lock the job control table for update.

The default value of **Path to Job Synchronization Lock Stored Process** is `/System/Applications/SAS Warranty Analysis/Warranty Analysis 4.3/Stored Processes/SAS Warranty Analysis driver`.

**CAUTION:**

**Do not change this value unless instructed by on-site SAS Support Personnel.**

---

## Queuing > SAS Share Server

**Use SHARE Server for Mid Tier**

This property specifies whether the SAS/SHARE server is used via its JDBC driver from the middle tier.

If you specify the value of **False**, all SQL queries go through the pooled workspace server, which requires more pooled workspace server connections.

The default value of **Use SHARE Server for Mid Tier** is **True**.

---

## Queuing > Teradata Server

### ***Use Teradata Server from Mid Tier***

This property specifies whether the SAS middle tier tries to use the Teradata server.

The default value of **Use Teradata Server from Mid Tier** is **False**.

### ***Connection URL***

This property specifies the JDBC connection string that enables the SAS middle tier to communicate with the Teradata JDBC driver.

The default value of **Connection URL** is blank.

### ***Teradata JDBC Driver Class Name***

This property specifies the Java class name for the Teradata JDBC driver.

The default value of **Teradata JDBC Driver Class Name** is **com.ncr.teradata.TeraDriver**.

### ***Database User ID***

This property specifies the user ID used for logging into the Teradata system.

The default value of **Database User ID** is blank.

### ***Database User Password***

This property specifies the user password used for logging into the Teradata system.

The default value of **Database User Password** is blank.

---

## Table Display

### ***Lookup Table Page Size***

This property specifies the maximum number of rows that a lookup table displays per page.

The SAS Warranty Analysis system does not show large amounts of data in one scrollable screen.

The default value of **Lookup Table Page Size** is **300**.

### ***Maximum Table Pages for PDF Export***

This parameter specifies the maximum number of pages that a table can occupy in a PDF file.

The SAS Warranty Analysis system does not write all of the rows of a large table into a PDF document. If a user wants to see all of the rows, then the client program's paging table views can be used.

The default value of **Maximum Table Pages for PDF Export** is 2.

---

## Other Property Values That You Can Override

### Overview

You can add additional properties to SAS Management Console to override the default values for certain aspects of SAS Warranty Analysis. You add these properties and their values on the **Advanced** tab of the Warranty Analysis Properties dialog box in SAS Management Console. For more information, see the SAS Management Console online Help.

### General Initialization

#### Overview

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

#### ***metaserver.trusteduserid***

This property 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 specified when the SAS Configuration Wizard runs.

#### ***metaserver.trusteduserpw***

This property 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 specified when the SAS Configuration Wizard runs.

#### ***metaserver.deploymentname***

This property 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 specified when the SAS Configuration Wizard runs.

#### ***metaserver.authdomainname***

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

The default value of metaserver.repositoryname is **Foundation**.

**share.selectHostName**

This property specifies the name of the SAS/SHARE server to use with SAS Warranty Analysis. You need to specify this value only when there are multiple SAS/SHARE servers configured in the environment.

*Note:* After changing this value, you must restart the Web application server that contains the SAS Warranty Analysis Web application.

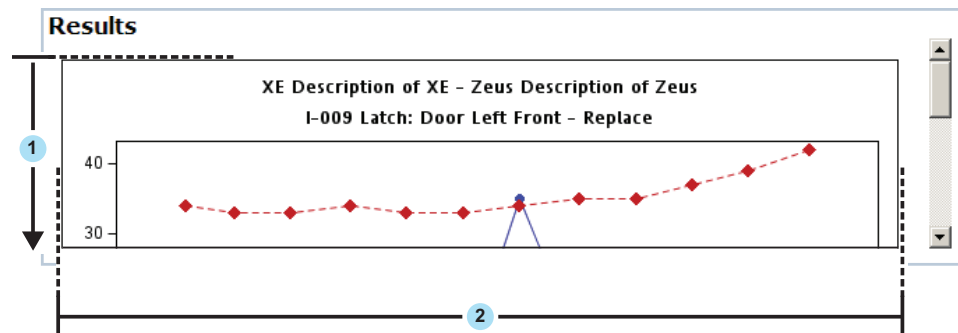
**SAS Warranty Analysis Thin Client Initialization****Overview**

The user interface initialization properties specify certain aspects of the SAS Warranty Analysis thin client user interface.

**alerts.graphics.height**

This property specifies the height in pixels for an Emerging Issues reports graph. If the graph is taller than the frame, vertical scroll bars are displayed.

**Figure 12.1** Illustration of alerts.graphics.height and alerts.graphics.width



1 alerts.graphics.height

2 alerts.graphics.width

For PDF output in the SAS Warranty Analysis thin client or rich client, this property specifies the height in pixels for an Emerging Issues reports graph. If the graph is taller than alerts.graphics.height, the graph is clipped.

The default value of alerts.graphics.height is **400**.

**alerts.graphics.width**

This property specifies the width in pixels for an Emerging Issues reports graph. If the graph is wider than the frame, horizontal scroll bars are displayed.

For an illustration of alerts.graphics.width, see [Figure 12.1 on page 83](#).

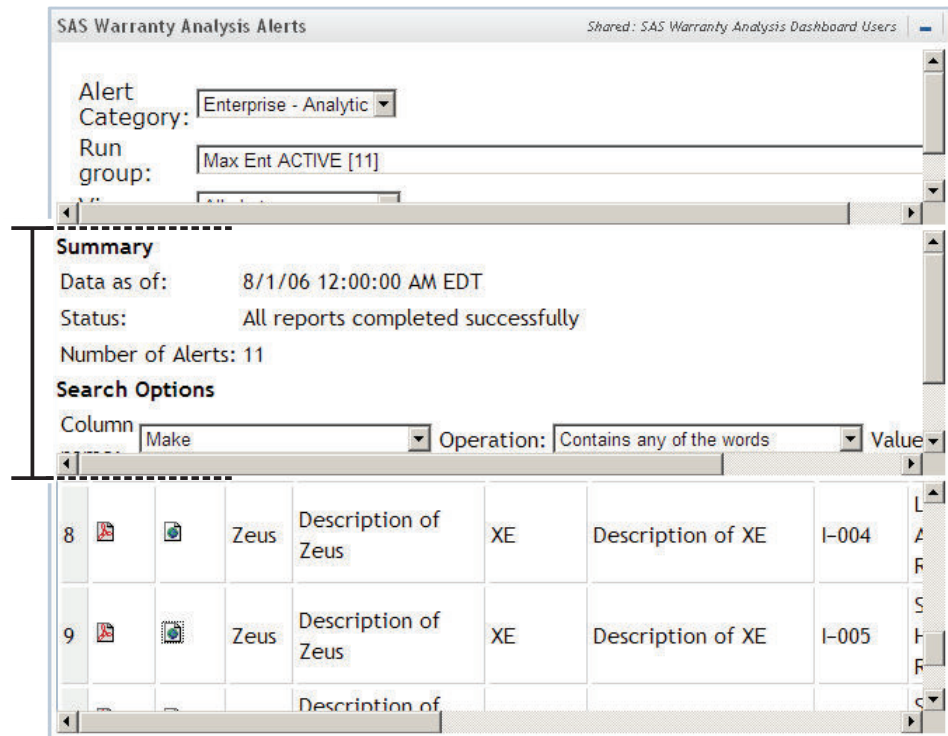
For PDF output in the SAS Warranty Analysis thin client or rich client, this property specifies the width in pixels for an Emerging Issues reports graph. If the graph is wider than `alerts.graphics.width`, the graph is clipped.

The default value of `alerts.graphics.width` is 600.

### ***eiFilterTableHeight***

This property specifies the height in pixels of the frame that contains the controls to search for Emerging Issues alerts.

**Figure 12.2** Illustration of *eiFilterTableHeight*

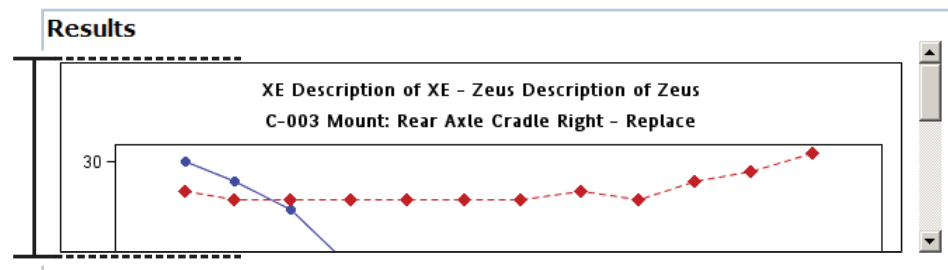


The default value of `eiFilterTableHeight` is 175.

### ***eiGraphHeight***

This property specifies the scaling height of the frame that contains a graph when there is no table displayed with a graph in the HTML view of an Emerging Issues alert.

**Figure 12.3** Illustration of *eiGraphHeight*

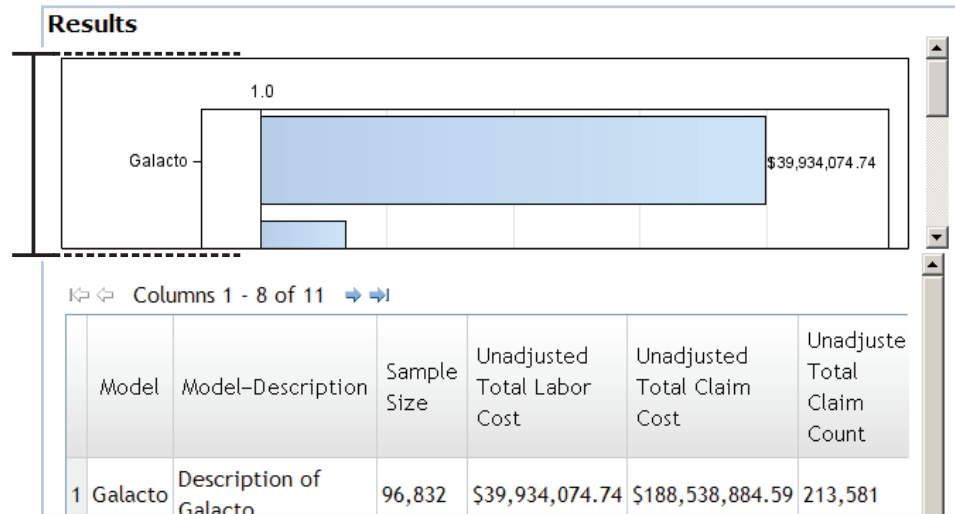


The default value of `eiGraphHeight` is 100%.

***eiGraphHeightWithTableView***

This property specifies the scaling height of the frame that contains a graph when there is a table displayed with a graph in the HTML view of an Emerging Issues alert.

**Figure 12.4** Illustration of *eiGraphHeightWithTableView*

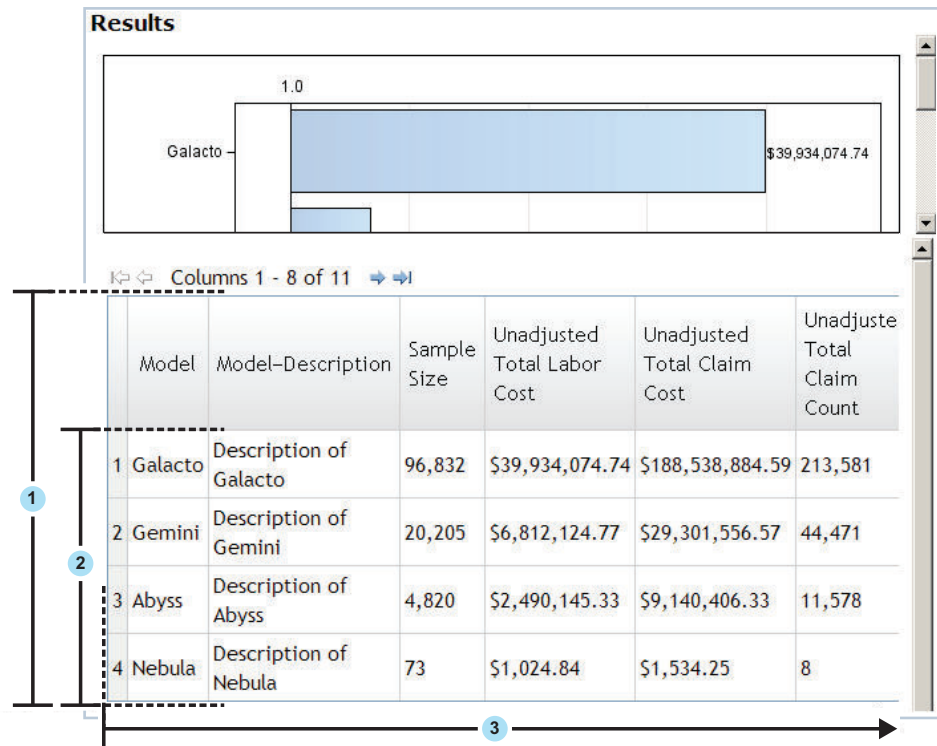


The default value of *eiGraphHeightWithTableView* is 50%.

***eiGraphTableColumnPageSizeValue***

This property specifies the number of columns to display in the table that is displayed in the HTML view of an Emerging Issues alert. If the table is wider than the frame that contains it, horizontal scroll bars appear.

**Figure 12.5** Illustration of *eiGraphTableColumnPageSizeValue*, *eiGraphTableRowPageSizeValue*, and *eiGraphTableHeight*



- 1 *eiGraphTableHeight*
- 2 *eiGraphTableRowPageSizeValue*
- 3 *eiGraphTableColumnPageSizeValue*

Tables appear most commonly in Ad Hoc - Threshold alerts.

The default value of *eiGraphTableColumnPageSizeValue* is 8. To display all columns, specify the value -1.

### ***eiGraphTableHeight***

This property specifies the scaling height of each table that is displayed in the HTML view of an Emerging Issues alert.

For an illustration of *eiGraphTableHeight*, see [Figure 12.5 on page 86](#).

The default value of *eiGraphTableHeight* is 100%.

### ***eiGraphTableRowPageSizeValue***

This property specifies the number of rows to display in a table that is displayed in the HTML view for an Emerging Issues alert. If the table is taller than the frame that contains it, vertical scroll bars appear.

For an illustration of *eiGraphTableRowPageSizeValue*, see [Figure 12.5 on page 86](#).

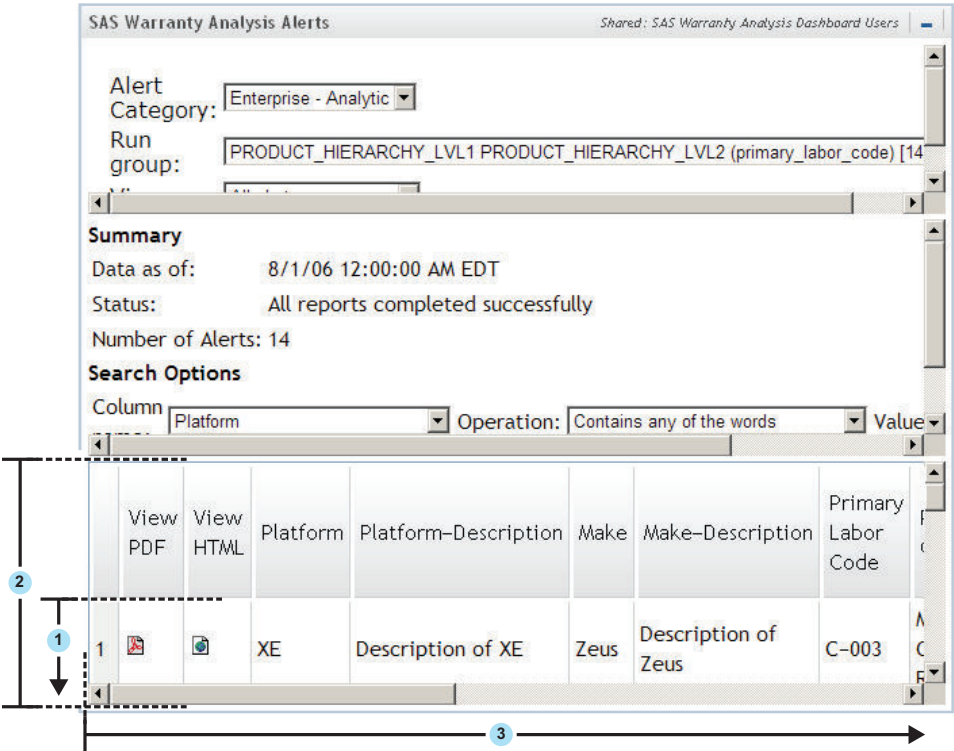
Tables appear most commonly in Ad Hoc - Threshold alerts.

The default value of *eiGraphTableRowPageSizeValue* is 10. To display all rows, specify the value -1.

**eiTableColumnPageSizeValue**

This property specifies the number of columns per page to display in the Emerging Issues alerts table.

**Figure 12.6** Illustration of eiTableColumnPageSizeValue, eiTableHeight, and eiTableRowPageSizeValue



- 1 eiTableColumnPageSizeValue
- 2 eiTableHeight
- 3 eiTableRowPageSizeValue

The default value of eiTableColumnPageSizeValue is -1. To display all columns, specify the value -1.

**eiTableHeight**

This property specifies the height in pixels of the frame that contains the Emerging Issues alerts table.

For an illustration of eiTableHeight, see [Figure 12.6 on page 87](#).

The default value of eiTableHeight is 180.

**eiTableRowPageSizeValue**

This property specifies the number of rows of data to display in the Emerging Issues alerts table.

For an illustration of eiTableRowPageSizeValue, see [Figure 12.6 on page 87](#).

The default value of eiTableRowPageSizeValue is 50. To display all rows, specify the value -1.

***jre.download.url.anchor.url***

This property specifies the URL to the Java runtime environment (JRE) download page, which is shown when there is no JRE installed on the client computer.

The default value of `jre.download.url.anchor.url` is ' `href=http://java.sun.com/products/archive/j2se/5.0_12/index.html` '.

***initial-heap-size***

This property specifies the initial amount of memory allocated to the SAS Warranty Analysis client.

The default value of `initial-heap-size` is **64M**.

See also “[max-heap-size](#)” on page 89.

***linksTableHeight***

This property specifies the height in pixels of the frame that displays the links table. The links table appears in the SAS Information Delivery portal on the **SAS Warranty Analysis Home Page** tab.

**Figure 12.7** Illustration of `linksTableHeight`

SAS Warranty Analysis News

Shared: SAS Warranty Analysis Dashboard Users

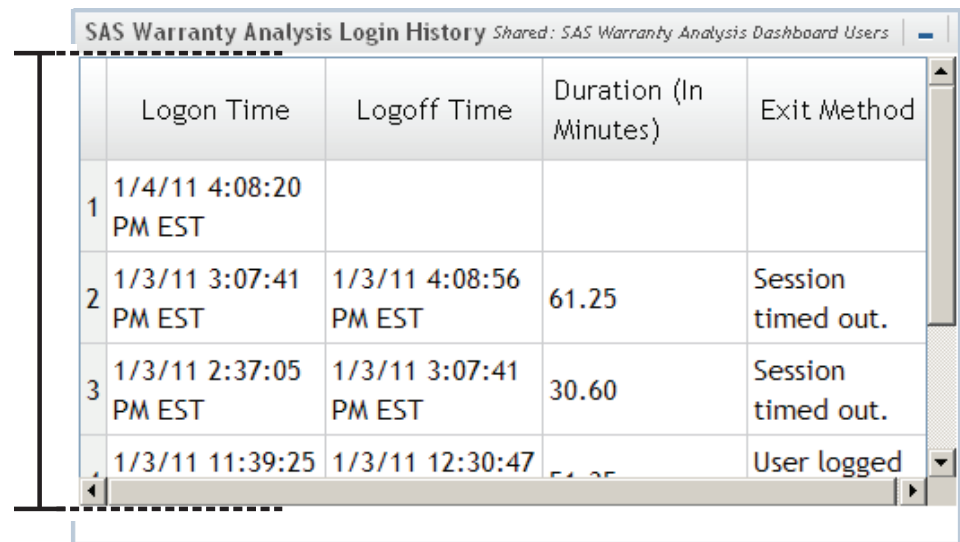
	Data Source ID	Warehouse Refresh Date	ETL Refresh Date
1	1	8/1/06 12:00:00 AM EDT	8/1/06 12:00:00 AM EDT

Message Text	Expiration Date
No messages found.	

The default value of `linksTableHeight` is **200**.

***loginHistoryTableHeight***

This property specifies the height in pixels of the frame that displays the login history table.

**Figure 12.8** Illustration of loginHistoryTableHeight


	Logon Time	Logoff Time	Duration (In Minutes)	Exit Method
1	1/4/11 4:08:20 PM EST			
2	1/3/11 3:07:41 PM EST	1/3/11 4:08:56 PM EST	61.25	Session timed out.
3	1/3/11 2:37:05 PM EST	1/3/11 3:07:41 PM EST	30.60	Session timed out.
4	1/3/11 11:39:25	1/3/11 12:30:47	51.25	User logged

The default value of loginHistoryTableHeight is 200.

### **max-heap-size**

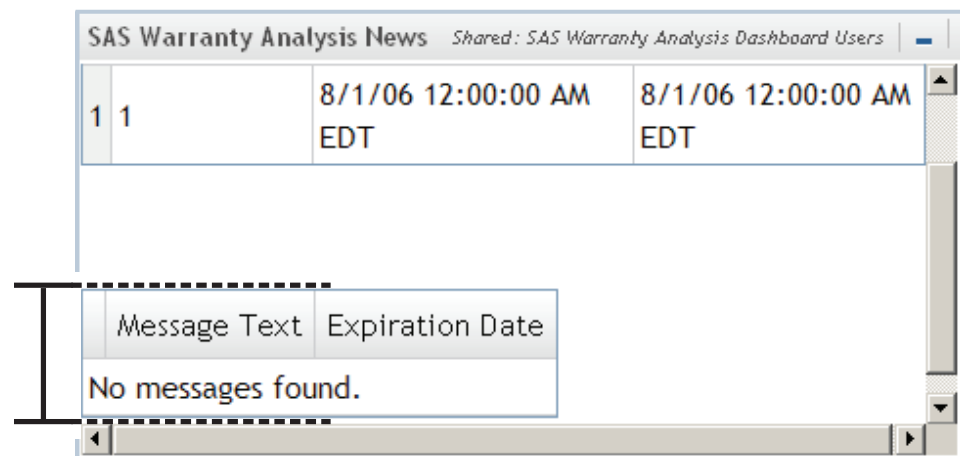
This property specifies the maximum amount of memory allocated to the SAS Warranty Analysis client.

The default value of metaserver.authdomainname is 512M.

See also “initial-heap-size” on page 88.

### **messagesTableHeight**

This property specifies the height in pixels of the frame that displays the messages table. The messages table appears in the SAS Information Delivery Portal on the **SAS Warranty Analysis Home Page** tab.

**Figure 12.9** Illustration of messagesTableHeight


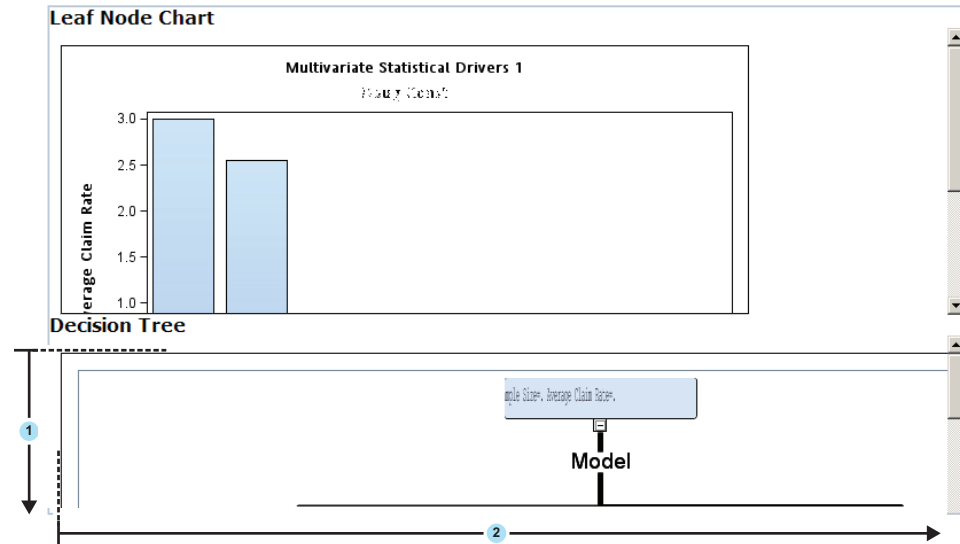
	Message Text	Expiration Date
1 1	8/1/06 12:00:00 AM EDT	8/1/06 12:00:00 AM EDT

The default value of messagesTableHeight is 200.

***public.reports.graphics.DecisionTree.height***

This property specifies the height in pixels for the decision tree graph in a multivariate analysis. This value applies to both HTML and PDF versions on the multivariate analysis.

**Figure 12.10** Illustration of *public.reports.graphics.DecisionTree.height* and *public.reports.graphics.DecisionTree.width* in HTML view



1 *public.reports.graphics.DecisionTree.height*

2 *public.reports.graphics.DecisionTree.width*

The default value of *public.reports.graphics.DecisionTree.height* is 800.

***public.reports.graphics.DecisionTree.width***

This property specifies the width in pixels for the decision tree graph in a multivariate analysis. This value applies to both HTML and PDF versions on the multivariate analysis.

For an illustration, see [Figure 12.10 on page 90](#).

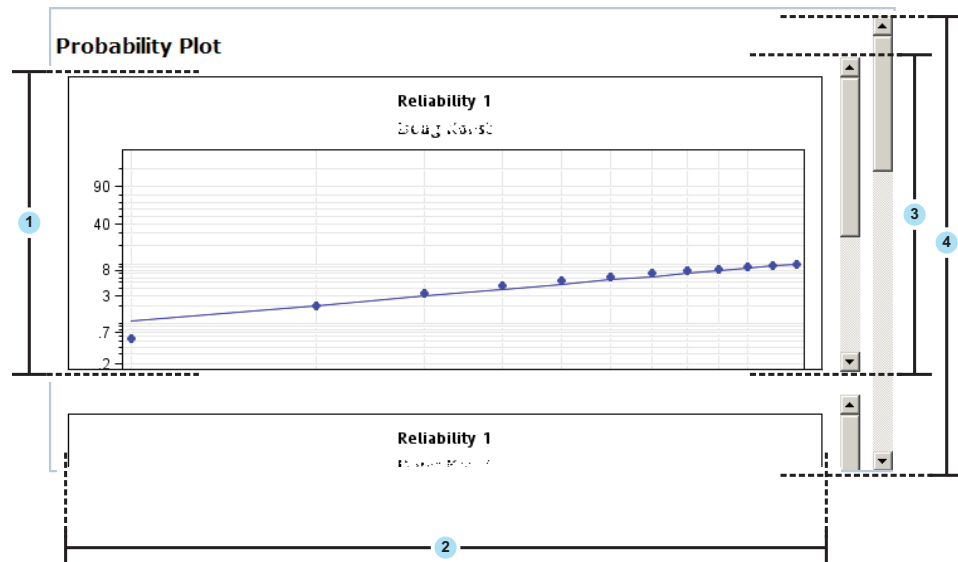
The default value of *public.reports.graphics.DecisionTree.width* is 1200.

***public.reports.graphics.height***

This property specifies the height in pixels for a graph in a report.



**Figure 12.11** Illustration of `public.reports.graphics.height`, `public.reports.graphics.width`, `publicReportsGraphHeight`, and `publicReportsAllGraphsTableHeight`



- 1 `public.reports.graphics.height`
- 2 `public.reports.graphics.width`
- 3 `publicReportsGraphHeight`
- 4 `publicReportsAllGraphsTableHeight`

The default value of `public.reports.graphics.height` is 400.

### **`public.reports.graphics.width`**

This property specifies the width in pixels for graphs in reports.

For an illustration see, [Figure 12.11 on page 91](#).

The default value of `public.reports.graphics.width` is 600.

### **`public.reports.subsetselectiondetails.attribute.allvalues.height`**

This property specifies the number of lines that are displayed in the Attribute Selections browser window that displays all attribute selections. If the amount of text is taller than `public.reports.subsetselectiondetails.attribute.allvalues.height`, vertical scroll bars appear.

The default value of `public.reports.subsetselectiondetails.attribute.allvalues.height` is 30.

### **`public.reports.subsetselectiondetails.attribute.allvalues.width`**

This property specifies the number of columns of characters that are displayed in the Attribute Selections browser window that displays all attribute selections in the SAS Warranty Analysis thin client. If the text is wider than `public.reports.subsetselectiondetails.attribute.allvalues.width`, the text is wrapped.

The default value of `public.reports.subsetselectiondetails.attribute.allvalues.width` is 120.

### **`publicReportsAllGraphsTableHeight`**

This property specifies the scaling height of the frame that displays all of the graphs (in their individual frames) in a report.

For an illustration see, [Figure 12.11 on page 91](#).

The default value of `publicReportsAllGraphsTableHeight` is **45%**.

### ***publicReportsAllGraphsTableHeightWithoutTable***

This property specifies the scaling height of the frame that displays the output of a report that contains no tables.

The default value of `publicReportsAllGraphsTableHeightWithoutTable` is **60%**.

### ***publicReportsGraphHeight***

This property specifies the height in pixels of the frame that contains a graph in a report.

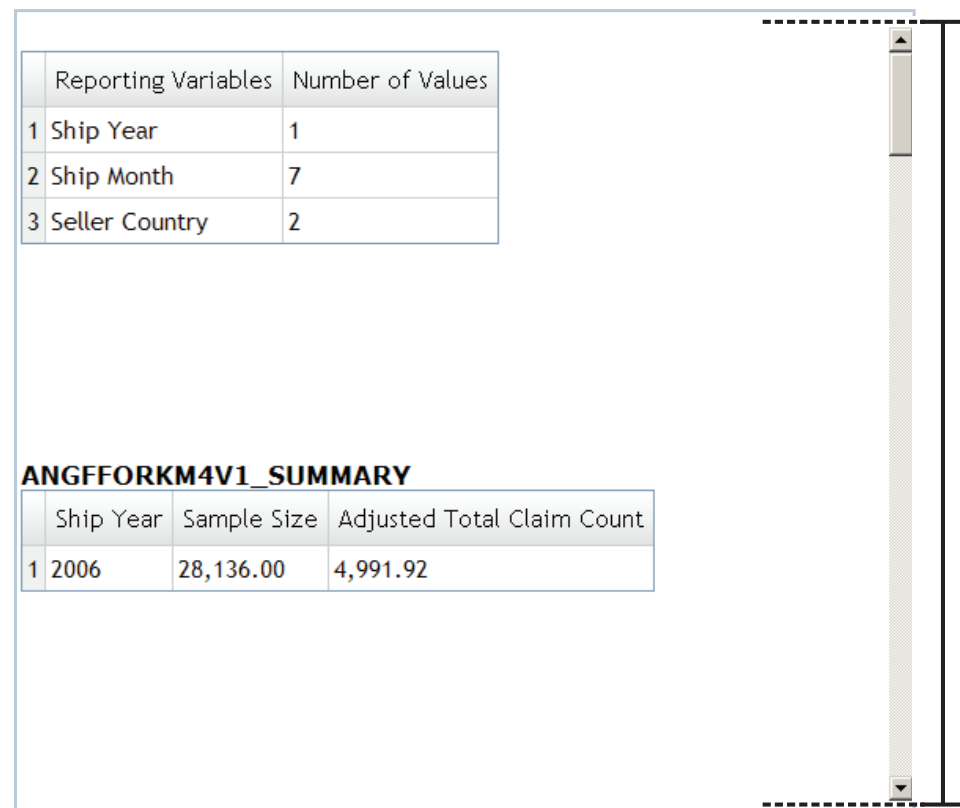
For an illustration, see [Figure 12.11 on page 91](#).

The default value of `publicReportsGraphHeight` is **250**.

### ***publicReportsGraphTableHeight***

This property specifies the scaling height of the frame that displays tables in a report that contains only tables.

**Figure 12.12** Illustration of `publicReportsGraphTableHeight`



	Reporting Variables	Number of Values
1	Ship Year	1
2	Ship Month	7
3	Seller Country	2

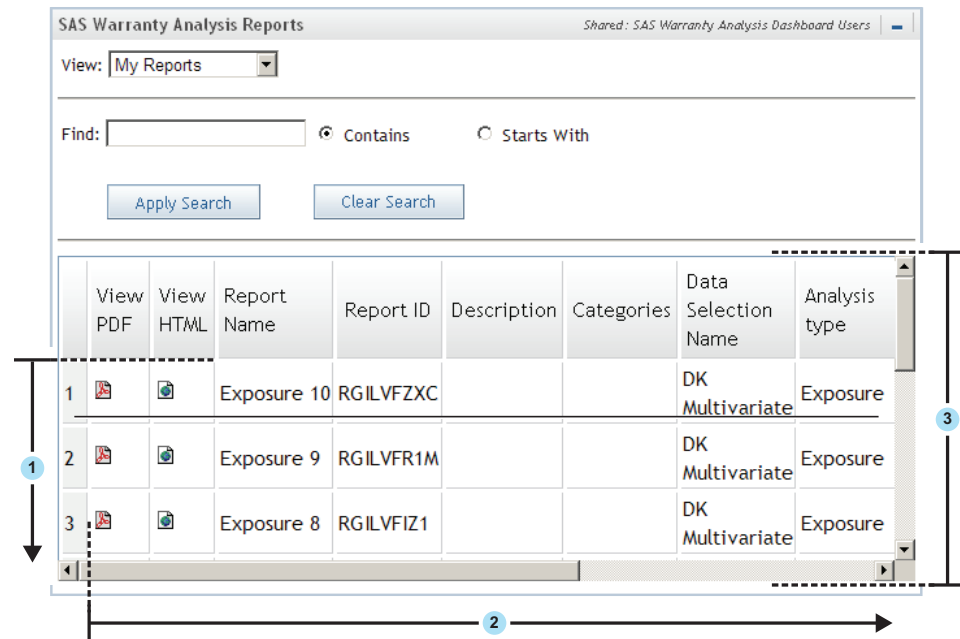
ANGFFORKM4V1_SUMMARY			
	Ship Year	Sample Size	Adjusted Total Claim Count
1	2006	28,136.00	4,991.92

The default value of `publicReportsGraphTableHeight` is **100%**.

### ***publicReportsTableColumnPageSizeValue***

This property specifies the number of columns that are displayed in the reports table.

**Figure 12.13** Illustration of `publicReportsTableRowPageSizeValue`, `publicReportsTableColumnPageSizeValue`, and `publicReportsTableHeight`



- 1 `publicReportsTableRowPageSizeValue`
- 2 `publicReportsTableColumnPageSizeValue`
- 3 `publicReportsTableHeight`

The default value of `publicReportsTableColumnPageSizeValue` is `-1`. To display all columns, specify the value `-1`.

### ***publicReportsTableHeight***

This property specifies the height in pixels of the frame that contains the table displaying the list of reports. If the list is longer than `publicReportsTableHeight`, vertical scroll bars appear.

For an illustration, see [Figure 12.13 on page 93](#).

The default value of `publicReportsTableHeight` is `250`.

### ***publicReportsTableRowPageSizeValue***

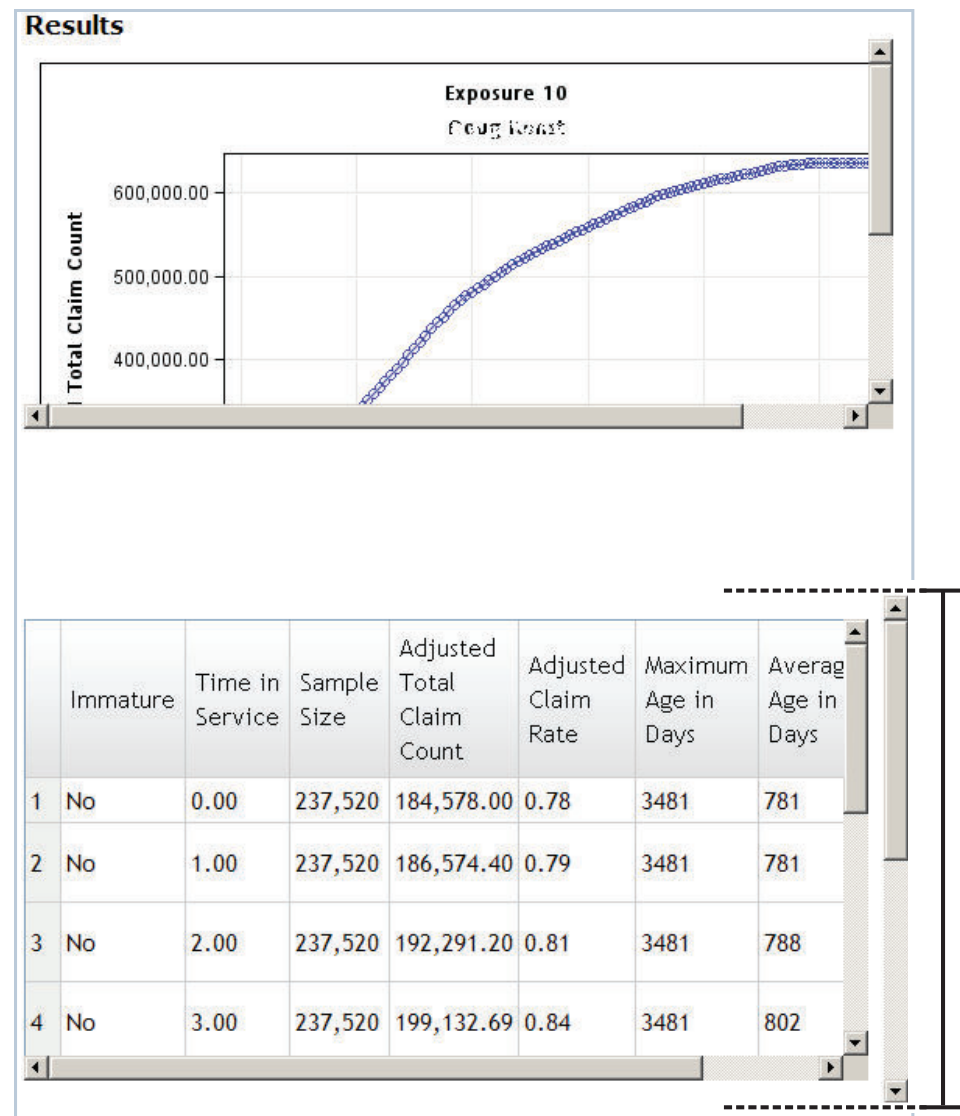
This property specifies the number of rows that are displayed in the reports table.

For an illustration, see [Figure 12.13 on page 93](#).

The default value of `publicReportsTableRowPageSizeValue` is `10`.

### ***publicReportsTableViewHeight***

This property specifies the scaling height of the frame that displays tables in a report that contains both tables and graphs.

**Figure 12.14** Illustration of `publicReportsTableViewHeight`

The default value of `publicReportsTableViewHeight` is **40%**.

### ***publicReportsTableViewHeightWithoutGraph***

This property specifies the scaling height of the frame that displays a Details Table analysis report, which contains no graphs.

The default value of `publicReportsTableViewHeightWithoutGraph` is **60%**.

## **Security**

### ***security.loginIdsCaseSensitive***

By default, SAS Warranty Analysis treats logon IDs as case insensitive. The default works fine in a Microsoft Windows environment. However, in a UNIX environment, you should change this value to **True** because UNIX authentication respects the case of the logon ID.

The default value of `security.loginIdsCaseSensitive` is **False**.

***user.login.history.threshold.limit***

SAS Warranty Analysis keeps track of logon attempts. This property 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.

***Application Server******Overview***

The application server properties specify the application server configuration properties and thread pooling properties.

***queue.genericservice.maxthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.genericservice.maxthreads` is 10.

***queue.genericservice.minthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.genericservice.minthreads` is 10.

***queue.informationsservice.maxthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.informationsservice.maxthreads` is 10.

***queue.informationsservice.minthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.informationsservice.minthreads` is 10.

***queue.shareserver.maxthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.shareserver.maxthreads` is 10.

***queue.shareserver.minthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.shareserver.minthreads` is 10.

***queue.teradataserver.minthreads******CAUTION:***

**Do not change this value unless instructed by on-site SAS Support Personnel.**

The default value of `queue.teradataserver.minthreads` is 10.

**queue.teradataserver.maxthreads****CAUTION:**

Do not change this value unless instructed by on-site SAS Support Personnel.

The default value of queue.teradataserver.maxthreads is 10.

**Storage Location Names****Overview**

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.

For more information, see [Chapter 5, “Security,” on page 23](#).

**warranty.analyticReportTreeName**

This property specifies the folder name to match the corresponding folder in 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.dataSelectionTreeName**

This property specifies the folder name to match the corresponding folder in 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.projectTreeName**

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

The default value of warranty.projectTreeName is **Projects**.

**Job Control Table Synchronization****Overview**

The job control table (JCT) synchronization is a process that the SAS Warranty Analysis middle tier runs every few minutes (see [“Minutes Between Job Control Synch Runs” on page 80](#)). The configuration properties 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 are 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 property, 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.

### ***jobDensity.reallySlowJobCutoff***

This property specifies the threshold to classify a job as very slow.

The Job Density view multiplies the `jobDensity.reallySlowJobCutoff` value by the reference job run time for jobs of the same type. If the job's run time exceeds this calculated value, then the job is considered a very slow job.

The default value of `jobDensity.reallySlowJobCutoff` is 5.

### ***jobDensity.slowJobCutoff***

This property specifies the threshold to classify a job as slow.

The Job Density view multiplies the `jobDensity.SlowJobCutoff` value by the reference job run time for jobs of the same type. If the job's run time is less than this calculated value, then the job is considered to be running at specification. If the job's run time exceeds this calculated value but is less than the `jobDensity.reallySlowJobCutoff` value, then the job is considered a slow job.

The default value of `jobDensity.slowJobCutoff` is 1.5.

### ***queue.housekeepingIntervalMinutes***

This property specifies 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.logging.jobLogNormalExpirationDaysOld***

This property specifies the number of days that a normal job log file must exist before it is automatically deleted.

The default value of `queue.logging.jobLogNormalExpirationDaysOld` is 5.

### ***queue.logging.jobLogSystemExpirationHoursOld***

This property specifies the number of hours that a system job log file must exist before it is automatically deleted.

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

### ***queue.logLocation***

This property specifies the middle-tier file location where the queuing job log files are written.

There is no default value.

**share.isActive**

This property enables the SAS Warranty Analysis middle tier to run JDBC queries directly against the SAS Share server.

This property specifies whether the SAS Warranty Analysis middle tier increases the number of users without having to increase the number of workspace server connections for the pooled workspace server. Share connections use fewer system resources than do workspace server connections.

The default value of share.isActive is **False**.

**queue.finishedJobExpirationIntervalMinutes**

This property specifies 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 property specifies 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 **120**.

## Numeric Formatting Options

**Overview**

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

**default.decimal.fractiondigits**

This property specifies the default fraction digits that are used in rounding the decimal values that are 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.decimal.fractiondigits is **0.001**.

**default.currency.fractiondigits**

This property specifies how many decimal fraction digits are used to round 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 property specifies how many decimal fraction digits are used to round 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.

To specify a value, supply an example of the number of digits. For example, to specify one digit, enter the value **.1**. To specify two digits, enter the value **.01**.

The default value of default.percent.fractiondigits is **0**.



***default.integer.fractiondigits***

This property specifies how many decimal fraction digits are used to round 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.

To specify a value, supply an example of the number of digits. For example, to specify one digit, enter the value `.1`. To specify two digits, enter the value `.01`.

The default value of `default.integer.fractiondigits` is `0`.

**Report Generation*****Overview***

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

***maxRowsForPdfExport***

This property specifies the maximum number of rows in a tables in a PDF report.

The default value of `maxRowsForPdfExport` is `300`.

***pdfgen.default.fontSize***

This property specifies the point size of all text in a PDF report.

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

***pdfgen.default.fontName***

This property specifies the font used in a PDF report.

The default value of `pdfgen.default.fontName` is **Arial Unicode MS**.

***pdfGen.pageSizeCode***

This property specifies the page size for a PDF report.

The default value of `pdfGen.pageSizeCode` is **LETTER**.

***report.autoUpdateCounter***

Each report that a user creates uses batch processing time. 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 property 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 `10,000`.

***report.deleteAfterNDays***

This property specifies the number of days that must elapse before an emerging issues ad hoc analytic or threshold report is automatically deleted.

The default value of `report.deleteAfterNDays` is `60`.

## Miscellaneous

### ***allowFilterSubsetDeleteFromUI***

This property enables users in the rich client to delete filter subsets within the Disk Usage utility.

The default value of `allowFilterSubsetDeleteFromUI` is **True**.

### ***eiAnalyticAlertTablePageSize***

This property specifies the number of rows of data retrieved per page for lookup value tables for Emerging Issues analytic alerts. If you enlarge the **Results** pane to show more rows in the table, the number of rows that show data is the value of `eiAnalyticAlertTablePageSize`.

The default value of `eiAnalyticAlertTablePageSize` is **20**.

*Note:* Increasing this value slows the responsiveness of the system but allows for a greater amount of data to be viewed at one time.

Compare to “[eiTableRowPageSizeValue](#)” on page 87.

### ***ETL.messageCounter***

This property specifies 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.

### ***logging.stripHtml***

This property 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**.

### ***metaserver.trusteduserlocale***

This property 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 82). This property is seldom used for batch reports, because they take on the locale of the analysis.

The default value of `metaserver.trusteduserlocale` is **en\_US**.

### ***queuing.stats***

This property 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**. By default, this value is commented out.

### ***sas.stps.driver.program***

This property specifies the path in the SAS Metadata Server to the SAS Warranty Analysis stored process that is used to start 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, on-site SAS support personnel change this value.

***sas.wsps.driver.program***

This property specifies the path in the SAS Metadata Server to the SAS Warranty Analysis stored process that is used to start 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, on-site SAS support personnel change this value.

***sharing.identityFilter.name***

This property specifies the name of a user account or user group to exclude from the SAS Warranty Analysis content-sharing user interface. For more information, see [“Sharing Groups” on page 71](#).



## Chapter 13

# Working With Technical Support

---

<b>Overview .....</b>	<b>103</b>
<b>Record the Problem Context .....</b>	<b>103</b>
<b>General System Questions .....</b>	<b>104</b>
<b>Gathering Information from the Log Files .....</b>	<b>104</b>

---

## Overview

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

Phone: 919-677-8008

E-mail: [support@sas.com](mailto: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. Include the steps that caused the problem.
2. Include a description of the data selection, if applicable, and its name.
3. Provide screen shots that illustrate the problem.
4. Provide a description of any recent changes or disruptions in the environment, such as the following:
  - 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 types 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 filename, which suggests the operation that was in progress when an error occurred.

For instructions to look up log files, see “[Anatomy of a Job Log File Listing](#)” on page 45. 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 `<SASConfig>/Lev1/Web/Logs/SASWarrantyAnalysis4.3/logs` directory and find the file in the user’s directory.





## Appendix 1

# Troubleshooting

---

<b>An Analysis Fails to Execute . . . . .</b>	<b>107</b>
<b>The Rich Client Fails to Open from the SAS Warranty Analysis Landing Page .</b>	<b>107</b>
<b>Deletion of a Data Selection Fails . . . . .</b>	<b>108</b>
<b>A User Is Granted Access to SAS Warranty Analysis, but the System Malfunctions . . . . .</b>	<b>108</b>
<b>Any User Logging into the System Receives the “System Experienced Fatal Error” Message . . . . .</b>	<b>108</b>
<b>The User Interface Is Unresponsive or an Interactive Job Does Not Progress . .</b>	<b>108</b>
<b>A Batch Job Does Not Progress . . . . .</b>	<b>109</b>
<b>JobSynchLock Is Locked . . . . .</b>	<b>109</b>

---

## 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:** Verify that full rights have been granted 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. For more information, see [Chapter 5, “Security,” on page 23](#).

---

## Any User Logging into the System Receives the “System Experienced Fatal Error” Message

**Problem:** If the “System Experienced Fatal Error” message displays, then it is likely that SAS Warranty Analysis failed to start properly.

**Solution:** Look in the error logs for specific information. Your on-site SAS support personnel 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 Job Monitoring** area) 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`**.

