



SAS[®] Visual Analytics 7.4: Administration Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS® Visual Analytics 7.4: Administration Guide*. Cary, NC: SAS Institute Inc.

SAS® Visual Analytics 7.4: Administration Guide

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

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

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

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

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

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

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

December 2018

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

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

7.4-P1:vaag

Contents

<i>Using This Book</i>	v
<i>What's New</i>	vii
<i>Accessibility</i>	ix
Chapter 1 / Getting Started	1
Orientation	1
Adding Users	3
Operating Servers	6
Chapter 2 / Loading Data	9
About Loading Data	10
Get Table Information	12
Administer LASR Tables	14
Administrator Load	15
Self-Service Import	16
Reload-on-Start	18
Autoload	22
Chapter 3 / Security	35
Permissions	36
Access to In-Memory Data	41
Row-Level Security	44
Key Actions Auditing	48
Locked-Down Servers	56
Access to SAS Mobile BI	56
Authentication	60
Encryption	61
Chapter 4 / Fine-Tuning	69
Administering SAS Home	70
Supporting Guest Access	73
Supporting Text Analytics	77
Supporting Geographic Maps	78
Supporting Stored Processes	80
Supporting Footers in Printed Reports	80
Supporting Report Distribution	82
Supporting the Search Features	83
Supporting User-Defined Formats	83
Managing Alerts and Notifications	84
Supporting the Monitoring Features	85
Adjusting the Logging Configuration	87
Making More Fonts Available on UNIX	89
Using Multiple SAS Application Servers	90
Setting User Preferences	91
Chapter 5 / SAS LASR Analytic Server	93
About SAS LASR Analytic Server	94
LASR-Related Metadata	95

In-Memory LASR Names	96
Add a SAS LASR Analytic Server	97
Add a LASR Library	98
Limit Space for Tables	100
Extended Attributes: SAS LASR Analytic Server	101
Advanced Options: SAS LASR Analytic Server	102
Distributed Server: Monitoring	105
Distributed Server: Parallel Load	108
Distributed Server: Co-located HDFS	111
Distributed Server: Legacy Co-located Providers	117
Distributed Server: High-Volume Access to Smaller Tables	117
Chapter 6 / Reports for Administrators	121
About the Predefined Reports	121
How to Provide Administrative Data	123
About auditRefresh	127
Appendix 1 / Reference	131
Software Components	131
Roles and Capabilities	133
Configuration Properties	139
High-Cardinality Constraints	148
Predefined LASR Libraries	153
Sample Objects	157
Appendix 2 / Adding Custom Polygon Data for Geographic Maps	161
Overview of Custom Polygon Data	161
Prerequisites for Adding Custom Polygon Data	162
Geographic Data Sets	162
Creating Custom Polygons	166
Testing Custom Polygons in SAS Visual Analytics	168
Example 1: Deploying German Postal Codes	169
Example 2: US Counties	175
Appendix 3 / Troubleshooting	181
Troubleshooting: SAS Visual Analytics	181
Troubleshooting: SAS Mobile BI	187
Permission Origins	188
Glossary	191
Index	193

Using This Book

Audience

This book documents administration of SAS Visual Analytics in a traditional on-premises deployment.

Documentation Conventions

SAS Configuration Directory

The phrase *SAS configuration directory* refers to a host path that includes a configuration name and level.

UNIX example:	<code>/opt/sas/config/Lev1</code>
---------------	-----------------------------------

Windows example:	<code>C:\sas\Config\Lev1</code>
------------------	---------------------------------

For more information, see [Overview of the Configuration Directory Structure](#) in the *SAS Intelligence Platform: System Administration Guide*.

Note: For directory paths that are identical on UNIX and Windows, this book uses UNIX style path delimiters (/ instead of \).

Short Forms and Labels

The following table documents the short-form terminology that is used in this book.

Long Form	Short Form	User Interface Labels*
SAS Home	the home page	Home
SAS Visual Analytics Hub		SAS Home
SAS Visual Analytics Administrator	the administrator	Administrator (Manage Environment)

Long Form	Short Form	User Interface Labels*
SAS Visual Analytics Explorer**	the explorer	Data Explorer (Create Exploration)
SAS Visual Analytics Designer	the designer	Report Designer (Create Report)
SAS Visual Analytics Graph Builder	the graph builder	Custom Graph Builder
SAS Visual Analytics Viewer	the web viewer	Report Viewer
SAS Visual Analytics Transport Service	transport service	-
SAS Visual Data Builder	the data builder	Data Preparation (Prepare Data) (Create Data Query)
distributed SAS LASR Analytic Server	distributed server	-
non-distributed SAS LASR Analytic Server	non-distributed server	-
SAS LASR Analytic Server library	LASR library	-
SAS LASR Analytic Server table	LASR table	-
Hadoop Distributed File System	HDFS	-

* Labels in parentheses are used only in the classic (Flash) presentation mode.

** Not all deployments of SAS Visual Analytics include the explorer.

What's New

What's New

General Administration

- You can define standard footer text that is added to all printed reports. See [“Supporting Footers in Printed Reports” on page 80](#).
- You can control whether missing values are ignored in counts when the Distinct operator is used in the designer and viewers. See the [va.IgnoreMissingInCountDistinct property on page 142](#).
- Custom polygon data can be added for geographic maps. See [Appendix 2, “Adding Custom Polygon Data for Geographic Maps,” on page 161](#).

Accessibility

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

x *What's New*

1

Getting Started

Orientation	1
Tasks	1
Tools	2
Adding Users	3
About Adding Users	3
How to Add a User	3
Host Account Privileges	5
Access Management	6
Results	6
Operating Servers	6
Operate a SAS LASR Analytic Server	6
Autostart a SAS LASR Analytic Server	7
Operate Other Servers	8

Orientation

Tasks

Get familiar with the software architecture.	See “Software Components” on page 131.
Get familiar with the software functionality.	See About SAS Visual Analytics in the <i>SAS Visual Analytics: User’s Guide</i> .
Register users.	See “Adding Users” on page 3.
Make data available.	See “About Loading Data” on page 10.
Ensure that backups occur.	See About Backups and Restores in the <i>SAS Intelligence Platform: System Administration Guide</i> .

Tools

The Administrator

Most tasks are performed in the administrator, a SAS Visual Analytics web application. To open the administrator, select **Administrator** or **Manage Environment** from the home page (for example, `http://host/SASVisualAnalyticsHub`), side menu, or banner.

For the exact URL, see the file [SAS-configuration-directory/Documents/Instructions.html](#) on the middle-tier machine. In the `Instructions.html` file, search for `SAS Visual Analytics Administrator`.

UNIX example: `/opt/sas/config/Levl/Documents/Instructions.html`

Windows example: `C:\sas\Config\Levl\Documents\Instructions.html`

Troubleshooting

Issue: You are redirected from the administrator's direct URL to the home page.

Resolution: Make sure that you have the required capabilities. In the standard configuration, members of the Visual Analytics Data Administrators group can access the administrator. See the instructions for the **Groups and Roles** tab in ["How to Add a User"](#) on page 3.

SAS Management Console

Some tasks are performed in SAS Management Console, a desktop application. To open SAS Management Console, use one of the following instructions:

UNIX example: From `/install/SASServer/SASHome/SASManagementConsole/9.4`, run `./sasmc`.

Windows example: From the **Start** button, select **All Programs** ► **SAS** ► **SAS Management Console**.

For more information, see [Administering SAS Management Console](#) in the *SAS Intelligence Platform: Desktop Application Administration Guide*.

SAS Environment Manager Administration

As an alternative to using SAS Management Console, you can use SAS Environment Manager (a web application) to set permissions, add users, add SAS LASR Analytic Servers, and add LASR libraries. To open SAS Environment Manager, go to `http://host:7080`, where *host* is the machine where the SAS Environment Manager server is installed. To access the metadata administrative functionality, select **Administration** from SAS Environment Manager's main menu.

For the exact URL, see the file [SAS-configuration-directory/Documents/Instructions.html](#) on the middle-tier machine. In the `Instructions.html` file, search for `SAS Environment Manager Web Console`.

For more information, see [SAS Environment Manager Administration: User's Guide](#).

TIP For information about using SAS Environment Manager to monitor resources, see [SAS Environment Manager: User's Guide](#).

Other Administrative Tools

For information about other administrative tools, see [Overview of the Administration Tools](#) in the *SAS Intelligence Platform: System Administration Guide*.

Adding Users

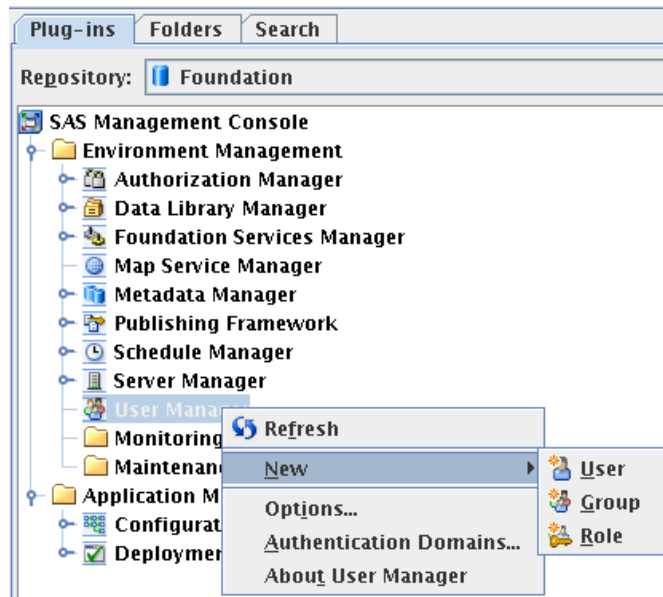
About Adding Users

The following instructions document one way to register a user. For alternatives, see [About User Administration](#) in the *SAS Intelligence Platform: Security Administration Guide*.

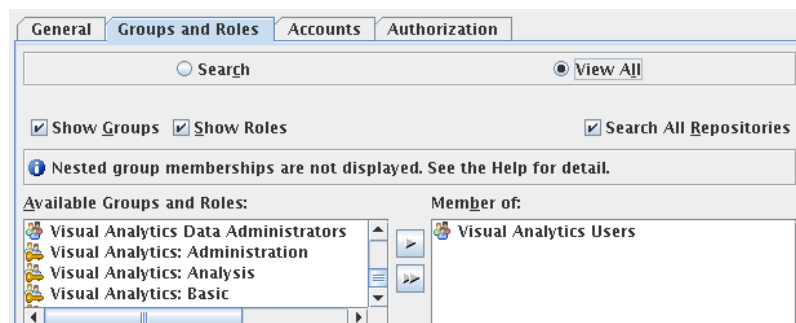
TIP If guest access is enabled, you do not have to register users who need only limited, anonymous access. See [“Supporting Guest Access” on page 73](#).

How to Add a User

- 1 Identify or create an account with which the user can access the SAS Metadata Server.
Note: In the simplest case, accounts are known to the metadata server's host. A metadata server on Windows usually authenticates users against Active Directory. A metadata server on UNIX might authenticate users against LDAP.
Note: If the user imports data, loads data, or starts and stops servers, make sure that the user's account has the necessary privileges. See [“Host Account Privileges” on page 5](#).
- 2 Log on to [SAS Management Console](#) as an administrator (for example, sasadm@saspw).
- 3 On the **Plug-ins** tab, right-click **User Manager**, and select **New** ► **User**.



- 4 On the **General** tab, enter a name for the user.
- 5 On the **Groups and Roles** tab, add direct memberships for the new user:
 - If the user does not perform administrative tasks, move the **Visual Analytics Users** group to the **Member of** list.



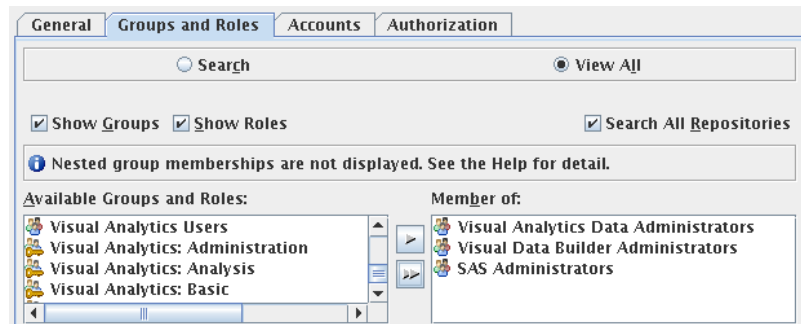
- If the user performs administrative tasks, move one or more of the following groups to the **Member of** list:

Visual Analytics Data Administrators (for suite-level administrative tasks)

Visual Data Builder Administrators (for data preparation tasks)

SAS Administrators (for platform-level administrative tasks)

TIP Unless you know you want to create a limited administrator, move all three groups to the **Member of** list.



- 6 On the **Accounts** tab, click **New** to add a login.
 - a Enter the user ID for the account from step 1. It is not necessary to store a password.
Windows Specifics: Enter the user ID in a fully qualified format (*user-ID@domain.extension*, *domain\user-ID*, or *machine\userID*).
 - b Select the **DefaultAuth** authentication domain. Click **OK**.
Note: If you know that web authentication has been set up, select the **web** authentication domain instead. Other authentication domains might be available at your site. See [Authentication Domains](#) in the *SAS Intelligence Platform: Security Administration Guide* for more information.
- 7 In the New User Properties window, click **OK**.

Host Account Privileges

Introduction

The requirements in this section apply to accounts that are used to import data, load data, or start and stop a SAS LASR Analytic Server. The requirements do not apply to users who only design reports, explore data, and view reports.

Host Directories

The account must be able to write to the [signature files](#) directory, the [va.lastActionLogPath](#) directory, and the PIDs directory that is beneath the [va.monitoringPath](#) directory.

SAS LASR Analytic Server

The account must be able to authenticate to the host for the [SAS LASR Analytic Server](#).

- For a non-distributed server, in most cases, no action is necessary. The credentials with which a user initially signs in are reused for authentication to the SAS LASR Analytic Server. For more complex environments, see [“Authentication” on page 60](#).
- For a distributed server, give the account passwordless SSH access to all of the machines in the cluster. See [Passwordless SSH](#) in the *SAS LASR Analytic Server: Reference Guide*.

Windows Compute Tier

To use a workspace server that runs on Windows, the account must have the local security policy **Log on as a batch job**. In a multi-machine deployment, set the policy on the compute tier (the machine that hosts the workspace server).

If an operating system group (such as SAS Server Users) has this policy, add the user's account to that group. Otherwise, see [Windows Privileges](#) in the *SAS Intelligence Platform: Security Administration Guide*.

Access Management

For registered users who have appropriate memberships, no access-related changes are required. To set up custom access patterns or troubleshoot any problems, see [“Permissions” on page 36](#), [“About Capabilities” on page 133](#), and [“Access to SAS Mobile BI” on page 56](#).

Results

To validate a registration, ask the user to sign in to the home page (<http://host/SASVisualAnalyticsHub>), and verify that the expected functionality is available.

For troubleshooting, see [“Access Issues” on page 181](#).

Operating Servers

Operate a SAS LASR Analytic Server

Get Server Information

To get status and other information for a SAS LASR Analytic Server:

- 1 From the main menu in the administrator, select **LASR ► Manage Servers**.
- 2 Select a server, right-click, and select **Get Status**.


The screenshot shows a window titled "LASR Servers" with a toolbar and a table. The table has columns for Server, Status, Tables Memory (MB), Tables Limit (MB), Host, Port, Started By, Description, and Started. Two servers are listed: "LASR Analytic Server" and "Public LASR Analytic Server". Both have a green status indicator and are running.

Server	Status	Tables Memory (MB)	Tables Limit (MB)	Host	Port	Started By	Description	Started
<input checked="" type="checkbox"/> LASR Analytic Server	●	4.29					SAS LASR Analytic se...	12/22/14 09:15 AM
<input checked="" type="checkbox"/> Public LASR Analytic Server ...	●	3,418.30					Visual Analytics Publi...	12/22/14 09:30 AM

Here are some details:

- The **Status** column indicates whether the server is running ●, stopped ■, or over capacity ▲.

Note: A server is over capacity when its tables memory value equals or exceeds its tables limit value. A server that is over capacity accepts requests for activities such as data retrieval and analysis, but rejects requests to load, import, append, or reload tables.



- The **Tables Limit** column can constrain the amount of memory that the server can use to host tables. By default, the cells in this column are blank, so no constraints are in effect. See [“Limit Space for Tables” on page 100](#).
- For a distributed server, a **Virtual Memory** column (not depicted) indicates how much of the total cluster memory is currently in use by each server process. See [“Distributed Server: Monitoring” on page 105](#).
- To get status and other information for multiple servers, select check boxes, and then click  in the tab toolbar.
- To show or hide a column, right-click on any column heading, and select the column.

Note: The displayed values are calculated as if all LASR star schemas are output as tables. See [Creating LASR Star Schemas](#) in the *SAS Visual Analytics: User's Guide*.

Start or Stop a Server

- 1 From the main menu in the administrator, select **LASR ► Manage Servers**.
- 2 Select a server, right-click, and select **Start** or **Stop**.

Here are some details:

- If you click  in the tab toolbar, all servers that have a selected (checked) check box are started.
- If you click  in the tab toolbar, all servers that have a selected (checked) check box are stopped.
- Starting a server reloads only those tables that participate in reload-on-start. See [“Reload-on-Start” on page 18](#).
- Stopping a server unloads all of its tables. See [“Server lifetime” on page 105](#).

Results

To view a log for the most recent interactive action for a server, right-click on the server, and select **Last Action Log**.

TIP On Windows, you can use keyboard commands to select the content of a last action log (Ctrl+A), copy the content (Ctrl+C), and paste the content elsewhere (Ctrl+V).

For troubleshooting, see [“Server Operation Issues” on page 183](#).

Autostart a SAS LASR Analytic Server

Introduction

A SAS LASR Analytic Server can start on demand if one or more of the server's LASR libraries enable autostart. Requests to an autostart-enabled LASR library start the associated SAS LASR Analytic Server if all of the following conditions are met:

- The server is not already running.



- The requesting user has the necessary privileges.
- The request is for a load or import action. Requests to open a data source, read data, or run a data query do not trigger autostart.

In the standard configuration, autostart is enabled for the **Visual Analytics Public LASR** library. To enable autostart for another library, use either of the following methods.

SAS Management Console

- 1 In [SAS Management Console](#), right-click on a LASR library, and select **Properties**.
- 2 On the **Extended Attributes** tab, set VA.AutoLoad.AutoStart to **yes**.

SAS Environment Manager

- 1 In the [SAS Environment Manager Administration](#) window, click or tap , and select **Libraries** from the side menu.
- 2 Open a LASR library.
- 3 From the library's drop-down list of property pages, select the **Extended Attributes** property page.
- 4 In the **Value** cell for VA.AutoLoad.AutoStart, enter **yes**.
- 5 Click .

Operate Other Servers

In addition to the SAS LASR Analytic Server, SAS Visual Analytics uses metadata, middle-tier, and compute servers that are provided by the underlying platform. See [“Software Components” on page 131](#).

Here are basic instructions for restarting the platform:

UNIX: From your equivalent of `/opt/sas/config/Lev1`, run `./sas.servers restart`.

Windows: Restart the machine.

If you have multiple machines, complete the preceding basic instruction on each machine, beginning with the machine that hosts the metadata server. Make sure that the metadata server is running before you proceed to other machines.

For details, exceptions, and alternatives, see [Operating Your Servers](#) in the *SAS Intelligence Platform: System Administration Guide*.

2

Loading Data

About Loading Data	10
Introduction	10
Load Methods	10
Reload Methods	11
Data Compression	11
Table and Column Names	12
Get Table Information	12
Administer LASR Tables	14
Unload, Reload, or Delete a Table	14
Replace a Source Table	14
Results	14
Additional Considerations	14
Administrator Load	15
Preparation	15
Load a Table	16
Self-Service Import	16
Introduction	16
Requirement: User Privileges	16
Requirement: SAS/ACCESS	17
How to Protect Imported Data	17
How to Limit Import Size	18
Reload-on-Start	18
Introduction	18
How Reload-on-Start Works	19
How to Enable Reload-on-Start	19
Additional Considerations	20
Reference	21
Autoload	22
Introduction	22
How Autoload Works	22
The Autoload Directories	23
Timing of Autoload	23
How to Start Autoload	24
How to Stop Autoload	25
How to Add an Implementation	25
Additional Considerations	30
Reference	30

About Loading Data

Introduction

Users can easily import data. See [Overview of Data Flow in SAS Visual Analytics](#) in the *SAS Visual Analytics: User's Guide*. This chapter documents administrative aspects of data loading.

SAS Visual Analytics uses data that is loaded to memory in a SAS LASR Analytic Server. Tables remain in memory until they are unloaded or the associated server stops. The following features help keep data available:


Table 2.1 *Optional Convenience Features*

Feature	Trigger	Result (Automated Action)
Autoload	A time interval elapses	In-memory data synchronizes against a designated directory.
Autostart	A load or import is requested	The associated server starts.
Reload-on-start	A server starts	Participating tables reload.

TIP See [SAS Visual Analytics: Getting Started with Data on Windows](#) for orientation and recommendations for beginners.

Load Methods

Load methods vary by data source.



Load Method	Data Source				
	Spreadsheet or Delimited	SAS Data Set	SASHDAT File*	Data Server	Other**
 Interactive load		✓	✓	✓	
Run a data query		✓	✓	✓	
Import from server		✓		✓	✓
Import a local file	✓	✓			
Autoload	✓	✓			

* Co-located HDFS or NFS-mounted MapR. See [“Distributed Server: Parallel Load”](#) on page 108.

** Data from Twitter, Google Analytics, or Facebook.

Reload Methods

Reload methods depend on how a table was initially loaded.

Reload Method	Eligible LASR Tables
 Interactive reload	<p>Tables that were interactively loaded ().</p> <p>Output from data queries.*</p> <p>Output from LASR star schemas.*</p> <p>SAS data sets that were imported from a server.</p>
Reload-on-start	<p>Participating tables from imports of local files.</p> <p>Participating tables from imports of Google Analytics, Facebook, or Twitter data.</p>

* Any input LASR tables must be available (loaded).

Reloading data requires access to either the current source data or a backing store copy of the original source data.

- Interactive reload runs against the current source data (using a job or query that was created by the initial load).

Note: Any data that was appended only in memory is not included in the reload.

- Reload-on-start runs against a copy of the original source data (using a data provider library that functions as a backing store).

To make a table that is not reloadable available:

- If the table was autoloading, wait for the next run of the scheduled task.
- Otherwise, repeat the action that initially loaded the data.

Data Compression

Effects of Compression

There is a trade-off between compression and performance. Compressing data conserves memory. However, it might take longer to retrieve data from a table that is compressed. See [Data Compression](#) in the *SAS LASR Analytic Server: Reference Guide*.

Support for Compression

Here is a summary of when compression occurs:

- Administrators and data builders can request compression when they load a table.

Note: If you load a SASHDAT file, you cannot request compression. The compression setting that already exists in the source SASHDAT file is honored. An exception is that an encrypted SASHDAT file is always

uncompressed when it is loaded. See [“On-Disk Encryption of SASHDAT Files” on page 64](#).

- Administrators and data builders can request compression when they add a table to co-located HDFS or NFS-mounted MapR.
- Administrators can request or remove compression for a LASR table, by using the **Change Source** action. See [“Replace a Source Table” on page 14](#).
- Data builders can request compression when they import a table or run a data query that outputs to a LASR table, co-located HDFS, or NFS-mounted MapR.
- Users who have access to the **Advanced** panel (in the designer or explorer) can request compression when they import a table. See the [“Build Data”](#) capability.
- Administrators can use an extended attribute to request compression of autoloaded data. See [“VA.AutoLoad.Compress.Enabled” on page 33](#).
- In reload-on-start, compression is used for tables that were compressed when they were initially loaded.
- Compression does not occur for small tables.
- Compression does not occur for tables that are loaded from encrypted SASHDAT files.

Table and Column Names

In general, names can include spaces and special characters. Exceptions include the following:

- For interactions with third-party data sources and operating systems, third-party name limitations apply.
- For LASR table names, the period character (.) is not supported. If you load a SAS data set that has a period in its name, the period is replaced with an underscore (_).
- SAS name limitations apply. See [Summary of Extended Rules for Naming SAS Data Sets and SAS Variables](#) in the *SAS Language Reference: Concepts*.

Note: When data is imported as a local file or autoloaded, any character that is not supported by SAS is replaced with an underscore.

Note: Column names in reports are case sensitive. If you change the case of a column name in a LASR table, then any reports that use that column will display an error message.

Get Table Information

To get information about a LASR table:

- 1 From the main menu in the administrator, select **LASR** ► **Manage Tables**.
- 2 Select a table, right-click, and select **Get Status**.

Table	Status	Size	Location	Server	Loaded	Description
<input checked="" type="checkbox"/> AUDIT_VISUALANALYTICS		3.81 MB	/Shared Data/SAS Visual ...	LASR Ana...	12/22/14 09:15 AM	Auto loaded December 22, 2014 9:15:13 AM
<input checked="" type="checkbox"/> ADVERSE		2.13 KB	/Shared Data/		12/23/14 05:37 AM	Adverse
<input checked="" type="checkbox"/> ALERT_INVESTIGATION...		1.26 MB	/Shared Data/L		12/23/14 05:37 AM	Alert investigation extract

Here are some details:

- To make sure you are seeing the most current information, repeat the get status action.
- To get information about multiple tables, select check boxes, and then click in the tab toolbar.
- Some columns are initially hidden. To show or hide a column, right-click on any column heading, and select the column.
- The **Status** column can contain the following icons:

Loaded

Loaded and [compressed](#)

Loaded, with additional [full copies](#)

Loaded, with additional full copies and compressed

Unloaded

- The **Size** column displays the in-memory size of each loaded table. If the table is compressed or loaded with additional full copies, a tooltip in the **Size** column provides details.

Note: For a LASR star schema that is output as a view, the displayed **Size** value is calculated as if the LASR star schema was output as a table. See [Creating LASR Star Schemas](#) in the *SAS Visual Analytics: User's Guide*.

- The **Loaded** column indicates when each table was initially loaded.
- The **Modified** column indicates when each table was most recently updated (for example, appended to, reloaded, or refreshed by autoloading).
- The **Loaded By** column displays the user ID that loaded a table (for a distributed server) or started the server (for a non-distributed server).

Note: If the FORCESIGNER= option is specified for the LASR table, then the **Loaded By** column is blank.

- The **LASR Name** column displays table names in the in-memory format `server-tag.table-name`. See [Figure 5.2 on page 96](#).
- The **Mapped Memory** column indicates how much memory is mapped to disk. The **Unmapped Memory** column indicates how much memory is in use.

Note: The **Mapped Memory** column and **Unmapped Memory** column are initially hidden. These columns are included only in deployments where a

distributed server can use highly efficient paging to read SASHDAT files. See [Memory Management](#) in the *SAS LASR Analytic Server: Reference Guide*.

Administer LASR Tables

Unload, Reload, or Delete a Table

- 1 From the main menu in the administrator, select **LASR** ► **Manage Tables**.
- 2 On the **LASR Tables** tab, right-click on a table, and select an action.
 - If most actions are disabled, select **Get Status**, and then right-click on the table again.
 - To delete or reload a table that is loaded, begin by unloading the table.

Replace a Source Table

To replace a source table, right-click on a LASR table, and select **Change Source**. You might use the change source action if an original source table is missing, or if you want to add or remove compression for a table.

Note: Not all tables support the change source action.

Note: If the replacement table differs from the original table in a way that affects a permission condition, data access problems can occur. To provide access, remove the permission condition from the LASR table. See [“Set a Row-Level Permission Condition”](#) on page 39.

Results

To view a log for the most recent interactive action for a table, open the **LASR Tables** tab, right-click on the table, and select **Last Action Log**.

TIP Not all actions generate a last action log. To determine which action generated a log, examine the log’s task summary and timestamp.


For troubleshooting, see [“Load, Reload, and Import Issues”](#) on page 185.

Additional Considerations

- Not all tables can be interactively reloaded. See [“Reload Methods”](#) on page 11.

Note: When you use the **Load a Table** action, a job object (named *source-table - Load Job date-and-time*) is created to support reloading of the table.

- The maximum length for a job name is 60 characters.
- If you edit a job, SAS Visual Analytics might not be able to use the job to perform an interactive reload. In this circumstance, a new job is created the next time you perform a new load of the table.

- To deploy a job for scheduling, see [Scheduling in SAS](#).
- The **Unload** action removes a table from memory, but it does not delete the corresponding metadata object. The **Delete** action deletes the metadata object that represents an in-memory table.
- Most of the tab toolbar buttons affect only tables that have a selected (checked) check box.
- Clicking on the cell that is next to a check box toggles the state of that check box. Before you use a tab toolbar button, make sure only the appropriate check boxes are selected. To clear all check boxes, click  in the tab toolbar.
- To perform an action on multiple tables, select check boxes, and then click an icon in the tab toolbar. To cancel all remaining actions in a multi-table operation, click **Cancel** in the tab toolbar.

Administrator Load

Preparation

Register Source Tables

In the administrator, only registered tables can be interactively loaded or staged. For alternative methods for making data available, see [“Autoload” on page 22](#) and [“Self-Service Import” on page 16](#).

- 1 In the administrator’s **Folders** pane, right-click on a library, and select **Register and Update Tables**.

TIP To display the **Folders** pane, select **View** ► **Folders** from the main menu.

- 2 In the Select Tables window, select the tables that you want to register. Click **OK**.
- 3 In the Register Tables window, make any necessary adjustments. Click **OK**.

Note: If you register a table that already exists in the specified metadata folder, that table’s metadata is updated.

Note: To add a library or learn about alternate methods for registering tables, see the [SAS Intelligence Platform: Data Administration Guide](#).

Stage a Registered Table

Note: This task is applicable only if you load data from co-located or NFS-mounted storage to a distributed SAS LASR Analytic Server. See [“Distributed Server: Parallel Load” on page 108](#).

- 1 In the **Folders** pane, right-click on a table, and select **Add to HDFS** (to stage to co-located HDFS) or **Add to Data Server** (to stage to NFS-mounted MapR or a legacy co-located provider).

TIP To display the **Folders** pane, select **View** ► **Folders** from the main menu.


- 2 In the Add Table window, make any necessary adjustments.

Note: Specify a table name that will also be appropriate as the LASR table name. (When you later load the staged table, the LASR table name will be the same as the name of the staged table.)

- 3 Click **OK**.

Load a Table

- 1 From the main menu in the administrator, select **LASR** ► **Manage Tables**.

- 2 In the tab toolbar, click .

- 3 In the Load a Table window:

- a Click **Browse**, and select a source table. For example, to load a sample table, navigate to `/Shared Data/SASHELP`, and select the CARS table.

- b In the **LASR Table** section, make any necessary adjustments.

Note: The location that you select affects access to the loaded table. Each table inherits permissions from its parent folder.

- c Click **OK**.

TIP You can also load a new table from the **Folders** pane (right-click on a table) or from the **LASR Servers** tab (right-click on a server).

Self-Service Import

Introduction

Data imports that are performed in the designer, the explorer, or the data builder are referred to as self-service imports. This topic provides information to help an administrator support self-service imports. For user instructions, see the [SAS Visual Analytics: User's Guide](#).

Requirement: User Privileges

- Individual data source-specific capabilities affect the availability of all self-service import actions. In the designer and the explorer, the Import and Load Data capability is a prerequisite for all self-service imports. For example, users who perform self-service imports from Oracle should have both of the following capabilities:
 - Import and Load Data
 - Import from Oracle

- Self-service import actions load data to memory, so users must have appropriate metadata-layer access to the target LASR library, server, and folder. See [Table 3.1 on page 37](#).
- Self-service import actions use a workspace server and a SAS LASR Analytic Server, so users must have appropriate host-layer access. See [“Host Account Privileges” on page 5](#).

Note: Self-service imports require a workspace server that supports the job execution service. See [“Using Multiple SAS Application Servers” on page 90](#).

Requirement: SAS/ACCESS

For most data sources, a SAS/ACCESS engine must be licensed, installed, and configured on the workspace server machine. For example, to perform a self-service import from Oracle, SAS/ACCESS Interface to Oracle is required.

TIP After you add or renew a license, you must update the SAS Installation Data (SID) file in metadata. See [Usage Note 49750](#).

If a SAS/ACCESS license for a data source is required but not available, that data source is not listed in the **Import Data** pane. This deployment-level exclusion affects all users, regardless of their capabilities.

Note: Imports from Salesforce use SAS/ACCESS Interface to ODBC and the Salesforce driver.

How to Protect Imported Data

User access to each data source is controlled by that data source’s authorization system.

Each self-service import action loads a source table to memory. The in-memory copy of the data is not subject to access controls from the original data source’s authorization system. Instead, access to in-memory data is controlled by metadata-layer permissions. Unless permissions are set directly on a LASR table, permissions on the LASR table’s parent folder determine access.

The following guidelines apply:

- Users who have privileged access to source data should import that data to only a location that has appropriate metadata-layer protections.
- Users who have fine-grained, identity-based access to source data should import that data to only a private location. For example, if UserA imports a source table that has salary information, and the source table has row-level controls that enable UserA to see only his salary, then the in-memory version of the imported table contains only information about UserA.

If your deployment supports self-service import of sensitive data, use the following measures:

- Give self-service import capabilities to only users who understand and can conform to the preceding guidelines.
- Set up an appropriately protected output location (metadata folder) for each distinct level of access. Ensure that users who have self-service import capabilities load data to the appropriate location.

TIP In the initial configuration, self-service import actions load data to a general-purpose location. Users can instead select a private location (My Folder). Only users who have the Build Data capability can select other locations.

How to Limit Import Size

Row Limit

To prevent users from importing extremely large DBMS tables, you can set a maximum number of rows for self-service imports of DBMS tables. If the number of rows in a DBMS source table exceeds the limit, no data is imported. In the initial configuration, no limit is imposed. See [“va.SelfService.ImportRowsHardCap” on page 143](#).

You can set a warning threshold for self-service import actions. If a user attempts to import a DBMS table that exceeds a specified number of rows (but does not exceed the maximum number of rows that can be imported), a warning message informs the user that the import might take a long time. The user can either continue the import or cancel the action. In the initial configuration, no warning threshold is set. See [“va.SelfService.ImportRowsSoftCap” on page 143](#).

File Size Limit

To set the maximum file size (in megabytes) that a user can import, see [“va.SelfServe.MaxUploadSizeInMegabytes” on page 142](#).

Tables Limit

To limit the total amount of space that a SAS LASR Analytic Server can use to host tables, see [“Limit Space for Tables” on page 100](#).

Reload-on-Start

Introduction

Reload-on-start is a special-purpose feature that can help keep some interactively loaded data available after certain types of server restart. Here are key points:

- Do not use reload-on-start as a general-purpose way of keeping data available. [Autoload](#) is the recommended general-purpose way of keeping data available.
- Before you use reload-on-start, review the limitations that are described in [“Additional Considerations” on page 20](#).
- See [SAS Visual Analytics: Getting Started with Data on Windows](#) for orientation and recommendations for beginners.

How Reload-on-Start Works

Here is an example of how reload-on-start works:

- 1 In the explorer, a user initiates an import of an XLS file.
- 2 SAS places a data set copy of the source data in the data provider library that is the designated backing store for the target LASR library.
- 3 SAS loads the data and creates a corresponding LASR table object.
- 4 Someone stops the server, so the table is unloaded.
- 5 Someone restarts the server from the **LASR Servers** tab, so the data is reloaded from the backing store.

Note: The reload is driven by the LASR table object's association to a LASR library that supports reload-on-start. That LASR library must be associated with a data provider library that contains a backing store copy of the original source data.

How to Enable Reload-on-Start

To enable reload-on-start, use either of the following methods.


SAS Management Console

- 1 In [SAS Management Console](#), right-click on a LASR library, and select **Properties**.
- 2 On the **Options** tab, in the **Data provider library** field, specify a Base SAS library. The specified library functions as the backing store for participating tables.
- 3 On the **Extended Attributes** tab, set attributes as follows:

VA.ReloadOnStart.Enabled	Yes
VA.ReloadOnStart.TableDefault	Yes
VA.ReloadOnStart.Method	Selective



- 4 (Optional) To selectively exclude a LASR table from participation, set VA.ReloadOnStart.Enabled to **No** on the table's **Extended Attributes** tab.

SAS Environment Manager

- 1 In the [SAS Environment Manager Administration](#) window, click or tap , and select **Libraries** from the side menu.
- 2 Open a LASR library.

- 3 From the library's drop-down list of property pages, select the **Options** property page. In the **Data provider library** field, specify a Base SAS library. The specified library functions as the backing store for participating tables.
- 4 From the library's drop-down list of property pages, select the **Extended Attributes** property page. Set attributes as follows:

VA.ReloadOnStart.Enabled	Yes
VA.ReloadOnStart.TableDefault	Yes
VA.ReloadOnStart.Method	Selective

- 5 Click .
- 6 (Optional) To selectively exclude a LASR table from participation:
 - a From the library's drop-down list of property pages, select the **Tables** property page.
 - b Open the LASR table.
 - c From the LASR table's drop-down list of property pages, select the **Extended Attributes** property page.
 - d In the **Value** cell for VA.ReloadOnStart.Enabled, enter `no`.
 - e Click .

Additional Considerations

- Not all tables can participate in reload-on-start. See [“Reload Methods” on page 11](#).
- A table that can participate in reload-on-start is reloaded only if *all* of the following additional requirements are met:
 - The table is not in a **My Folder** metadata location. Or, the table is in the **My Folder** metadata location that belongs to the identity who starts the server.

Note: Even an administrator who has access to another user's **My Folder** metadata location cannot reload a table to that location using reload-on-start.
 - The identity that starts the server has metadata-layer access to the table, its parent folder, and its parent library. See [“Permissions by Task” on page 37](#).
 - The identity that starts the server has host access to the table (in the associated data provider library).
- Reload-on-start occurs after the SAS LASR Analytic Server is started by autoloading, by an explicit start request in the administrator, or by a user action that triggers autostart.
- Only a Base SAS library can be used as a designated backing store for reload-on-start.

- If you enable reload-on-start for a library that contains sensitive data, you must protect the corresponding data provider library against unauthorized access.
- To increase protection of files in the backing store, see [“On-Disk Encryption of Reload-on-Start Files” on page 61](#).

Reference

Logs and Process IDs

The directory `va.monitoringPath/Logs` contains logs of reload actions.

The directory `va.monitoringPath/PIDs` contains text files that document process IDs.

See [“va.monitoringPath” on page 142](#).

Library-Level Attributes for Reload-on-Start

`VA.ReloadOnStart.Enabled` (`No` | `Yes`)
 specifies whether a LASR library supports reload-on-start. A `No` value for a library prevents participation by all of the library’s tables, regardless of any `Yes` values on the tables. For a new library, the value is `No`.

`VA.ReloadOnStart.TableDefault` (`No` | `Yes`)
 specifies whether tables that neither explicitly enable nor explicitly disable reload-on-start participate. For a new library, the value is `No`. Therefore, a table for which the extended attribute `VA.ReloadOnStart.Enabled` is not specified does not participate.

`VA.ReloadOnStart.Method` (`All` | `Selective`)
 affects table participation in reload-on-start.

`All` causes all eligible tables to participate, regardless of any contradictory table-level settings.

`Selective` causes any table-level settings (of `VA.ReloadOnStart.Enabled`) to be honored.

For a new library, the value is `All`.

Table-Level Attributes for Reload-on-Start

`VA.ReloadOnStart.Enabled` (`No` | `Yes`)
 affects whether the table participates in reload-on-start. For a new table, this attribute does not exist. Instead, table participation is determined by the library-level setting for `VA.ReloadOnStart.TableDefault`. If necessary, you can manually add the `VA.ReloadOnStart.Enabled` attribute to a table object.

This table-level setting is effective only if both of the following conditions are met:

- reload-on-start is enabled for the parent library
- the parent library’s `VA.ReloadOnStart.Method` is set to `selective`

Autoload

Introduction

You can use autoload to keep a set of source tables in memory. Users or processes place source tables in a specified host location (a drop zone). Corresponding in-memory data is periodically updated to reflect the contents of the drop zone.

Benefits of autoload include the following:

- You do not have to start the server. If a SAS LASR Analytic Server stops, the next run of autoload starts the server and loads data from the drop zone.
- You do not have to register the source tables in metadata.
- Browser-based constraints on the size of locally imported files do not apply to autoload.

For limitations of autoload, see [“Additional Considerations” on page 30](#).

For a concise summary of basic information, see [Autoload for Beginners](#) in *SAS Visual Analytics: Getting Started with Data on Windows*.

How Autoload Works

Here is a summary of how autoload works:

- 1 Autoload periodically scans the contents of a designated host directory, which is referred to as the *autoload data directory* or *drop zone*.
- 2 After each scan, autoload synchronizes in-memory data against source tables in the autoload data directory as follows:
 - For each delimited file and spreadsheet, a corresponding source table (SAS data set) is created. For a delimited file or spreadsheet that already has a newer corresponding source table, this step is omitted.
 - Source tables that are not already in memory are loaded.
 - Source tables that are newer than their corresponding in-memory tables are refreshed (unloaded and then reloaded).
 - Source tables that are in the Unload subdirectory and in memory when a run of autoload begins are unloaded in that run.
 - Source tables that are in the Append subdirectory and newer than their corresponding in-memory tables are appended to their corresponding in-memory tables. If a table in the Append subdirectory has no corresponding in-memory table, it is loaded as a new table.
 - Each Append table is also appended to its corresponding table in the autoload data directory. If no corresponding table exists, a new table is added to the autoload data directory.
 - To prevent redundant append actions, data in the Append subdirectory is compared to corresponding data in the autoload data directory. The append action is performed on only data in the Append subdirectory

that is newer than its corresponding data in the autoload data directory.

Note: To ensure that refresh and append actions occur for only source tables that are newer than their corresponding in-memory tables, autoload compares file timestamps of source tables to load timestamps of corresponding in-memory tables.

The Autoload Directories

Autoload Data Directory (Drop Zone)

In the standard configuration, autoload data directories are in the AppData branch of the [SAS configuration directory](#):

```
/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad
```

Each autoload data directory has four required subdirectories (Append, Formats, Logs, and Unload). The term *drop zone* refers to the autoload data directory and its subdirectories.

Note: The scheduler account and anyone who places tables in these directories must have Read and Write access to these directories.

Autoload Scripts Directory

In the standard configuration, autoload scripts directories are in the Applications branch of the [SAS configuration directory](#):

```
/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/
```

Note: The scheduler account must have Read and Write access to the autoload scripts directory and its contents.

Timing of Autoload

Autoload runs as a periodic scheduled task. In the standard configuration, a new run of autoload is started every 15 minutes. The timing is controlled by a setting in `schedule.sh` (on UNIX) or `schedule.bat` (on Windows) in the autoload scripts directory.

Here are additional details:

- A new run of autoload starts only after the previous run is complete.
- Starting the associated SAS LASR Analytic Server does not trigger an immediate run of autoload.
- Stopping the associated SAS LASR Analytic Server does not stop autoload activity. If the server is down when a run of autoload begins, autoload starts the server.

UNIX Specifics: The interval clock starts on the hour. For example, if the interval is 15 minutes, then autoload runs on the hour and at 15, 30, and 45 minutes after the hour.

Windows Specifics: The interval clock starts when autoload is scheduled. For example, if the interval is 15 minutes, then autoload runs 15 minutes after the schedule script is invoked, and every 15 minutes thereafter.

How to Start Autoload

To start scheduled runs for an implementation of autoload:

- 1 On the machine that hosts the implementation, identify or create a scheduler account.
 - Give the account the host-layer privileges that are required to start the associated SAS LASR Analytic Server and load data. See [“Host Account Privileges”](#) on page 5.
 - On UNIX, enable the account to run cron jobs.
 - In the [SAS configuration directory](#), give the account Read and Write access to the autoload directories and their contents. For the public implementation of autoload, the locations are as follows:

Data: /AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad

Scripts: /Applications/SASVisualAnalytics/VisualAnalyticsAdministrator

Note: For the public implementation, access to subdirectories for other implementations (for example, EVDMLA and VALIBLA) is not required.

- 2 In the metadata, create a corresponding individual metadata identity. (For the public implementation, the new identity does not need any explicit group memberships.) See [“How to Add a User”](#) on page 3.

Note: This requirement reflects the standard configuration. See [“Metadata Server Connection”](#) on page 31.

Make sure the scheduler account’s metadata identity has the required [metadata-layer permissions](#) on the target server, library, and folder.

For the public implementation, all registered users have sufficient access, so no adjustments are required. Here are the details:

Server:	Public LASR Analytic Server	RM, WM, A
Library:	Visual Analytics Public LASR	RM, R, WM, A
Folder:	<code>/Shared Data/SAS Visual Analytics/Public/LASR</code>	RM, R, WMM, W

- 3 Log on to the host as the scheduler account, navigate to the implementation’s scripts directory, and invoke `schedule.sh` (on UNIX) or `schedule.bat` (on Windows).

TIP You can change the schedule interval by editing the schedule script. For validation, an interval of 2 minutes is suggested.

- 4 Verify that the scheduled task is running.

Windows Specifics: Access the **Task Scheduler** (for example, select **Start** ► **Control Panel** ► **Administrative Tools** ► **Task Scheduler**). Locate the

task in the **Task Scheduler Library** (for example, `Visual Analyt Hi-Perf Cfg - Auto Load Scheduler`).

UNIX Specifics: Run the command: `crontab -l`

- 5 If necessary, edit the schedule script to adjust the interval. The standard interval is 15 minutes.
- 6 (Optional) Verify that tables that are placed in the autoload data directory are processed as described in [“How Autoload Works” on page 22](#).

For example, place a CSV file or a SAS data set in the autoload data directory. After 15 minutes, use the **LASR Tables** tab to verify that the data is loaded. See [“Get Table Information” on page 12](#).

How to Stop Autoload

To stop the scheduled task, use the scheduler account to invoke `unschedule.sh` (on UNIX) or `unschedule.bat` (on Windows). Stopping autoload does not stop the associated SAS LASR Analytic Server.

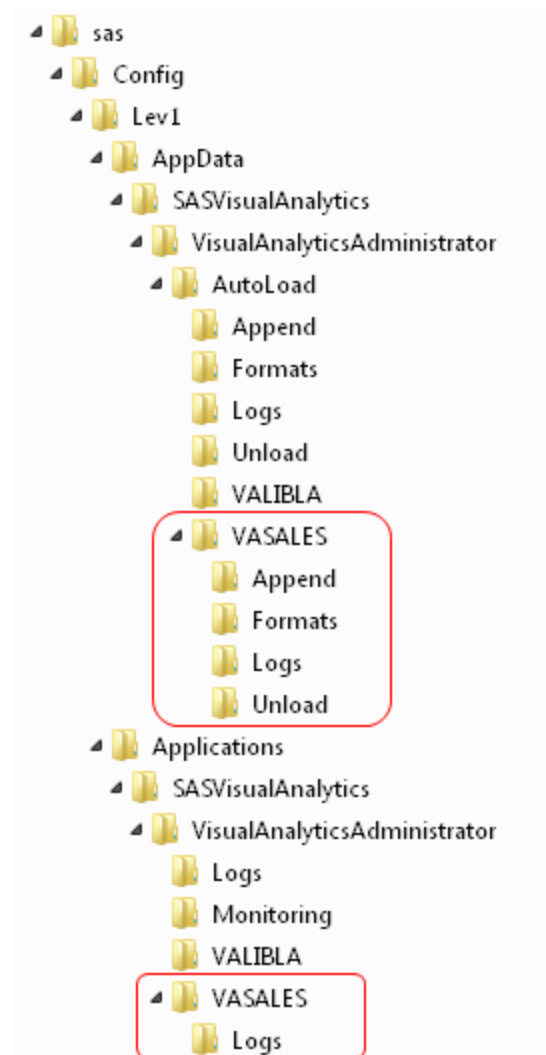
How to Add an Implementation

Introduction

Each LASR library that supports autoload must have its own implementation of autoload. For details about the predefined implementations, see [“Predefined LASR Libraries” on page 153](#).

This section creates an implementation of autoload for sales data. Here is a depiction of the host directories that this example creates:

Figure 2.1 Example: VASALES Implementation of Autoload



Note: In the preceding image, the new autoload data directory is above the new autoload scripts directory. Directories that are not essential to this example are omitted from the image.

Instructions: UNIX

- 1 Create a new autoload data directory as follows:

```
autoload-data-branch/VASALES
autoload-data-branch/VASALES/Append
autoload-data-branch/VASALES/Formats
autoload-data-branch/VASALES/Logs
autoload-data-branch/VASALES/Unload
```

- 2 Create a new autoload scripts directory.
 - a Make a sibling copy of an existing autoload scripts directory. For this example, copy `autoload-scripts-branch/VALIBLA` (or your equivalent of that predefined existing scripts directory) to a new directory named `autoload-scripts-branch/VASALES`.

VA.AutoLoad.Debug.Enabled	No
VA.AutoLoad.ExpandChars.Enabled	No
VA.AutoLoad.Import.Delimiter.TXT	TAB
VA.AutoLoad.Import.RowsToScan	500

* Set all 6 of the **Sync** attributes (**Enabled**, **Import**, **Load**, **Refresh**, **Append**, **Unload**) to **Yes**.

TIP If a new library's extended attributes are not initially visible, save and then reopen the library.

- 6 Start the new scheduled task by logging on to the host as the scheduler account and invoking `schedule.sh`.

Instructions: Windows

- 1 Create a new autoloading data directory as follows:

```
autoload-data-branch\VASALES
autoload-data-branch\VASALES\Append
autoload-data-branch\VASALES\Formats
autoload-data-branch\VASALES\Logs
autoload-data-branch\VASALES\Unload
```

- 2 Create a new autoloading scripts directory.

- a Make a sibling copy of an existing autoloading scripts directory. For this example, copy `autoload-scripts-branch\VALIBLA` (or your equivalent of that predefined existing scripts directory) to a new directory named `autoload-scripts-branch\VASALES`.

Note: Creating the new scripts directory beneath the existing autoloading scripts branch facilitates migration.

- b In the new `autoload-scripts-branch\VASALES\Logs` directory, delete any copied files.
- 3 In the new `autoload-scripts-branch\VASALES` directory, edit the copied files as follows:

AutoLoad.sas

Change the `%LET AL_META_LASRLIB=` value to the metadata name of the new implementation's LASR library. For example:

```
%LET AL_META_LASRLIB=SalesAutoload;
```

runsas.bat

Edit the `AUTOLOAD_ROOT=` value to reference the new autoloading scripts directory. For example:

```
AUTOLOAD_ROOT="autoload-scripts-branch\VASALES"
```

Verify that the appropriate configuration files are referenced. See [“Configuration Files for Autoload” on page 31](#).

schedule.bat and unschedule.bat

Edit the `RUNSAS_PATH=` value to reference the new implementation's autoload scripts directory. For example:

```
RUNSAS_PATH="autoload-scripts-branch\VASALES\runsas.bat"
```

In the `schedule.bat` and `unschedule.bat` files, change the name of the scheduled task. For example, if you began by copying scripts from the public implementation of autoload, the task name in the copied files is initially `Visual Analyt Hi-Perf Cfg - Auto Load Scheduler`. Change that name to any different value, such as `Private Autoload Scheduler\`". (The name change is necessary because the Windows Task Scheduler requires that each task name is unique.)

- 4 In [SAS Management Console](#) or [SAS Environment Manager](#), identify or create a metadata folder for generated LASR table objects (in this example, `/Shared Data/SAS Visual Analytics/Autoload/SALES`).
- 5 Configure a LASR library to support autoload.
 - If you do not want to modify an existing library, [add a LASR library](#). If you do not want to assign the new library to an existing server, [add a SAS LASR Analytic Server](#).
 - The library's name must exactly match the value that you entered in the `AutoLoad.sas` file in step 3 (in this example, `SalesAutoLoad`).
 - The library must be in a metadata folder that has appropriate permission settings (in this example, `/Shared Data/SAS Visual Analytics/Autoload/SALES`).
 - Set the library's [extended attributes](#) as follows:

VA.AutoLoad.Location	<code>autoload-data-branch\VASALES</code>
VA.Default.MetadataFolder	<code>/Shared Data/SAS Visual Analytics/Autoload/SALES</code>
VA.AutoLoad.AutoStart	Yes
VA.AutoLoad.Enabled	Yes
VA.AutoLoad.Sync.Action*	Yes
VA.AutoLoad.Compress.Enabled	No (or, to enable compression, Yes)
VA.AutoLoad.Debug.Enabled	No
VA.AutoLoad.ExpandChars.Enabled	No
VA.AutoLoad.Import.Delimiter.TXT	TAB
VA.AutoLoad.Import.RowsToScan	500

* Set all 6 of the **Sync** attributes (**Enabled**, **Import**, **Load**, **Refresh**, **Append**, **Unload**) to **Yes**.

TIP If a new library's extended attributes are not initially visible, save and then reopen the library.

- 6 Start the new scheduled task by logging on to the host as the scheduler account and invoking `schedule.bat`.

Additional Considerations

- Not all tables can be autoloaded. See [“Load Methods” on page 10](#).
- Autoload is supported for both distributed and non-distributed servers. However, you cannot autoload data from co-located storage.
- A new log file is generated for each run. The `autoload-scripts/Logs` directory must be periodically emptied.
- You cannot interactively reload an autoloaded table. You can instead interactively unload the table, and then wait for the next run of the autoload scheduled task, which refreshes (unloads and then reloads) the table.
- You cannot autoload multiple tables that have the same base name. For example, if the files `abc.xls` and `abc.xlsx` are placed in an autoload data directory, only one data set (`abc.sas7bdat`) is loaded.
- In a multi-machine deployment, autoload-related files are on the machine that hosts the workspace server.
- If you move a delimited file or spreadsheet from the autoload data directory to the Unload subdirectory, remember to also delete the file’s corresponding SAS data set (from the autoload data directory and, if applicable, from the Append subdirectory).
- If a table exists in both the autoload data directory and the Unload subdirectory, the table is repeatedly loaded and unloaded by alternating runs of autoload.
- If the metadata name of a LASR library that supports autoload includes UTF-8 characters, the corresponding `AutoLoad.sas` program must be saved in UTF-8 encoding. (In the `AutoLoad.sas` program, the `%LET AL_META_LASRLIB=` parameter specifies the library’s metadata name.)
- All synchronization actions create and update corresponding LASR table objects as needed. However, autoload does not delete LASR table objects.
- Autoload runs a SAS session directly from SAS Foundation. To modify session behavior for autoload, set SAS options (such as `MEMSIZE`) in an appropriate location. See [“Configuration Files for Autoload” on page 31](#).

Reference

Logs and Process IDs for Autoload

Comprehensive logs and any list output are written to the `autoload-scripts/Logs` directory. Each run of autoload generates a separate log, with a filename in the format `AutoLoad_date-and-time-stamp`.

Additional logs and any debug output are written to the `autoload-data/Logs` directory. Each run of autoload generates a new log (in both data set and text format) that overwrites the previous log.

Autoload process ID (PID) text files are written to the `va.monitoringPath/PIDs` directory in the format `autoload_library-name.pid` (for example,

autoload_VisualAnalyticsPublicLASR.pid). See [“va.monitoringPath” on page 142](#).

UNIX Specifics: An additional PID file (autoload.pid) is written to the autoload scripts directory. This additional PID file is used to prevent the runsas script from starting again if it is already running.

Metadata Server Connection

In the standard configuration, no metadata connection options are specified in the AutoLoad.sas program. Connection information is obtained as follows:

- The metadata repository name is obtained from the associated sasv9.cfg file. See [“Configuration Files for Autoload” on page 31](#).
- The metadata server’s machine name and port are obtained from the file that the sasv9.cfg file references in its METAPROFILE setting. This is the preferred approach, because it supports both clustered and unclustered metadata servers.
- The account that schedules autoload also runs autoload and connects to the metadata server. This is the preferred approach, because it does not require specifying credentials in any host file.

Note: For information about metadata server connection options, see [SAS Language Interfaces to Metadata](#).

Configuration Files for Autoload

Although autoload does not run in a SAS Application Server, autoload can borrow settings from server configuration files. This borrowing can reduce the need to set the same option in multiple locations. Each implementation of autoload has its own list of references to configuration files.

Windows Specifics: The list is in the AutoLoad.cfg file in the implementation’s autoload scripts directory.

UNIX Specifics: The list is in the SASCFGPATH= variable in the implementation’s runsas script.

The standard list references the following files in the following order:

- 1 The sasv9.cfg file for the SAS Application Server that is designated in the implementation’s runsas script (for example, SERVER_CONTEXT= **SASApp**). The designated SAS Application Server and the autoload implementation must be on the same machine.
- 2 The sasv9_usermods.cfg file for the designated SAS Application Server.
- 3 The implementation’s AutoLoad.cfg file.
- 4 The implementation’s AutoLoad_usermods.cfg file.

The preceding list is in reverse precedence order. If an option is set in multiple configuration files, the setting in the last-listed file has precedence. For example, settings in an AutoLoad_usermods.cfg file override any conflicting settings in other configuration files.

You can add, remove, or adjust options in the referenced configuration files as needed. See [Reference: Configuration Files for SAS Servers](#) in the *SAS Intelligence Platform: System Administration Guide*.

User-Defined Formats for Autoload

For general information, see “[Supporting User-Defined Formats](#)” on page 83.

Any format catalogs that are made available through a referenced configuration file are available to autoload.

If you want to make certain user-defined formats available exclusively to a particular implementation of autoload, place format catalogs in that implementation’s `autoload-data-branch/Formats` directory. Catalogs in that directory have precedence over same-named catalogs that are available to autoload through configuration files.

Library-Level Attributes for Autoload

The following attributes support autoload:

VA.AutoLoad.Location

sets the autoload data directory. If you change the location, make sure you create the required subdirectories. For a new library, the suggested value is `autoload-data-branch/LIBNAME`.

VA.Default.MetadataFolder

sets the metadata location for the LASR table objects that autoload generates. For a new library, the initial value is your equivalent of `/shared Data/SAS Visual Analytics/Autoload`.

VA.AutoLoad.Enabled

specifies whether the library supports any autoload features. For a new library, the initial value is `No`.

Note: Setting this attribute to `yes` does not disable interactive loading. You can interactively load data to a library that supports autoload.

VA.AutoLoad.Sync.Enabled

specifies whether synchronization actions are enabled. This is a parent setting (and a prerequisite) for other `*.Sync.*` attributes. For a new library, the initial value is `No`.

To preview synchronization actions, set this value to `No`, run autoload, and then examine the autoload log file.

VA.AutoLoad.Sync.Import

specifies whether the import action is enabled. For a new library, the initial value is `No`.

VA.AutoLoad.Sync.Load

specifies whether the load action is enabled. For a new library, the initial value is `No`.

VA.AutoLoad.Sync.Refresh

specifies whether the refresh action is enabled. For a new library, the initial value is `No`.

VA.AutoLoad.Sync.Append

specifies whether the append action is enabled. For a new library, the initial value is `No`.

VA.AutoLoad.Sync.Unload

specifies whether the unload action is enabled. For a new library, the initial value is `No`.

VA.AutoLoad.Compress.Enabled
specifies whether compression is used when data is autoloaded. The default value is `NO`. (For the administrative reporting library, EVDMLA, the initial value is `YES`.)

VA.AutoLoad.Debug.Enabled
specifies whether debugging is enabled for autoload. The default value is `NO`.

VA.AutoLoad.ExpandChars.Enabled
specifies whether autoload supports expansion of character variable lengths. The default value is `NO`. To enable character expansion, set the value to `YES`.

Note: Character expansion occurs when a SAS data set that is not UTF-8 encoded is autoloaded to a server that uses UTF-8 encoding. For more information, see [Avoiding Character Data Truncation By Using the CVP Engine](#) in the *SAS National Language Support (NLS): Reference Guide*.

CAUTION! Format widths are not expanded with character variable lengths. If you enable character expansion, in-memory data might appear to be truncated. In the designer and explorer, you can adjust formats as needed. To minimize the potential impact, enable character expansion in a separate LASR library that contains only tables that require character expansion. For more information, see the technical paper "[Processing Multilingual Data with the SAS 9.2 Unicode Server](#)".

VA.AutoLoad.Import.Delimiter.TXT
specifies the delimiter to use when autoload imports TXT files. The default value is `TAB`, which specifies to use the Tab character as the delimiter. You can specify a single character (for example, `|`, `!`, or `&`), `SPACE` (to use a space delimiter), or a hexadecimal code (for example, `'09'x`).

VA.AutoLoad.Import.RowsToScan
specifies the number of rows to scan to determine the data type and length for each column in an imported table. You can specify a positive integer or the value `ALL`. The default value is 500. (For the administrative reporting library, EVDMLA, the initial value is `ALL`.)

TIP The header row counts. For example, to scan one row of data, specify 2 as the value.

The following attribute is used by autoload (but is not exclusive to autoload):

VA.AutoLoad.AutoStart
specifies whether the associated SAS LASR Analytic Server starts on demand for load requests against this library. For a new library, the initial value is `NO`.

Note: Except where otherwise specified, the supported values are `NO` and `YES`.

Note: Changes take effect on the next run of autoload. For information about how tables that are already loaded are affected, see "[How Autoload Works](#)" on [page 22](#).

To set library-level extended attributes, access a LASR library's metadata definition in SAS Management Console or SAS Environment Manager.

Processing of Delimited Files and Spreadsheets

In general, autoload processes delimited files and spreadsheets in the same way that these files are processed during a self-service import. For information

about supported file types, requirements, missing values, and valid names, see the [SAS Visual Analytics: User's Guide](#).

The following details are specific to autoloading:

- The file size limitation for interactive import is not applicable to autoloading.
- You cannot autoloading a ZIP file.
- Autoloading always reads column names from the first row and begins data import on the second row.
- When you autoloading a spreadsheet that has multiple worksheets, only the first worksheet is loaded.
- For append actions, column data types and lengths in both files must match.
- Autoloading of XLSB and XLSM files is supported only on Windows. The 64-bit version of Microsoft Access Database Engine (formerly known as Microsoft Office Access Connectivity Engine, or ACE) is required.
- To autoloading files that use a delimiter (other than a comma or the Tab character), use the TXT file extension and specify the delimiter in the [VA.AutoLoad.Import.Delimiter.TXT](#) extended attribute.

3

Security

Permissions	36
About Permissions	36
Permissions by Task	37
Grant or Deny a Permission	39
Set a Row-Level Permission Condition	39
View Authorization Information	40
Access to In-Memory Data	41
SAS LASR Authorization Service	41
Signature Files	43
Server Tags	44
Row-Level Security	44
Introduction	44
Permission Precedence	45
Syntax (Enhanced Editor)	46
Syntax (Basic Editor, Batch)	46
Key Actions Auditing	48
Introduction	48
How to Safely Enable Auditing	49
Audit Content and Coverage	50
Locked-Down Servers	56
Access to SAS Mobile BI	56
About Mobile Device Management	56
How to Manage Mobile Devices	57
Change How Devices Are Managed	58
About the Mobile Devices Tab	59
Protections for Mobile Content	60
Authentication	60
Introduction	60
Shared Accounts for Self-Service Imports	60
Policy for Concurrent User Logins	61
Encryption	61
Introduction	61
On-Disk Encryption of Reload-on-Start Files	61
On-Disk Encryption of SASHDAT Files	64

Permissions

About Permissions

Key Points

Here are the key points about permissions:

- SAS Visual Analytics uses the platform's metadata authorization layer to manage access to objects such as reports, explorations, tables, libraries, servers, and folders.
- SAS Visual Analytics supports row-level security. SAS Visual Analytics does not support column-level security.

Note: Do not set denials of the ReadMetadata permission on individual columns within a table. If a table is loaded by a user who lacks access to one or more columns, duplicate metadata entries are created for the unavailable columns.

- In the administrator, you can set folder, library, table, and row-level permissions. For alternate interfaces, see [Access Management](#) in the *SAS Intelligence Platform: Security Administration Guide*.

Note: In the administrator, you cannot view or set permissions on the objects that support metadata-bound libraries (secured library folders, secured libraries, and secured tables).

- Access to each object is displayed as part of the object's properties. Not all permissions are relevant for all objects.
- Do not block ReadMetadata access for the SAS Trusted User (for example, sastrust@saspw). To preserve access, grant the ReadMetadata permission to the SAS System Services group.
- For simplicity, set permissions on folders, not on individual objects. Most objects (including tables) inherit permissions from their parent folders. To learn how to customize the metadata folder structure, see [Working with SAS Folders](#) in the *SAS Intelligence Platform: System Administration Guide*.
- For simplicity, assign permissions to groups, not to individual users. The broadest group is called PUBLIC. The SASUSERS group includes all registered users. To learn how to manage permissions systematically using access control templates, see [Access to Metadata Folders](#) in the *SAS Intelligence Platform: Security Administration Guide*.

Permission Definitions

The following table documents permissions that have a special purpose in SAS Visual Analytics, and introduces some of the standard permissions.

Permission	Affected Actions
Administer (A)	On a LASR library, load and import tables. On a SAS LASR Analytic Server, stop the server or set a tables limit.

Permission	Affected Actions
Read (R)	On a LASR table, read data. On a LASR library, load and import tables. On an encrypted SASHDAT library , add, delete, or load data.
Write (W)	On a LASR table, unload and reload the table; append and delete rows; and edit computed columns.
ReadMetadata (RM)	View an object. For example, to see an exploration, report, table, or library, you need the ReadMetadata permission for that object.
WriteMetadata (WM)	Edit, rename, set permissions for, or delete an object; create certain associations among objects.
WriteMemberMetadata (WMM)	On a folder, add or remove objects.

For more information, see [Metadata Authorization Model](#) in the *SAS Intelligence Platform: Security Administration Guide*.

Permissions by Task

LASR Tables and Servers

The following table documents metadata-layer permissions for working with LASR tables and SAS LASR Analytic Servers.

Table 3.1 Permissions for Working with LASR Tables and Servers

Task	Server	Library	Folder	Table
Read data	RM	RM	RM	RM, R
Append or delete rows	RM	RM	RM	RM, R, W
Edit computed columns	RM	RM	RM	RM, R, W
Load or import a table*	RM	RM, R, WM, A	RM, R, WMM, W	-
Load a stop list	RM, WM	RM, R, WM, A	RM, R, WMM, W	-
Reload a table	RM	RM	RM	RM, R, WM, W
Unload a table	RM	RM	RM	RM, R, W
Start a server	RM	-	-	-
Stop a server	RM, A	-	-	-
Set a server's tables limit	RM, WM, A	-	-	-
Assign a library to a server	RM, WM	RM, WM	-	-
Register a table in metadata	-	RM, WM	RM, WMM	-

Task	Server	Library	Folder	Table
Update a table's metadata	-	RM	RM	RM, WM
Delete a table from metadata	-	RM, WM	RM, WMM	RM, WM

* An initial load (or import) creates a new LASR table object. Read and Write permissions on the folder support actions against the new table.

Explorations and Reports

The following table documents metadata-layer permissions for working with reports and explorations.

Table 3.2 Permissions for Working with Reports and Explorations

Task	Server	Table	Folder	Report	Exploration
Open a report or exploration	RM	RM, R	-	RM	RM
Export a report or exploration	RM	RM, R	-	RM	RM
Modify a report or exploration	RM	RM, R	-	RM, WM	RM, WM
Save a new report or exploration	-	RM	RM, WMM	-	-
Delete a report or exploration	-	RM	RM, WMM	RM, WM	RM, WM

To create, update, or delete a report, access to the SAS Content Server is also required. See [Administering the SAS Content Server](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*. To view the data that populates a report, ReadMetadata permission for the library where the data is stored is also required. See the first row in [Table 3.1 on page 37](#).

Data Queries and LASR Star Schemas

The following table documents metadata-layer permissions for working with data queries and LASR star schemas.

Table 3.3 Permissions for Working with Data Queries and LASR Star Schemas

Task	Server	Table**	Folder	Query or Schema
Save a new query or schema*	RM	RM	RM, WMM	-
Run a query or schema*	RM	RM	-	RM
Edit and save a query or schema	RM	RM	RM	RM, WM
Delete or rename a query or schema	RM	-	RM, WMM	RM, WM

* These tasks create new LASR tables, so the permission requirements for loading a LASR table must also be met. See [Table 3.1 on page 37](#).

** This column refers to any source tables that are represented in metadata. To run a query or schema against a LASR table, Read permission for the LASR table is also required.

Read access to data in a LASR star schema is not affected by permissions for input tables. Instead, Read access to data in a LASR star schema is affected by the Read and ReadMetadata permissions for the output table or view. ReadMetadata permission for the associated server, library, and folder is also required. See the first row in [Table 3.1 on page 37](#).

Note: You can set explicit access controls (including permission conditions) on the output table or view for a LASR star schema. Any explicit access controls persist when you rerun the LASR star schema.

Grant or Deny a Permission

To set an explicit grant or denial:

- 1 In the administrator's **Folders** pane, right-click on an object, and select **Authorization**.

TIP To display the **Folders** pane, select **View** ► **Folders** from the main menu.

- 2 In the **Effective Permissions** table, locate the identity to which you want to assign an explicit control. If the identity is not listed, click **+** to open the Add Identities window.

Note: In the Add Identities window, only user administrators can successfully search by user ID. Regular users cannot see other users' IDs.

Note: An explicit grant of the ReadMetadata permission is automatically set for each identity that you add.

- 3 Double-click on a cell. From the cell's drop-down list, select either **Deny** or **Grant**.

Identity	ReadMetadata	Read	WriteMetadata
PUBLIC			
SAS Administrators	• (no explicit control)		
SAS System Services	Deny		
SASUSERS	Grant		
	Show Origins		

When the drop-down list collapses, notice that the cell contains an explicit control indicator .

Note: If the selected identity is an unrestricted user, all permissions are granted and you cannot make changes.

- 4 If you changed a group's access, review the impact on the other listed identities. Controls that you add for a group can affect access for all members of that group.
- 5 In the toolbar at the top of the tab, click .

Set a Row-Level Permission Condition

To limit Read access to rows in a LASR table:

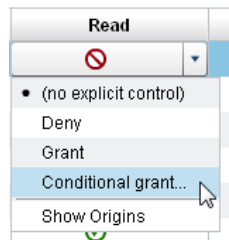
- 1 In the administrator's **Folders** pane, right-click on a LASR table, and select **Authorization**.

TIP To display the **Folders** pane, select **View** ► **Folders** from the main menu.

- 2 In the **Read** column, double-click on the cell for the identity whose row-level access you want to constrain. (Or, if the identity is not listed, click **+** at the right edge of the table.)

Note: An explicit grant of the ReadMetadata permission is automatically set for each identity that you add.

- 3 From the cell's drop-down list, select **Conditional grant**.






Note: If **Conditional grant** is not in the drop-down list, the table does not support row-level security. Only LASR tables support row-level security.

Note: If **Conditional grant** is already selected, select **Conditional grant** again to view or edit the existing condition.

- 4 In the New Permission Condition window, create a condition that specifies which rows the identity can see. See [“Syntax \(Enhanced Editor\)”](#) on page 46.

Note: Conditions from releases prior to 6.2 or from batch tools use a basic editor. In the basic editor, syntax is not validated. See [“Syntax \(Basic Editor, Batch\)”](#) on page 46.






- 5 Click **OK**. Notice that the cell contains the conditional grant icon  with an explicit control indicator .
- 6 If you set a permission for a group, review the impact on the other listed identities. Constraints that you add for a group can affect access for all members of that group.
- 7 In the toolbar at the top of the tab, click .









TIP When you test conditions in another application (such as the explorer), refresh the data source in that application so that your changes are reflected. See [“Caching”](#) on page 43.

A permission condition constrains Read access to rows within a LASR table. For more information, see [“Row-Level Security”](#) on page 44.

View Authorization Information

Here are some details about the **Authorization** page:

- Each object's **Authorization** page describes access to that object. The displayed effective permissions are a calculation of the net effect of all applicable metadata-layer permission settings. To identify the source of an effective permission, double-click on its cell, and select **Show Origins** from the drop-down list. See "[Permission Origins](#)" on page 188.
- Icons indicate grants , conditional (row-level) grants , and denials .
- The explicit indicator icon  indicates an access control that is explicitly set on the current object and explicitly assigned to the selected identity.
- The ACT indicator icon  indicates an access control that comes from an applied ACT whose pattern assigns the grant or denial to the selected identity.
- In combination, icons provide information as follows:

Icon	Meaning
	Grant from an explicit control
	Grant from a directly applied ACT
	Grant from an indirect source (such as a parent group or parent object)
	Conditional grant from an explicit control
	Conditional grant from an indirect source (a parent group)
	Denial from an explicit control
	Denial from a directly applied ACT
	Denial from an indirect source (such as a parent group or parent object)

- To compare permissions for two tables, open them both, and then select **View** ► **Tab Layout** ► **Stacked** from the main menu.

Access to In-Memory Data

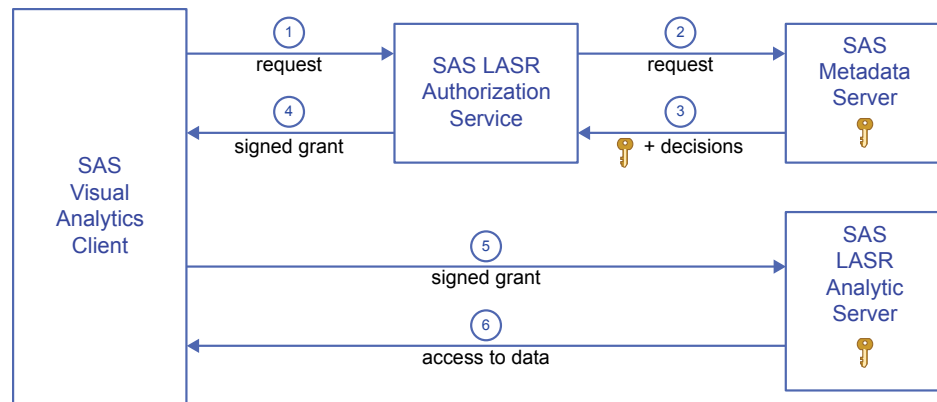
SAS LASR Authorization Service

Overview

The SAS LASR Authorization Service collaborates with the metadata authorization layer to manage user access to in-memory data.

The following figure depicts the authorization process:

Figure 3.1 Authorization Process



- 1 In a SAS Visual Analytics client, a user performs an action that uses a SAS LASR Analytic Server. In this example, the request is to read data. The client sends the request to the authorization service.

Note: Other examples of actions include requesting analysis of data, loading tables, appending rows, and stopping the server.

- 2 The authorization service requests the following information from the metadata server:
 - authorization decisions that indicate whether the requesting user has the effective metadata-layer permissions that are required to perform the requested action. See [“Permissions by Task” on page 37](#).
 - the security key for the target SAS LASR Analytic Server
- 3 The authorization service receives the authorization decisions and security key from the metadata server. If the requesting user has a conditional grant of the Read permission, the authorization service also receives a clause (or set of clauses) that specifies which rows the user can access.
- 4 If the requesting user has effective grants of all permissions that are required for the requested action, the authorization service provides a signed grant to the client.

Note: The authorization service uses the security key to create the signed grant. The signed grant includes the table name, the type of action (for example, Table Info, Summary Statistics, or Regression), and any applicable row-level security conditions.
- 5 The client submits the signed grant to the SAS LASR Analytic Server.
- 6 The SAS LASR Analytic Server uses its knowledge of the security key to validate the signed grant that the client supplies. If the signed grant is valid, the server provides access to the requested in-memory table (conforming to any row-level security conditions in the signed grant).

Security Keys

A LASR security key is a unique, shared secret between a SAS LASR Analytic Server and the metadata server. LASR security keys are created and stored as follows:

- When a SAS LASR Analytic Server is started, a key is generated. In the SAS LASR Analytic Server, the key is stored in memory. The key is also stored in metadata in the password field of a login object that is associated with the server's connection object.
- If a SAS LASR Analytic Server is stopped, the associated key remains in the metadata. If the server connection is restarted, a new key is generated. The new key replaces the existing key in the metadata.

Note: A LASR security key is a SAS internal construct. Do not confuse LASR security keys with encryption key passphrases. See [“On-Disk Encryption of SASHDAT Files” on page 64](#).

Caching

To avoid making repeated queries to the metadata server for a security key, the authorization service caches the key. When the cache interval has expired, the authorization service removes the key from the middle-tier cache. When the next request is made for in-memory data, the authorization service again obtains the key from the metadata server and repopulates the cache.

To enhance performance, the authorization service caches information about users and permissions. When a SAS Visual Analytics user accesses a data source in the SAS LASR Analytic Server, a user object is created and cached. A permission object is also created and cached for the data source. These are middle-tier, session-based caches.

The duration of each cache is set by the `las.caching.*` properties. See [“Configuration Properties” on page 139](#).

Signature Files

Signature files are created when a SAS LASR Analytic Server is started (server signature files) and when a table is loaded (table signature files). The location for each server's signature files is specified in the server's metadata definition.

Manage access to the signature files directory as follows:

- Anyone who performs task that generate signature files must have Write access to the directory.
- Any service accounts that perform tasks that generate signature files must have Write access to the directory. For example, if you use automated data loading, the account under which the scheduled task runs must have this access.
- Nobody else needs access to signature files. (Access from SAS Visual Analytics clients to the SAS LASR Analytic Server and its in-memory data is controlled by metadata permissions.)
- Host-layer access controls on signature files determine access for any requests that are not mediated by the SAS LASR Authorization Service. For this reason, it is important to restrict access to signature files.

Host-protect the signature files directory as follows:

- 1 In [SAS Management Console](#), right-click on a SAS LASR Analytic Server, and select **Properties**.
- 2 On the **Options** tab, click the **Advanced Options** button.

- 3 In the Advanced Options window, select the **Additional Options** tab. Note the path that is specified in the **Signature files location on server** field.
- 4 Host-protect the directory, using the following guidelines:

Windows Specifics: Limit Read and Write access as described above.

UNIX Specifics: For a distributed server, the UMASK value of the TKGrid determines the permissions on signature files. Set the TKGrid UMASK to 077. For a non-distributed server, set the personal UMASK to 077. These settings prevent any user other than the file owner (creator) from gaining access to signature files.

Server Tags

Server tags are identifiers that help the SAS LASR Authorization Service map each in-memory table to a corresponding metadata object. See [“In-Memory LASR Names” on page 96](#).

Each LASR library’s server tag must be defined as follows:

- If the LASR library’s data is loaded from co-located HDFS or NFS-mounted MapR, the server tag must be the source path in dot-delimited format. Here are some examples:

Source Path	Corresponding Server Tag
/hps	hps
/hps/special	hps.special
/sales	sales

- If the LASR library’s data is loaded using SAS Embedded Process, the server tag must be valid as a SAS libref. For example, the server tag cannot be `MyServerTag` (more than eight characters) or `user.sasdemo` (more than one level).
- If the LASR library’s data is loaded from a legacy co-located provider, the server tag must be the source library’s libref (for example, **TDLIB** or **GPLIB**).
- Otherwise, the server tag can be any unique string. If you do not supply a server tag in a LASR library’s metadata definition, the tag `WORK` is used.

CAUTION! Within a server instance (a host-port combination), each server tag must be unique.

Row-Level Security




Introduction

Row-level security enables you to control who can access particular rows within a LASR table, and it is defined by data filter expressions. Row-level access

distinctions can be based on a simple attribute (such as security clearance level) or on a more complex expression that consists of multiple criteria.

Row-level security affects access to subsets of data within a resource. To establish row-level security, you add constraints called permission conditions to explicit grants of the Read permission. Each permission condition filters a particular LASR table for a particular user or group. Each permission condition constrains an explicit grant of the Read permission so that the associated user or group can see only those rows that meet the specified condition.

When row-level security is used, there are three possible authorization decision outcomes for a user request to view data:

-  **Grant**
The requesting user can see all rows.
-  **Conditional grant**
The requesting user can see only those rows that meet the specified filtering conditions.
-  **Denial**
The requesting user cannot see any rows.

TIP When you test conditions in a SAS Visual Analytics application (such as the explorer), refresh the data source in that application (so that the results reflect your saved changes to permission conditions). See “Caching” on page 43.

Permission Precedence

Here are some key points about how permission conditions are incorporated into the metadata-layer access control evaluation process:

- A permission condition is applied only if it is on the setting that is closest to the requesting user. Other permission conditions that are relevant because of further-removed group memberships do not provide additional, cumulative access.
- If there is an identity precedence tie between multiple groups at the highest level of identity precedence, those tied conditions are combined in a Boolean OR expression. If the identity precedence tie includes an unconditional grant, access is not limited by any conditions.

The following table provides examples:

Table 3.4 Precedence for Permission Conditions

Principle	Scenario	Outcome and Explanation
If there are multiple permission conditions that apply to a user because of the user's group memberships, then the identity that has the highest precedence controls the outcome.	A condition on TableA limits Read permission for GroupA.	The user can see only the rows that GroupA is permitted to see. GroupA has a higher level of identity precedence than SASUSERS, so the filters that are assigned to GroupA define the user's access.
	Another condition on TableA limits Read permission for the SASUSERS group.	
	The user is a member of both GroupA and SASUSERS.	


Principle	Scenario	Outcome and Explanation
If there are multiple permission conditions at the highest level of identity precedence, then any data that is allowed by any of the tied conditions is returned.	<p>A condition on TableA limits Read permission for GroupA.</p> <p>Another condition on TableA limits Read permission for GroupB.</p> <p>The user is a first-level member of both GroupA and GroupB.</p>	The user can see any row that is permitted for either GroupA or GroupB.

Syntax (Enhanced Editor)

TIP To access the enhanced editor, see “Set a Row-Level Permission Condition” on page 39.

- On the **Visual** tab, you can drag and drop operators and data items from the left panes.

Note: When you enter values, do not enclose them in quotation marks. The editor adds any necessary quotation marks for you.
- On the **Text** tab, you can enter text directly. Use only those operators that are available on the **Visual** tab.

Note: The **Text** tab does not use the same syntax as the basic editor and the batch tools. For hints, select the **Text** tab, and then click  in the window toolbar.

Syntax (Basic Editor, Batch)

Introduction

This topic is applicable to permission conditions created in the following contexts:

- In SAS Visual Analytics Administrator 6.1 and earlier.
- In the batch tools for metadata authorization. See [Batch Tools for Metadata Authorization](#) in the *SAS Intelligence Platform: Security Administration Guide*.

General Guidelines

- Enclose non-numerical character values within quotation marks.
- The symbol `||` is not supported. Instead, use the keyword `OR`.
- Expressions with months or dates are not supported.
- Do not include the `WHERE` keyword in any expression.

Supported Syntax

Table 3.5 Supported Syntax

Syntax Element	Example
AND, OR, NOT	Toy_Type='cars' OR Toy_Type='dolls'
IN, NOTIN	Toy_Type IN ('dolls' 'cars' 'animals')
CONTAINS, ?	Toy_Type CONTAINS 'cars'
BETWEEN, NOT BETWEEN	Toy_Price BETWEEN 20 AND 30
LIKE	Toy_Type LIKE 'dolls'
=, >, <, >=, <=, <>	Toy_Price=25
^=, NE	Toy_Price^=30

Identity-Driven Properties

The following table introduces properties that you can use to create identity-driven permission conditions. When you use these properties in a permission condition, values are dynamically substituted into the condition at run time, based on the metadata identity of each requesting user.

Table 3.6 Identity-Driven Properties

Syntax Element	Description
SUB::SAS.Userid	<p>returns the requesting user's authenticated ID, normalized to the uppercase format USERID or USERID@DOMAIN.</p> <p>Here is an example for use in the batch tools:</p> <pre>-condition "empID='SUB::SAS.Userid'"</pre>
SUB::SAS.IdentityGroups	<p>returns the requesting user's group and role memberships (direct, indirect, and implicit). The returned list contains group and role names (not display names).</p> <p>Here is an example for use in the batch tools:</p> <pre>-condition "FacilityRegion IN ('SUB::SAS.IdentityGroups')"</pre>
SUB::SAS.PersonName	<p>returns the requesting user's name (as specified in the Name field on the General tab of the user's metadata definition).</p>
SUB::SAS.ExternalIdentity	<p>returns a site-specific identifier for the requesting user. External identity values are populated by the platform's user import macros (if you bulk load user information into metadata).</p>

For example, if a LASR table has an empID column with values that match the user IDs with which users authenticate, you might use the condition `empID='SUB::SAS.Userid'`. Each affected user's ID is substituted into the right side of the condition. In a request from the sasdemo user, the condition

resolves as `empID='sasdemo'`, so only those rows where the value in the `empID` column is `sasdemo` are returned to the `sasdemo` user. If you assign the condition to a group, each member's access is restricted to those rows where the `empID` value matches his or her authenticated user ID. Here is an example of the full command for the use in batch tools:

```
sas-set-metadata-access -profile Admin "/Shared Data/LASRtableA(Table)"
-grant sasusers:Read -condition "empID='SUB::SAS.Userid'"
```

Note: Two additional properties (`SAS.IdentityName` and `SAS.IdentityGroupName`) are not documented here because they are less frequently useful. See [About Identity-Driven Properties](#) in the *SAS Intelligence Platform: Security Administration Guide*.

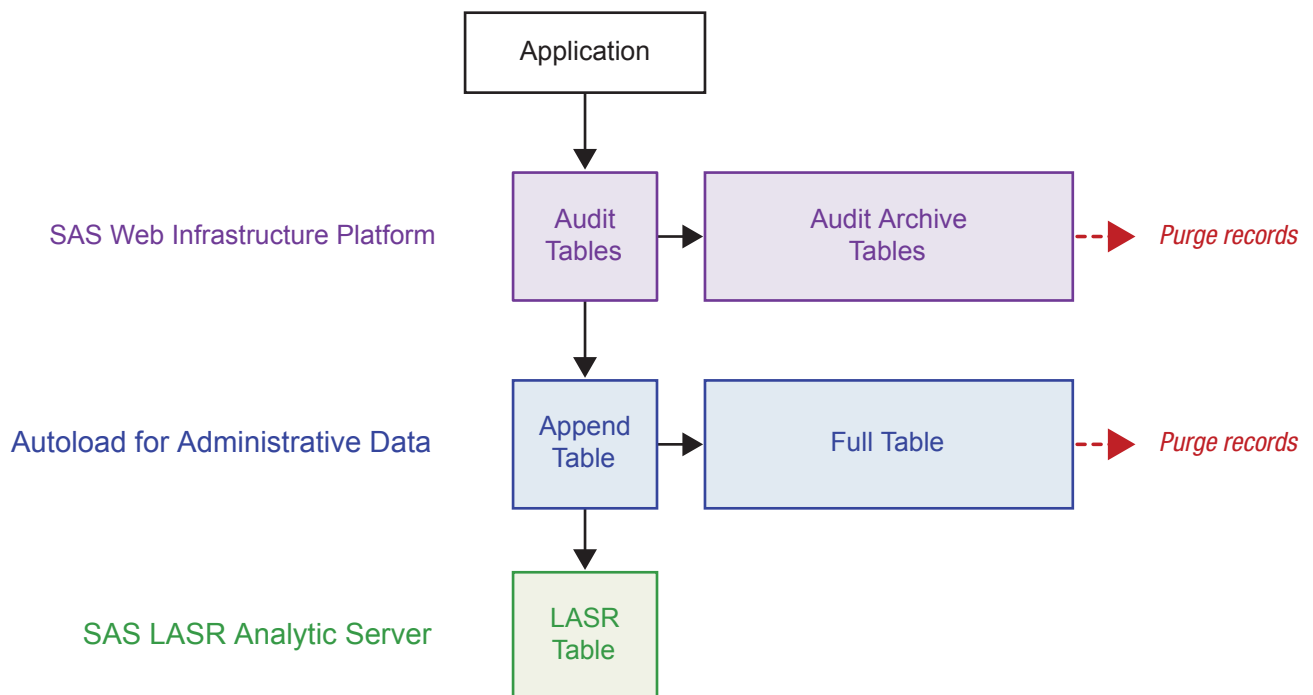
Key Actions Auditing

Introduction

This topic provides information that is specific to SAS Visual Analytics. For general information, see [Configuring Auditing for SAS Web Applications](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

The following figure depicts the flow of audit data:

Figure 3.2 Flow of Audit Data



The vertical flow provides data for an administrative report:

- 1 If auditing is enabled and a user performs an auditable action, the current application generates one or more audit records. The audit records are written to the `SAS_AUDIT` and `SAS_AUDIT_ENTRY` tables in the public

schema of the SharedServices database in the SAS Web Infrastructure Platform.

- 2 The next time the [extraction process](#) runs, it notices that the audit tables in the SAS Web Infrastructure Platform's SharedServices database have data that does not exist in the corresponding autoloading data directory (drop zone). The extraction process writes the new data to the Append subdirectory of the EVDMLA autoloading data directory. The scope of extraction consists of all records for a fixed set of object types. See ["Audit Content and Coverage"](#) on [page 50](#). Audit records for logon and logoff actions are not extracted.
- 3 The next time the [autoloading process](#) runs, it notices that the audit data in the Append subdirectory is newer than the corresponding LASR table. The autoloading process appends the new data to the existing LASR table.

The horizontal flows manage audit data as follows:

- To manage the sizes of the audit tables in the SAS Web Infrastructure Platform's SharedServices database, audit archive rules move specified records of a specified age to the audit archive tables.
- To manage the sizes of the audit archive tables in the SAS Web Infrastructure Platform's SharedServices database, you must periodically purge records from those tables.
- To ensure that data is available in memory after the server restarts, appended data is immediately written to a second location, the AUDIT_VISUALANALYTICS table in the autoloading data directory. This is standard autoloading append behavior. See ["How Autoloading Works"](#) on [page 22](#).
- To manage the size of the AUDIT_VISUALANALYTICS tables in the autoloading drop zone, a scheduled task deletes records of a specified age, and then refreshes the corresponding LASR table. In the initial configuration, the task runs daily and deletes records that are more than 30 days old.

How to Safely Enable Auditing

CAUTION! Audit data can consume significant amounts of disk space and processing capacity. To safely enable auditing, complete all of the following steps.

- 1 In the SAS Web Infrastructure Platform's SharedServices database, make sure that appropriate audit archive rules are in place. See [Archive Process for Audit Records](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Note: Predefined rules cause audit records that are older than 30 days to be archived, subject to the following constraints:

- Predefined rules apply to only actions that SAS Visual Analytics audits, sign-in actions, and sign-out actions. (Sign-in and sign-out actions are not included in the SAS Visual Analytics extraction of audit records or the SAS Visual Analytics administrative reports.)
- Predefined rules are not provided for sites that use a SAS Web Infrastructure Platform database other than PostgreSQL.
- Predefined rules do not replace any existing site-specific custom archive rules.

Note: For ID values to use in the archive rules, see [Table 3.9 on page 55](#). The suggested value for `FREQUENCY_NO` is 2592000000 milliseconds (30 days).

- 2 In the SAS Web Infrastructure Platform's SharedServices database, establish a procedure to periodically purge records from the audit archive tables. See [Purging Audit Records](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.
- 3 On the autoloader host, start the scheduled task that deletes old records from the full `AUDIT_VISUALANALYTICS` table. See ["Start auditRefresh" on page 128](#).
- 4 Set the property `va.AuditingEnabled` to `true`. See ["How to Set Configuration Properties" on page 139](#).
- 5 Restart the SAS Web Application Server.

Audit Content and Coverage

The following tables describe SAS Visual Analytics audit records. Here are some key points:

- To visualize audit information, see [Chapter 6, "Reports for Administrators," on page 121](#).
- In some cases, multiple audit records are written for a single user interaction. For example, if UserA opens ReportA, and ReportA uses TableA and TableB, records that are written include `[Report.BI]Open`, multiple `[Table]Read` records for TableA, and multiple `[Table]Read` records for TableB.
- In the `audit_info` field, `Security access denied` indicates that a permissions-based access denial from the LASR authorization service occurred. `Capacity access denied` indicates that a capacity-based access denial from the LASR authorization service occurred. See ["Limit Space for Tables" on page 100](#).
- The `server_app` field is populated for actions that use the transport service. For example, when a user prints a report object, the `executor_nm` value identifies the client (for example, Visual Analytics Viewer 7.4) and the `server_app` value identifies the underlying component (for example, Visual Analytics Transport Service 7.4).
- The `email_recipients` field is not populated for actions that are performed in SAS Mobile BI.
- For some of the specialized fields, `new` and `old` values are recorded. The `new` value reflects current information.

Table 3.7 Audit Content

Field	Description	Example Value
General:		
audit_id	Identifier for the audit record	871
timestamp_dttm	Date and time (GMT)	08:06:2014 06:42:59:219
user_id	Metadata name of the identity that performed the action	sasadm
action_type	Name of action	Add
object_type	Object type (in the audit service's type classification scheme)	Report.BI
executor_nm	Application name, device type (if applicable), and version	Visual Analytics Designer 7.4
action_success_flg	Whether the action succeeded (Y) or failed (N)	Y
audit_info	Information about a failed action, additional details	LASR_ACTION=TASK_TABLEINFO; Security access denied
Specialized:		
location	metadata path and type, or local filename	SBIP://METASERVER/User Folders/ncjoe/My Folder/MyReport(MyReport)
lasr_server_name	Machine name and port of a SAS LASR Analytic Server	abc.mycompany.com:7300
table_name	Server tag and name of a LASR table	HPS.CARS
client_id	IP address or mobile device ID	12.34.56.789
report_elements	Identifiers for successfully printed objects (or, all)	ve2

Field	Description	Example Value
server_app	Underlying component or service	Visual Analytics Transport Service 7.4
elapsed_time	Time for the execute method in a query (<i>seconds.milliseconds</i>)	27.829
export_output	Output type	XLSX
export_rows	Number of rows exported (or, all)	250
export_object	Name of the report object from which data is exported	List Table 2
email_sender	Email address	joe@company.com
email_recipients	One or more email addresses	tara@company.com,joy@company.com

Table 3.8 Audit Coverage

Audited Activity	[object_type] action_types	Specialized Fields
Subscribe to a report on a mobile device.	[BIReportSubscription] Create	client_id, location, server_app
Delete a report from a mobile device.	[BIReportSubscription] Delete	client_id, location, server_app
Open, create, save, save as, or delete a report.	[Report.BI] Open, Create, Save, Delete	client_id, location, oldlocation (for save as)
Move, copy and paste, or rename a report.	[Report.BI] Move, Copy, Rename	client_id, location, oldlocation
Send a link to a report via email.	[Report.BI] SendEmail	client_id, location, email_sender, email_recipients
Export data from an object within a report.	[Report.BI] Export	client_id, location, export_object, export_rows, export_output
Print some or all objects in a report to PDF.	[Report.BI] Print	client_id, location, report_elements, server_app
Automatic refresh of a report by the transport service.	[Report.BI] Execute	client_id, location, server_app
Start a server (or trigger autostart) from the UI.	[Server.LASR] Start	client_id, lasr_server_name
Stop a server.	[Server.LASR] Cancel	client_id, lasr_server_name
Read a LASR table.	[Table] Read	client_id, location, lasr_server_name, table_name
Read a source table (before import or load).	[Table] Read	client_id, location (of the source table)
Load, import, or reload a LASR table.	[Table] Add	client_id, location, lasr_server_name, table_name
Add a table to co-located or NFS-mounted storage.	[Table] Add	client_id, location (in metadata, new table)
Unload a LASR table.	[Table] Release	client_id, location, lasr_server_name, table_name

Audited Activity	[object_type] action_types	Specialized Fields
Delete a physical table from co-located HDFS.	[Table] Delete	client_id, location (in HDFS)
Append, modify, or delete rows; add computed columns.	[Table] Update	client_id, location, lasr_server_name, table_name
Open, create, save, save as, or delete a data query.	[VisualDataQuery] Open, Create, Save, Delete	client_id, location, oldlocation (for save as)
Move or rename a data query.	[VisualDataQuery] Move, Rename	client_id, location, oldlocation
Run a data query.	[VisualDataQuery] Execute	client_id, location, elapsed_time
Open, create, save, save as, or delete an exploration.	[VisualExploration] Open, Create, Save, Delete	client_id, location, oldlocation (for save as)
Move, copy and paste, or rename an exploration.	[VisualExploration] Move, Copy, Rename	client_id, location, oldlocation
Send a link to an exploration via email.	[VisualExploration] SendEmail	client_id, location, email_sender, email_recipients
Export data from an object within an exploration.	[VisualExploration] Export	client_id, location, export_object, export_rows, export_output
Print some or all objects in an exploration to PDF.	[VisualExploration] Print	client_id, location, report_elements
Access encrypted SASHDAT (use of passphrase).	[Library] or [Server.Hadoop] Read	client_id, library_name, or hadoop_server_name

Table 3.9 Audit Type IDs

Object Type (ID)	Action Types (ID)
Server.LASR (206)	Start (34), Cancel (47)
Server.Hadoop (208)	Read (45)
Library (31)	Read (45)
Table (32)	Read (45), Add (36), Release (48), Delete (2), Update (1)
BIReportSubscription (827000)	Create (0), Delete (2)
Report.BI (106)	Create (0), Delete (2), Open (13), Save (53), Move (39), Rename (40), Copy (16), SendEmail (44), Export (26), Print (7), Execute (35)
VisualExploration (101)	Create (0), Delete (2), Open (13), Save (53), Move (39), Rename (40), Copy (16), SendEmail (44), Export (26), Print (7)
VisualDataQuery (826001)	Create (0), Delete (2), Open (13), Save (53), Move (39), Rename (40), Execute (35)

Locked-Down Servers

You can limit the reach and activities of certain SAS servers. For more information, see [Locked-Down Servers](#) in the *SAS Intelligence Platform: Security Administration Guide*.

If you choose to lock down a server that is used by SAS Visual Analytics, make sure that the following directories are accessible to that server:

- `SAS-configuration-directory/Applications/SASVisualAnalytics`
- For a non-distributed server, the signature files directory. See [“Signature files location on server”](#) on page 104.
- For a distributed server, each user’s home directory (~) to provide access to SSH keys. See [Passwordless SSH](#) in the *SAS LASR Analytic Server: Reference Guide*.
- The directory where process IDs are written. See [“va.monitoringPath”](#) on page 142.
- The directory where last action logs are written. See [“va.lastActionLogPath”](#) on page 142.
- The directory that contains geographic data sets. See [“Geographic Data Sets”](#) on page 80.
- The directory that contains the SAS linguistic files for text analytics. See [“Supporting Text Analytics”](#) on page 77.
- Any directory to which users export code. See [“Record actions as SAS statements”](#) on page 92.
- Any directory that serves as a data provider for reload-on-start. See [“Reload-on-Start”](#) on page 18.
- Any directory from which users import non-local data. See [“Self-Service Import”](#) on page 16.
- The directory where scheduled jobs for SAS Visual Data Builder are written. (The standard location is in the [SAS configuration directory](#) at your equivalent of `SAS-application-server\SASEnvironment\SASCode\Jobs`.)
- The autoload drop zone for administrative reporting (if the lockdown affects the pooled workspace server within the default SAS Application Server). See [“How to Provide Administrative Data”](#) on page 123.

Access to SAS Mobile BI

About Mobile Device Management

Here are the key points:

- To manage device access to SAS Mobile BI, select **Tools** ► **Manage Devices** from the main menu in the administrator. You can manage devices either by exclusion or by inclusion.
 - If you manage by exclusion, all devices can use SAS Mobile BI, except those that are on the blacklist.
 - If you manage by inclusion, only devices that are on the whitelist can use SAS Mobile BI.
- A deployment enforces only one list (either the blacklist or the whitelist). In a new deployment the blacklist is enforced, so there are no device-level barriers to participation.
- You can modify both lists. Making changes to a list that is not currently enforced can help accommodate a future change.
- These lists affect devices, not users. To manage what a particular user can see or do, use permissions and capabilities.

How to Manage Mobile Devices

Blacklist a Device

Note: These instructions have an effect only if the blacklist is enforced.

To prevent a mobile device from using SAS Mobile BI:

- 1 In the main menu bar, select **Tools** ► **Manage Devices**.
- 2 On the **Mobile Devices** tab, select the **Blacklist** tab.
- 3 At the right edge of the tab, click **+**.
- 4 In the Add Device To Blacklist window, enter the ID of the device that you want to exclude from using SAS Mobile BI. (Or, to add multiple device IDs, click **Add List**.) Click **OK**.

Note: The information that you supply is not validated by the software.

TIP For a device that has already connected (or attempted to connect), you can initiate this task from the **Logon History** tab. Select the device, right-click, and select **Add to Blacklist**.

To remove a device from the blacklist, select it on the **Blacklist** tab, right-click, and select **Move to Whitelist**.

Whitelist a Device

Note: These instructions have an effect only if the whitelist is enforced.

To enable a mobile device to use SAS Mobile BI:

- 1 In the main menu bar, select **Tools** ► **Manage Devices**.
- 2 On the **Mobile Devices** tab, select the **Whitelist** tab.
- 3 At the right edge of the tab, click **+**.

- 4 In the Add Device To Whitelist window, enter the ID of the device that you want to enable to use SAS Mobile BI. (Or, to add multiple device IDs, click **Add List**.) Click **OK**.

Note: The information that you supply is not validated by the software.

TIP For a device that has already connected (or attempted to connect), you can initiate this task from the **Logon History** tab. Select the device, right-click, and select **Add to Whitelist**.

To remove a device from the whitelist, select it on the **Whitelist** tab, right-click, and select **Move to Blacklist**.

Determine Which List Is Enforced

In the toolbar at the top of the **Mobile Devices** tab, the **Enforced** drop-down list indicates which list is enforced.

In addition, text at the top of either the **Blacklist** tab or the **Whitelist** tab indicates the list that is not currently enforced.

TIP You can also verify the current configuration in SAS Management Console. The blacklist is enforced unless the `viewerservices.enable.whitelist.support` property is set to `true`. See [“viewerservices.enable.whitelist.support” on page 146](#).

Determine When a Device Was Blacklisted

Here is one way to determine when a device was blacklisted:

- 1 On the **Blacklist** tab, right-click on the device, and select **Copy Device ID**.
- 2 On the **Management History** tab, select **Device ID** from the **Filter** drop-down list.
- 3 Click in the text field, and enter Ctrl+V from the keyboard. (You cannot perform the paste action from the pop-up menu.)
- 4 Click **Apply**.

TIP You can also copy a device ID from the **Whitelist** tab. You can also paste a device ID into the **Device ID** filter on the **Logon History** tab.

Change How Devices Are Managed

CAUTION! These are deployment-level instructions that affect all access to SAS Mobile BI.

To switch from enforcing one list to enforcing the other:

- 1 Select **Tools** ► **Manage Devices** from the main menu.
- 2 Verify that the list that you intend to enforce is appropriately populated.
 - If you enforce the whitelist, the whitelist should contain all eligible devices. The blacklist is ignored.

- If you enforce the blacklist, the blacklist should contain all excluded devices. The whitelist is ignored.
- 3 In the toolbar at the top of the **Mobile Devices** tab, make a selection from the **Enforced** drop-down list. In the confirmation window, click **Yes**.

About the Mobile Devices Tab

Here are some details about the **Mobile Devices** tab:

- On the **History** tabs, you can filter by selecting an item from a **Filter** drop-down list, specifying a value, and clicking **Apply**.
 - The **Logon History** tab displays logon events. By default, only one logon event for each device is displayed. To view previous logon events, select the **Include device history** check box. The following occurrences are logon events:
 - A connection attempt that comes from a new source (a unique combination of device ID and user ID).
 - A connection attempt that is accompanied by a device change (such as a new operating system version or application version).
 - On the **Logon History** tab, the **Status** column provides information about a logon event. The **Status** column does not indicate the current status of a device connection.
 - When you right-click on a device on the **Logon History** tab, remember that only one list is in use. Adding a device to the list that is not in use has no immediate effect. For example, if your deployment uses the blacklist, adding a device to the whitelist has no immediate effect.
 - On the **Blacklist** and **Whitelist** tabs, each cell in the **User ID** column contains the user ID that connected (or attempted to connect) to SAS Mobile BI from the associated device. The user ID is provided for the purpose of helping you identify a device. If no user has attempted to connect from a particular device, no user ID is listed for that device. If multiple users have attempted to connect from a particular device, all of those user IDs are listed.
 - On the **Manage** tabs, you can right-click on a device ID, and select **Copy Device ID**. On the **History** tabs, you can paste a device ID into the text field next to the **Filter** drop-down list.
- Note:** A device ID is a unique identifier (usually a hardware device number) that is determined and communicated by the connecting mobile application.
- The **Management History** tab displays device management events, such as adding a device to a list or removing a device from a list. The **Admin ID** column provides the user ID of the administrator who performed each action.
 - When you right-click on a device in the blacklist or whitelist, you can choose either a move action or a remove action. In terms of immediate effect, there is no difference between these two actions.

Note: Authentication from SAS Office Analytics to SAS Visual Analytics does not use the SAS Visual Analytics Transport Service. For this reason, actions and information about the **Mobile Devices** tab in the administrator have no impact on SAS Enterprise Guide, SAS Add-in for Microsoft Office, and SAS Web Parts for SharePoint.

Protections for Mobile Content

In addition to the blacklist and whitelist functionality, protections that are specific to mobile content include the following:

- To prevent offline access to mobile data, assign users or groups to a role that has the **Purge Mobile Report Data** capability. See [“Purge Mobile Report Data” on page 137](#).
- To limit offline access to mobile data, assign users or groups to a role that has the **Limit Duration of Offline Access** capability. See [“Limit Duration of Offline Access” on page 137](#).
- To require knowledge of an application passcode, assign users or groups to a role that has the **Require Passcode on Mobile Devices** capability. See [“Require Passcode On Mobile Devices” on page 137](#).
- Content on a mobile device is encrypted by the device’s operating system. For information about encrypted communication for mobile devices, see the [SAS Intelligence Platform: Middle-Tier Administration Guide](#).

Authentication

Introduction

SAS Visual Analytics uses platform-level functionality for authentication. See [Authentication Model](#) in the *SAS Intelligence Platform: Security Administration Guide*. For information about authentication for mobile devices, see the [SAS Intelligence Platform: Middle-Tier Administration Guide](#).

This topic provides details that are specific to SAS Visual Analytics.

Shared Accounts for Self-Service Imports

To enable users to import data under a shared account, configure SAS token authentication for a general purpose workspace server. See [SAS Token Authentication](#) in the *SAS Intelligence Platform: Security Administration Guide*.

To set up multiple levels of access, use multiple shared accounts. Here is a summary of one approach:

- 1 For each distinct set of secured resources, create a service account that can authenticate to the SAS LASR Analytic Server. Make sure the account has the privileges that are required to operate the server and load data. See [“Host Account Privileges” on page 5](#).
- 2 For each service account, create a SAS Application Server that contains a standard workspace server. See [“Add a New Server” on page 90](#).
- 3 Configure each standard workspace server for SAS token authentication. For each standard workspace server, use a different service account as the launch credential. See [How to Configure SAS Token Authentication](#) in the *SAS Intelligence Platform: Security Administration Guide*.

- 4 For each SAS Application Server, create a corresponding SAS LASR Analytic Server instance. Assign a unique signature files directory to each instance. Give each service account exclusive host access to the signature files directory for its server instance. See [“Add a SAS LASR Analytic Server” on page 97](#).
- 5 For each SAS LASR Analytic Server instance, create one or more LASR libraries. Assign each library to the SAS Application Server that corresponds to the library’s SAS LASR Analytic Server instance. See [“Add a LASR Library” on page 98](#) and [“Which Server is Used?” on page 90](#).
- 6 On the **Authorization** tab for each SAS Application Server and SAS LASR Analytic Server instance, limit ReadMetadata access. See [Hide Server Definitions](#) in the *SAS Intelligence Platform: Security Administration Guide*.

Note: Keep the initial SAS Application Server (for example, **SASApp**) available for general use.

Policy for Concurrent User Logins

SAS Visual Analytics does not support `deny` or `logout` values for the `Policy.ConcurrentUserLogins` property. For successful interactions with the SAS LASR Analytic Server, make sure this property is set to `allow`.

The `Policy.ConcurrentUserLogins` property is documented in [Disabling Concurrent Sign In Sessions](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Encryption

Introduction

SAS Visual Analytics uses platform-level functionality to encrypt sensitive data in transit and on disk. See [Encryption Model](#) in the *SAS Intelligence Platform: Security Administration Guide*.

This topic helps you get started with AES encryption of data that SAS Visual Analytics writes to disk.

On-Disk Encryption of Reload-on-Start Files

Overview

To increase protection of data in a [reload-on-start](#) backing store, bind the backing store to metadata and enable encryption on the corresponding secured library.

CAUTION! Binding physical data to metadata is an advanced technique.

Before you configure encryption, see [Overview of Metadata-Bound Libraries](#) in *SAS Guide to Metadata-Bound Libraries* and review the following key points.

Key Points

- Access to in-memory data is unaffected by encryption of corresponding backing store files. Encrypted backing store files are not read or written as quickly as unencrypted backing store files.
- Each metadata-bound backing store is represented twice in metadata:
 - One representation is a *traditional library* that is assigned as the backing store for a particular LASR library.
 - The other representation is a *secured library* to which the physical backing store is bound.

To read from or write to an encrypted backing store, you must have sufficient metadata-layer permissions on both the traditional library and the secured library.

- Passphrases (**Encrypt Key** values) and passwords are not promoted. After the initial import of a secured library, you must re-apply the passphrase and password (or passwords) in the target environment. See [Promoting Secured Data Folders, Secured Library Objects, and Secured Table Objects](#) in the *SAS Intelligence Platform: System Administration Guide*.
- To use AES, SAS/SECURE must be installed and available. See [SAS/SECURE](#) in *Encryption in SAS*.

Encrypt a Backing Store Library

- 1 Identify or create a backing store for a LASR library that supports reload-on-start and will contain sensitive data. See [“How to Enable Reload-on-Start” on page 19](#).

Note: A backing store is a host directory that is registered in metadata and assigned to a LASR library as its data provider library.

- 2 Log on to [SAS Management Console](#) as someone who has the following privileges:
 - Host-layer control of the target directory:
 - On Windows, you must have full control of the directory.
 - On UNIX, you must be an owner of the directory.
 - Metadata-layer access to the **Secured Libraries** folder. The SAS Administrators group usually has the necessary access.
- 3 On the **Folders** tab, navigate to **System** ► **Secured Libraries**, right-click, and select **New** ► **Secured Library**.

Note: As an alternative, you can first create a secured library folder, and then create the new secured library within that folder. If you are creating multiple secured libraries, it is usually more efficient to create one or more folders, so that each secured library inherits effective permissions from a parent folder. Each secured table inherits effective permissions from its parent secured library. See [Object Creation, Location, and Inheritance](#).

- 4 On the **General** page, enter a name and description. Click **Next**.
- 5 On the **Connection Data** page, provide information as follows:
 - a Select a SAS Application Server. For **Library Path**, click **Browse**, and select your target directory.

- b Enter and confirm a library password.

CAUTION! If you lose a library password, you cannot unbind or modify the library. Keep track of the password (or passwords) that you enter.

Note: The password must be a valid SAS name. (It must begin with a letter or an underscore. It can include letters, underscores, and numeric digits. It is not case sensitive. It cannot be longer than 8 characters.) If you need to create a longer, compound password, select the **Specify multiple passwords** check box and specify multiple passwords.

- c Select the **Require Encryption** check box and its **Yes** radio button. With this setting, the following files are encrypted:
 - Any unencrypted tables that already exist in the directory.
 - Tables that are later added to the directory during imports that participate in reload-on-start.
 - Tables that are later added to the directory directly through SAS code. (Do not use a host copy utility to add tables to the directory.)
- d Select the **Encryption Type** check box and its **AES** radio button.
- e Leave the first **Encrypt Key** field blank. That field is not applicable when you create a secured library for a directory that is empty or contains only unencrypted files.

Enter a value in the **New Encrypt Key** and **Confirm Encrypt Key** fields. Here are some details:

- Keep track of the value that you enter.
 - The value that you enter functions as a passphrase that is used to create the actual key with which AES encrypts the target tables.
 - The value that you enter is automatically enclosed in quotation marks when it is saved, so the value is case sensitive. (Do not include quotation marks when you enter a value.) For more information, see [ENCRYPTKEY=](#) in the *SAS Data Set Options: Reference*.
- f Click **Finish**. When prompted, click **Yes** to review the log.

- 6 Review and adjust metadata-layer access to the new secured library.

- a Right-click on the new secured library, and select **Properties**.

Note: If you are managing permissions at the folder level, right-click on the appropriate secured library folder.

- b On the **Authorization** tab, use one of the following techniques:
 - Grant all permissions to a broad group, such as PUBLIC, SASUSERS, or Visual Analytics Users. This simple approach uses the secured library only to provide on-disk encryption.
 - Grant permissions in a more selective, limited manner. This advanced approach uses the secured library to provide enhanced enforcement of authorization constraints, as well as on-disk encryption. See [Permissions for Metadata-Bound Data](#). Here are some examples:
 - To import a table that participates in reload-on-start, a user must have the Create Table permission on the corresponding secured library object.

Note: If a same-named table already exists in the metadata-bound backing store, the user must also have the Alter Table permission on the corresponding secured table object.

- To reload a table (using reload-on-start), the user who triggers the SAS LASR Analytic Server to start must have the ReadMetadata and Select permissions on the corresponding secured table object.

7 To verify the results:

- In the data builder, explorer, or designer, import a participating table. For example, import a local file to a LASR library that supports reload-on-start from an AES-encrypted backing store.
- In the administrator, stop and then start a SAS LASR Analytic Server that is associated with a LASR library that supports reload-on-start from an AES-encrypted backing store.
- In SAS code, run the CONTENTS procedure against the backing store library. The procedure output indicates whether tables are encrypted.
- For further verification, see [Validating a Metadata-Bound Library](#).

Additional Information

This topic is intended to help you get started. For alternate methods and related tasks, see the chapter [Implementation of Metadata-Bound Libraries](#) in the *SAS Guide to Metadata-Bound Libraries*.

For example:

- To update a passphrase, see [Changing a Metadata-Bound Library's Encryption Options](#).
- To remove encryption, see [Unbinding a Metadata-Bound Library](#).

On-Disk Encryption of SASHDAT Files

Overview

To increase protection of SASHDAT files, enable on-disk AES encryption for a library that uses the SASHDAT engine.

CAUTION! Encrypting SASHDAT files can significantly impact data availability and memory consumption. Before you configure encryption, review the following sections.

Key Points

- Access to in-memory data is unaffected by encryption of corresponding SASHDAT files. Encrypted SASHDAT files are not read or written as quickly as unencrypted SASHDAT files.
- Encrypted SASHDAT files are available to only requests that are authorized by the SAS LASR Authorization Service (which is also referred to as the *signer*). For authorized requests, the authorization service retrieves the encryption passphrase from metadata and provides it to the SASHDAT engine. This enables the SASHDAT engine to encrypt and unencrypt data as needed. Here are the related requirements:

- The connection object for the associated data server must enable the authorization service. For SAS Visual Analytics, encryption of SASHDAT files is always signer-managed.
- In an encrypted SASHDAT library, users who add, delete, or load associated data must have the Read permission.
- Within an environment, each Hadoop server must have a unique host name. Within a Hadoop server, each SASHDAT library must have a unique host path.
- Encrypted SASHDAT files always consume unmapped memory when they are loaded. Memory mapping is not available for LASR tables that are loaded from encrypted SASHDAT files.
- Encrypted SASHDAT files are always uncompressed when they are loaded.

Note: You can use compression to conserve disk space for an encrypted SASHDAT file. However, compressing an encrypted SASHDAT file does not conserve memory. Before an encrypted file is loaded, it must be decrypted—decryption requires that the data be uncompressed.
- Changes that you make to SASHDAT encryption settings do not affect existing SASHDAT files.
- If you want to centralize SASHDAT encryption configuration, specify encryption settings at the server level, and configure each associated library to inherit its settings.
- Passphrases (**Encrypt Key** values) are not promoted. After the initial import of an encrypted SASHDAT library or server, you must use SAS Management Console to re-apply the passphrase in the target environment.

Note: If both the source and the target environment reference the same physical data instance, then you do not have to copy and replace that data (because that data remains encrypted).
- To encrypt SASHDAT files, the following requirements must be met:
 - The SAS TGrid Encryption Extension must be installed and available. See the *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.
 - To use AES, SAS/SECURE must be installed and available. See [SAS/SECURE](#) in *Encryption in SAS*.

Protect Encryption Settings

To protect SASHDAT encryption settings, limit WriteMetadata access to the SASHDAT library.

Limiting WriteMetadata access is necessary because anyone who has WriteMetadata access to an encrypted SASHDAT library can modify its VA.Encryption.Enabled extended attribute. That attribute is intended for exclusively internal purposes. Nobody should directly set, modify, or delete the VA.Encryption.Enabled attribute. Instead, unrestricted users can manage settings from the library's **Options** tab, as instructed below.

Note: Limiting WriteMetadata access has side effects. Users who lack WriteMetadata access to a library cannot register tables in or delete tables from that library.

For example, for maximum protection, you might give the PUBLIC group an explicit denial of WriteMetadata on the **Authorization** tab of an encrypted SASHDAT library. With that setting, only an unrestricted user has WriteMetadata access to the library. Actions that add or remove SASHDAT table metadata for that library must be performed by an unrestricted user.

Encrypt a SASHDAT Library

- 1 Identify a SASHDAT library that references an empty target directory.

Note: These instructions are for an existing SASHDAT library. To create a new library that uses the SASHDAT engine, see the [SAS Intelligence Platform: Data Administration Guide](#).
- 2 Log on to [SAS Management Console](#) as an unrestricted user (for example, sasadm@saspw).
- 3 On the library, set encryption options and adjust metadata-layer permissions.
 - a On the **Plug-ins** tab, expand the **Data Library Manager** node and then the **Libraries** node. Right-click the target library, and select **Properties**.
 - b On the **Options** tab, make the following changes:
 - i In the **Enable Encryption** field, select the **Yes** radio button.

TIP To instead make the library inherit encryption settings from its associated data server, select the **Inherit from server** radio button. Then, verify that encryption is enabled on the data server's **Options** tab. Inherited settings are dynamic. Server-level changes affect all associated libraries that are configured to inherit server-level settings.

- ii Enter a value in the **New Encrypt Key** and **Confirm Encrypt Key** fields.

CAUTION! If the passphrase is lost, all access to the encrypted data is irretrievably lost. Keep track of the passphrase that you enter.

Here are some details:

- The value that you enter functions as a passphrase that is used to create the actual key with which AES encrypts the target tables.
 - The value that you enter is automatically enclosed in quotation marks when it is saved, so the value is case sensitive. (Do not include quotation marks when you enter a value.) For more information, see [ENCRYPTKEY=](#) in the *SAS Data Set Options: Reference*.
- c On the **Authorization** tab, grant the Read permission to users who add data to the encrypted library, load data from the encrypted library, or delete data from the encrypted library. In most cases, it is sufficient to grant the Read permission to the following groups:
 - Visual Analytics Data Administrators
 - Visual Data Builder Administrators

Note: For an unencrypted SASHDAT library, the Read permission is not required or enforced.

Note: You can grant the Read permission on a parent folder, rather than directly on the library.

- d On the **Authorization** tab, make sure that WriteMetadata access is limited. See [“Protect Encryption Settings” on page 65](#).
 - e Click **OK**.
- 4 On the associated server’s connection object, enable the LASR authorization service.

CAUTION! If the LASR authorization service is not enabled, added tables are not encrypted and encrypted tables are not available.

- a On the **Plug-ins** tab, expand **Server Manager**, and select the target data server.
 - b In the right pane, right-click the server’s connection object, and select **Properties**.
 - c On the **Options** tab, make sure the **Use LASR authorization service** check box is selected.
- 5 To verify the results:
- Add tables to the SASHDAT library.
 - Load tables from the SASHDAT library to a SAS LASR Analytic Server.
 - For SASHDAT files in co-located HDFS, examine each table’s **Encryption** property on the administrator’s **HDFS** tab. See [“About the HDFS Tab” on page 112](#).
 - In SAS code, run the CONTENTS procedure against the SASHDAT library. The procedure output indicates whether tables are encrypted.

Additional Information

To update a passphrase:

- 1 If the target directory currently contains tables, move those tables to an alternate location.

TIP One approach is to load the existing tables to memory and then delete both the physical tables and the corresponding metadata definitions.

CAUTION! If you delete table metadata, you must manually repair or re-create any affected objects (for example, explicit and row-level permissions).

- 2 Log on to [SAS Management Console](#) as an unrestricted user (for example, sasadm@saspw). On the appropriate server or library, enter a new value in the **New Encrypt Key** and **Confirm Encrypt Key** fields.
- 3 If you moved tables in step 1, move them back to the target directory. As the tables are written back to the target directory, they are encrypted using the new encryption key (which is generated from the updated passphrase).

TIP If you loaded tables from co-located HDFS or NFS-mounted MapR in step 1, you can use the data builder to save the tables back to the target directory.

For more information, see [Data Encryption](#) in the *SAS LASR Analytic Server: Reference Guide*.

4

Fine-Tuning

Administering SAS Home	70
Introduction	70
Home Page: Roles and Capabilities	70
Home Page: Configuration Properties	71
Home Page: Manage Shared Content	73
Supporting Guest Access	73
Introduction	73
Limit Content	74
Limit Functionality	74
Enable or Disable Guest Access	75
Customize the Home Page for Guests	75
Customize the Web Viewer for Guests	76
Supporting Text Analytics	77
Introduction	77
Linguistic Files	77
Stop Lists	77
Supporting Geographic Maps	78
Introduction	78
OpenStreetMap Server	78
Esri Server	79
Geographic Data Sets	80
Supporting Stored Processes	80
Supporting Footers in Printed Reports	80
Supporting Report Distribution	82
Introduction	82
About the Scheduling Server	82
About the Size of Email Attachments	82
Supporting the Search Features	83
Supporting User-Defined Formats	83
Managing Alerts and Notifications	84
Introduction	84
Requirements	84
How to Delete Other Users' Alerts	85
Supporting the Monitoring Features	85
Introduction	85
Network Name Resolution	86

Managing the Monitoring Server	86
Logging for the Monitoring Server	87
Adjusting the Logging Configuration	87
Log Directories	87
How to Change Log Levels	88
How to Log Submitted Code	88
How to Log LASR Actions	88
Logging of Access Denials	89
Logging for SAS Mobile BI	89
Logging for Platform Servers	89
Making More Fonts Available on UNIX	89
Using Multiple SAS Application Servers	90
Which Server is Used?	90
Add a New Server	90
Setting User Preferences	91
About User Preferences	91
User Preference: SAS Application Server	91
Other User Preferences in the Administrator	92

Administering SAS Home

Introduction

Note: Home page administration is documented in [The SAS Visual Analytics Home Page](#) in the *SAS Intelligence Platform: Web Application Administration Guide*. As a convenience, documentation for the 7.4 release of the home page is reproduced here.

The home page is a SAS web application that provides integrated access to participating licensed SAS solutions, including SAS Visual Analytics. The home page's official product and software component names (**SAS Visual Analytics Hub** and **Visual Analytics Hub**) reflect the original scope of the home page.

Home Page: Roles and Capabilities

Here are the predefined roles for the home page:

Home: Usage

provides non-administrative functionality for the home page. In the standard configuration, the SASUSERS group is a member of this role.

Home: Administration

provides all capabilities for the home page. In the standard configuration, the Visual Analytics Data Administrators group is a member of this role.

Table 4.1 Home Page: Capabilities by Role

Capability	Usage Role	Administration Role
Administer Hub		✓
Add and View Comments	✓	✓
Create Collections	✓	✓
Personalization	✓	✓

Here are descriptions of the home page capabilities:

Administer Hub

manage shared content on the home page.

Add and View Comments

add comments, view comments, and edit your own comments.

Note: The home page's modern [presentation mode](#) does not support comments.

Note: To delete comments and edit other users' comments, you need the capabilities that are listed under **SAS Application Infrastructure** ► **Comments**. Consider making any users that need these capabilities members of the **Comments: Administrator** role.

Create Collections

create groups of bookmarks that reference objects. (The Personalization capability is also required.)

Personalization

use individualized features such as settings, favorites, and lists of recent objects.

Note: In the standard configuration, the property [sas.home.exclude.capability.users](#) blocks individualized home page capabilities for the service identity that provides guest access (the SAS Anonymous Web User).

Home Page: Configuration Properties

The home page software component (**Visual Analytics Hub**) is a sibling of the **Visual Analytics** software component, immediately below **SAS Application Infrastructure**. This relationship is reflected on the **Plug-ins** tab in SAS Management Console as follows:

```
-Application Management
--Configuration Manager
---SAS Application Infrastructure
----Visual Analytics
----Visual Analytics Hub
```

Configuration properties for the home page include the following:

App.AllowGuest

enables or disables [guest access](#). Valid values are `true` and `false`.

sas.home.allow.anonymous.user.personalization

determines whether the service account that supports guest access can modify the home page. Valid values are `true` and `false`. The default is `false`.

Note: This property enables an administrator to temporarily override the capabilities block that the property `sas.home.exclude.capability.users` implements.

sas.home.exclude.capability.users

blocks individualized home page capabilities for one or more users—regardless of group and role memberships. In the standard configuration, this property is not specified. Set this property only at the direction of SAS Technical Support.

- To block the SAS Anonymous Web User (and any other configured anonymous users), leave this property unspecified so that the default value, `ANONYMOUS_USER`, is in effect.
- To block a different user, specify the user ID that is stored for that user in metadata.
- To block multiple users, specify a comma-separated list of user IDs.
- To unblock all users, leave this property unspecified and set the property `sas.home.allow.anonymous.user.personalization` to `true`.

home.ui.mode

enables an administrator to force the use of a particular presentation mode for the home page, regardless of individual user preference settings. In the initial configuration, this property is not specified, so individual user preference settings are effective.

(unspecified) If the `home.ui.mode` property is not specified, the modern presentation mode is used, unless a user selects **Classic** (in his or her **Default Appearance** setting for the home page) or uses a web browser that does not support the modern presentation mode (for example, Microsoft Internet Explorer 9).

`classic` Forces the use of the Flash presentation mode.

`modern` Forces the use of the HTML5 presentation mode unless your web browser does not support that mode.

Note: The modern presentation mode does not support right-to-left (RTL) languages.

TIP To force the use of a particular presentation mode for the web viewer, set the property `vav.ui.mode` on the **Visual Analytics Viewer** software component.

va.supportSharedThumbnails

determines whether the home page displays available thumbnails (specific preview images for reports and explorations).

Note: Thumbnails are available for reports and explorations that were rendered while the `va.supportSharedThumbnails` property was set to `true` at the [suite-level](#) for SAS Visual Analytics.

Valid values are as follows:

- false No thumbnails are displayed. This is the default value.
- true Any available thumbnails are displayed. In the modern presentation mode, thumbnails are not immediately visible. Instead, thumbnails are displayed on the **Preview** tab within the object inspector.

CAUTION! Shared thumbnails reflect the most recently rendered version of an object and are not subject to each requesting user's data access permissions. Before you enable shared thumbnails, consider the security implications.

TIP To make thumbnails available, you must set `va.supportSharedThumbnails` to `true` on two separate software components: **Visual Analytics Hub** and **Visual Analytics**.

To set configuration properties for the home page:

- 1 Open SAS Management Console, and click the **Plug-ins** tab.
- 2 Navigate to **Configuration Manager** ► **SAS Application Infrastructure** ► **Visual Analytics Hub**.
- 3 Right-click **Visual Analytics Hub**, and select **Properties**.
- 4 Make changes on the **Advanced** tab.
- 5 Restart the SAS Web Application Server.

Home Page: Manage Shared Content

Administrators can add, organize, and remove shared content on the home page. For instructions, access the online Help from the home page's banner or menu bar. For the classic mode, see also [About the Classic SAS Visual Analytics Home Page](#) in the *SAS Visual Analytics: User's Guide*.

See also "Customize the Home Page for Guests" on page 75.

Supporting Guest Access

Introduction

Note: For the most current and comprehensive information about guest access, see [Configuring Guest Access](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*. As a convenience, some information about guest access is provided here.

Guest access is an optional feature that provides anonymous access to a subset of resources and functionality. In guest access, there is no individualized authentication of the requesting user, so there are no requirements for individual user accounts or metadata identities. Instead, all guest users are authenticated as the same service account (the SAS Anonymous Web User). That service account functions as the single surrogate identity for all guest users. All guest

users can see what the SAS Anonymous Web User can see, and do what the SAS Anonymous Web User can do.

To provide guest access within an intranet only, place the applications behind a firewall. See [Best Practices for Configuring Your Middle Tier](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

To use guest access with web authentication, additional middle-tier configuration is required. See [Fallback to SAS Form-based Authentication](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Limit Content

Any content that the SAS Anonymous Web User can access is available to all guest users.

CAUTION! Grants to the SASUSERS and PUBLIC groups can introduce additional content at the guest access URLs. If your deployment supports guest access, it is important to review access that is granted to the SASUSERS and PUBLIC groups. The SAS Anonymous Web User is an implicit member of those groups, so any content that you make available to those groups is potentially available at the guest access URLs.

Here are some guidelines for managing access:

- Do not expect user- or group-based access distinctions (such as row-level security) for guests. Guest access provides only generic, lowest-common-denominator access to content.
- Review the metadata-layer [permissions on page 36](#) that are granted to the SASUSERS and PUBLIC groups. You can use either of the following approaches to exclude content from guest access:
 - Where access is granted to the SASUSERS or PUBLIC group, add denials for the SAS Anonymous Web User.
 - Replace grants to the SASUSERS or PUBLIC group with grants to the Visual Analytics Users group and the SAS System Services group.
- Do not revoke the SAS Anonymous Web User's ReadMetadata access to the `/system` folder.

Limit Functionality

Guest access functionality corresponds to the capabilities of the SAS Anonymous Web User.

- For the home page, the property [sas.home.allow.anonymous.user.personalization](#) ensures that guest access does not include individualized capabilities.
- For SAS Visual Analytics, the Visual Analytics: Basic role provides an appropriate set of guest access capabilities. Do not permanently give the Personalization capability to the Visual Analytics: Basic role. Failure to conform to this guideline causes each user's experience to reflect the activities of the previous user.

CAUTION! Any capabilities that the SASUSERS or PUBLIC group has can expand guest access functionality. This expansion of functionality can cause unintended results. If your deployment supports guest access, it is important to

review the capabilities of the SASUSERS and PUBLIC groups. The SAS Anonymous Web User is an implicit member of those groups.

Enable or Disable Guest Access

Enable Guest Access

The preferred method for configuring guest access is to make the following choices during installation:

- Create a SAS Anonymous Web User.
- Enable guest access for the home page and SAS Visual Analytics.

If you enable guest access during installation, the home page, the web viewer, and transport service (SAS Mobile BI) allow users to connect as the guest user. Users can choose to sign in to those applications as a guest.

Note: For the home page and the web viewer, users can instead explicitly specify a guest access URL. For example:

```
http://host/SASVisualAnalyticsHub/guest.jsp
```

For the exact URL, see the file [SAS-configuration-directory/Documents/Instructions.html](#) on the middle-tier machine.

If you need to configure guest access as a post-installation task:

- 1 If the SAS Anonymous Web User does not already exist in your deployment, create that service identity. See [Using the SAS Anonymous Web User with SAS Authentication](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.
- 2 In SAS Management Console's Configuration Manager, set the property [App.AllowGuest](#) to `true` on the **Visual Analytics Hub**, **Visual Analytics Viewer**, and **Visual Analytics Transport Service** nodes.
- 3 Restart the SAS Web Application Server.

Disable Guest Access

To disable guest access for an individual application, set the application's `App.AllowGuest` property to `false`, and restart the SAS Web Application Server.

To disable guest access system-wide, set the property `Policy.DisallowGuestAccess` to `true` for Logon Manager.

Customize the Home Page for Guests

TIP If guest access is enabled, it is a good practice to periodically access the home page as a guest to verify that only the intended resources and functionality are available to guests.

To customize the [home page](#) for guests:

- 1 Identify the changes that you want to make.

TIP This step helps you minimize the period of time in which another user might sign in as the guest user and inadvertently affect the guest access configuration.

- a Access the home page as yourself, and familiarize yourself with the available customizations. You can make changes such as the following:
 - Add, remove, or reorganize collections, shortcuts, and links.
 - Change application settings by clicking your name (in the upper right), and then selecting **Settings**.

Note: These instructions are for the modern mode. For details or classic mode instructions, see the online Help.
 - b Access the home page as a guest, and examine the current configuration. Notice that if you click **SAS Anonymous Web User** in the banner, the **Settings** menu item is not available.
- 2 Temporarily enable the SAS Anonymous Web User to modify the home page.
 - a Set the property `sas.home.allow.anonymous.user.personalization` to `true`.
 - b Restart the SAS Web Application Server.
 - 3 Access the home page as a guest. Notice that if you click **SAS Anonymous Web User** in the banner, the **Settings** menu item is available because anonymous user personalization is now enabled.
 - 4 Make the changes that you identified in step 1.
 - 5 Set the property `sas.home.allow.anonymous.user.personalization` to `false`, and restart the SAS Web Application Server.
 - 6 Access the home page as a guest.
 - a Verify that the results are as expected.
 - b Verify that you (as the SAS Anonymous Web User) cannot make any further customizations.

Customize the Web Viewer for Guests

To customize the web viewer for guests:

- 1 Temporarily add the Personalization capability to the Visual Analytics: Basic role.
 - a Log on to SAS Management Console as someone who has user administration capabilities (for example, `sasadm@saspw`).
 - b On the **Plug-ins** tab, select **User Manager**.
 - c In the right pane, right-click the **Visual Analytics: Basic** role, and select **Properties**.

- d On the **Capabilities** tab, expand the **Visual Analytics** node, and select the check box for the **Personalization** capability. Click **OK**.
- 2 Access the web viewer as a guest.
- 3 As soon as your session is established, remove the Personalization capability from the **Visual Analytics: Basic** role.

TIP Minimizing the period of time in which the Personalization capability is granted to the **Visual Analytics: Basic** role reduces the risk of another user inadvertently affecting the guest access configuration.

- 4 Change web viewer settings for the SAS Anonymous Web User as needed, and then sign out.
 - 5 Access the web viewer as a guest.
 - a Verify that the results are as expected.
 - b Verify that you (as the SAS Anonymous Web User) cannot make any further customizations.

Supporting Text Analytics

Introduction

This topic provides information to help administrators support text analysis features. For user instructions, see the *SAS Visual Analytics: User's Guide*.

Linguistic Files

To enable a server to access required linguistic files, set a server-level extended attribute. See [“Extended Attributes: SAS LASR Analytic Server” on page 101](#).

Stop Lists

To omit certain words from text analysis that is performed by a SAS LASR Analytic Server, register and load a stop list for that server. For example, you can filter out noise by omitting commonly used words. For instructions, see [Load a Stop List](#) in the *SAS Visual Analytics: User's Guide*.

TIP To find the location of the stop lists that SAS provides, examine a predefined SAS LASR Analytic Server's **Extended Attributes** tab. The field names are `VA.TextAnalyticsStopList` and `VA.TextAnalyticsStopList.de`.

Supporting Geographic Maps

Introduction

Use of geo maps requires a connection to a supported geo map server.

Note: If the geo map server URLs that you use are not in your deployment's cross domain proxy servlet whitelist, geo map functionality might be incomplete. See [Configuring the Cross Domain Proxy Servlet through a Whitelist](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

OpenStreetMap Server

Hosted by SAS

By default, SAS Visual Analytics retrieves mapping tiles from OpenStreetMap servers that SAS hosts. The OpenStreetMap servers that SAS hosts support replication and failover, providing reliable and dependable access.

Maps are rendered internally within SAS Visual Analytics. Only the following information is exchanged:

- Requests for tile numbers (in a URL format) are sent from SAS Visual Analytics to an OpenStreetMap server.
- Map images are returned from an OpenStreetMap server to SAS Visual Analytics.

To change the protocol that SAS Visual Analytics uses to connect to the OpenStreetMap servers that SAS hosts, set the property [“va.SASGeomCommunicationProtocol”](#) on page 142.

Note: The OpenStreetMap servers that are hosted by SAS log each request that is made. Each log entry includes only the requesting IP address and the specific tile that is requested.

Hosted Elsewhere

As an alternative to using the OpenStreetMap servers that SAS hosts, you can install, configure, host, and maintain an OpenStreetMap server at your site. This is a complex task that should be attempted only after you have carefully evaluated the requirements, needs, benefits, and maintenance responsibilities at your site. For information about OpenStreetMap servers, see www.openstreetmap.org.

To reference an alternate OpenStreetMap server from SAS Visual Analytics:

- 1 Set the property [“va.GeoMapServerUri”](#) on page 141.
 - a On the **Plug-ins** tab in SAS Management Console, navigate to **Application Management** ► **Configuration Manager** ► **SAS Application Infrastructure** ► **Visual Analytics**. Right-click, and select **Properties**.

- b On the **Advanced** tab, in the `va.GeoMapServerUrl` property, specify a comma-delimited list of alternate URLs.

Note: The protocol that you specify (`http` or `https`) determines the protocol that SAS Visual Analytics uses to connect to the alternate OpenStreetMap servers.

- 2 On the **SAS Application Infrastructure** node, update the advanced property `sas.web.cdps.knownHosts` to include the alternate URLs.
- 3 Restart the SAS Web Application Server.

Esri Server

Using an Esri server is an optional additional configuration. In deployments that reference an Esri server, the explorer and the designer provide user preferences and per-object settings that determine which server (OpenStreetMap or Esri) is used.

Note: Only Esri maps published using the WKID (well-known ID) format are supported.

To reference an Esri server from SAS Visual Analytics:

- 1 Set the property "[va.SASGeomapEsriURL](#)" on [page 142](#).
 - a On the **Plug-ins** tab in SAS Management Console, navigate to **Application Management** ► **Configuration Manager** ► **SAS Application Infrastructure** ► **Visual Analytics**. Right-click, and select **Properties**.
 - b On the **Advanced** tab, in the `va.SASGeomapEsriURL` property, specify a URL that includes a protocol, the Esri server's host name, and the REST endpoint of the server.
 - You can reference an Esri server (ArcGIS for Server, 10.1 or later) that you install, configure, host, and maintain at your site. For example: `http://my.arcgis.com:6080/arcgis/rest/services`
 - You can reference public ArcGIS Online sample map services that do not require authentication. For example: `http://services.arcgisonline.com/ArcGIS/rest/services`
 - You cannot reference an ArcGIS site that requires authentication.
 - You cannot reference an individual subfolder or map service. You must reference the REST endpoint of the server.
 - The protocol that you specify (`http` or `https`) determines the protocol that SAS Visual Analytics uses to connect to the Esri server.
- 2 On the **SAS Application Infrastructure** node, update the advanced property `sas.web.cdps.knownHosts` to include the Esri URL.
- 3 Restart the SAS Web Application Server.

Geographic Data Sets

SAS provides data sets that contain geographic information for several geographic domains (for example, states in the United States and ZIP codes for cities in the United States). The data sets (ATTRLOOKUP and CENTLOOKUP) are in the [SAS configuration directory](#) at your equivalent of `/SASApp/Data/valib/`. A corresponding library (for example, **SASApp - valib**) is registered in metadata.

In addition to the predefined geographical roles that use the SAS geographic data sets, you can define custom geographical roles for your data. If your data contains latitude and longitude values, then you can assign custom geographical roles using those values.

- For instructions for the explorer, see [Define a Geography Data Item By Using a Custom Geographic Role](#) in the *SAS Visual Analytics: User's Guide*.
- For instructions for the designer, see [Create a Custom Geography Data Item](#) in the *SAS Visual Analytics: User's Guide*.

Supporting Stored Processes

A stored process is a SAS program that is stored on a server and defined in metadata. For information about how stored processes are incorporated in SAS Visual Analytics, see the [SAS Visual Analytics: User's Guide](#). For information about how to create and register a stored process, see [Managing Stored Process Metadata](#) in the *SAS Stored Processes: Developer's Guide*.

The following considerations are specific to the administration of stored processes for SAS Visual Analytics:

- Stored processes can use any available data source (not only LASR tables). However, running stored processes against large LASR tables is not a high-performance operation. Any referenced LASR tables must be read from the SAS LASR Analytic Server into a SAS session in the SAS Stored Process Server. Using a stored process to read large tables from memory is not a high-performance operation.
- Most SAS procedures are available to only sites that license additional software (such as Base SAS). For a site that licenses only SAS Visual Analytics, most stored processes do not run.

Supporting Footers in Printed Reports

Administrators can define standard footer text that is added at the bottom of all printed reports. The footers do not appear when reports are viewed online, only when reports are printed.

The footer definition is stored in a SAS Visual Analytics Transport Service property and can include basic formatting options such as font selection and size and attributes such as bold, italic, or underline. Footers cannot contain images.

To define a footer, follow these steps:

- 1 Log on to SAS Management Console as an administrator (for example, sasadm@saspw).
- 2 Expand the **Application Management** folder, and select **SAS Application Infrastructure** ► **Visual Analytics 7.4** ► **Visual Analytics Services 7.4** ► **Visual Analytics Transport Service 7.4**.
- 3 Right-click **Visual Analytics Transport Service 7.4**, and select **Properties**.
- 4 In the Visual Analytics Transport Service 7.4 Properties window, select the **Advanced** tab.
- 5 Select the Printing.Footer.Content.Formatted property, and unlock the property value by selecting the lock icon on the right.
- 6 Add the desired footer content as the property value.

The following rules apply to the footer definition:

- The footer definition must consist of one or more HTML `` elements. The spans cannot be nested.
- You can use the `style=` attribute in the span element to specify formatting for the text in the span.

The following Cascading Style Sheets (CSS) attribute values are supported:

color:

Values must be specified as hex colors (for example, #000000 for black).

font-family:

Values must be available font or font family names (for example, **sans**).

font-style:

Supported values are **normal** or **italic**.

font-size:

Values must be specified in pt units (for example, 10pt).

font-weight:

Supported values are **normal** or **bold**.

text-decoration:

Supported values are **underline** or **none**.

- You can use the `
` element between spans to force a line break.

Here is a sample footer definition:

```
<span style="font-weight: bold;">Company Confidential.</span><br />
<span style="font-style: italic; font-size: 10pt;">This document contains
sensitive information.</span>
```

Supporting Report Distribution

Introduction

This topic helps administrators support the report distribution feature that the designer provides. For user instructions, see [Sharing Reports with Other Users](#) in the *SAS Visual Analytics: User's Guide*.

Here are the key points:

- In the designer, the **File** ► **Distribute Reports** menu item is available to only those users who have the Distribute Reports capability.
- Use only the designer to schedule and distribute SAS Visual Analytics reports. (If you use the Schedule Manager plug-in to SAS Management Console, and you set an option that is not available in the designer, the report job might be incompatible with the designer.)
- Log output for report distribution is in the SAS Visual Analytics Hyperlink Services log. The logging context is `com.sas.bicommon.distribution`. See [“Adjusting the Logging Configuration” on page 87](#).
- Report jobs and distributions (job flows) are stored in user-specific folders beneath a folder that is referenced by a configuration property. See [“va.baseSchedulingFolder” on page 140](#).

About the Scheduling Server

Report distribution uses distributed in-process scheduling. See [Setting Up Scheduling Using SAS Distributed In-Process Scheduling](#) in *Scheduling in SAS*.

A SAS Java Batch Server of the sub-type `Visual Analytics Scheduled Distribution` is required. The predefined SAS Java Batch Server is named `Visual Analytics Scheduled Distribution`.

About the Size of Email Attachments

How Size Limits Are Defined

Reports are distributed as PDF attachments to email messages. Size limits for emails are established as follows:

- Most email systems limit the size of attachments. If an email system rejects an attachment due to size, report distribution uses special handling for any future attempts to send emails that are the same size (or larger). Report distribution's knowledge of email system size limits is reset when the SAS Web Application Server restarts.
- You can use the configuration properties `va.distribution.email.aggregate.attachments.mb` and `va.distribution.email.attachment.mb` to set explicit size limits. See [“How to Set Configuration Properties” on page 139](#).

TIP Because report distribution adapts to actual rejections from an email system, it is rarely necessary to set explicit size limits.

How Oversize Email Attachments Are Handled

For a report distribution email with attachments that exceed a known size limit, the following results occur:

- If there are multiple attachments, report distribution attempts to distribute the attachments among multiple messages. The user who requests the report distribution is notified that multiple emails have been sent.
- Any individual attachment that exceeds the size limit is removed from the email. The email indicates that an attachment was omitted due to a size limit.

If a report distribution email is rejected by an email system due to a (previously unknown) size limit, report distribution's knowledge of email system's size limits is updated to reflect the rejection.

Note: If size limits are explicitly set (in configuration properties), the new limits that are in effect are lower than the configured limits.

Supporting the Search Features

See [Administering the Search Facility](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Supporting User-Defined Formats

Formats are instructions that SAS uses to write data values. Formats are used to control the written appearance of data values, or, in some cases, to group data values together for analysis.

User-defined formats are specialized formats that are stored in a custom format library. To incorporate user-defined formats, make them available as required by the clients that you are using.

- The explorer, the designer, and the viewers rely on formats that are available when data is loaded. To make user-defined formats available during autoloading, see ["User-Defined Formats for Autoload"](#) on page 32.
- Other clients (such as the data builder and SAS Enterprise Guide) rely on formats that are available when data is accessed. Any user-defined formats must be known to the appropriate SAS Application Server. The preferred method for making user-defined formats available to a SAS Application Server is to use a standard name and location for the custom format catalog as follows:

- 1 Name the format catalog `formats.sas7bcat`.
- 2 On each machine that hosts a workspace server, place the format catalog in the [SAS configuration directory](#) under `/SAS-application-server/SASEnvironment/SASFormats`.

Note: To use a nonstandard name or location or to make multiple custom format catalogs available, see [Create a User-Defined Formats Configuration File](#) in the *SAS Intelligence Platform: Data Administration Guide*.

Note: If you use load-balanced workspace servers, see [Managing Data and Catalogs for Servers on Multiple Machines](#) in the *SAS Intelligence Platform: Application Server Administration Guide*.

Managing Alerts and Notifications

Introduction

This topic provides information to help administrators manage the alerts that users can create in the designer. For user instructions, see [Working with Alerts for Report Objects](#) in the *SAS Visual Analytics: User's Guide*.

Requirements

Evaluation of data-driven alerts requires the following conditions:

- The associated SAS LASR Analytic Server is running.
- The target LASR table is loaded.
- The SAS Trusted User (for example, sastrust@saspw) has metadata-layer access to the target LASR table. The standard method for providing the necessary access is to grant the ReadMetadata and Read permissions to the SAS System Services group.

TIP Alert evaluations are performed by SAS Visual Analytics Hyperlink Services, so any errors are reported in your equivalent of `/Web/Logs/server/SASVisualAnalyticsHyperlinkVersion.log`.

For information about delivery, see [SAS Web Infrastructure Platform](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*. Here are some tips:

- To support email or SMS text message delivery, use either of the following approaches:
 - On the **Settings** tab of the **SAS Application Infrastructure Properties** dialog box in SAS Management console, add **E-mail** and **SMS** to the list of selected alert notification types. For details, see [Setting Global Properties for SAS Applications](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.
- Note:** With this approach, alert recipients should leave **Administrator default** (in the modern web viewer) or **Use the system default for alert notifications** (in the classic web viewer) selected in their preference settings.
- Instruct each alert recipient to change his or her **Alert Notifications** preference setting to **Send e-mail messages** or **Send SMS text messages**.

TIP To access the setting in the modern web viewer, click your name (on the right edge of the banner), select **Settings**, and then select **Report Viewer** ► **Alert Notifications** in the navigation pane.

TIP To access the setting in the classic web viewer, select **File** ► **Preferences** from the main menu. In the Preferences window, select **SAS Visual Analytics** ► **General**.

- Email delivery of alert notifications requires that the subscriber's metadata user definition includes a valid email address.
- Text message delivery of alert notifications requires that the subscriber's metadata user definition includes a valid mobile phone number that has **SMS** as its assigned type value. The phone number must be specified as an email address, in the appropriate carrier-specific format (for example, `1234567899@mobile.att.net`).
- To customize alert and notification behaviors, see [“Alerts Properties” on page 143](#).

How to Delete Other Users' Alerts

To delete other users' alerts:

- 1 From the main menu in the administrator, select **Tools** ► **Manage Alerts**.
- 2 On the **Alerts** tab, select one or more alerts, right-click, and select **Delete**.

Here are some details:

- Anyone who has the **Manage Environment** capability can access the **Alerts** tab.
- Alerts are stored in the SharedServices database in the middle tier.

Supporting the Monitoring Features

Introduction

Note: This topic is applicable to only distributed servers.

Features that depend on the monitoring server (SAS LASR Analytic Server Monitor) include the following:

- table details about the **Process Monitor** tab
- per-instance memory gauges on the **LASR Servers** tab
- certain information about the **HDFS** tab

Network Name Resolution

Successful functioning of the monitoring server requires network name resolution on the middle-tier machine.

Insufficient network name resolution can cause a log entry such as the following:

```
Exception caught in LASRClient (lasr=null, host=hostname, port=nnnn)
java.net.UnknownHostException: hostname
action=TableInfo
```

The log entry is written to the `bihpgrdc.monitor.console.log` file, which is in the [SAS configuration directory](#) at `/Applications/SASVisualAnalytics/HighPerformanceConfiguration/Logs`.

The host name that the log entry references requires name resolution on the middle-tier machine. You can alias the host name to the IP address for the grid host that is used for the SAS LASR Analytic Server.

Managing the Monitoring Server

Commands

Use the following commands to manage the process that monitors a distributed server:

UNIX:	LASRMonitor.sh	start stop restart status
Windows:	LASRMonitor.bat	start stop restart status pause resume

The script is in the [SAS configuration directory](#) under `/Applications/SASVisualAnalytics/HighPerformanceConfiguration`.

Note: On Windows, the monitoring process runs as a service (for example, **SAS [config-Lev1] LASR Analytic Server Monitor**). You can manage the process using the Windows Services interface.

Requirement: Passwordless SSH

The monitoring process account must have passwordless SSH access to all machines on the cluster.

To provide passwordless SSH access to a monitoring process that runs on Windows:

- Determine which Windows account the service runs as. Here are sample instructions:
 - Select **Start** ► **Control Panel** ► **Administrative Tools** ► **Services**.
 - Right-click on the service, and select **Properties**.
 - On the **Log On** tab, next to the **This account** radio button, note the user ID.
- Make sure that the following requirements are met:

- The account must have a copy of the SSH keys that belong to a corresponding UNIX account on the distributed server. See the section about Windows clients in [Passwordless SSH](#) in the *SAS LASR Analytic Server: Reference Guide*.
- In the `HighPerformanceConfiguration\wrapper\LASRMonitor.conf` file, the `set.USERNAME` field must specify the user name of the corresponding UNIX account.

Logging for the Monitoring Server

Note: Changes to logging should be made only under the direction of SAS Technical Support.

Generated logs are in the [SAS configuration directory](#) under `/Applications/SASVisualAnalytics/HighPerformanceConfiguration/Logs`.

The following log excerpt indicates that the account that attempted to start the monitoring server is not configured correctly for passwordless SSH:

```
NOTE: Unable to enumerate grid.
java.io.IOException
    at com.sas.grid.broker.monitor.ConnectionManager.startTKGridMon(
ConnectionManager.java:228)
    at com.sas.grid.broker.core.BrokerCore.main(BrokerCore.java:257)
ERROR: ERROR: Monitor thread failed to start.
```

Adjusting the Logging Configuration

Log Directories

In the standard configuration, logs are in the [SAS configuration directory](#) as follows:

- Log configuration files are in `/Web/Common/LogConfig`.
 - Note:** Log configuration files that have an `_apm` suffix are used by SAS Environment Manager. Do not make changes to those files.
- Generated logs are in `/Web/Logs/server`. For exact locations for generated logs, see the file [SAS-configuration-directory/Documents/Instructions.html](#) on the middle-tier machine (for web applications) and the compute tier machine (for servers).

UNIX example: `/opt/sas/config/Lev1/Documents/Instructions.html`

Windows example: `C:\sas\Config\Lev1\Documents\Instructions.html`

Note: SAS Visual Statistics logs are included in logs for SAS Visual Analytics Explorer.

How to Change Log Levels

CAUTION! Excessive logging can degrade performance. Do not use the TRACE and DEBUG logging levels unless you are directed to do so by SAS Technical Support. Remember to revert your logging level after diagnosis is complete, or as directed by SAS Technical Support.

The preferred method for changing a logging level is to make a temporary, dynamic change in SAS Web Administration Console. See [Administering Logging for SAS Web Applications](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

As an alternative to making a dynamic change, you can directly edit the appropriate log configuration file (after making a backup copy of the file).

The following example demonstrates how to change the log level to debug the explorer.

- 1 In the SASVisualAnalyticsExplorer-log4j.xml file, change the log level to DEBUG:

```
<category additivity="false" name="com.sas.biv">
<level value="DEBUG"/>
<appender-ref ref="SAS_CONSOLE"/>
<appender-ref ref="SAS_INFO_FILE"/>
</category>
```

- 2 Restart the SAS Web Application Server.

How to Log Submitted Code

To include SAS logs in the data builder and administrator logs:

- 1 In the SASVisualDataBuilder-log4j.xml file and the SASVisualAnalyticsAdministrator-log4j.xml file, uncomment the `SAS Job submission` section.
- 2 Restart the SAS Web Application Server.

TIP Related functionality is provided by a user preference. See [“Record actions as SAS statements”](#) on page 92.

How to Log LASR Actions

To log commands that are issued to a SAS LASR Analytic Server from the designer or the web viewer:

- 1 In the `logging contexts` section of the SASVisualAnalyticsDesigner-log4j.xml file and the SASVisualAnalyticsViewer-log4j.xml file, uncomment the tags for the `com.sas.lasr.command` category.
- 2 Restart the SAS Web Application Server.

Logging of Access Denials

Metadata-layer access denials are logged as follows:

- For requests from the administrator and the data builder, access denials are logged in the LASR authorization log (`SASLASRAuthorizationVersion.log`).
- For requests from other SAS Visual Analytics applications, access denials are logged in the application's log file (for example, `SASVisualAnalyticsExplorerVersion.log`).

Log entries provide details about the cause of the access denial, indicating which user lacks which permissions for which metadata object.

Logging for SAS Mobile BI

The logging configuration for SAS Mobile BI is defined in the `SASVisualAnalyticsTransport-log4j.xml` file.

You can use the `viewerservices.validate.schema.*` properties to increase logging for the rendering of reports in SAS Mobile BI. See [“Transport Service Properties” on page 145](#).

Logging for Platform Servers

See [Administering Logging for SAS Servers](#) in the *SAS Intelligence Platform: System Administration Guide*.

Making More Fonts Available on UNIX

When a report or exploration is printed to PDF, font substitution occurs for fonts that are unavailable to the transport service that generates the PDF. To make more fonts available to a service that runs on UNIX, use Fontconfig (version 2.8 or later). See www.freedesktop.org/wiki/Software/fontconfig/.

Note: The font file must be installed on the middle-tier machine that hosts the transport service. If you have a clustered middle tier, the font file must be installed on every middle-tier machine.

Note: ODS destinations such as stored processes register fonts using Fontreg, not Fontconfig.

Using Multiple SAS Application Servers

Which Server is Used?

Preliminary Requirements

Only a SAS Application Server that meets both of the following requirements can be used:

- The server is registered with the job execution service.
- The server is visible to the requesting user. (The user must have ReadMetadata access to the server.)

Automatic Selection

In most cases, a server is automatically selected using the following logic:

- 1 Use an associated server. For example, for a load request, follow the associations from the target LASR library to its SAS Application Server assignments.

Note: For a request to start or stop a SAS LASR Analytic Server, follow the associations from the target SAS LASR Analytic Server to its LASR libraries to their SAS Application Server assignments.

If there is no associated server that is registered with the job execution service and visible to the requesting user, proceed to step 2.

- 2 Use the suite-level default server. See [“va.defaultWorkspaceServer” on page 141](#).

If the suite-level default server is not registered with the job execution service or not visible to the requesting user, proceed to step 3.

- 3 Use any server that is registered with the job execution service and visible to the requesting user.

Specific Designation

For the administrator and data builder, users can opt out of automatic selection and instead force the use of a particular server. See [“User Preference: SAS Application Server” on page 91](#).

Note: For example, if you schedule data queries in the data builder, you might need to force the use of a SAS Application Server that includes a SAS DATA Step Batch Server.

Add a New Server

To add a new SAS Application Server, see [Managing SAS Application Servers](#) in the *SAS Intelligence Platform: Application Server Administration Guide*.

The following details are specific to SAS Visual Analytics:

- You must register the server with the job execution service. See [Job Execution Service](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.
- To support geo maps in the classic (Flex) applications, include a SAS Stored Process Server within the SAS Application Server. (SAS Visual Analytics uses the stored process server internally to read boundary and lookup data for geo maps.)
- To support any of the following activities, include a SAS Pooled Workspace Server within the SAS Application Server:
 - Use geo maps. The pooled workspace server loads the [geographic lookup data sets](#).
 - Extract and feed data for [administrative reports](#).
 - Access the **Import Data** panel in the designer, explorer, or data builder. (The pooled workspace server pre-screens available SAS/ACCESS licenses and then populates the **Import Data** panel.)
- To support scheduled queries in the data builder, include a SAS DATA Step Batch Server within the SAS Application Server.
- To support scheduled report distribution in the designer, include a Java batch server within the SAS Application Server.
- To support customized formats for data, see [“Supporting User-Defined Formats” on page 83](#).
- To support interactions with a SAS LASR Analytic Server, make sure the configured monitoring path exists on the machine that hosts the new SAS Application Server. Either ensure that the standard location exists on that machine, or add an extended attribute to the SAS LASR Analytic Server to specify an alternate location. See [“va.monitoringPath” on page 142](#).

TIP To reduce the need for back-end accounts, consider configuring the workspace server within the new SAS Application Server to use SAS token authentication. See [“Shared Accounts for Self-Service Imports” on page 60](#).

Setting User Preferences

About User Preferences

This topic documents user preferences that are specific to the administrator.

To review or set preferences, select **File** ► **Preferences** from the main menu in the administrator.

User Preference: SAS Application Server

In the Preferences window, under **SAS Visual Analytics Administrator** ► **Application Server**, the value in the **Application server** drop-down list specifies how a SAS Application Server is selected for requests in the administrator.

(auto-select)

causes an appropriate server to be automatically selected for each request. See “Which Server is Used?” on page 90.

server-name

forces use of a specified SAS Application Server (for example, **SASApp**). Only servers that are registered with the job execution service are listed.

Other User Preferences in the Administrator

In the Preferences window, under **SAS Visual Analytics Administrator** ► **Manage Environment**, the following settings are available:

Resource monitor sample rate (ms)

Specifies, in milliseconds, the sampling rate that the resource monitor uses for polling the machines in the cluster. This setting is not applicable to a non-distributed server.

Process monitor sample rate (ms)

Specifies, in milliseconds, the sampling rate that the performance monitor uses for polling application instances. This setting is not applicable to a non-distributed server.

Show the processes that measure performance

Controls whether processes that measure performance are included in the process-monitoring graphs. To include performance measurement processes in the graphs, select the check box. If several instances of performance measurement processes are running, they can negatively impact performance. This setting is not applicable to a non-distributed server.

Record actions as SAS statements

Saves the SAS code that the administrator generates when you perform certain tasks. You can save all recorded code in a single file or you can save the recorded code for each task in its own file.

If you enable recording, the following actions are recorded:

- Start or stop a SAS LASR Analytic Server.
- Load, reload, or unload a table.
- Add a table to co-located HDFS (**Add to HDFS**) or NFS-mounted MapR (**Add to Data Server**).
- Delete a table from co-located HDFS.

Note: You can modify and schedule recorded statements. However, metadata server connection information is not recorded. For information about the metadata server connection options, see [SAS Language Interfaces to Metadata](#).

5

SAS LASR Analytic Server

About SAS LASR Analytic Server	94
Introduction	94
Distributed or Non-distributed?	94
Non-distributed Server Dependencies	94
LASR-Related Metadata	95
In-Memory LASR Names	96
Add a SAS LASR Analytic Server	97
Introduction	97
Instructions	97
Optional Features	98
Add a LASR Library	98
Introduction	98
Instructions	99
Optional Features	100
Limit Space for Tables	100
Introduction	100
Over Capacity	100
How to Set a Tables Limit	101
Extended Attributes: SAS LASR Analytic Server	101
Introduction	101
Instructions	101
Reference	102
Advanced Options: SAS LASR Analytic Server	102
Introduction	102
Instructions	103
Version Information	103
Memory Limits	103
Logging Options	104
Additional Options	104
Distributed Server: Monitoring	105
Memory Gauges	105
Resource Monitor	106
Process Monitor	107
Memory Usage: A Closer Look	108
Distributed Server: Parallel Load	108
Introduction	108

Method: Co-located Storage	109
Method: NFS-Mounted Storage	109
Method: SAS Embedded Process	109
Example Depictions	110
Distributed Server: Co-located HDFS	111
Introduction	111
About the HDFS Tab	112
How to Introduce an Additional Directory	116
How to Delete an HDFS Table	116
Distributed Server: Legacy Co-located Providers	117
Distributed Server: High-Volume Access to Smaller Tables	117
Introduction	117
Instructions	118
Extended Attribute	118
Example	119

About SAS LASR Analytic Server

Introduction

SAS LASR Analytic Server is an analytic platform that provides secure, multi-user concurrent access to in-memory data. With high-performance, multi-threaded, analytic code that processes client requests at extraordinarily high speeds, the server enables business analysts to easily explore data and discover relationships. The server handles both big data and smaller sets of data. For comprehensive information, see the [SAS LASR Analytic Server: Reference Guide](#).

Distributed or Non-distributed?

A SAS LASR Analytic Server can be distributed or non-distributed.

- A distributed SAS LASR Analytic Server runs on multiple blades in a chassis.
- A non-distributed SAS LASR Analytic Server runs on a single machine. All of the in-memory analytic features that are available for a distributed server are also available for a non-distributed server. A non-distributed server does not support memory gauges, the **Resource Monitor** tab, or the **Process Monitor** tab.

Non-distributed Server Dependencies

When a non-distributed server is started interactively by the administrator by clicking or by using autostart, two additional SAS components are used:

- The job execution service in the SAS middle tier.
- A SAS Object Spawner in the server tier.

These two components are used to start a workspace server in the server tier. The workspace server submits the code that starts the non-distributed server and it remains running.

The execution sequence creates a dependency between a SAS LASR Analytic Server, the middle tier, and the SAS Object Spawner. If any of the following are stopped or restarted, a non-distributed server can stop or become unresponsive:

- the SAS Web Application Server (SASServern_m)
- the related instance of the job execution service on the SAS Web Application Server
- the SAS Object Spawner

TIP A server that is configured to perform autoload does not have a dependency on the job execution service or on the SAS Object Spawner.

LASR-Related Metadata

Metadata objects that are related to the SAS LASR Analytic Server include the following:

LASR Analytic Server

a metadata definition for a LASR Analytic Server process

LASR Analytic Server connection

a metadata representation of one instance of a LASR Analytic Server

LASR library

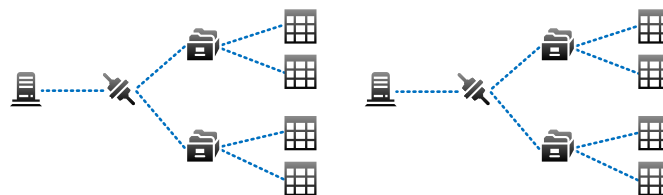
a metadata representation of a data library that is associated with a LASR Analytic Server connection

LASR table

a metadata representation of a table that has been loaded to memory in a LASR Analytic Server

The following figure depicts the relationships among these metadata objects.

Figure 5.1 Server, Connection, Libraries, and Tables



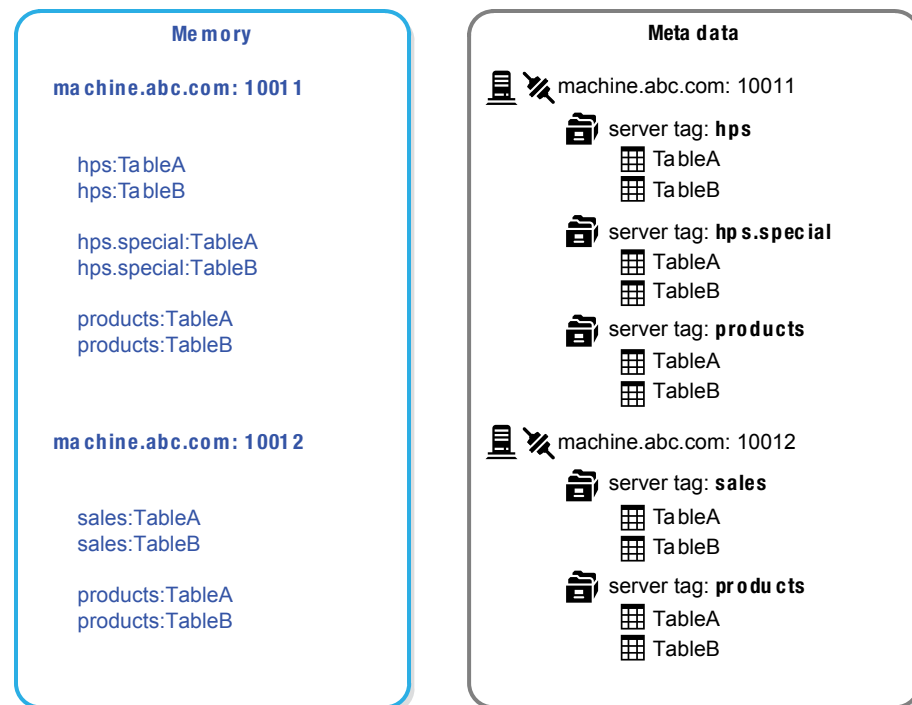
The preceding figure illustrates these points:

- Each server has one (multi-user) connection.
- Each connection can have multiple libraries.
- Each library can have multiple tables.
- Each deployment can have multiple servers.

In-Memory LASR Names

The following figure depicts an example of the mapping of metadata objects to corresponding in-memory data.

Figure 5.2 Uniqueness Requirements



The preceding figure illustrates these uniqueness requirements:

- Each host-port combination must be unique.
- Within a server instance (a host-port combination), each server tag must be unique. See “[Server Tags](#)” on page 44.
- Within a server tag, each table name must be unique.
- The fully qualified in-memory name for a table (in the format *host-name:port/ server-tag.table-name*) must be unique.

Note: The metadata does not always reflect the current state of the SAS LASR Analytic Server. For example, when you unload a table from memory, the corresponding table object is not deleted from metadata.

Add a SAS LASR Analytic Server

Introduction

This topic explains how to define an additional instance of a SAS LASR Analytic Server in metadata. Adding a server instance facilitates separation for management and ease-of-use purposes.

- Each server instance has a unique, multi-user connection to the hardware on which the SAS LASR Analytic Server process runs.
- Each server instance has a distinct set of associated LASR libraries and provides access to only those tables that are in an associated LASR library.

Instructions

TIP The following instructions use SAS Management Console. To instead use [SAS Environment Manager](#), see [Register a SAS LASR Analytic Server Using SAS Environment Manager](#) in the *SAS Intelligence Platform: Data Administration Guide*.

- 1 On the **Plug-ins** tab in [SAS Management Console](#), expand **Environment Management**. Right-click **Server Manager**, and select **New Server**.
- 2 In the New Server wizard, select **SAS Servers** ► **SAS LASR Analytic Server**. Click **Next**.
- 3 Enter a name for the server. Click **Next**.
- 4 Set properties as follows:

Single machine server	For a distributed server, select No . For a non-distributed server, select Yes .
High-Performance Analytics environment install location	Specify the host path where files that define the cluster are located (for example, <code>/opt/TKGrid</code>). This field is applicable to a distributed server only.
Number of machines to use	Accept the default value (ALL). This field is applicable to a distributed server only.

Click the **Advanced Options** button.

- 5 In the Advanced Options window, select the **Additional Options** tab. Review the settings and make any necessary adjustments. See [“Advanced Options: SAS LASR Analytic Server”](#) on page 102.

TIP Make sure that **Signature files location on server** field references a directory that has appropriate host protection.

Click **OK** to close the Advanced Options window. In the wizard, click **Next**.

- 6 Enter connection properties as follows:

Port number	Enter a unique port number. See “In-Memory LASR Names” on page 96 .
High-Performance Analytics environment host	Enter the fully qualified machine name of the host (for example, <code>va.abc.com</code>).*
Use LASR authorization service	Leave this check box selected. See “SAS LASR Authorization Service” on page 41 .

* The prepopulated **LASR authorization service location** field identifies the host. For example, if the displayed location is `http://va.abc.com:80/SASLASRAuthorization`, then the host value is usually `va.abc.com`.

Click **Next**.

- 7 If you want to adjust the default grants of the Administer permission that the wizard applies to the server, move identities from one list to the other. Click **Next**.

Note: Only users who have the Administer permission for the server can stop the server or set its tables limit. The server inherits settings from the repository ACT (default ACT), so it might not always be essential to add explicit grants.

- 8 Click **Finish**.

Optional Features

- To support text analytics, see [“Supporting Text Analytics” on page 77](#).
- To limit the amount of space that the server can use to host tables, see [“Limit Space for Tables” on page 100](#).

Add a LASR Library

Introduction

Here are the main reasons for creating a new LASR library:

- You want additional separation for management or ease-of-use purposes.
- You added a new directory within co-located HDFS or NFS-mounted MapR. You already created the new SASHDAT library. Now, you need to create the corresponding LASR library.

Instructions

TIP The following instructions use SAS Management Console. To instead use SAS Environment Manager, see [Register a SAS LASR Analytic Server Library Using SAS Environment Manager](#) in the *SAS Intelligence Platform: Data Administration Guide*.

- 1 On the **Plug-ins** tab in [SAS Management Console](#), expand **Data Library Manager**. Right-click **Libraries**, and select **New Library**.
- 2 In the New Library wizard, select **High-Performance Analytics** ► **SAS LASR Analytic Server Library**. Click **Next**.
- 3 Enter a name (for example, `sales LASR`). If necessary, adjust the location. Click **Next**.
- 4 (Optional) Assign the library to one or more SAS Application Servers. Click **Next**.

Note: Assigning a LASR library to a SAS Application Server facilitates interactions from clients such as SAS Enterprise Guide. Assignments can also affect which SAS Application Server is used for interactions with this LASR library and its associated SAS LASR Analytic Server. See [“Which Server is Used?”](#) on page 90.

- 5 Set library properties as follows. Click **Next**.

Libref	Enter an identifier of your choice (for example, <code>SALESLIB</code>).
Engine	This field is not editable. The value (<code>SASIOLA</code>) is the engine name for LASR libraries.
Server tag	See “Server Tags” on page 44.
Data provider library	If you want participating tables to be reloaded each time the associated server starts, specify a Base SAS library to function as the backing store for this LASR library. See “Reload-on-Start” on page 18.

- 6 Assign the library to a SAS LASR Analytic Server by entering settings as follows. Click **Next**.

Database Server	Select a server from the drop-down list.
Connection	Use the pre-selected value (which prepends the selected server name with the string <code>Connection:</code>).
Default Login	This field is not editable. The value is <code>None</code> .

- 7 If you want to adjust the default grants of the Administer permission that the wizard applies to the library, move identities from one list to the other. Click **Next**.

Note: Only users who have the Administer permission for the library can load new tables to memory. The library inherits settings from its parent folder, so it might not be essential to add any explicit grants.

- 8 Click **Finish**.

Optional Features

- If you want the associated server to start on demand for data load and import requests against the new library, enable autostart for the library. See [“Autostart a SAS LASR Analytic Server” on page 7](#).
- If you want locally imported files to reload each time the associated server restarts, enable reload-on-start for the library. See [“Reload-on-Start” on page 18](#).
- If you want to automatically synchronize the library’s in-memory data against source tables in a host directory, set up a corresponding implementation of autoload. See [“Autoload” on page 22](#).
- If the library is associated with a distributed server and provides high-volume access to small tables, consider setting an extended attribute that optimizes performance. See [“VA.TableFullCopies” on page 118](#).

Limit Space for Tables

Introduction

To limit the amount of space that a SAS LASR Analytic Server can use to host tables, set a tables limit. For example, to limit the total amount of data that can be loaded or imported to a general-purpose or public server, you might set a tables limit of 500 megabytes for that server. The limit helps ensure sufficient memory availability for other processes that run on the same machine (or cluster).

Over Capacity

Definition

If the sum of the sizes of loaded tables on a particular server equals or exceeds the server’s tables limit, the server is over capacity. A server that is over capacity accepts requests for activities such as data retrieval and analysis, but rejects requests to load, import, append, or reload tables. In other words, the tables limit does not constrain total memory usage; it constrains only the amount of memory that a particular server can use to host tables.

Note: Memory that is mapped for tables counts toward the limit. Memory that is used for temporary tables does not count toward the limit.

Note: Because the tables limit is enforced only after a server is over capacity, it does not establish an absolute constraint. A load that causes a server to be over capacity succeeds, regardless of the size of the table. Only subsequent loads are prevented.

Feedback

In most cases, a request that is rejected because a server is over capacity generates a message that indicates that the server is over capacity. However, for the following requests, the message indicates that the metadata server denied access to the operation:

- autoloading
- create a table as output from a data query

TIP For information about how the administrator displays the actual use of memory for tables, tables limits, and over capacity status, see [“Get Server Information” on page 6](#).

How to Set a Tables Limit

To set a server's tables limit:

- 1 From the administrator's main menu, select **LASR** ► **Manage Servers**.
- 2 In the **Tables Limit** column, click a cell, and enter a number.

Note: Any cells that you are authorized to modify have an edit indicator (a small triangle). You must have both the Administer and WriteMetadata permissions for a server to add, update, or remove its tables limit.

- 3 To save the change, press **Enter** (or click anywhere else in the interface).

Note: These instructions affect the [VA.MaxTotalMemoryForTables on page 102](#) extended attribute.

Extended Attributes: SAS LASR Analytic Server

Introduction

This topic documents extended attributes for a SAS LASR Analytic Server. Extended attributes for autostart, autoloading, and reload-on-start are at the library level, not the server level. See [Table 2.1 on page 10](#).



Instructions

To view or set the extended attributes `VA.MonitoringPath` and `VA.TextAnalyticsBinaryLocation`, use either of the following methods.

SAS Management Console

- 1 In [SAS Management Console](#), on the **Plug-ins** tab, expand **Environment Management** ► **Server Manager**.
- 2 Right-click a SAS LASR Analytic Server, and select **Properties**.
- 3 In the Properties dialog box, select the **Extended Attributes** tab.

SAS Environment Manager

- 1 In the [SAS Environment Manager Administration](#) window, click or tap , and select **Servers** from the side menu.
- 2 Open a SAS LASR Analytic Server.
- 3 From the server's drop-down list of property pages, select the **Extended Attributes** property page.
- 4 Modify extended attributes as needed, and then click .

Reference

VA.MonitoringPath

specifies a custom directory for monitoring artifacts for this SAS LASR Analytic Server. By default, this attribute is not set. If this attribute is set, it overrides the corresponding suite-level property (“[va.monitoringPath](#)”) for this server. If this attribute is set, the specified directory must exist on a workspace server host. The directory must have two required subdirectories: **PIDs** and **Logs**.

VA.MaxTotalMemoryForTables (tables limit)

specifies (in bytes) how much of a server's memory can be used by tables. By default, this attribute is not set, so no limit is in effect. The best practice for setting this attribute is to use the administrator. See “[Limit Space for Tables](#)” on page 100.

VA.TextAnalyticsBinaryLocation

location of SAS linguistic files. In the standard configuration, the files are in the SAS installation directory. Here are some examples:

UNIX Specifics: `/SASFoundation/Version/misc/tktg` (for a non-distributed server), `/opt/TKTGDat` (for a distributed server)

Windows Specifics: `\SASFoundation\Version\tktg\sasmisc`

Advanced Options: SAS LASR Analytic Server

Introduction

This topic documents advanced options in the metadata definition for a SAS LASR Analytic Server. For information about the basic options, see “[Add a SAS LASR Analytic Server](#)” on page 97.

Note: For a non-distributed server, the only applicable advanced options are **Server lifetime**, **Signature files location on server**, and **Enable logging**.



Instructions

To view or set advanced options, use either of the following methods.

SAS Management Console

- 1 In [SAS Management Console](#), on the **Plug-ins** tab, expand **Environment Management** ► **Server Manager**.
- 2 Right-click a SAS LASR Analytic Server, and select **Properties**.
- 3 In the Properties dialog box, select the **Options** tab.
- 4 On the **Options** tab, click **Advanced Options**.

SAS Environment Manager

- 1 In the [SAS Environment Manager Administration](#) window, click or tap , and select **Servers** from the side menu.
- 2 Open a SAS LASR Analytic Server.
- 3 From the server's drop-down list of property pages, select the **Options** property page.
- 4 Make changes as needed, and then click .

Version Information

The options are for descriptive purposes only.

Memory Limits

The following options affect the circumstances in which a distributed SAS LASR Analytic Server rejects certain tasks:

Data loading (%)

specifies a percentage of used physical memory above which tables cannot be loaded to memory. If total memory use (by all processes on the cluster) exceeds the specified limit, operations that add tables or append rows fail. For example, if the value for this field is 80, and more than 80% of memory is already in use, tables cannot be loaded.

Note: Tables that are loaded from co-located HDFS or NFS-mounted MapR do not count toward this limit.

TIP To limit available memory for tables on a particular server, see [“Limit Space for Tables”](#) on page 100.

External processes (%)

specifies a percentage of used physical memory above which external processes (such as SAS High-Performance Analytics procedures) cannot

retrieve data. If total memory use (by all processes on the cluster) exceeds the specified limit, affected processes cannot retrieve data. For example, if the value for this field is 80, and more than 80% of memory is already in use, affected processes cannot retrieve data.

Note: If you do not specify values, the default value (75) is used for both options.

Logging Options

The logging options are as follows:

Enable logging

Enables logging in a SAS LASR Analytic Server.

Path to log files

The path where the log file for a distributed server is placed.

Note: For a non-distributed server, log files are always written to the signature files directory.

Maximum file size (MB)

Specifies the size of the log file (in megabytes) before the log file is rolled over. The default is 100 MB.

Maximum rollover files

Specifies how many rotating log files can be used before older log files are overwritten. The default is 10.

Keep log files when the server terminates

Select **Yes** to leave log files in the file system when the server terminates. The default value is **No**, and the files are removed.

Additional logging parameters

This field is reserved for use in the future.

Additional Options

The **Additional Options** tab includes the following items:

Vendor

SAS

Associated Machine

Select the server's host. If the host is not listed, click **New** to add it.

Force overwrite of server description file

This field is not used.

Signature files location on server

The host directory where signature files are written. The location is set during installation.

CAUTION! It is important to protect the specified directory. See “Signature Files” on page 43.

Note: Do not use a signature files path that contains double-byte character set (DBCS) characters.

Note: For a distributed server, the specified path must be located on the SAS High-Performance Analytics environment root node.

Server lifetime

By default, the server runs forever. This is appropriate in most environments.

To set a maximum run time, specify a value in seconds. For example, if you specify 3600, the server stops after it runs for 60 minutes.

For a distributed server, you can also set a time-out period so that the server stops after an interval of inactivity. The time-out is specified in parentheses after the first value. For example, if you specify the value as 3600 (600), then after the server runs for 60 minutes, it starts tracking any inactivity. If no action requests are received within 10 minutes, the server stops.

Display detailed diagnostics

By default, detailed diagnostics are not displayed.

The connection object's Advanced Options window includes this option:

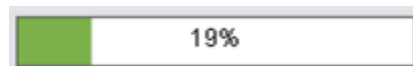
Server description file

Leave this field blank. SAS Visual Analytics does not use this setting.

Distributed Server: Monitoring

Memory Gauges

For a distributed server, an overall memory gauge is displayed in the main menu bar in the administrator. The overall gauge indicates how much of the server host's total physical memory is currently in use. The overall gauge is refreshed every minute.



Here are some details:

- If a specified percentage of memory is used, a distributed server rejects requests to load tables or append rows. See [“Memory Limits” on page 103](#).
- The gauge provides information for only the distributed server that is referenced in the service.properties file in the [SAS configuration directory](#) (at /Applications/SASVisualAnalytics/HighPerformanceConfiguration).

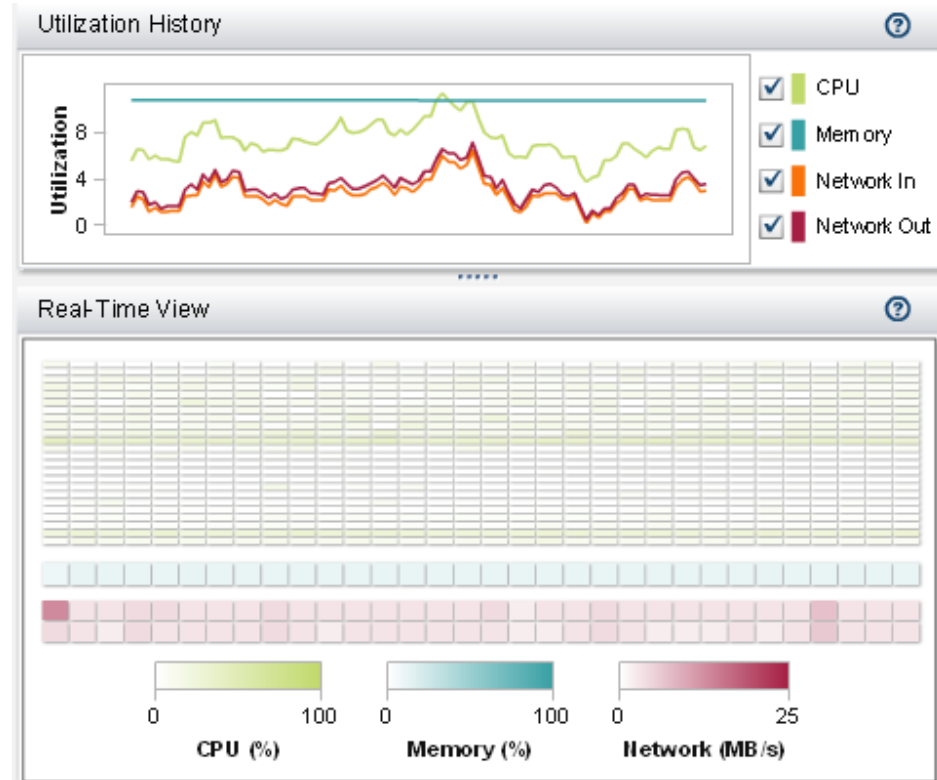
Note: This constraint also applies to additional memory usage information for distributed servers (on the **Monitor** tabs and the **LASR Servers** tab).

- For a distributed server, individual memory gauges are displayed in the **Virtual Memory** column on the **LASR Servers** tab. Each individual gauge indicates how much of the cluster's total virtual memory is being used by a particular server instance (process). The individual gauges are refreshed every minute after the **LASR Servers** tab is opened.

Note: The calculation behind the overall memory gauge differs from the calculation behind the individual memory gauges. For details, see [“Memory Usage: A Closer Look” on page 108](#).

Resource Monitor

For a distributed server, you can monitor resource utilization by selecting **LASR** ► **Monitor Resources** from the main menu in the administrator.



In the upper half of the **Resource Monitor** tab, the **Utilization History** graph plots utilization against time as follows:

- CPU and memory utilization are plotted as percentages of capacity. Under high demand, the upper bound can reach 100%. Under low demand, the upper bound can drop below 10%.
- Network input and output utilization is displayed as two line plots. The plots show the transfer rate in megabytes per second.

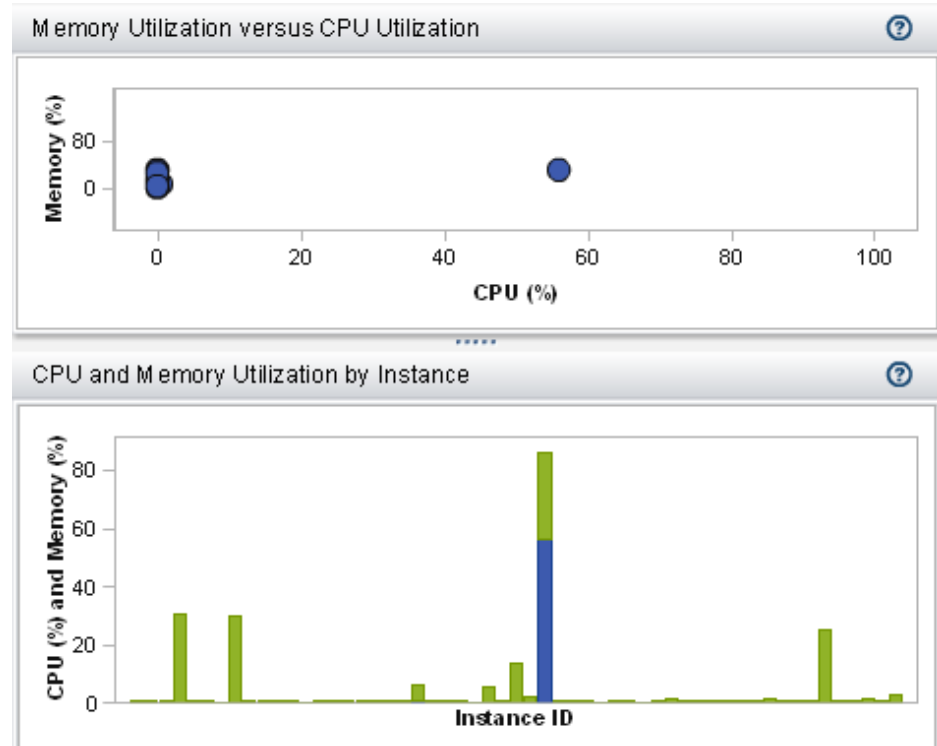
To view resource utilization for a particular sampling period, place your pointer over a line. To select that sampling period in the entire display, click a line. You can then place your pointer over the sampling period on each line to view details.

In the lower half of the **Resource Monitor** tab, the **Real-Time View** heat map contains a column for each machine in the cluster. To view the host name and details, place your pointer over a cell.

- The top and middle sections show CPU utilization and memory utilization, respectively. The color reflects the workload (more saturated color indicates heavier use).
- The bottom section shows network output transfer rate (in the first row) and network input transfer rate (in the last row). The color reflects the transfer rate between 0 and 25 megabytes per second. If the transfer rate exceeds 25 MBps, the color is purple.

Process Monitor

For a distributed server, you can monitor per-process utilization by selecting **LASR ► Monitor Processes** from the main menu in the administrator.



At the top of the **Process Monitor** tab, the **Selection** and **Filter** controls enable you to specify which server instances to display.

The **Process Monitor** tab displays two graphs:

- **Memory Utilization versus CPU Utilization**

Each server instance is represented by a bubble. The size of the bubble represents the number of processes for that instance. The location of the bubble indicates the resource utilization for that instance. Ideally, an instance has one process for each machine in the cluster.

- **CPU and Memory Utilization by Instance**

Each bar shows CPU and memory utilization for a server instance. If a bar is vertically divided, CPU utilization is shown in the lower section, and memory utilization is shown in the upper section.

To view details, place your pointer over a bubble or a bar. For machine-level information, click on a bubble or bar. A window lists host names, ranks the hosts (by the column that you most recently sorted), and shows used memory and CPU. For table-level information, click **Show tables** within the window. The window lists loaded tables, the user ID of the person who loaded them, and the number of rows and columns in each table.

Note: Per-process utilization is calculated from the traditional systems perspective. See the following section.

Memory Usage: A Closer Look

For a distributed server, the administrator provides two distinct expressions of memory usage:

- In the main menu bar, the overall memory gauge provides a practical estimate of effective capacity.

The value for the overall memory gauge is calculated as follows:

$$\frac{\text{(total allocations - SASHDAT allocations)}}{\text{(total memory for the cluster)}}$$

total allocations all memory allocations for all processes on the cluster.

SASHDAT allocations memory that is allocated for tables that have been loaded from HDFS. These tables are stored in a highly efficient manner that keeps them instantly available on demand but consumes memory only when the data is accessed. For this reason, the overall gauge does not count SASHDAT allocations as used memory.

- In the **Virtual Memory** column on the **LASR Servers** tab, each individual gauge indicates how much memory is currently being used by a particular server instance (process).

The individual gauges express memory usage from the traditional systems perspective, disregarding the memory-sparing efficiencies of SASHDAT allocations. The individual gauges can help you analyze capacity for the hypothetical scenario in which all tables are accessed simultaneously.

The value for each individual gauge is calculated as follows:

$$\frac{\text{(total allocations for this instance)}}{\text{(total memory for the cluster)}}$$

Note: The per-instance information about the **Process Monitor** tab also uses the preceding calculation to express memory usage.

TIP On the **LASR Servers** tab, the sum of the individual gauges matches the overall memory gauge only if there are no SASHDAT allocations.

Distributed Server: Parallel Load

Introduction

SAS Visual Analytics loads data in parallel whenever possible. This topic outlines the parallel load methods that SAS Visual Analytics can support.

Note: Not all methods and providers are configured and available in all deployments. See the [SAS Visual Analytics: Installation and Configuration Guide \(Distributed SAS LASR\)](#).

Method: Co-located Storage

Topology:	The storage and analytics nodes must be on the same machines.
Provider:	Co-located HDFS or a legacy co-located provider .
Pattern:	Symmetric. There must be a one-to-one mapping between storage and analytics nodes.
SASHDAT:	In co-located HDFS, data is staged in SASHDAT format.
Server tag:	The HDFS source path in dot-delimited format or the legacy libref.
Usage:	See “ Administrator Load ” on page 15 or use the data builder.

Method: NFS-Mounted Storage

Topology:	The storage cluster can be separate from the analytics cluster.*
Provider:	MapR. See MapR Distribution for Apache Hadoop in the <i>SAS LASR Analytic Server: Reference Guide</i> .
Pattern:	Asymmetric. One-to-one mapping between storage and analytics nodes is not required.
SASHDAT:	Data is staged in SASHDAT format.
Server tag:	The NFS source path in dot-delimited format.
Usage:	See “ Administrator Load ” on page 15 or use the data builder.

* Regardless of topology, the SAS LASR Analytic Server accesses data as if it is co-located.

Method: SAS Embedded Process

Topology:	The storage cluster can be separate from the analytics cluster.
Provider:	Various.**
Pattern:	Asymmetric. One-to-one mapping between storage and analytics nodes is not required.
SASHDAT:	Data is not staged in SASHDAT format.
Server tag:	Any valid libref.
Usage:	See “ Administrator Load ” on page 15, use the data builder, or use an import action.*

* Load is parallel if embedded processing is available, LASR table name matches source table name, and server tag is valid as a SAS libref.

** See the [SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide](#).

Example Depictions

The following figures depict staging to and loading from co-located HDFS:

Figure 5.3 Stage to Co-located Storage

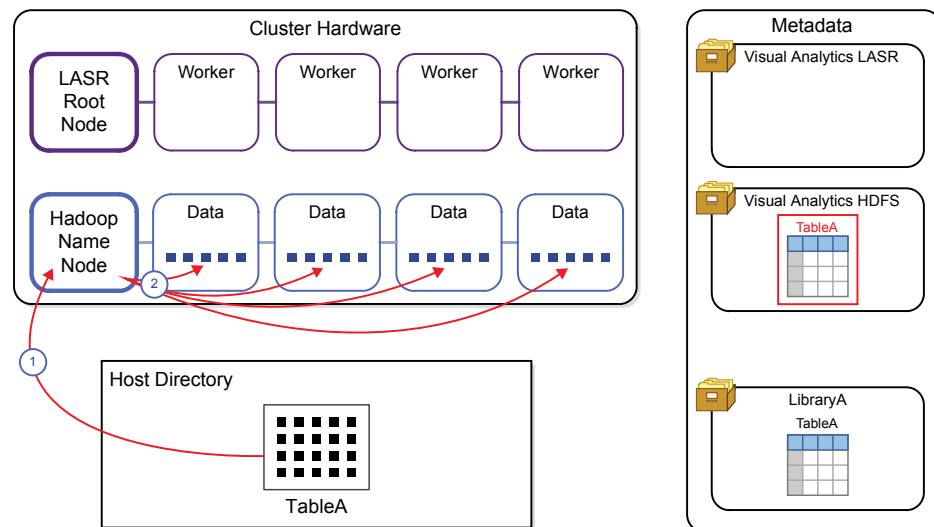
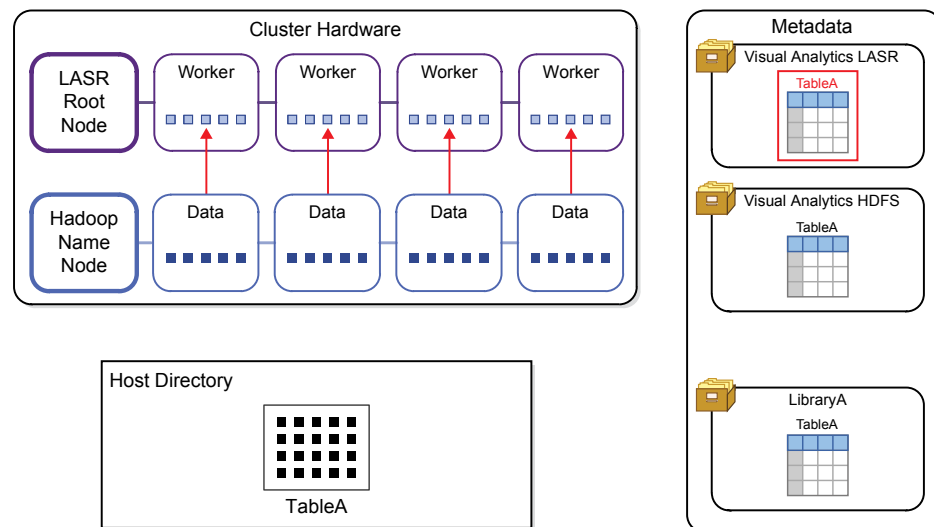


Figure 5.4 Load from Co-located Storage

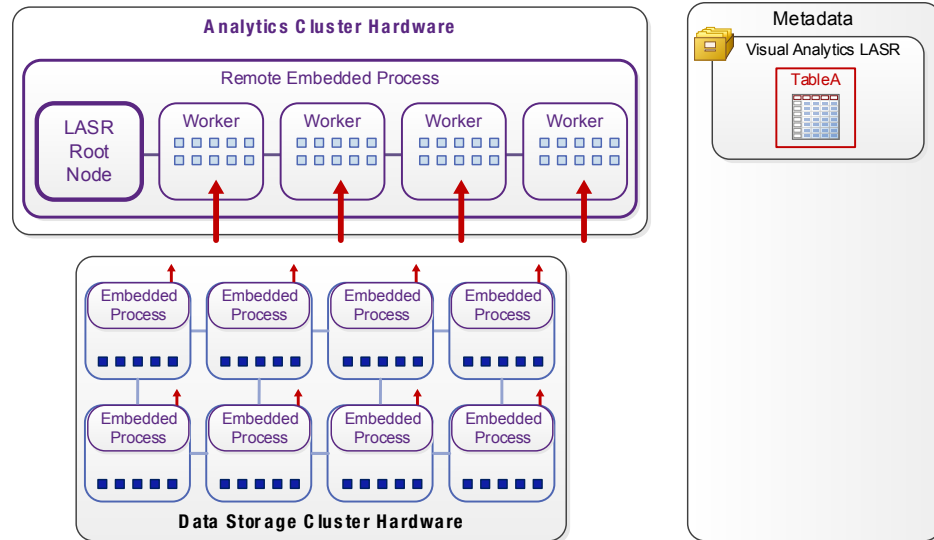


For NFS-mounted MapR, the stage and load processes are similar to the preceding example, except as follows:

- The storage and analytics clusters can be separate.
- The metadata objects would have different names.

The following figure depicts an import action that uses SAS Embedded Process:

Figure 5.5 Import Using SAS Embedded Process



Distributed Server: Co-located HDFS

Introduction

Co-located HDFS is a deployment of Hadoop that meets the following criteria:

- The deployment runs on the same hardware as a distributed SAS LASR Analytic Server.
- The deployment incorporates services that SAS High-Performance Deployment of Hadoop provides.

SAS High-Performance Deployment of Hadoop adds services to Apache Hadoop (and other supported Hadoop distributions) to provide the following integrated functionality:

- SAS uses a special file format (with the filename suffix SASHDAT) to store tables in HDFS. Like any file that is stored in HDFS, a SASHDAT file is distributed as a series of blocks. Copies of blocks are stored to provide data redundancy.
- SAS enhances the block distribution algorithm to make sure that blocks are distributed evenly. Because SAS LASR Analytic Server reads blocks of data directly, the even block distribution contributes to an even workload on the machines in the cluster.

This integration enables a distributed SAS LASR Analytic Server to use HDFS to read SASHDAT tables in parallel very efficiently.

TIP Basic HDFS commands are documented in the [SAS LASR Analytic Server: Reference Guide](#).

About the HDFS Tab

Introduction

To open the **HDFS** tab, select **Tools** ► **Explore HDFS** from the main menu in the administrator.

Note: The **HDFS** tab is available in deployments that use co-located HDFS. Only users that have the Browse HDFS capability can access the **HDFS** tab.

The **HDFS** tab provides a host-layer view of HDFS folders and tables. The view is not mediated by metadata or by your permissions. Instead, a privileged Hadoop account retrieves the information that this tab displays.

You can use the **HDFS** tab to perform the following tasks:

- Browse HDFS folders and tables.
- View row count, columns, column information, and block information for tables that have been added to HDFS. Information about block distribution, block redundancy, and measures of block utilization is provided.
- Delete HDFS tables that are stored in SASHDAT format. (Files that are not SASHDAT files are listed, but they cannot be deleted.)

System Properties

To view HDFS system properties, click . The following table describes the fields:

Table 5.1 HDFS System Properties

Property	Description
Command for setting permissions	This setting is not used.
Set permissions as root?	This setting is not used.
Command for getting file information	This setting is not used.
Data directories	Specifies the directory that is used to store blocks.
Name Node	Specifies the host name of the machine that is used as the Hadoop NameNode.
Live Data Nodes	Specifies the number of Hadoop DataNodes that are reachable.
Dead Data Nodes	Specifies the number of Hadoop DataNodes that are not available.

Basic File Information

To view basic file information, select a file. The following information is provided:

Table 5.2 Basic File Information

Field	Description
Name	Specifies the name of the file.
Size	Specifies the file size. This value includes the disk space required to store the data in blocks and metadata about the file.
Date Modified	Specifies the date on which the file was created or replaced.
Path	Specifies the HDFS directory.
Description	Specifies the description that is stored with the data. The description is displayed beside the table name in the explorer interface.
Copies	Specifies the number of redundant copies of the data.
Block Size	Specifies the number of bytes that are used to store each block of data.
Number of Variables	Specifies the number of columns in the HDFS table.
Owner	Specifies the user account that added the data to HDFS.
Group	Specifies the primary UNIX group for the user account.
Permissions	Specifies the Read, Write, and Execute access permissions for owner, group, and other.
SASHDAT file?	Specifies whether the file is in the SASHDAT format. Yes indicates that the file is in the SASHDAT format.
Compression	Specifies whether the file is compressed. Yes indicates that the file is compressed.
Encryption	Specifies whether the file is encrypted. Yes indicates that the file is encrypted.

Note: The **HDFS** tab might display multiple files for a table as the table is being added to HDFS. After the table is added, the multiple files disappear.

Table Information


To view column information, select a table, and click . The following information is provided:

Table 5.3 Column Information

Field	Description
Column Name	Specifies the column name from the source table.
Label	Specifies the label for the data set column when the table was added to HDFS.
Type	Numeric or Character. Numeric variables are encoded as 1.
Offset	Specifies the starting position for the variable in the SASHDAT file.
Length	Specifies the storage used by the variable.
Format	Specifies the format associated with the variable.
Format Length	Specifies the format length of the format that existed on the variable when it was added to HDFS. This value is zero if the variable did not have a format when it was added to HDFS.
Precision	Specifies the precision portion of the format for number formats.
Length (Formatted)	Specifies the length of the variable when formatting is applied.


To view the row count, select a table, and click . The following information is provided:

Table 5.4 Row Count Information

Field	Description
Rows	Specifies the number of rows in the data.
Blocks	Specifies the number of HDFS blocks that are used to store the data.
Allocated	Specifies the number of bytes allocated to store the data. The value is a multiple of the block size and the number of blocks. This value is smaller than the file size because it does not include the space needed for the SASHDAT file header.
Used	Specifies the number of bytes within the allocated blocks that are used for storing rows of data.

Field	Description
Utilization	Specifies the percentage of allocated space that is used for storing rows of data.

Block Detail Information


To view block details, select a file, and click . The following information is provided:

Table 5.5 Block Detail Information

Field	Description
Host Name	Specifies the machine in the cluster that stores the block of data.
Block Name	Specifies the filename for the block.
Path	Specifies the directory to the block.
Record Length	Specifies the sum of the column lengths for the variables in the data.
Records	Specifies the number of rows stored in the block. Because redundant blocks are listed in the table, the sum of the records listed does not equal the number of rows in the data.
Owner	Specifies the user account that added the data to HDFS.
Group	Specifies the primary UNIX group for the user account that stored the data.
Permissions	Specifies the Read, Write, and Execute access permissions for owner, group, and other.

You can sort by the column headings to identify anomalies. It is normal for several blocks to be stored on the same machine. However, it is not normal for the values of **Record Length**, **Owner**, **Group**, or **Permissions** to be different from row to row.

The files added to HDFS are stored as blocks. One block is the preferred block, and additional copies of the blocks are used to provide data redundancy. The Block Distribution dialog box offers two ways to view this information. The **Block Detail View** tab enables you to select a block number and view the host names that store the original or redundant blocks. The **Node Detail View** enables you to select a host name and view the block numbers that are stored on the machine.

Block Distribution Information


To view the block distribution, select a table, and click . The following information is provided:

Table 5.6 *Block Distribution Information*

Field	Description
File Size	Specifies the size of the file in bytes.
Block Size	Specifies the block size for the file.
Blocks	Specifies the number of blocks used to store the original copy of the data.
Machines Used	Specifies the number of machines in the cluster that have original or redundant blocks for the file.
Copies	Specifies the number of redundant block copies of the data.

On the **Block Detail View** tab, you can select a block number. This enables you to view how many copies of the block exist and the host names for the machines that store the blocks. The value in the **Total Copies** column equals the number of redundant copies of the block plus the original block. You can select the column heading to sort the rows. In an ideal distribution, the number of total copies is equal for all blocks.

On the **Host Detail View** tab, you can expand a host name node, and then view the block numbers that are stored on that machine. When you select the block number, the host name and any additional machines with copies of the block are identified in the host name list.

How to Introduce an Additional Directory

Each co-located HDFS directory that you use must be represented in metadata by a library that uses the SASHDAT engine. To create the required metadata, see the chapter [Connecting to Common Data Sources](#) in the *SAS Intelligence Platform: Data Administration Guide*.


Here are some key points:

- Each directory in co-located HDFS must also have a corresponding LASR library. See [“Add a LASR Library” on page 98](#).
- The server tag for the corresponding LASR library must be the source path in dot-delimited format. See [“Server Tags” on page 44](#).
- To facilitate parallel loads, use single-level paths that have only eight or fewer characters. For example, use `/sales` instead of `/dept/sales` or `/sales_department`. The path is the basis for the server tag, and the server tag is used as a libref in parallel loads.

How to Delete an HDFS Table

- 1 Right-click on the table in the **Folders** pane, and select **Delete**.

- 2 In the confirmation window, if you want to delete the physical table with the metadata object that represents it, select the **Remove from HDFS storage** check box.

TIP You can also delete an HDFS table from the **HDFS** tab. Select the table, and click  in the tab's toolbar.

Distributed Server: Legacy Co-located Providers

If your deployment uses a legacy provider (co-located Greenplum or co-located Teradata), consider the following points:

- To stage a table, right-click the table (in the administrator's **Folders** pane), and select **Add to a Data Server**.
- For data that is loaded from a legacy provider, SAS Visual Analytics uses SAS variable names as data item names.
- Each legacy provider library must have a corresponding LASR library.
- The middle-tier machine is configured as a client of the legacy provider, and must have network name resolution for host names.

Distributed Server: High-Volume Access to Smaller Tables

Introduction

This topic addresses the specialized situation where all of the following circumstances exist:

- You must support high-volume Read access to smaller tables.

Note: Smaller is a relative concept. Tables that are less than 2 GB are good candidates. Tables that are between 2 GB and 20 GB might be good candidates, depending on factors such as server capacity, amount of free memory, and number of nodes.
- High inter-machine network communication (relative to table size) is negatively impacting data retrieval performance.
- You are willing to separate your frequently accessed smaller tables into a separate LASR library.

For smaller tables, in-memory access is faster when data is consolidated rather than distributed. For example, if a smaller table serves as the data source for a report, retrieval of that report is faster if the table is available in its entirety on a single machine rather than distributed across multiple machines. For reports that are widely and frequently accessed, the difference in retrieval performance can be worth the effort of managing a separate library for smaller tables.

To optimize retrieval performance for smaller tables, a distributed SAS LASR Analytic Server can keep multiple consolidated (full non-distributed) copies of each table. Each copy is written to and retrieved from a single machine. Each machine launches its own non-distributed server processes as needed to fulfill load and access requests. Load balancing and reuse of the non-distributed server processes further enhance performance.

For more information, see [High Volume Access to Smaller Tables](#) in the *SAS LASR Analytic Server: Reference Guide*.

Instructions

To optimize high-volume access to smaller tables in a distributed SAS LASR Analytic Server:

- 1 Identify or [create](#) a LASR library that is exclusively for smaller tables.
 - Give the library a name that helps users recognize that they should never load or import large tables into it.
 - Associate the library with a distributed SAS LASR Analytic Server.
- 2 Set the LASR library's extended attribute "[VA.TableFullCopies](#)" to a positive integer. (You can use either SAS Management Console or SAS Environment Manager to set extended attributes for a LASR library.)
- 3 To verify results, load a table to the LASR library. On the **LASR Tables** tab, verify the table's status. See "[Get Table Information](#)" on page 12.

Extended Attribute

The following library-level extended attribute enables smaller-table optimization and controls the number of in-memory instances per table.

VA.TableFullCopies

specifies how many complete, in-memory, single-node instances are created for each loaded table. By default, no value is specified, so no full copy instances are created. If you have a LASR library that contains only smaller tables and is associated with a distributed server, set the value to a positive integer.

CAUTION! If you specify a high value or if someone loads a large table to the library, server memory could be rapidly consumed. Consider initially specifying a value less than 4 (and increasing the value incrementally if needed), setting a [tables limit](#) for the associated server, and limiting the Administer permission on the library.

Here are some additional details:

- Autoload supports this attribute.
- You cannot append data to tables that are loaded with additional full copies.
- LASR star schemas, imports from Twitter, and imports from Facebook ignore this attribute.
- Non-distributed SAS LASR Analytic Servers ignore this attribute.

- In general, it is not beneficial to use compression for tables that are loaded with additional full copies.

Example

Scenario

- LibraryA is a LASR library that contains only smaller tables.
- LibraryA is associated with ServerA, a distributed SAS LASR Analytic Server.
- LibraryA's **Extended Attributes** tab specifies a value of 3 for VA.TableFullCopies.

Results

- When TableA is loaded to LibraryA, three of the nodes on ServerA get a full copy of TableA.
- When access to TableA is requested, one of those three nodes provides its full copy of TableA.
- TableA is also loaded in the usual distributed manner. However, no access requests are fulfilled from the distributed instance of TableA.
- You cannot append to TableA.

6

Reports for Administrators

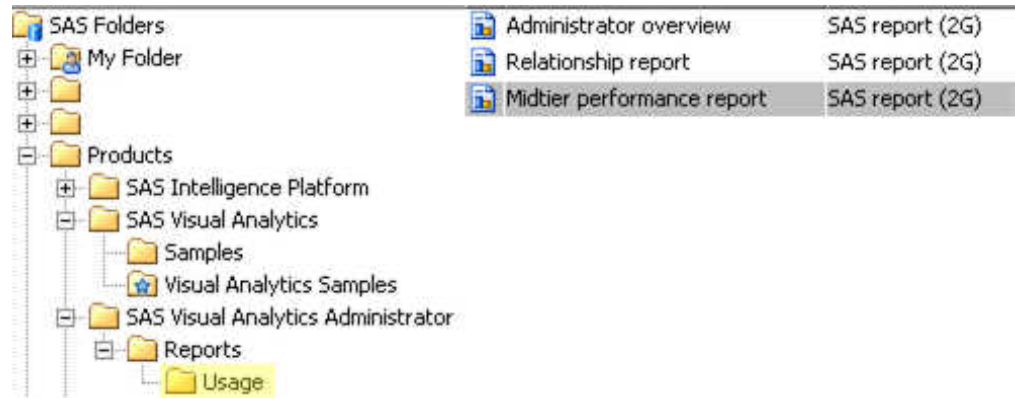
About the Predefined Reports	121
Location	121
Access	122
Data Currency	123
Interacting with Reports	123
About Custom Reports	123
How to Provide Administrative Data	123
Orientation	123
1. Start Autoload	124
2. Feed Data to Autoload	125
About auditRefresh	127
Introduction	127
Start auditRefresh	128
Stop auditRefresh	128
Reduce the Task Frequency	128
Change the Purge Boundary	128
Logs and Process IDs for auditRefresh	129

About the Predefined Reports

Location

Predefined reports are in the folder `/Products/SAS Visual Analytics Administrator/Reports/Usage`. In the administrator, you can access the predefined reports by selecting **View** ► **Usage Reports** from the main menu.

Administrator overview	Provides usage information by application, user, and object. Uses data from the middle-tier audit service.
Relationship report	Provides lineage and dependency information for selected object types. Uses data from the middle-tier relationship service.
Midtier performance report	Provides system metrics for participating SAS server machines. Uses agent collected metric (ACM) data.



Note: Each predefined report is populated with data only after its data feed is fully enabled and operational. Not all predefined reports are supported in all deployments. See “[How to Provide Administrative Data](#)” on page 123.

Access

Initial Configuration

In the standard configuration, only administrators and unrestricted users can access predefined reports and their underlying data. Details are as follows:

- The Visual Analytics Data Administrators and Visual Data Builder Administrators groups have ReadMetadata and Read access to the data.
- The Visual Analytics Data Administrators and Visual Data Builder Administrators groups have ReadMetadata access to the reports.
- The Visual Analytics Data Administrators and SAS Administrators groups have WriteMemberMetadata access to the `/Products/SAS Visual Analytics Administrator/Reports/Usage` folder.
- An explicit denial of the WriteMetadata permission on each predefined report prevents modification or deletion by anyone other than an unrestricted user.

How to Modify Access

Here is one way to make predefined reports more widely available:

- 1 Grant the ReadMetadata permission on the folder that contains the reports (`/Products/SAS Visual Analytics Administrator/Reports/Usage`).
- 2 Grant the ReadMetadata and Read permissions on the folder that contains the data (`/Shared Data/SAS Visual Analytics/Autoload/EVDMLA`).
- 3 Grant the ReadMetadata permission on the library **Environment Manager Data Mart LASR**.

TIP If your site does not use the reports and underlying data, you can hide them from all restricted users by adding denials of the ReadMetadata permission.

Data Currency

Data currency is affected by the following factors:

- frequency of data collection or generation by the source system
- frequency of data extraction from the source system to a drop zone
- frequency of data loading from the drop zone to a SAS LASR Analytic Server

For example, in the standard configuration, the SAS Visual Analytics key actions audit data is usually less than 30 minutes old. Audit records are continuously generated, audit data is extracted every 15 minutes, and the extracted data is loaded every 15 minutes.

Note: For data extraction and loading, a new run begins only after the preceding run is completed. This can cause occasional exceptions to the timing that is described here.

Interacting with Reports

For information about viewing and interacting with reports, see the [SAS Visual Analytics: User's Guide](#).

About Custom Reports

Do not modify the predefined reports. Instead, use the designer to create custom reports.

Before you create a custom report that uses data structures that SAS provides, or data that SAS generates, review the following considerations:

- Data structures, data generation, and available tables are subject to change in future releases (of SAS Visual Analytics or of any underlying component). Any custom reports that you create might require revision before they can be used in a future release.
- Before you perform any software upgrades, migrations, or new installations, you should save a backup copy of any custom reports.


How to Provide Administrative Data

Orientation

What Data is Needed?

Before data is available in an administrative report, the data must be collected or generated by a source system, extracted to a drop zone, and autoloading to memory. To determine which data source is used by a particular report section, open the report in the designer, select the report section that you are interested in, and then select the **Data** tab in the left pane.

What Data is Already Available?

- 1 In the administrator, select **LASR** ► **Manage Tables** from the main menu.
- 2 Right-click on any column heading, and make sure the **LASR Name** column is selected.
- 3 In the tab toolbar, select **LASR name** from the drop-down list, and enter *EVDM* in the search field.
- 4 Check the check box in the first column heading so that all tables are selected. Then, click  in the tab toolbar. To interpret the display, see “[Get Table Information](#)” on page 12.

Deployment Topology Considerations

Support for administrative data is limited as follows:

- If the middle tier and server tier are deployed on separate machines with dissimilar operating systems (one tier on Linux and the other tier on Windows), the audit and relationship data feeds are not supported.
- If the middle tier and server tier are deployed on separate machines with similar operating systems (both on Linux or both on Windows), the autoloading data directory (drop zone) for administrative data must be available to both machines.

Windows Specifics: Share the directory. Change the EVDMLA library’s VA.AutoLoad.Location extended attribute to be a UNC path.

UNIX Specifics: Put the directory in an NFS-mounted location. Change the EVDMLA library’s VA.AutoLoad.Location extended attribute to point to the NFS-mounted location.

Life Cycle for Administrative Data

For audit data, see “[Key Actions Auditing](#)” on page 48.

You can periodically retire other tables by deleting them from the autoloading data directory (`/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad/EVDMLA`) or moving them to a backup location.

1. Start Autoload

To start autoloading for administrative data:

- 1 On the machine that hosts the [administrative reporting library](#), identify or create a scheduler account.
 - Give the account the host-layer privileges that are required to start the associated SAS LASR Analytic Server and load data. See “[Host Account Privileges](#)” on page 5.
 - On UNIX, enable the account to run cron jobs.
 - In the [SAS configuration directory](#), give the account Read and Write access to the following autoloading directories and their contents:

Data: `/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad/EVDMLA`

Scripts: `/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/EVDMLA`

- 2 In the metadata, create a corresponding individual metadata identity. See [“Adding Users” on page 3](#).

Note: This requirement reflects the standard configuration. See [“Metadata Server Connection” on page 31](#).

Give the scheduler account’s metadata identity the required [metadata-layer permissions on page 36](#) on the target server, library, and folder. A simple approach is to add the scheduler account’s metadata identity to the Visual Analytics Data Administrators group. An alternative is to grant access to the metadata identity as follows:

Server:	LASR Analytic Server	RM, WM, A
Library:	Environment Manager Data Mart LASR	RM, R, WM, A
Folder:	<code>/Shared Data/SAS Visual Analytics/Autoload/EVDMLA</code>	RM, R, WMM, W

- 3 Log on to the host as the scheduler account, navigate to the scripts directory for EVDMLA, and invoke `schedule.sh` (on UNIX) (or `schedule.bat` (on Windows)).

TIP The `schedule` script that is in the `VisualAnalyticsAdministrator` directory starts a different library’s implementation of `autoload`. You must invoke the script that is in `/VisualAnalyticsAdministrator/EVDMLA`. See [“Autoload” on page 22](#).

- 4 Verify that the scheduled task is running.

Windows Specifics: Access the **Task Scheduler** (for example, select **Start** ► **Control Panel** ► **Administrative Tools** ► **Task Scheduler**). Locate the task *Environment Manager LASR - Auto Load Scheduler* in the **Task Scheduler Library**.

UNIX Specifics: Run the command: `crontab -l`

2. Feed Data to Autoload

To efficiently enable multiple data feeds, defer server restart and verification steps until all [configuration property](#) changes have been made.

Audit Data

To feed audit data to the drop zone:

- 1 Initiate generation and periodic extraction of audit data. See [“How to Safely Enable Auditing” on page 49](#).

Note: In addition to starting data collection, this step extracts certain audit records from the audit service’s database, and feeds that data to the Append directory in the administrative reporting drop zone.

Note: Until autoloading is started, the data feed occurs only one time.

- 2 Perform some tasks that generate audit records. For example, load or import a table, or create and save a report or exploration.
- 3 After 30 minutes, verify that the LASR table EVDM.AUDIT_VISUALANALYTICS is loaded.

Here are some details:

- Audit data is provided by the audit service. See [“Audit Content and Coverage” on page 50](#).
- Extraction occurs only if the data builder is licensed, installed, and running.
- Extraction uses the pooled workspace server in the suite-level default SAS Application Server. See [“va.defaultWorkspaceServer” on page 141](#).
- The extraction process must have Read and Write access to the autoloading data directory and that directory’s Append subdirectory.

Note: Autoloading’s append action is used, so the extracted table is written to the Append subdirectory.

- If autoloading is not running, the data feed to the drop zone occurs only one time.
- A separate scheduled task manages the size of the full AUDIT_VISUALANALYTICS table in the drop zone. See [“About auditRefresh” on page 127](#).

Relationship Data

To feed relationship data to the drop zone:

- 1 Enable collection and extraction of relationship data.
 - a To set up scheduled data collection (or verify that data collection is already scheduled), see [Configuring Automatic Relationship Loading and Cleaning](#) in the *SAS Intelligence Platform: System Administration Guide*.
 - b To enable periodic extraction, set [“va.extractRelationshipData”](#) to `true`. The extraction interval is 3 hours.
 - c Restart the SAS Web Application Server.
- 2 After a sufficient interval has elapsed, verify that the LASR table EVDM.RELATIONSHIPS_VISUALANALYTICS is loaded.

Here are some details:

- Relationship data is provided by the relationship service. See [Using the Batch Relationship Reporting Tools](#) in the *SAS Intelligence Platform: System Administration Guide*.
- Relationship data is extracted for the following object types:
 - 2G reports (Report.BI objects)
 - Explorations (VisualExploration objects)
 - Data queries (VisualDataQuery objects)
 - Data sources (Table objects)

Note: Relationship data is also extracted for objects that are directly referenced by any of the preceding object types.

- The extraction process must have Read and Write access to the autoloading data directory.

Note: Autoload's refresh action is used, so the extracted table is written to the autoloading data directory.

Agent-Collected Metrics

Note: Agent-collected metrics do not provide information about distributed SAS LASR Analytic Servers.

To feed agent-collected metrics (ACM) to the drop zone:

- 1 Enable data collection. See [Initializing and Enabling the Service Management Architecture](#) in the *SAS Environment Manager: User's Guide*.
- 2 Enable data transfer to the EVDMLA drop zone. See [Feeding Data From the Data Mart into SAS Visual Analytics](#) in the *SAS Environment Manager: User's Guide*.

TIP A supporting format catalog must be available. When you enable this data feed, the required format catalog is added to the `/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad/EVDMLA/Formats` directory. For some applications, the format catalog must also be added to the path of the appropriate SAS Application Server. See ["Supporting User-Defined Formats"](#) on page 83.

- 3 After a sufficient interval has elapsed, verify that the expected ACM tables are loaded. For information about the source data, see [ACM Tables](#) in the *SAS Environment Manager: User's Guide*. (The list of tables is subject to change in future releases of SAS Environment Manager.)

About auditRefresh

Introduction

To manage the size of the full AUDIT_VISUALANALYTICS table in the autoloading data directory (drop zone) for administrative data, auditRefresh periodically performs these tasks:

- delete records of a specified age
- refresh the corresponding LASR table

Note: auditRefresh functionality is provided by scripts, programs, and directories that are very similar to an implementation of autoloading. Unlike autoloading, auditRefresh is applicable to only one LASR library, the administrative data library (EVDMLA). There is no reason to create additional implementations of the auditRefresh task.

Note: auditRefresh addresses one of several measures for managing audit data. For the full picture, see ["Key Actions Auditing"](#) on page 48.

Start auditRefresh

- 1 Log on to the autoloader host as the scheduler account that you identified or created when you started the administrative data implementation of autoloader.
- 2 Navigate to the associated scripts directory.

```
/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/EVDMLA
```
- 3 Invoke `auditRefreshSchedule.sh` (on UNIX) or `auditRefreshSchedule.bat` (on Windows).
- 4 Verify that the scheduled task is running.

Windows Specifics: Access the **Task Scheduler** (for example, select **Start** ► **Control Panel** ► **System and Security** ► **Administrative Tools** ► **Task Scheduler**). Locate the task *Environment Manager LASR - Audit Table Refresh* in the **Task Scheduler Library**.

UNIX Specifics: Run the command: `crontab -l`

Stop auditRefresh

To stop `auditRefresh`, use the scheduler account to invoke `auditRefreshUnschedule.sh` (on UNIX) or `auditRefreshUnschedule.bat` (on Windows). Stopping `auditRefresh` does not stop the audit data feed, autoloader, or the associated SAS LASR Analytic Server.

Reduce the Task Frequency

Initially, `auditRefresh` runs daily. To purge old records less frequently, modify the associated schedule script. It is a good practice to make a backup copy of the script before you make changes.

Windows Specifics: In the `auditRefreshSchedule.bat` script, change the value for `TIME_INTERVAL_DAYS=`. The initial value is `1`.

UNIX Specifics: In the `auditRefreshSchedule.sh` script, change the string that is highlighted in the following line:

```
cat <(fgrep -i -v $AUDIT_RUNSAS_PATH <(crontab -l))
<(echo "0 0 * * * $AUDIT_RUNSAS_PATH") | crontab -
```

The initial syntax schedules the task to run nightly, at midnight. You might find it helpful to use a site such as www.cronchecker.net to verify your cron command syntax.

Change the Purge Boundary

Initially, records that are more than 30 days old are purged. To purge younger or older records, change the `auditinterval=` setting in the `AuditRefresh.sas` file.

Logs and Process IDs for auditRefresh

Logs and process ID (PID) files for auditRefresh are co-located with the EVDMLA autoloader logs and PID files. See [“Logs and Process IDs for Autoloader” on page 30](#).

Appendix 1

Reference

Software Components	131
Roles and Capabilities	133
About Capabilities	133
Predefined Roles	133
Capability Definitions	136
Standard Memberships	138
Configuration Properties	139
How to Set Configuration Properties	139
Suite-Level Properties	139
Alerts Properties	143
Explorer Properties	144
Transport Service Properties	145
Web Viewer Properties	147
High-Cardinality Constraints	148
Introduction	148
Client-Side Thresholds for High-Cardinality Data	148
Middle-Tier Thresholds for High-Cardinality Data	149
Configuration Properties for High-Cardinality Thresholds	150
Predefined LASR Libraries	153
Sample Objects	157
Configured Samples	157
Tips and Guidelines	158
Manual Configuration	158

Software Components

Here is an introduction to selected components:

mobile viewers

mobile apps that support native interactions with reports and dashboards on mobile devices. See the [SAS Mobile BI](#) page on the SAS support site.

web applications

provide role-based access to an integrated suite of functionality.

SAS LASR Authorization Service

enforces data access permissions.

SAS Visual Analytics Hyperlink Service supports functionality such as report distribution, linking, and alerts.

SAS Visual Analytics Transport Service supports communication from SAS Mobile BI, provides integration with SAS Office Analytics (SAS Enterprise Guide, SAS Add-In for Microsoft Office, and SAS Web Parts for Microsoft SharePoint), and supports printing of reports.

SAS LASR Analytic Server provides secure, multi-user, concurrent access to in-memory data. See [Chapter 5, “SAS LASR Analytic Server,” on page 93.](#)

SAS LASR Analytic Server Monitor supports monitoring of a distributed server and browsing of co-located HDFS content, if applicable. See [“Supporting the Monitoring Features” on page 85.](#)

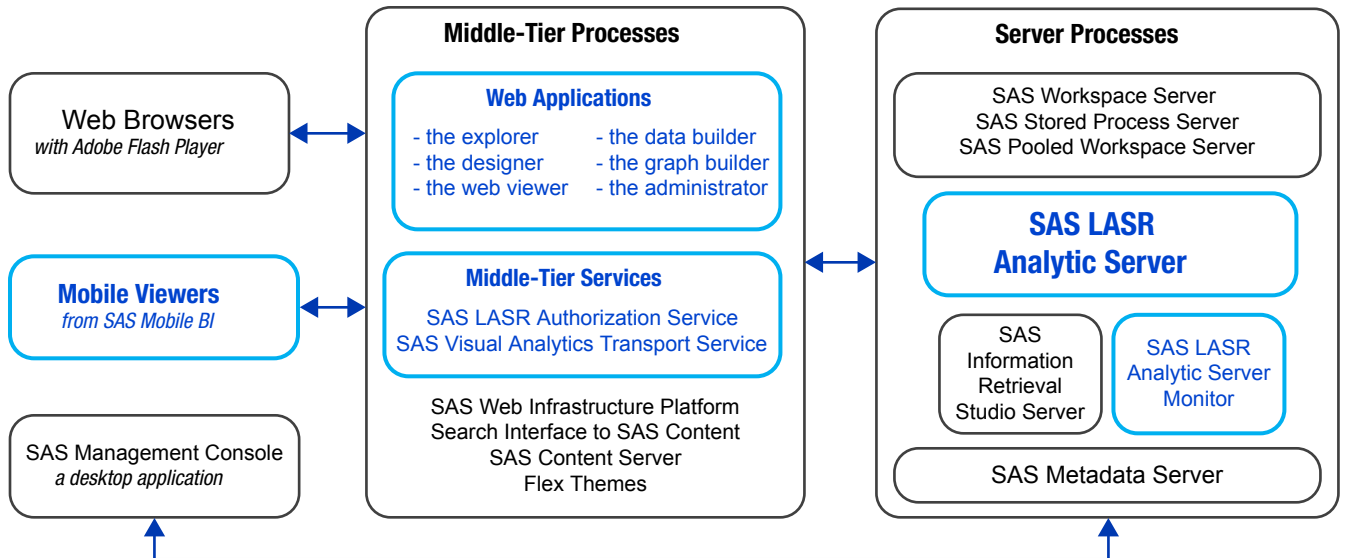
SAS Intelligence Platform

servers and services that support SAS solutions. Here are some examples of how SAS Visual Analytics uses platform servers:

- The metadata server provides metadata management.
- The SAS Content Server stores digital content in the middle tier. Reports are stored in both metadata and the content server. Explorations are stored exclusively in metadata.
- SAS Information Retrieval Studio and Search Interface to SAS Content index SAS content and support search features on the home page.
- The workspace server supports tasks such as registering tables, staging data, importing data, loading data, and starting or stopping the SAS LASR Analytic Server.
- For more examples, see [“Add a New Server” on page 90.](#)

Here is a conceptual view of selected components:

Figure A1.1 Clients, Middle-Tier, and Servers



Roles and Capabilities

About Capabilities

Here are the key points about capabilities:

- Unlike permissions, which affect access to data, content, and metadata, capabilities affect access to features and functionality.
- Capabilities are assigned to roles. Users get their capabilities through their memberships.
- You cannot deny a capability to a user. Instead, make sure that user is not a member of any role that provides the capability.
- If the standard distribution of capabilities is not optimal for your environment, consider creating custom roles. Here are some tips:
 - If you create a specialized administrative role, remember to provide the Manage Environment capability in addition to any specific functional capabilities.
 - If you create a global administrative role, make the **Visual Analytics: Administration** role a contributing role for the new custom role. In addition, add the Build Data capability to the custom role.

Predefined Roles

Here are the predefined roles for SAS Visual Analytics:

Visual Analytics: Basic

provides functionality for guest access (if applicable) and entry-level users. This role enables all registered users to view reports in the web viewer. This role does not provide commenting or personalization features. See [“Supporting Guest Access” on page 73](#).

Visual Analytics: Report Viewing

provides commenting and personalization features, in addition to basic functionality.

Visual Analytics: Analysis

provides the ability to create reports and explorations, in addition to report viewing functionality. If SAS Visual Statistics is licensed, provides the Build Analytical Model capability.

Visual Analytics: Data Building

provides the ability to prepare data, in addition to analysis functionality.

Visual Analytics: Administration

provides the ability to perform tasks in the administrator, in addition to most other capabilities.

Table A1.1 Capabilities by Role

Capability	Basic	Report Viewing	Analysis	Data Building	Administration
Visual Analytics					
View Report and Stored Process	✓	✓	✓	✓	✓
Create Report			✓	✓	✓
Explore Data			✓	✓	✓
Build Custom Graph			✓	✓	✓
Add and View Comments		✓	✓	✓	✓
Export Data			✓	✓	✓
Export or Print as PDF	✓	✓	✓	✓	✓
Email	✓	✓	✓	✓	✓
Personalization		✓	✓	✓	✓
Visual Analytics: Self-Service Import					
Import and Load Data			✓	✓	✓
Import Local Files			✓	✓	✓
Import SAS Data Sets from a Server			✓	✓	✓
Import from <i>data-source</i>			✓	✓	✓
Visual Analytics: Advanced					
Build Data				✓	

Capability	Basic	Report Viewing	Analysis	Data Building	Administration
Manage Environment					✓
Manage Mobile Devices					✓
Distribute Reports			✓		✓
Visual Analytics Transport Service					
Purge Mobile Report Data					
Require Passcode On Mobile Devices					
Limit Duration of Offline Access					
Visual Analytics Explorer					
Refresh Data			✓	✓	✓
Export as Image			✓	✓	✓
Export as Report			✓	✓	✓
Build Analytical Model			✓		
Visual Analytics Admin					
Manage LASR Analytic Server					✓
Monitor LASR Analytic Server					✓
Manage Authorization					✓
Browse HDFS					✓

For role membership information, see [“Standard Memberships” on page 138](#).

To manage roles, see [SAS Management Console: Guide to Users and Permissions](#) or [SAS Environment Manager Administration: User’s Guide](#).

See Also

[“Home Page: Roles and Capabilities” on page 70](#)

Capability Definitions

Here are descriptions of the SAS Visual Analytics capabilities:

Visual Analytics

View Report and Stored Process

Access the viewer. View reports and stored process output. (Access to SAS Mobile BI is also affected by device-level constraints. See [“Access to SAS Mobile BI” on page 56](#).)

Create Report

Access the designer. Create and modify reports.

Explore Data

Access the explorer. Create and modify explorations. (In some contexts, the explorer is a separately licensed add-on product.)

Build Custom Graph

Access the graph builder. Create and modify graph template objects for use in the designer.

Add and View Comments

Add comments, view comments, and edit your own comments.

Note: In order to delete comments and edit other users’ comments, you need the capabilities that are listed under **SAS Application Infrastructure ▶ Comments**. Consider adding those capabilities to the **Visual Analytics: Administration** role or making any users that need these capabilities members of the **Comments: Administrator** role.

Export Data

Export data to other applications.

Export or Print as PDF

Export or print reports and explorations as PDF files.

Email

Send a link to a report or exploration via email.

Personalization

Use individualized features such as setting preferences, accessing recently viewed objects, and managing favorites.

Visual Analytics: Self-Service Import

Import and Load Data

A prerequisite for access to self-service import functionality in the designer and the explorer. See [“Self-Service Import” on page 16](#).

Import Local Files

Import spreadsheets, delimited files, and SAS data sets from your computer.

Import SAS Data Sets from a Server

Import remote data sets.

Import from *data-source*

Import data from a third-party data source (for example, Import from Oracle).

Visual Analytics: Advanced**Build Data**

Access the data builder. Set advanced load options in the explorer and the designer.

Manage Environment

Access the administrator. Additional capabilities are required to perform particular tasks.

Manage Mobile Devices

Blacklist or whitelist mobile devices. (The Manage Environment capability is also required.)

Distribute Reports

Schedule and manage the distribution of reports.

Visual Analytics Transport Service**Purge Mobile Report Data**

Causes cached data in SAS Mobile BI to be purged when reports are closed. For users who do not have this capability, cached data is retained locally on the mobile device for use in offline mode. In previous releases, this capability was named Purge Mobile Report Data.

Note: For unrestricted users, mobile data is always purged when reports are closed.

Require Passcode On Mobile Devices

Requires users to enter an application passcode on their devices when they use SAS Mobile BI. For users who do not have this capability, an application passcode is not required.

Note: Unrestricted users are always subject to the application passcode requirement.

See [“viewerservices.passcode.attempts” on page 147](#) and [“viewerservices.passcode.timeout” on page 147](#).

Limit Duration of Offline Access

Causes a time limit for offline access to be enforced. A user who has this capability and has been offline for a certain period of time must sign in to SAS Mobile BI to access any mobile report data. The time limit is specified in the property [viewerservices.offline.limit.days](#).

Note: For unrestricted users, mobile data is always purged when reports are closed.

Visual Analytics Explorer**Refresh Data**

Refresh data for explorations.

Export as Image

Export images of explorations to a local machine.

Export as Report

Export explorations as reports to SAS folders.

Build Analytical Model

Create and modify analytical models in the explorer using SAS Visual Statistics (a separately licensed add-on).

Visual Analytics Admin

The Manage Environment capability provides access to the administrator, and is a prerequisite for all tasks that are performed in the administrator.

Manage LASR Analytic Server

Access the **LASR** tabs and the folders tree. For a distributed server, this capability makes a link to the SAS High-Performance Computing Management Console available from the **Tools** menu.

Monitor LASR Analytic Server

Access the **Monitor** tabs. This capability is applicable to deployments that use a distributed server.

Manage Authorization

Set metadata-layer permissions.

Browse HDFS

Access the **HDFS** tab. This capability is applicable to deployments that use co-located HDFS.

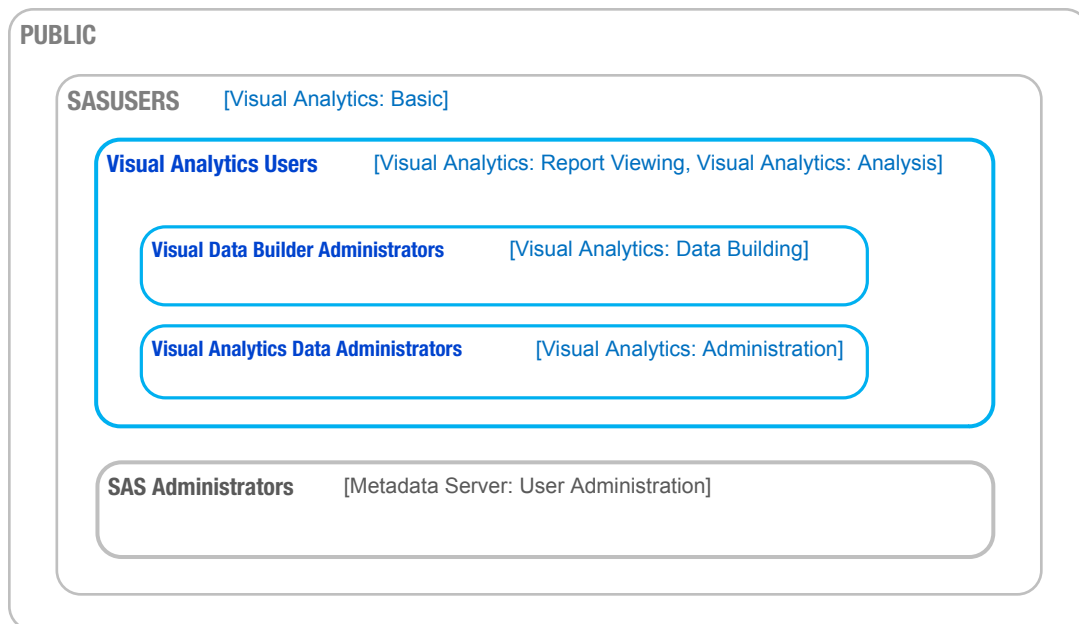
Note: For conciseness, version numbers are omitted in this topic.

Standard Memberships

The following figure depicts selected group and role relationships in the standard membership structure. Here are some details about the figure:

- Containers indicate nested group memberships. For example, the Visual Analytics Data Administrators group is a direct member of the Visual Analytics Users group.
- Bracketed text indicates role assignments. For example, the SASUSERS group is a direct member of the **Visual Analytics: Basic** role.

Figure A1.2 Standard Membership Structure



Configuration Properties

How to Set Configuration Properties

- 1 On the **Plug-ins** tab in [SAS Management Console](#), navigate to **Application Management** ► **Configuration Manager** ► **SAS Application Infrastructure** ► **Visual Analytics**. Expand nodes as needed, right-click on the appropriate node, and select **Properties**.
 - Set suite-level properties on the **Visual Analytics** node.
 - Set explorer properties on the **Visual Analytics Explorer** node.
 - Set alerts properties on the **Visual Analytics Hyperlink Service** node.
 - Set SAS Mobile BI properties on the **Visual Analytics Transport Service** node.
 - Set web viewer properties on the **Visual Analytics Viewer** node.

Note: For conciseness, version numbers are omitted from the instructions and figure in this topic.
- 2 On the **Advanced** tab of the appropriate Properties dialog box, add or set values.
- 3 To make changes take effect, restart the SAS Web Application Server. One approach is to restart all instances from your equivalent of `sas-configuration-directory/Web/Scripts/AppServer/`.

```
UNIX      appsrvconfig.sh restart
```

```
Windows  appsrvconfig.cmd restart
```

For details and alternatives, see [Understanding SAS Web Application Server Management](#) and [Using Configuration Manager](#) in the *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Suite-Level Properties

TIP Use the **Visual Analytics** node (except where otherwise specified).

App.AllowGuest
enables or disables guest access. Valid values are `true` and `false`. See [“Supporting Guest Access” on page 73](#).

Note: In new deployments, this property is set on the **Visual Analytics Transport Service** node and the **Visual Analytics Viewer** node, not at the suite level.

TIP To enable or disable guest access for the home page, set this property on the **Application Management** ► **Configuration Manager** ► **SAS Application Infrastructure** ► **Visual Analytics Hub** node.

`las.caching.key.lifetime`

sets the duration of time (in seconds) for which a LASR security key is cached in the middle tier. The default is 180 seconds (3 minutes). Do not set a custom value unless you are directed to do so by SAS Technical Support.

`las.caching.permission.lifetime`

sets the duration of time (in seconds) for which permission information is cached by the LASR authorization service. The default is 900 seconds (15 minutes). Do not set a custom value unless you are directed to do so by SAS Technical Support.

`las.caching.user.lifetime`

sets the duration of time (in seconds) for which user information is cached by the LASR authorization service. The default is -1 (the cache does not have a time-based expiration period). With the default setting, user objects remain in the cache until the requesting user's session ends. Do not set a custom value unless you are directed to do so by SAS Technical Support.

`lasr.mgmt.server.monitor.refresh`

sets the refresh interval (in seconds) for the LASR management service's information cache. This setting affects timing on the **LASR Servers** and **LASR Tables** tabs. The default is 60. The default value provides a trade-off (among responsiveness, consumption of system resources, and currency of information) that is appropriate for most deployments.

Note: If you set this property to 0, no caching of LASR management service information occurs. When the cache is disabled, changes to a server's tables limit take effect immediately, and information is retrieved on demand for each request. Response time for an information request is increased.

`va.AuditingEnabled`

specifies whether applications write audit records. Valid values are `true` and `false`. The default is `false`.

CAUTION! Audit data can consume significant amounts of disk space and processing capacity. If you enable auditing, it is essential that you manage the sizes of tables that contain audit data. See ["How to Safely Enable Auditing" on page 49](#).

`va.baseSchedulingFolder`

specifies the name of the parent folder for jobs and flows that are used in report distribution. The default is `/System/Applications/SAS Visual Analytics/ScheduledDistribution`. See ["Supporting Report Distribution" on page 82](#).

Note: To schedule a report, users must have `WriteMemberMetadata` access to the specified base folder.

`va.ComparisonEpsilon`

specifies a small number to be used to account for floating-point rounding error in the following numeric comparisons: equals, not equals, less than, greater than, less than or equals, greater than or equals. Valid values are doubles. The default is `1e-12`. In the unusual circumstance in which users find that some values are being compared as equal when they should not be (or vice versa), consider changing this value. The epsilon comparison is relative to the size of the numbers that are being compared (it is not

absolute). When the following expression is true, a and b are considered to be equal:

$$\text{ABS}(a-b) \leq \text{epsilon} * \text{MAX}(\text{ABS}(a), \text{ABS}(b))$$

- va.dataServer.PublicLibrary
identifies the standard library for your configured data server. If you change the name of the library, you must also update this property.
 - va.defaultLASRLibrary
identifies the predefined LASR library for the **Visual Analytics LASR** server. This property is no longer in use.
 - va.defaultPublicFolder
identifies the standard metadata location for LASR tables that are generated by data import and load activities (for example, `/Shared Data/SAS Visual Analytics/Public/LASR`). If you change the name of the referenced folder, you must also update this property.
 - va.defaultWorkspaceServer
identifies the suite-level default SAS Application Server. If you change the name of the referenced server (for example, `SASAPP`), you must also update this property. See [“Which Server is Used?” on page 90](#).
 - va.distribution.email.aggregate.attachments.mb
sets a maximum combined size (in megabytes) for all attachments in a report distribution email. If this property is not defined, the default value of 20 is in effect. If a value of -1 is specified for this property, no limit is in effect. See [“Supporting Report Distribution” on page 82](#).
 - va.distribution.email.attachment.mb
sets a maximum size (in megabytes) for an individual attachment in a report distribution email. Initially, a value of -1 is specified, so no limit is in effect. If this property is not defined, the default value (20) is in effect. See [“Supporting Report Distribution” on page 82](#).
 - va.extractRelationshipData
enables periodic extraction of data about relationships among SAS Visual Analytics objects. Relationship data is extracted from the relationship service to the autoloading data directory (drop zone) for administrative reporting. Valid values are `true` and `false`. The default is `false`. See [Chapter 6, “Reports for Administrators,” on page 121](#).
 - va.GeoMapMaxResolution
for an alternate OpenStreetMap server (specified in the property `va.GeoMapServerUrl`), sets the resolution value for the farthest out zoom level in each geo map. The default is `156543.0339`.
 - va.GeoMapNumResolutions
for an alternate OpenStreetMap server (specified in the property `va.GeoMapServerUrl`), sets the number of levels in each geo map. The default is `18`.
- TIP** Each level corresponds to an increment by which a user can zoom in. Adding a level doubles the resolution and quadruples the number of tiles. In general, a value higher than 23 is not practical.
- va.GeoMapServerUrl
specifies a comma-delimited list of URL addresses that reference alternate OpenStreetMap servers (for example, `http://serverA.org, http://`

`serverB.org`, `http://serverC.org`). To use the OpenStreetMap servers that SAS hosts, leave the value for this property blank. See [“OpenStreetMap Server” on page 78](#).

va.IgnoreMissingInCountDistinct

determines whether missing values are ignored in counts when the Distinct operator is used. Valid values are `true` and `false`. The default is `false` (missing values are included in distinct counts).

Note: This property affects count behavior in the designer and viewers, but not in the explorer.

va.LASRMonitor.HostPort

specifies the machine name and port for the process that monitors a distributed SAS LASR Analytic Server. The value is in the format `host:port` (for example, `machine.company.com:9971`).

va.lastActionLogPath

specifies the location of last action logs. See [“Get Server Information” on page 6](#) and [“Get Table Information” on page 12](#). The standard location is within the [SAS configuration directory](#) at `/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/Monitoring/Logs`. In a multi-machine deployment, the specified location exists on the middle-tier host.

va.MaxTiesToIncludeOnRank

sets the maximum number of identically ranked values that can be returned in a rank operation. Valid values are integers. The default is 100.

va.monitoringPath

specifies the location for certain process ID files and logs. The standard location is within the [SAS configuration directory](#) at `/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/Monitoring`. The specified location must exist on a workspace server host.

Note: To specify a custom directory for monitoring artifacts for a particular SAS LASR Analytic Server, set an extended attribute for that server. See [“VA.MonitoringPath” on page 102](#).

va.publicLASRLibrary

identifies the general purpose library for data import and load activities (for example, `Visual Analytics Public LASR`). If you change the name of the referenced library, you must also update this property.

va.publicLASRServer

identifies the server that is associated with the `va.publicLASRLibrary` (for example, `Public LASR Analytic Server`).

va.SASGeomapCommunicationProtocol

sets the protocol for connections between SAS Visual Analytics and the OpenStreetMap servers that SAS hosts. Valid values are `http` and `https`.

va.SASGeomapEsriURL

references a supported Esri server. The value must be a URL that specifies a protocol, the server’s host name, and the REST endpoint of the server. See [“Esri Server” on page 79](#).

va.SelfServe.MaxUploadSizeInMegabytes

sets the maximum file size (in megabytes) that a user can import. This property affects importing local files in the data builder, the explorer, and the designer. The default, 4096, corresponds to browser-based constraints. To

further constrain import activities, set a lower value for this property. You cannot use this property to circumvent browser-based constraints.

- va.SelfService.ImportGoogleRowLimit
sets a maximum number of rows for an import from Google Analytics. The default is 100000.
- va.SelfService.ImportRowsHardCap
sets a maximum number of rows for a self-service import action. If this value is exceeded, no data is imported. No initial value is set (initially, no limit is imposed).

Note: Enforcing a threshold requires a query to the data provider for each import action, so setting a value for this property can negatively impact performance.
- va.SelfService.ImportRowsSoftCap
sets the number of rows that triggers a warning message for a user who is performing a self-service import action. The message indicates that the import action might take a long time. No initial value is set (initially, no limit is imposed).

Note: Enforcing a threshold requires a query to the data provider for each import action, so setting a value for this property can negatively impact performance.
- va.supportSharedThumbnails
determines whether the designer and explorer generate thumbnails (specific preview images). Valid values are `true` and `false`. The default is `false` (specific preview images are not generated).

Note: Generated thumbnails are displayed only if this property is also set to `true` on the **Visual Analytics Hub** software component. See [“Home Page: Configuration Properties” on page 71](#).

Alerts Properties

TIP Use the **Visual Analytics Services** ► **Visual Analytics Hyperlink Service** node.

- va.Alert.DefaultEvaluationIntervalMilliseconds
specifies the evaluation interval (how frequently the system makes a determination about whether the alert’s conditions have been met). The default is 600000 milliseconds (10 minutes).

Note: This property affects only alerts that do not use a custom interval. In the designer’s Edit Alert window, the **Use the system default** setting causes the value for this property to be used.

Note: Long intervals increase the risk of a missed incident (where the alert’s conditions are met intermittently between one evaluation and the next). Short intervals consume more resources and can negatively impact the performance of the entire SAS Visual Analytics suite of applications.
- va.Alert.DefaultMaxEvaluationTimeMilliseconds
specifies how long an individual evaluation can run before it terminates and restarts. The default is 1800000 milliseconds (30 minutes).

- `va.Alert.EvaluationCycleMilliseconds`
specifies how frequently the system verifies that alerts are running. The default is 30000 milliseconds (30 seconds). If a large number of alerts are registered, consider increasing the value to reduce the use of resources.
- `va.Alert.Eventgen.disabled`
specifies whether alerts generate notifications. The default is `false`. To disable notifications, set this property to `true`.
- `va.Alert.SMS.showServerName`
specifies whether to append *Server: server-name* to the end of an SMS text message that is generated by an alert. The default is `true`. If the server name is not useful in your environment, or if you want to reduce the possibility of message truncation, set this property to `false`.
- `va.AlertThreadPool.CoreSize`
specifies the number of threads that are available in normal circumstances (for concurrent evaluation of alerts). The default is 3.
- `va.AlertThreadPool.IdleTimeoutSeconds`
specifies how long excess threads can be idle before they are terminated. The purpose of terminating idle excess threads is to reduce the number of threads to the specified `CoreSize`. The default is 1800 seconds (30 minutes).
- Note:** This property is applicable only if the `MaxSize` is greater than the core size.
- `va.AlertThreadPool.MaxSize`
specifies the maximum number of threads that can be used (for concurrent evaluation of alerts). If the load is heavy, additional threads are temporarily added to the `CoreSize` (up to the value that is set for this property). The default is 3.
- `va.AlertThreadPool.QueueSize`
specifies the maximum number of tasks that can be queued. The default is 100000.

Explorer Properties

TIP Use the **Visual Analytics Explorer** node.

- `vae.DecisionTreeTimeout`
affects how long (in seconds) the explorer waits for a response after the explorer makes a decision tree request. The default is 300.
- `vae.PageRowCount`
limits the amount of data that can be returned for a table visualization. If table sorting is enabled, the `vae.PageRowCount` limit is applicable only if its value exceeds the value that is specified for the `vae.SortResultLimit` property. The default is 10000.
- Note:** For example, if the value is 10000, then 10000 rows of data are returned to the client. If the user scrolls through the data, and passes the row that is numbered 10001, the client prompts the SAS LASR Analytic Server for the next 10000 rows.
- `vae.PathingPathLengthLimit`
sets the server-side maximum path length for a Sankey diagram. The value specifies the maximum number of events (nodes) in a single path. If the

longest path length equals or exceeds the specified value, the explorer displays a message indicating that the Sankey diagram excludes paths longer than the specified value. If this property is not defined, the default value (2000) is in effect.

Note: Client-side controls in the explorer interface provide additional, more stringent limits (to facilitate quick display of Sankey diagrams).

Note: Each path in a Sankey diagram has a limit of 32,767 characters for the event values. Depending on the width of an event, the effective path length limit might be less than the value that you specify for `vae.PathingPathLengthLimit`. A numeric value uses a width of 40 for this calculation.

`vae.PathingTopKLimit`

sets the server-side maximum number of paths that are selected by path ranking in a Sankey diagram. If this property is not defined, the default value (1000) is in effect.

`vae.PathingTransactionIdsLimit`

sets the server-side maximum number of unique values for the transaction identifier in a Sankey diagram. If this property is not defined, the default value (10000) is in effect.

`vae.TableSortingEnabled`

specifies whether users in the explorer can click on a column heading to sort the items. Valid values are `true` and `false`. The default is `true`.

Transport Service Properties

Note: For the most current and comprehensive information about transport service properties, see [Configuration Properties: SAS Mobile BI in SAS Intelligence Platform: Middle-Tier Administration Guide](#). As a convenience, some information about transport service properties is provided here.

TIP Some transport service properties affect multiple components, such as the modern web viewer and SAS Mobile BI.

TIP Use the **Visual Analytics Services** ► **Visual Analytics Transport Service** node.

`Printing.Timeout`

sets a maximum wait time (in milliseconds) that affects printing reports from applications such as the designer and the viewer. The default is 900000 milliseconds (15 minutes). To disable this property, set its value to 0.

Note: This setting does not affect the first phase of a print request, which generates a report package. This setting affects only the second phase of a print request, which uses a stored process call to execute the print routine.

`viewerservices.data.default.interactive.drill.depth`

determines how much data is sent to SAS Mobile BI for offline drilling. This property is applicable to visualizations that reference a hierarchy. The default is 3 (users can drill down three levels). If certain reports require users to have the ability to drill down more than three levels into a hierarchy, modify the value.

viewerservices.company.banner.logoUrl
this property is not currently supported.

viewerservices.company.banner.message
this property is not currently supported.

viewerservices.company.banner.title
this property is not currently supported.

viewerservices.default.max.cells.produced
sets the maximum number of data cells that can be delivered to SAS Mobile BI for a single data query. The default is 250000 data cells, which is sufficient for most environments and does not cause the web application server to crash. In very rare scenarios, you might need to modify the value.

Note: If the number of data cells in a query exceeds the value specified for this property, the data that is returned to SAS Mobile BI is truncated. Data in the displayed report is not complete.

viewerservices.enable.whitelist.support
controls which approach is used to manage access to SAS Mobile BI. Valid values are:

false causes the blacklist to be enforced and the whitelist to be ignored. With this setting, all mobile devices can use SAS Mobile BI except for those devices that are on the blacklist. This is the default.

true causes the whitelist to be enforced and the blacklist to be ignored. With this setting, only mobile devices that are on the whitelist can use SAS Mobile BI.

CAUTION! Enabling the whitelist can disrupt existing users. Make sure that all valid mobile devices are on the whitelist before you make the change.

TIP As an alternative to setting this property explicitly, you can set it from within the administrator. See [“Change How Devices Are Managed” on page 58](#).

viewerservices.image.default.max.bytes
sets the maximum size of images (PNG, BMP, JPEG, or GIF) that can be delivered to a mobile device. Larger images are resized on the server side before delivery. The default is 307200 (300 KB), which is sufficient for most environments. In very rare scenarios when you want to change this constraint, consider modifying the value. To entirely disable resizing of images in the middle tier, set the value to 0. However, to ensure faster download times and smaller memory footprints on the mobile device, do not increase the value of this property or set the value to 0.

Note: Users can customize image resizing on their devices by setting the **Scale type** option (under **Insert** ► **Other** ► **Image**). If the option is set to **None**, the user’s device is exempt from middle-tier resizing.

viewerservices.lasr.socketTimeout.milliseconds.interactions
sets the maximum wait time for when SAS Mobile BI attempts to contact SAS LASR Analytic Server. This property is applicable to live requests from a mobile device for tasks such as filtering, brushing, and drilling. The default is 30000 milliseconds (30 seconds), which is sufficient for most environments. If sessions between SAS Mobile BI and SAS LASR Analytic Server are timing out, consider modifying the value.

- `viewerservices.lasr.socketTimeout.milliseconds.subscribe`
sets the maximum wait time for a response to a query in a subscribed report when SAS Mobile BI contacts the SAS LASR Analytic Server. The default is 300000 milliseconds (5 minutes), which is sufficient for most environments. If the queries within some reports take an excessive amount of time for completion, consider modifying the value.
- `viewerservices.offline.limit.days`
specifies how long downloaded mobile report data remains available to a user who is not signed in to SAS Mobile BI. A user who is offline for the specified number of days must sign in before accessing any mobile report data. The default is 15. This property affects only those users who have the Limit Duration of Offline Access capability.
- `viewerservices.passcode.attempts`
limits the number of sequential failed attempts to enter a passcode for SAS Mobile BI. The default is 5. If a user reaches the limit, the user is locked out of the app for 15 minutes. After the lockout interval, the user can again attempt to enter his or her passcode. If the user reaches the limit again, all custom content (data, reports, settings, and connection information) is removed from the mobile device.
- Note:** This property is applicable to only those users who are subject to the capability [“Require Passcode On Mobile Devices”](#).
- `viewerservices.passcode.timeout`
specifies, in minutes, how frequently a user must re-enter his or her passcode in SAS Mobile BI. The default is 15.
- `viewerservices.validate.schema.create`
enables XML schema validation when reports are rendered in SAS Mobile BI. When this property is set to `true`, all actions that apply to the creation of reports are captured in the transport log. The default is `false`. Set this property only if SAS Technical Support instructs you to do so.
- `viewerservices.validate.schema.read`
enables XML schema validation when reports are rendered in SAS Mobile BI. When this property is set to `true`, all actions that apply to opening and viewing reports are captured in the transport log. The default is `false`. Set this property only if SAS Technical Support instructs you to do so.
- `viewerservices.validate.schema.write`
enables XML schema validation when reports are rendered in SAS Mobile BI. When this property is set to `true`, all actions that apply to the writing of reports are captured in the transport log. See [“Adjusting the Logging Configuration” on page 87](#). The default is `false`. Set this property only if SAS Technical Support instructs you to do so.

Web Viewer Properties

TIP Use the **Visual Analytics Viewer** node.

- `vav.ui.mode`
enables an administrator to force the use of a particular presentation mode for the web viewer.

(unspecified)	If the <code>vav.ui.mode</code> property is not specified, the modern presentation mode is used, unless a user selects Classic (in his or her Default Appearance setting for the web viewer) or uses a web browser that does not support the modern presentation mode (for example, Microsoft Internet Explorer 9).
classic	Forces the use of the Flash presentation mode.
modern	Forces the use of the HTML5 presentation mode, unless your web browser does not support that mode.

Note: The modern presentation mode does not support right-to-left (RTL) languages.

TIP To force the use of a particular presentation mode for the home page, set that application's presentation mode property. See [“Home Page: Configuration Properties” on page 71](#).

See Also

- [“Configuration Properties for High-Cardinality Thresholds” on page 150](#)
- [“Home Page: Configuration Properties” on page 71](#)

High-Cardinality Constraints

Introduction

High-cardinality data has one or more columns that contain a very large number of unique values. For example, user names, email addresses, and bank account numbers can be high-cardinality data items.

SAS Visual Analytics supports billions of values that are aggregated to thousands of values. If the billions of values in a table have millions of unique identifiers, then a column that contains those identifiers is a high-cardinality data item.

To help ensure that users get meaningful results in a timely fashion, the number of unique values that can be returned for certain visualizations and report objects is constrained. When a user selects a high-cardinality data item, the outcome is determined by any applicable thresholds, the number of unique values in the data, and the user's selections.

The following topics provide information about two distinct levels of thresholds: client-side thresholds and middle-tier thresholds.

Client-Side Thresholds for High-Cardinality Data

Client-side thresholds are specific to an individual application (such as the explorer), or to a group of applications (such as the designer and the viewer). For some requests that exceed a client-side threshold, an error is displayed, and

no results are returned. For some requests that exceed a client-side threshold, but do not exceed a middle-tier threshold, adapted results are returned.

Note: In general, client-side thresholds are fixed. An exception is that a user can select a low, medium, or high threshold level as a user preference in the explorer. On a computer that has low memory availability, setting the client-side threshold to **Low** can help prevent events such as system crashes.

Client-side thresholds for visualizations and report objects are documented in the [Data Limits](#) appendix in the *SAS Visual Analytics: User's Guide*. The appendix explains the adapted responses that clients provide for certain requests that exceed a client-side threshold (but do not exceed a middle-tier threshold).

Middle-Tier Thresholds for High-Cardinality Data

Middle-tier thresholds have a wider scope, affecting all instances of the specified visualization or report object. Compared to client-side thresholds, middle-tier thresholds are less granular and less restrictive. For requests that exceed a middle-tier threshold, an error message is displayed, and no results are returned. The default thresholds work in almost all environments. In general, users filter or group any high-cardinality data items, so requests rarely exceed a middle-tier threshold.

In the following table, the second column indicates the maximum number of unique values (not the maximum volume of data).

Table A1.2 Middle-Tier Thresholds

Visualization or Report Object	Rows
Decision trees*	10,000
Crosstabs	50,000
Tables (in the designer and the viewers)	50,000
Box plots: at least one measure, no categories**	50,000
Bar charts: single category	50,000
Heat maps: single category	50,000
Line charts: at least one measure, single category (numeric, date, time, or string)	50,000
Bubble plots: three measures, grouped	50,000
Bubble plots: three measures, grouped with animation category	50,000
Bubble plots: three measures, not grouped, horizontal or vertical series (or both)	50,000
Bubble plots: three measures, no categories	100,000
Scatter plots	100,000

Visualization or Report Object	Rows
Tables (in the explorer)	100,000

* There is also a time-out period for decision tree calls. See “[vae.DecisionTreeTimeout](#)” on page 144.

** If there is no category, one box is applied for each measure, up to 400 measures.

Configuration Properties for High-Cardinality Thresholds

CAUTION! Increasing a middle-tier threshold can affect performance and stability. The default settings are appropriate in most environments. Do not set excessively high thresholds. If you have questions about adjusting the following properties, contact SAS Technical Support.

Note: For instructions, see “[How to Set Configuration Properties](#)” on page 139.

The following properties affect middle-tier thresholds:

va.DistinctCountServerLimit

sets the distinct count limit for graphs. By default, there is no distinct count limit for graphs. The default is -1.

Scope: Entire suite

va.DistinctCountDataPanelLimit

sets the distinct count limit for data that is displayed in a data panel. This property affects only the data panel, not the distinct count limits within graphs. The default is 5,000.

Scope: Entire suite

va.CardinalityLimitForGroupByTempTable

for all high cardinality rank requests that exceed the specified limit (number of unique values), prevents processing and returns an error. Set this property only in the unusual circumstance in which high cardinality ranks cause the SAS LASR Analytic Server to hang. For example, to block rank requests against data that contains more than 2 million unique values, set this property to 2000000. If you choose to set this property, the suggested value is 3000000.

Scope: Entire suite

va.CardinalityLimitForGroupByCountDistinctTempTable

for only distinct count high cardinality rank requests that exceed the specified limit (number of unique values), prevents processing and returns an error. Set this property only in the unusual circumstance in which distinct count high cardinality ranks cause the SAS LASR Analytic Server to hang. (This property affects only distinct count requests, providing a narrower constraint than the `va.CardinalityLimitForGroupByTempTable` property.) If you choose to set this property, the suggested value is 1000000.

Scope: Entire suite

va.SortResultServerLimit

sets the maximum number of values that can be returned for detail queries that are run with sorting. This property affects only results in list tables for which details are turned on.

Scope: Entire suite, except for the explorer

va.CategoryCardinalityServerLimit

sets the maximum number of values for category crossings. Only a fixed (and finite) number of category crossings are supported. For example, if you drag and drop "First name" and "Last name" onto the population of the United States, the server might generate 200 million different values. This property determines how high the cardinality can be and still allow the server to process and return results to the client. If the number of values for category crossings exceeds this limit, the query is not run.

Scope: Entire suite, except for the explorer

va.SummaryServerRowLimit

sets the maximum number of values that can be returned to the middle tier for further processing. For example, for high-cardinality data that is sorted by first name, the number of values computed could be very large.

Scope: Entire suite, except for the explorer (which uses `vae.SummaryServerLimit`)

va.MidtierCellLimit

sets the maximum size of a crosstab.

Scope: Entire suite, except for the explorer

va.maxPeriodCalculations

specifies the maximum number of calculated columns that are constructed for period calculations. If this limit is exceeded for a particular period measure, excess calculations are excluded, and existing calculations (for that particular period measure) are replaced with missing values. The user is prompted to apply a filter to reduce the number of calculations. The default is 800.

Note: Software optimizations reduce the number of calculations before this limit is applied, so this limit is rarely exceeded. An example of the effect of this property is a distinct count calculation with cumulative periods (the number of unique date values that are visible cannot exceed the specified limit).

Scope: designer, viewer, transport service

va.MaxSparkTables

sets the maximum number of spark tables. The default is 300.

Scope: Entire suite, except for the explorer

va.CheckCardinalityBeforeQuery

controls whether cardinality pre-checks occur. The default value is -1 (which disables this constraint). By default, pre-checks do not occur.

Scope: Entire suite, except for the explorer

va.CheckCardinalityWithinQuery

controls whether SAS LASR Analytic Server enforces cardinality limits. By default, these checks do occur.

Scope: Entire suite, except for the explorer

vae.BoxPlotServerLimit

sets middle-tier thresholds for box plots that have at least one measure and no more than one category.

Scope: Explorer only

- `vae.DecisionTreeServerLimit`
sets the middle-tier threshold for decision trees.
Scope: Explorer only
- `vae.FetchRowsServerLimit`
sets middle-tier thresholds for tables.
Scope: Explorer only
- `vae.FrequencyServerLimit`
sets middle-tier thresholds for bar charts that have a single category. This constraint is applied before a selection list of values is displayed.
Scope: Explorer only
- `vae.modeling.ClassCardinalityLimit`
sets the maximum number of distinct levels in a model. This property limits the cumulative total of classification effects and interaction terms in a model. For example, if you set this property to 800, a user can neither specify an effect variable that contains more than 800 distinct levels nor add an effect variable that would cause the total number of distinct levels to exceed 800. The initial value is 2048.
Scope: SAS Visual Statistics add-on (if licensed)
- `vae.modeling.DecisionTreePredictorBinsCardinalityLimit`
sets the maximum number of bins for a measure variable in a decision tree. The initial value is 1024.
Scope: SAS Visual Statistics add-on (if licensed)
- `vae.modeling.DecisionTreePredictorCardinalityLimit`
sets the maximum number of distinct levels for a category variable in a decision tree. The initial value is 1024.
Scope: SAS Visual Statistics add-on (if licensed)
- `vae.modeling.DecisionTreeResponseCardinalityLimit`
sets the maximum number of distinct levels for the response category variable in a decision tree. In the initial configuration, this property is not set, so the default value (100) is in effect.
Scope: SAS Visual Statistics add-on (if licensed)
- `vae.modeling.GroupByCardinalityLimit`
sets the maximum number of distinct levels for the group-by variables in a model. This property limits the cumulative total for group-by variables in a model. For example, if the value of this property is set to 800, users can neither specify a group-by variable that contains more than 800 distinct levels nor add a group-by variable that would cause the total number of distinct levels to exceed 800. The initial value is 1024.
Scope: SAS Visual Statistics add-on (if licensed)
- `vae.RealScatterServerLimit`
sets middle-tier thresholds for scatter plots and bubble plots that have three measures and no categories.
Scope: Explorer only
- `vae.ScatterPlotServerLimit`
sets middle-tier thresholds for heat maps that have exactly one category.
Scope: Explorer only

`vae.SummaryServerLimit`

sets middle-tier thresholds for the following visualization types:

- crosstabs
- line charts that have at least one measure and a single category (numeric, date, time, or string)
- bubble plots that are grouped with no series, grouped with animation, or with series and not grouped

Scope: Explorer only (other applications use `va.SummaryServerRowLimit`)

Predefined LASR Libraries

The following tables document the initial configuration of the predefined LASR libraries in a new deployment.

Note: Paths that begin with `/AppData` or `/Applications` are host locations within a [SAS configuration directory](#).

Note: Paths that begin with `/Products` or `/Shared Data` are metadata folders.

Table A1.3 General-Purpose Library

Name	Visual Analytics Public LASR (libref: LASRLIB, server tag: VAPUBLIC) The library name must match the value of a configuration property. See “va.publicLASRLibrary” on page 142.
Location	/shared Data/SAS Visual Analytics/Public The location must match the value of a configuration property. See “va.defaultPublicFolder” on page 141.
Data server	Public LASR Analytic Server The server name must match the value of a configuration property. See “va.publicLASRServer” on page 142.
Intended use	The default output library for import and load actions for all registered users (SASUSERS). This library is sometimes referred to as the public LASR library.
Autostart	Enabled
Autoload	Enabled (To use autoload, start the scheduled task. See “Autoload” on page 22.) Data: /AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad Scripts: /Applications/SASVisualAnalytics/VisualAnalyticsAdministrator LASR table objects: /Shared Data/SAS Visual Analytics/Public/LASR
Reload-on-start	Enabled Library: Visual Analytics Public Data Provider (libref: DPPUBLIC) Directory: /AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/PublicDataProvider LASR table objects: /Shared Data/SAS Visual Analytics/Public/LASR

Table A1.4 Restricted Library

Name	Visual Analytics LASR (libref: VALIBLA, server tag: HPS)
Location	/Products/SAS Visual Analytics Administrator
Data server	LASR Analytic Server
Intended use	An output library to which only administrators can import and load data. This library is sometimes referred to as the non-public, private, or limited availability LASR library. All registered users have Read access to this library.
Autostart	Not enabled
Autoload	Not enabled (To use autoload, set extended attributes, and start the scheduled task. See “Autoload” on page 22.) Data: /AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad/VALIBLA Scripts: /Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/VALIBLA LASR table objects: /Shared Data/SAS Visual Analytics/AutoLoad/VALIBLA
Reload-on-start	Not enabled

Table A1.5 Administrative Reporting Library

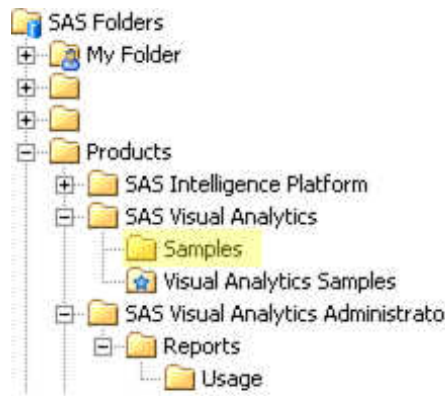
Name	Environment Manager Data Mart LASR (libref: EVDMLA, server tag: EVDM)
Location	/Shared Data/SAS Visual Analytics/AutoLoad/EVDMLA
Data server	LASR Analytic Server
Intended use	Drop zone for administrative reporting data. See Chapter 6, “Reports for Administrators,” on page 121.
Autostart	Enabled
Autoload	Enabled (To use autoload, start the scheduled task. See “Autoload” on page 22.) Data: /AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad/EVDMLA Scripts: /Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/EVDMLA LASR table objects: /Shared Data/SAS Visual Analytics/AutoLoad/EVDMLA
Reload-on-start	Not enabled

Sample Objects

Configured Samples

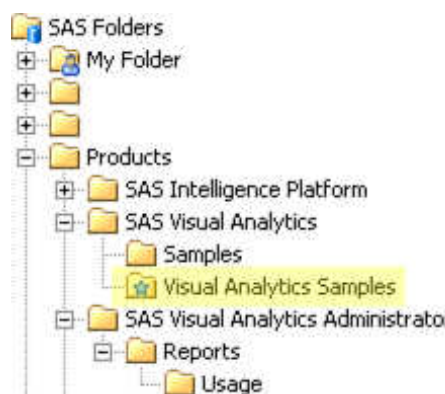
If you chose to configure samples when you installed SAS Visual Analytics, the following objects exist:

- The metadata folder `/Products/SAS Visual Analytics/Samples`, which contains a sample exploration (if your deployment includes the explorer), sample tables, and sample reports.



Small Insight Sample	Visual exploration
VA_SAMPLE_WARRANTY_CLAIMS	Table
VA_SAMPLE_TELCO03	Table
VA_SAMPLE_SMALLINSIGHT	Table
VA_SAMPLE_RETAIL_RO_PROMO_EFFECTV	Table
VA_SAMPLE_RETAIL_RO_MD_CANDIDATE	Table
VA_SAMPLE_RETAIL_RO_COST_GT_COMP	Table
VA_SAMPLE_K12_STUDENT	Table
VA_SAMPLE_CASINO_REV_FCST2	Table
VA_SAMPLE_BANKCAMPAIGN	Table
VA_SAMPLE_ACME_BANK	Table
Wireless Call Quality Analysis Sample	SAS report (2G)
Warranty Analysis Sample	SAS report (2G)
Student Analysis Sample	SAS report (2G)
SAS Visual Analytics Sample Includes Exploration Content	SAS report (2G)
SAS Visual Analytics Sample	SAS report (2G)
Revenue Optimization Sample	SAS report (2G)
Casino Floor Performance Sample	SAS report (2G)
Capital Exposure and Risk Sample	SAS report (2G)

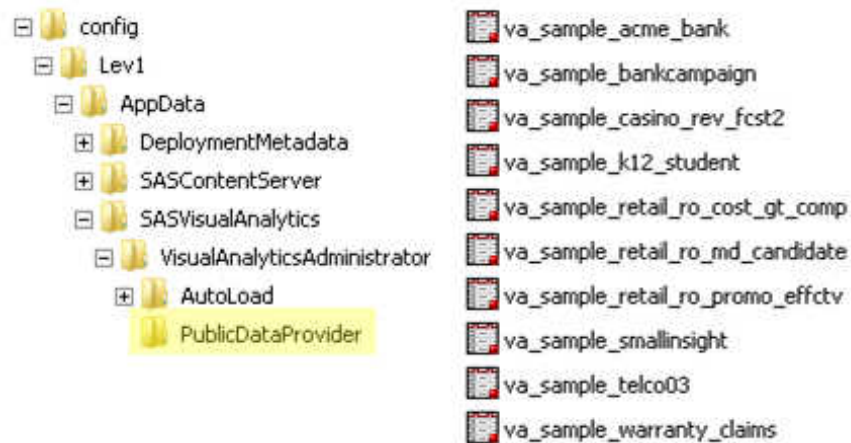
- The metadata favorites folder `/Products/SAS Visual Analytics/Visual Analytics Samples`, which provides access to the sample reports and is displayed on the home page as a collection.



Wireless Call Quality Analysis Sample	SAS report (2G)
Warranty Analysis Sample	SAS report (2G)
Student Analysis Sample	SAS report (2G)
SAS Visual Analytics Sample Includes Exploration Content	SAS report (2G)
SAS Visual Analytics Sample	SAS report (2G)
Revenue Optimization Sample	SAS report (2G)
Casino Floor Performance Sample	SAS report (2G)
Capital Exposure and Risk Sample	SAS report (2G)

- Physical tables (`va_sample_*.sas7bdat`), stored in the [SAS configuration directory](#) at the following location:

`/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/PublicDataProvider`



Note: The specified location is the reload-on-start backing store for the public LASR library (Visual Analytics Public LASR).

Tips and Guidelines

- The sample data is loaded when the associated server (Public LASR Analytic Server) starts. See [“Start or Stop a Server” on page 7](#).

Note: If the tables are interactively unloaded, you can reload the tables by stopping and then restarting the server. You cannot interactively reload the tables. (A log entry or message might indicate that you are not authorized to interactively reload the tables. However, the tables are not interactively reloadable, regardless of your permissions.)

Note: In the designer, explorer, and data builder, users can trigger an autostart of the associated server by performing a self-service import action. See [“Host Account Privileges” on page 5](#).

- Initially, all registered users can see the sample objects. To limit access, set denials of the ReadMetadata permission on the folder `/Products/SAS Visual Analytics/Samples`. For example, if you want the sample objects to be visible to only unrestricted users, set a denial of the ReadMetadata permission for the PUBLIC group. See [“Grant or Deny a Permission” on page 39](#).

Note: Each user can hide the sample collection. An administrator can unpublish the sample collection. See the Help for the home page.

- You can delete the sample objects (from metadata) and the associated physical tables (from the file system).

Manual Configuration

If you chose not to configure samples when you installed SAS Visual Analytics, the folders, objects and physical tables that are described in the preceding sections do not exist. However, sample-related files exist in the following locations regardless of whether you chose to configure samples:

- SAS package (SPK) files for the sample objects:

Table A1.6 SPK Locations

UNIX (example)	<code>/opt/sas/SASHome/SASVisualAnalyticsAdministrator/7.4/Config/Deployment/ Packages</code> <code>/opt/sas/SASHome/SASVisualAnalyticsExplorer/7.4/Config/Deployment/Packages*</code>
Windows (example)	<code>C:\sas\SASHome\SASVisualAnalyticsAdministrator\7.4\Config\Deployment \Packages</code> <code>C:\sas\SASHome\SASVisualAnalyticsExplorer\7.4\Config\Deployment\Packages*</code>

* If your deployment includes the explorer.

Note: The SPK files are on the machine that hosts SAS Visual Analytics.

- Backup copies of the physical tables:

UNIX (example)	<code>/opt/sas/SASHome/SASFoundation/9.4/samples/bimva</code>
Windows (example)	<code>C:\sas\SASHome\SASFoundation\9.4\bimva\sample</code>

Here is one way to configure samples after installation:

- 1 In SAS Management Console, add metadata objects as follows:
 - a On the **Folders** tab, right-click the `/Products/SAS Visual Analytics` folder, and select **New ► Folder**. Add a **Samples** folder. On the new folder's **Authorization** tab, add the following direct access controls for the SASUSERS group: deny WriteMetadata, grant Read, and grant Write.
 - b Right-click the **Samples** folder, and select **Import SAS Package**.
 - i From the SASVisualAnalyticsAdministrator [branch](#), import VASampleTable.spk and VAAAdditionalSampleTables.spk. Then, import VASampleReport.spk and VAAAdditionalSampleReports.spk.
 - ii If your deployment has the SASVisualAnalyticsExplorer [branch](#), import VASampleReportExpl.spk. If your deployment also has SAS Visual Statistics, import VASampleBankTable.spk. (It is not necessary to import VASampleTable.spk again.)
- 2 Copy the physical tables (va_sample_*.sas7bdat) from your `bimva` location to the `PublicDataProvider` directory for your [Visual Analytics Public LASR](#) library.
- 3 On the modern SAS Home page, create, add, and publish a collection that references the sample reports.
 - a Sign in to the home page as a member of the SAS Administrators group.
 - b Click **Collection**.
 - c Select **Create a New Collection**.
 - d Enter the name `Visual Analytics Samples`.

- e Select **SAS Folders** ▶ **Products** ▶ **SAS Visual Analytics**.
- f Select **Publish this collection for all users**.
- g Click **Save**.
- h To add the sample reports to the new collection, edit the collection.

Appendix 2

Adding Custom Polygon Data for Geographic Maps

<i>Overview of Custom Polygon Data</i>	161
<i>Prerequisites for Adding Custom Polygon Data</i>	162
General Prerequisites	162
Considerations for Non-English Character Data	162
<i>Geographic Data Sets</i>	162
ATTRLOOKUP Data Set	162
CENTLOOKUP Data Set	164
Custom Polygon Data Set	164
<i>Creating Custom Polygons</i>	166
Overview of Creating Custom Polygons	166
MAPSCSTM Library	167
Custom Polygon Deployment Program	167
Output Data Sets	168
<i>Testing Custom Polygons in SAS Visual Analytics</i>	168
<i>Example 1: Deploying German Postal Codes</i>	169
<i>Example 2: US Counties</i>	175

Overview of Custom Polygon Data

To add custom polygon data into an existing SAS Visual Analytics 7.4 environment, there are certain requirements. By default, SAS Visual Analytics supports country- and state-level polygons for regional overlays in geographic visualizations. Data for these levels is provided as part of the included GfK license.

The necessary steps to access your own custom polygons in SAS Visual Analytics are provided. Polygon data is usually provided in shape files and needs to be imported into SAS.

Examples are provided for US counties and for German provinces, but the information applies to all polygon sets for which the correct data structure is provided. Some steps might need modification depending on the environment, file types, or formats used.

CAUTION! Steps to modify an existing SAS configuration and related data files are included. Be sure to back up the current SAS environment before loading polygon data.

Prerequisites for Adding Custom Polygon Data

General Prerequisites

Before you can add custom polygon data to SAS Visual Analytics, you must meet the following prerequisites:

- access to a SAS programming interface
- polygon data in Esri shapefile file format
- a license for SAS/GRAPH software
- access to the SAS Visual Analytics configuration directory
- administrative rights to restart services

You must also add a MAPSCSTM library to your SAS Web Application Server's autoexec file. See ["MAPSCSTM Library" on page 167](#).

Considerations for Non-English Character Data

If your custom polygon data contains non-English characters, then the ENCODING= system option for your SAS Workspace Server and the encoding of the SAS LASR Analytic Server must be configured for an encoding that is compatible with the custom polygon data. It is recommended that you configure your SAS Workspace Server and SAS LASR Analytic Server for UTF-8 encoding.

Note: As a workaround, you can transcode your custom polygon data by using the CORRECTENCODING= option for the DATASETS procedure. See ["CORRECTENCODING= Option" in SAS National Language Support \(NLS\): Reference Guide](#).

Geographic Data Sets

SAS Visual Analytics generates its polygons based on a series of rules and data contained within separate data sets. There are two important data sets that you will need to modify.

ATTRLOOKUP Data Set

The ATTRLOOKUP data set contains information about the individual regions and their polygons. It has 13 variables that, combined, provide information about the identifiers of the region. As part of the custom polygon registration, you add additional rows to this data set identifying the new custom regions.

By default, the ATTRLOOKUP data set is located in `SAS-config-dir/Lev1/SASApp/Data/valib`.

IDLABEL

specifies a unique name for the region. When registering custom regions, you need to ensure that the values are unique throughout the custom region. This might mean that you have to concatenate higher levels. For example, registering a list of US counties would require you to include the related US state in the name because there are county names that exist in multiple states.

Note: The values for IDLABEL cannot contain special characters.

ID

specifies the identifier code for the region. For the original polygons available in SAS Visual Analytics, it is a concatenation of the values of the two-letter country identifier (for example: AU for Australia, US for United States, UK for United Kingdom), and a numeric count indicating which region (for example, New South Wales = AU-1, Victoria = AU-2). This is based on level 1 values. Level values are explained below.

Note: The values for ID must be unique throughout the ATTRLOOKUP data set.

Note: The values for ID cannot contain single quotation or double quotation marks.

IDNAME

specifies a unique name for the region. For custom polygons, the IDNAME and IDLABEL values are identical.

Note: The values for IDNAME cannot contain special characters.

ID1NAME

identifies the IDNAME value of the region that a subregion belongs to. For example, the ID1NAME of North Carolina is United States and the ID1NAME of London is United Kingdom.

Note: The values of ID1NAME must be unique throughout the custom region and cannot contain special characters.

ID2NAME

identifies the region that level 2 subdivisions belong to. This variable is not used in SAS Visual Analytics 7.4.

ID3NAME

identifies the region that level 3 subdivisions belong to. This variable is not used in SAS Visual Analytics 7.4.

ISO

specifies the ISO international country code standards for the default values. The numbers are determined for each country by the ISO. The list of current country ISO values used by SAS Visual Analytics can be found at <http://support.sas.com/va73geo>. For custom polygons, this value should be an unused value within the map file.

ISONAME

specifies the ISO standard name for the default values. For custom polygons, the value must be less than 55 characters and must be an uppercase value.

key

specifies a lookup key. The value is a concatenation of the IDNAME and ID1NAME variables, separated by a pipe (|) character.

ID1

specifies the ID value of the region that a subregion belongs to. For example, the ID1 of North Carolina is US and the ID1 of London is UK. The ID1 value must be unique throughout the ATTRLOOKUP data set.

ID2

specifies the ID value of the region that a level 2 subregion belongs to. This variable is not used in SAS Visual Analytics 7.4.

ID3

specifies the ID value of the region that a level 3 subregion belongs to. This variable is not used in SAS Visual Analytics 7.4.

level

specifies the level of the region. A level 0 region is a country. A level 1 region is a subdivision, such as a state or province. Level 2 and level 3 regions are not currently supported in SAS Visual Analytics.

The custom polygon deployment program creates a new region (level=0) representing the entire custom polygon area, and subdivisions (level=1) representing each individual custom region or polygon.

CENTLOOKUP Data Set

The CENTLOOKUP data set contains information about the centroids (center points) of individual regions and the map file that contains their polygon data. Information about centroids is used for the bubble plot and scatter plot geographic visualizations in SAS Visual Analytics.

By default, the CENTLOOKUP table is located in *SAS-config-dir/Lev1/SASApp/Data/valib*.

MAPNAME

specifies a SAS data set reference, including the library. This is under the standard method of LIBNAME.DATASETNAME, and is restricted by all standard SAS naming conventions.

Note: The data set name containing the polygon files must end in a “1” for SAS Visual Analytics to correspond these polygons to a level 1 variable. For example, MAPSGFK.VAOCEANIA1.

ID

specifies the short code identifier of the region. This ID value must be identical to the value in the ATTRLOOKUP table for the region. This ID value will also be in the map.

X

specifies the longitude coordinate for the center point of the region.

Y

specifies the latitude coordinate for the center point of the region.

Custom Polygon Data Set

The custom polygon data set needs to contain the necessary information for SAS Visual Analytics to plot a polygon based on the relevant ID variable. Each variable in the data set corresponds to a single point in the polygon. SAS Visual Analytics connects each of the points corresponding to the ID variable to create a polygon based on this file.

ID

specifies the short coded identifier value for the region. The ID value must be identical to the value in the ATTRLOOKUP and CENTLOOKUP data sets for the region.

SEGMENT

specifies a numeric value to identify segments of the polygons. This is used to classify areas that are separated from the rest of the region that they belong to. For example, state polygons that contain multiple islands would need to specify a different segment to prevent SAS Visual Analytics from drawing a connection between the mainland and the island.

IDNAME

specifies a unique name for each region. It must be unique within the map file and identical to the corresponding IDNAME and IDLABEL values in the ATTRLOOKUP data set.

LONG

contains the unprojected longitude value for the point specified in this region. SAS Visual Analytics uses this value for its polygons. Therefore, this value is required to be present and accurate in the final file.

LAT

contains the unprojected latitude value for the point specified in this region. SAS Visual Analytics uses this value for its polygons. Therefore, this value is required to be present and accurate in the final file.

X

contains the projected longitude Albers value for the point specified in this region. SAS Visual Analytics uses the unprojected longitude values for its polygons. Therefore, this value is not required to be accurate for the purposes of custom polygons.

Y

contains the projected latitude Albers value for the point specified in this region. SAS Visual Analytics uses the unprojected latitude values for its polygons. Therefore, this value is not required to be accurate for the purposes of custom polygons.

ISO

specifies the ISO international country code standards for the default values. The numbers are determined for each country by the ISO. The list of current country ISO values used by SAS Visual Analytics can be found at <http://support.sas.com/va73geo>. For custom polygons, this value should be an unused value within the map file.

DENSITY

refers to the values output from running the map file through the GREduce procedure. PROC GREduce calculates the density of the points on the outline of the polygon. The higher the density, the higher the number, and the more detailed the polygon. For the purpose of visualizations in SAS Visual Analytics, high-density polygons are unnecessary. Therefore, you should limit the density of the map file to densities of less than 2. For more information, see [“GREduce Procedure” in SAS/GRAPH: Mapping Reference](#).

RESOLUTION

specifies the level of map detail of the map file. This variable is currently unused by the polygon generation process.

LAKE

specifies whether this section of the polygon is a body of water. This variable is currently unused by the polygon generation process.

ISOALPHA2

specifies the identifier of the higher level associated with this polygon. For example, the ISO alpha-2 of the New South Wales polygon is AU and the ISO alpha-2 of London is UK. This variable should match the corresponding ID1 in the ATTRLOOKUP data set.

Admintype

specifies the type of polygon for SAS Visual Analytics, whether it corresponds to a state or province or to a country. For custom polygons, the Admintype Region is sufficient.

Creating Custom Polygons

Overview of Creating Custom Polygons

Deploying custom polygons in SAS Visual Analytics involves a series of steps. A provided SAS program performs most of the steps, and only minor modifications are required in some circumstances. You might need to modify the code depending on the polygon data set used and the environment in which the code is executed.

Once the registration is done, the SAS Web Application Server requires a restart before the custom polygons are accessible in SAS Visual Analytics.

Note: Prior to any processing, you should create a backup of the ATTRLOOKUP and CENTLOOKUP tables. In the event of any unexpected results, you will then be able to restore the original tables.

Perform the following high-level steps to create custom polygons:

- 1 Create the library that contains your custom polygon data sets. See [“MAPSCSTM Library” on page 167](#).
- 2 Obtain the polygon data that you want to add. The data must be in Esri shapefile file format.
- 3 Download the custom polygon deployment program at support.sas.com/rnd/datavisualization/vageo/va74polygons.sas.
- 4 In the custom polygon deployment program, define the library references and macro variables for your custom polygons.
- 5 If necessary, modify the PROC SQL statement in step 2 of the custom polygon deployment program to ensure that the lookup values are unique.
- 6 Run the custom polygon deployment program.
- 7 Restart the web application server.

MAPSCSTM Library

Before you can create custom polygons, you must define a MAPSCSTM library to contain the custom polygon data sets. The recommended location for this library is at `SASHOME/SASFoundation/9.4/mapscstm` or within the same directory location as the default MAPSGFK library. Include a LIBNAME statement in your `appserver_autoexec_usermods.sas` file, located by default at `config-dir/Lev1/SASApp` that references this directory. If your SAS Visual Analytics deployment runs on multiple server contexts, then add the LIBNAME statement to the file for each server context.

After you have modified your autoexec file, restart the SAS Object Spawner.

Note: You must have Write permissions for the directory that contains the MAPSCSTM library to create the output polygon data sets.

Custom Polygon Deployment Program

Custom polygons can be deployed to SAS Visual Analytics by using a SAS program that is available at support.sas.com/rnd/datavisualization/vageo/va74polygons.sas. The example programs in this chapter also feature the custom polygon deployment program.

The first section of the SAS program sets the values of macro variables that specify the inputs and outputs of the program.

Specify the following macro variables:

REGION_SHP_FILE

specifies the name of the shapefile files on the server. PROC MAPIMPORT is later used to import the file into SAS.

REGION_SHP_ID_COL

specifies the column in the polygon data set that represents the unique ID of each polygon.

REGION_SHP_NAME_COL

specifies the column in the polygon data set that represents the name of each polygon.

REGION_SHP_DENSITY

specifies the coordinate density for the output polygon data. Specify a value between 1 (low) and 10 (high). Typically, a value of 5 or less is recommended for standard reporting. Higher density values generally decrease report-rendering performance and require more browser memory.

REGION_LABEL

specifies the label for the set of custom polygons that are being added. The label is used in SAS Visual Analytics to identify these custom polygons.

REGION_PREFIX

specifies a two-letter identifier for your custom polygon that is not previously used in the ATTRLOOKUP data set. If the identifier value is already in use, then the deployment program displays an error message.

REGION_ISO

specifies any numeric identifier, between 0 and 999 (inclusive) for your custom polygon that is not previously used in the ATTRLOOKUP data set. If

the identifier value is already in use, then the deployment program displays an error message.

REGION_DATASET

specifies the location and name for the output polygon data set.

TIP To read the contents of an Esri shapefile without ArcGIS software, use the MAPIMPORT procedure to read the shapefile into SAS, and then use the CONTENTS or PRINT procedure to view the data. For details, see “MAPIMPORT Procedure” in *SAS/GRAPH: Mapping Reference*.

Note: The MAPSCSTM library used for your polygon data set needs to be manually created. For more information, see “MAPSCSTM Library” on page 167.

You must also ensure that the values for IDLABEL (IDNAME) are unique. In step 2 of the deployment program, edit the PROC SQL statement if needed to make the values unique. For example, step 2 in “Example 2: US Counties” on page 175 concatenates two columns from the shapefile to create unique values.

Output Data Sets

The successful completion of the program creates two data sets in the MAPSCSTM library and appends the necessary records to the ATTRLOOKUP and CENTLOOKUP tables.

The primary data set is specified by the REGION_DATASET macro variable in the polygon deployment program. For example, the primary data set from Example 1 is MAPSCSTM.DE_ZIP1. The primary data set contains all of the necessary ID variables and coordinate values that are necessary to generate the custom polygons.

The validation data set has the same name as the primary data set, with the string `_VALIDATE` appended to the end of the data set name. For example, the validation data set from Example 1 is MAPSCSTM.DE_ZIP1_VALIDATE. The validation data set contains a single observation for each polygon. It contains a value corresponding to the ID and IDNAME variable in the primary data set, as well as a randomly generated measure value. The validation data set can be used in SAS Visual Analytics to test and validate the generation of each individual polygon and the custom polygon data as a whole. See “Testing Custom Polygons in SAS Visual Analytics” on page 168.

Testing Custom Polygons in SAS Visual Analytics

Once the program has been successfully run, and the web application server has been restarted, SAS Visual Analytics will have access to these new polygons. The program will have output a data set in the MAPSCSTM library that contains a single record for each region generated by the program. The following steps will enable the testing of individual or multiple polygons based on assigned filters.

- 1 Sign in to SAS Visual Analytics as a user with exploration and data upload capabilities.

- 2 Create a new exploration.
- 3 Select a data source. Select **Import Data** ► **Server** ► **SAS Data Set**
- 4 Navigate to the MAPSCSTM location, and select the file that has the polygon name that you want to test (for example, US_COUNTIES1_VALIDATE.sas7bdat).
- 5 In the **Data** pane, right-click ID, and then select **Geography** ► **Subdivision (State, Province) SAS Map ID Values**. More information about geographic data items can be found in the online documentation for SAS Visual Analytics.
- 6 In the **Data** pane, right-click NAME, and then select **Geography** ► **Subdivision (State, Province) Names**. From the **Country or Region** drop-down list, select the region label that you specified in the custom polygon deployment program.
- 7 Create a new geo map visualization, and then assign either the NAME data item or the ID data item. On the **Roles** tab, change the **Map Style** to **Regions** and assign a measure to the **Color** role.
- 8 The geo map visualization should now display the custom polygons. Some polygons might not be displayed if the polygons are too close. If there are too many polygons, then an error message might appear.
- 9 On the **Filters** tab, add the ID or NAME data items as filters to inspect the custom polygons more closely.

Example 1: Deploying German Postal Codes

The following code uses macro variables to read and register the custom polygons. The following example uses polygon data obtained from <https://www.suche-postleitzahl.org/downloads> to render German postal code polygons. LIBNAMES and macro variables might need to be modified depending on your environment or preferences.

Note: Only the highlighted portions of the code should be changed. Be sure to change the highlighted code in Step 2.

```
libname MAPSCSTM "C:\Data\mapscstm";
libname VALIB "C:\SAS\VA\Lev1\SASApp\Data\valib";

%let REGION_LABEL=DE ZIP Codes; /* The label for the custom region */
%let REGION_PREFIX=G1; /* unique ISO 2-Letter Code - make sure it doesn't conflict with any code! */
%let REGION_ISO=001; /* unique ISO Code - make sure it doesn't conflict with any other code! */
%let REGION_DATASET=MAPSCSTM.DE_ZIP1; /* Name of the polygon data set to be created - be sure
to use suffix "1" */
%let REGION_SHP_FILE=C:\shp\Germany\plz-gebiete.shp; /* From
https://www.suche-postleitzahl.org/downloads */
%let REGION_SHP_ID_COL=plz; /* the column representing the unique ID of a polygon */
%let REGION_SHP_NAME_COL=name; /* the column representing the name of the polygon */
%let REGION_SHP_DENSITY = 3; /* Value between 1 (low) - 10 (high) describing the detail level/density. */
```

```

/* Utility macro to stop processing in case of any validation errors */
%let ERROR_COUNT=0; /* 0 indicates no error */
%let ERROR_MESSAGE=;

%MACRO ErrorTest;
  %GLOBAL ERROR_COUNT;

  %IF &ERROR_COUNT gt 0 %THEN
    %DO;
      %PUT ERROR: &ERROR_MESSAGE;

      %abort cancel;
      %let ERROR_COUNT=0; /* reset */
    %END;
  %MEND;

/* [BACKUP] Create a backup of the original lookup tables if required */
%macro backupLookupTable(dsn);
  %if %sysfunc(exist(&dsn._original)) %then
    %do;
      %put INFO: Backup table &dsn found.;
    %end;
  %else
    %do;
      data &dsn._original;
        set &dsn;
      run;
    %end;
  %mend backupLookupTable;

%backupLookupTable(valib.attrlookup);
%backupLookupTable(valib.centlookup);

/* [VALIDATION STEP] Check for unique 2-letter ISO code */
proc sql noprint;
  /* count and create error code if greater than zero */
  select count(ID) as count,
    case
      when calculated count > 0 then "2-letter ISO code &REGION_PREFIX. is not unique in ATTRLOOKUP. Please specify a different code."
      else ""
    end
  into :ERROR_COUNT, :ERROR_MESSAGE
  from valib.attrlookup_original
  where upcase(ID) eq upcase("&REGION_PREFIX.");
quit;

%ErrorTest;

/* [VALIDATION STEP] Check for unique ISO code */
proc sql noprint;
  /* count and create error code if greater than zero */
  select count(ISO) as count,
    case
      when calculated count > 0 then "ISO code &REGION_ISO. is not unique in ATTRLOOKUP. Please specify a different ISO code."
      else ""
    end

```

```

into :ERROR_COUNT, :ERROR_MESSAGE
  from valib.attrlookup_original
  where ISO eq "&REGION_ISO.";
quit;

%ErrorTest;

/* Remove any previous entries if required */
proc sql;
  delete from valib.attrlookup where ID like "&REGION_PREFIX.%";
  delete from valib.centlookup where ID like "&REGION_PREFIX.%";
quit;

/* Step 1: Import the SHP file into SAS */
PROC MAPIMPORT DATAFILE="&REGION_SHP_FILE."
  OUT=&REGION_DATASET.;
  ID &REGION_SHP_ID_COL.;
RUN;

/* Add the DENSITY column to the polygon data set used to reduce the polygon density in the visualization */
proc greduce data=&REGION_DATASET. out=&REGION_DATASET.;
  id &REGION_SHP_ID_COL.;
run;

/* Determine the column type of REGION_SHP_ID_COL */
data _null_;
  set &REGION_DATASET.(obs=1);
  call symputx('REGION_SHP_ID_COL_TYPE', vtype(&REGION_SHP_ID_COL.));
run;

/* Step 2: Region lookup can either be via IDLABEL (IDNAME) or ID. In case we lookup by IDLABEL let's generate a
  unique name. The following steps might need to be adjusted depending on your source data.
*/
proc sql;
  create table work.idname_fmt as
  select distinct
    "idname" as FMTNAME,
    "&REGION_SHP_ID_COL_TYPE." as TYPE,
    &REGION_SHP_ID_COL. as START,
    trim(left(&REGION_SHP_NAME_COL.)) as LABEL
  from &REGION_DATASET.;
quit;

/* [VALIDATION STEP] Check for duplicate names in the IDNAME lookup data set. With duplicate entries here the idname based lookup would not work.
If you have names listed here, revisit the previous step to make sure each LABEL is unique, for example, by merging other parent level details */
proc sql noprint;
  create table work.nonunique_names
  as select distinct LABEL, count(LABEL) as count
  from work.idname_fmt
  group by LABEL
  having calculated count > 1
  order by count desc;

/* count and create error code if greater than zero */
select count(*) as count,

```

```

case
  when calculated count > 0 then compress(put(calculated count,8.))
  || " duplicate names have been found. Please check the idname_fmt table for duplicates."
  || " A list of duplicate names are in WORK.NONUNIQUE_NAMES."
  else ""
end
into :ERROR_COUNT, :ERROR_MESSAGE
  from work.nonunique_names;
quit;

%ErrorTest;

proc format cntlin=work.idname_fmt;
run;

/* [VALIDATION STEP] Check for required columns in the input polygon data set. */
data _null_;
  dsid = open("&REGION_DATASET.");
  array colNames(5) $50 _temporary_ ("&REGION_SHP_ID_COL.", "X", "Y", "SEGMENT", "DENSITY");

  do i = lbound(colNames) to hbound(colNames);
    check = varnum(dsid, colNames(i));

    if check eq 0 then
      do;
        call symputx('ERROR_COUNT', 1);
        call symputx('ERROR_MESSAGE',"Column " || compress(colNames(i)) || " is required in &REGION_DATASET.");
        stop;
      end;
    end;
  end;
run;

%ErrorTest;

/* [VALIDATION STEP] Check for conflicting columns in the input polygon data set. */
data _null_;
  array colNames(7) $50 _temporary_ ("IDNAME", "LONG", "LAT", "RESOLUTION", "LAKE", "ISOALPHA2", "AdminType");

  do i = lbound(colNames) to hbound(colNames);
    if colNames(i) eq "&REGION_SHP_ID_COL." then
      do;
        call symputx('ERROR_COUNT', 1);
        call symputx('ERROR_MESSAGE',"Column name for REGION_SHP_ID_COL conflicts with predefined column " || compress(colNames(i)) || ".");
        stop;
      end;
    end;
  end;
run;

%ErrorTest;

/* Step 3: Create the final polygon data set with the required set of columns. We are going to create a new
shorter ID column here as the original ID columns are often long and unnecessarily complex. Note, we also apply
the previously created idname format to the IDNAME column.
*/
data &REGION_DATASET.;
  set &REGION_DATASET.(keep=&REGION_SHP_ID_COL. X Y SEGMENT DENSITY);

```

```

/* create new (shorter) ID */
length _TMP_ $60. _SID_ 8.;

if (_n_ eq 1) then
  do;
    _SID_ = 1;
    _TMP_ = &REGION_SHP_ID_COL.;
  end;

if _TMP_ ne &REGION_SHP_ID_COL. then
  _SID_ = _SID_ + 1;
  _TMP_ = &REGION_SHP_ID_COL.;
retain _TMP_ _SID_;
drop _TMP_ _SID_;
_RID_ = compress("&REGION_PREFIX.-" || put(_SID_,8.));
IDNAME = put(&REGION_SHP_ID_COL., idname.);
LONG = X;
LAT = Y;
ISO = "&REGION_ISO.";
RESOLUTION = 1;
LAKE = 0;
ISOALPHA2 = "&REGION_PREFIX.";
AdminType = "regions";
where density<&REGION_SHP_DENSITY.;
keep _RID_ SEGMENT IDNAME LONG LAT X Y ISO DENSITY RESOLUTION LAKE ISOALPHA2 AdminType;
rename _RID_=ID;

run;

/* Step 4: Add a custom region (country) to ATTRLOOKUP - a new entry will appear in the country selection list */
proc sql;
  insert into valib.attrlookup
  values (
    "&REGION_LABEL.",          /* IDLABEL=State/Province Label */
    "&REGION_PREFIX.",        /* ID=SAS Map ID Value */
    "&REGION_LABEL.",        /* IDNAME=State/Province Name */
    "",                        /* ID1NAME=Country Name */
    "",                        /* ID2NAME */
    "&REGION_ISO.",          /* ISO=Country ISO Numeric Code */
    "&REGION_LABEL.",        /* ISONAME */
    "&REGION_LABEL.",        /* KEY */
    "",                        /* ID1=Country ISO 2-Letter Code */
    "",                        /* ID2 */
    "",                        /* ID3 */
    "",                        /* ID3NAME */
    0                          /* LEVEL (0=country level, 1=state level) */
  );

quit;

/* Register each single region in the lookup data set */
proc sql;
  insert into valib.attrlookup
  select distinct
    IDNAME,                    /* IDLABEL=State/Province Label */
    ID,                        /* ID=SAS Map ID Value */
    IDNAME,                    /* IDNAME=State/Province Name */

```

```

"&REGION_LABEL.",          /* ID1NAME=Country Name */
"",                        /* ID2NAME */
"&REGION_ISO.",          /* ISO=Country ISO Numeric Code */
"&REGION_LABEL.",        /* ISONAME */
trim(IDNAME) || "&REGION_LABEL.", /* KEY */

"&REGION_PREFIX.",       /* ID1=Country ISO 2-Letter Code */
"",                      /* ID2 */
"",                      /* ID3 */
"",                      /* ID3NAME */
1                        /* LEVEL (1=state level) */
from &REGION_DATASET.;
quit;

/* Register custom region in CENTLOOKUP */
proc sql;
/* Add custom region */
insert into valib.centlookup
select distinct
"&REGION_DATASET." as mapname,
"&REGION_PREFIX." as ID,
avg(x) as x,
avg(y) as y
from &REGION_DATASET.;

/* Add custom provinces */
insert into valib.centlookup
select distinct
"&REGION_DATASET." as mapname,
ID as ID,
avg(x) as x,
avg(y) as y
from &REGION_DATASET.
group by id;
quit;

/* Step 5: (optional) Creates a validation data set which can be used to validate the newly added regions in SAS Visual Analytics */
proc sql;
create table &REGION_DATASET._VALIDATE as
select distinct ID as ID,
IDNAME as NAME
from &REGION_DATASET.;
create table &REGION_DATASET._VALIDATE as
select *,
round(ranuni(1) * 10000) as measure1,
round(ranuni(1) * 100000) as measure2 format=dollar20.0
from &REGION_DATASET._VALIDATE
group by ID, NAME
order by ID, NAME;
quit;

```


Example 2: US Counties

The following code uses macro variables to read and register the custom polygons. The following example uses polygon data obtained from https://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html to render US counties. LIBNAMES and macro variables might need to be modified depending on your environment or preferences.

Note: Only the highlighted portions of the code should be changed. Be sure to change the highlighted code in Step 2.

```
libname MAPSCSTM "C:\Data\mapscstm";
libname VALIB "C:\SAS\VA\Lev1\SASApp\Data\valib";

%let REGION_LABEL=US Counties; /* The label for the custom region */
%let REGION_PREFIX=CS; /* unique ISO 2-Letter Code - make sure it doesn't conflict with any code! */
%let REGION_ISO=000; /* unique ISO Code - make sure it doesn't conflict with any other code! */
%let REGION_DATASET=MAPSCSTM.US_COUNTIES1; /* Name of the polygon data set to be created - be sure
to use suffix "1" */
%let REGION_SHP_FILE=C:\shp\US\cb_2015_us_county_500k.shp; /* From
https://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html */
%let REGION_SHP_ID_COL=GEOID; /* the column representing the unique ID of a polygon */
%let REGION_SHP_NAME_COL=NAME; /* the column representing the name of the polygon */
%let REGION_SHP_DENSITY = 3; /* Value between 1 (low) - 10 (high) describing the detail level/density. */

/* Utility macro to stop processing in case of any validation errors */
%let ERROR_COUNT=0; /* 0 indicates no error */
%let ERROR_MESSAGE=;

%MACRO ErrorTest;
  %GLOBAL ERROR_COUNT;

  %IF &ERROR_COUNT gt 0 %THEN
    %DO;
      %PUT ERROR: &ERROR_MESSAGE;
      %abort cancel;
      %let ERROR_COUNT=0; /* reset */
    %END;
%MEND;

/* [BACKUP] Create a backup of the original lookup tables if required */
%macro backupLookupTable(dsn);
  %if %sysfunc(exist(&dsn._original)) %then
    %do;
      %put INFO: Backup table &dsn found.;
    %end;
  %else
    %do;
      data &dsn._original;
        set &dsn;
      run;
    %end;
%mend backupLookupTable;
```

```

%backupLookupTable(valib.attrlookup);
%backupLookupTable(valib.centlookup);

/* [VALIDATION STEP] Check for unique 2-letter ISO code */
proc sql noprint;
  /* count and create error code if greater than zero */
  select count(ID) as count,
         case
           when calculated count > 0 then "2-letter ISO code &REGION_PREFIX. is not unique in ATTRLOOKUP. Please specify a different code."
           else ""
         end
  into :ERROR_COUNT, :ERROR_MESSAGE
  from valib.attrlookup_original
  where upcase(ID) eq upcase("&REGION_PREFIX.");
quit;

%ErrorTest;

/* [VALIDATION STEP] Check for unique ISO code */
proc sql noprint;
  /* count and create error code if greater than zero */
  select count(ISO) as count,
         case
           when calculated count > 0 then "ISO code &REGION_ISO. is not unique in ATTRLOOKUP. Please specify a different ISO code."
           else ""
         end
  into :ERROR_COUNT, :ERROR_MESSAGE
  from valib.attrlookup_original
  where ISO eq "&REGION_ISO.";
quit;

%ErrorTest;

/* Remove any previous entries if required */
proc sql;
  delete from valib.attrlookup where ID like "&REGION_PREFIX.%";
  delete from valib.centlookup where ID like "&REGION_PREFIX.%";
quit;

/* Step 1: Import the SHP file into SAS */
PROC MAPIIMPORT DATAFILE="&REGION_SHP_FILE."
  OUT=&REGION_DATASET.;
  ID &REGION_SHP_ID_COL.;
RUN;

/* Add the DENSITY column to the polygon data set used to reduce the polygon density in the visualization */
proc greduce data=&REGION_DATASET. out=&REGION_DATASET.;
  id &REGION_SHP_ID_COL.;
run;

/* Determine the column type of REGION_SHP_ID_COL */
data _null_;
  set &REGION_DATASET.(obs=1);
  call symputx('REGION_SHP_ID_COL_TYPE', vtype(&REGION_SHP_ID_COL.));
run;

```

/* Step 2: Region lookup can either be via IDLABEL (IDNAME) or ID. In case we lookup by IDLABEL let's generate a unique name. The following steps need to be adjusted depending on your source data.

As there are multiple counties across the US with the same name - the code snippet below is adjusted to include the county ID. This combination of name and id makes the IDNAME unique across the regions.

```
*/
proc sql;
  create table work.idname_fmt as
  select distinct
    "idname" as FMTNAME,
    "&REGION_SHP_ID_COL_TYPE." as TYPE,
    &REGION_SHP_ID_COL. as START,
    trim(left(&REGION_SHP_NAME_COL.)) || " (" || trim(left(&REGION_SHP_ID_COL.)) || ")" as LABEL
  from &REGION_DATASET.;
quit;
```

/* [VALIDATION STEP] Check for duplicate names in the IDNAME lookup data set. With duplicate entries here the idname based lookup would not work. If you have names listed here, revisit the previous step to make sure each LABEL is unique, for example, by merging other parent level details */

```
proc sql noprint;
```

```
  create table work.nonunique_names
  as select distinct LABEL, count(LABEL) as count
  from work.idname_fmt
  group by LABEL
  having calculated count > 1
  order by count desc;
```

/* count and create error code if greater than zero */

```
select count(*) as count,
  case
    when calculated count > 0 then compress(put(calculated count,8.))
    || " duplicate names have been found. Please check the idname_fmt table for duplicates."
    || " A list of duplicate names are in WORK.NONUNIQUE_NAMES."
  else ""
  end
into :ERROR_COUNT, :ERROR_MESSAGE
from work.nonunique_names;
quit;
```

```
%ErrorTest;
```

```
proc format cntlin=work.idname_fmt;
run;
```

/* [VALIDATION STEP] Check for required columns in the input polygon data set. */

```
data _null_;
  dsid = open("&REGION_DATASET.");
  array colNames(5) $50 _temporary_ ("&REGION_SHP_ID_COL.", "X", "Y", "SEGMENT", "DENSITY");

  do i = lbound(colNames) to hbound(colNames);
    check = varnum(dsid, colNames(i));

    if check eq 0 then
      do;
        call symputx('ERROR_COUNT', 1);
      end;
  end;
```

```

        call symputx('ERROR_MESSAGE','Column " || compress(colNames(i)) || " is required in &REGION_DATASET. ');
        stop;
    end;
end;
run;

%ErrorTest;

/* [VALIDATION STEP] Check for conflicting columns in the input polygon data set. */
data _null_;
    array colNames(7) $50 _temporary_ ("IDNAME", "LONG", "LAT", "RESOLUTION", "LAKE", "ISOALPHA2", "AdminType");

    do i = lbound(colNames) to hbound(colNames);
        if colNames(i) eq "&REGION_SHP_ID_COL." then
            do;
                call symputx('ERROR_COUNT', 1);
                call symputx('ERROR_MESSAGE','Column name for REGION_SHP_ID_COL conflicts with predefined column " || compress(colNames(i)) || ". ');
                stop;
            end;
        end;
    end;
run;

%ErrorTest;

/* Step 3: Create the final polygon data set with the required set of columns. We are going to create a new
shorter ID column here as the original ID columns are often long and unnecessarily complex. Note, we also apply
the previously created idname format to the IDNAME column.
*/
data &REGION_DATASET.;
    set &REGION_DATASET.(keep=&REGION_SHP_ID_COL. X Y SEGMENT DENSITY);

    /* create new (shorter) ID */
    length _TMP_ $60. _SID_ 8.;

    if (_n_ eq 1) then
        do;
            _SID_ = 1;
            _TMP_ = &REGION_SHP_ID_COL.;
        end;

    if _TMP_ ne &REGION_SHP_ID_COL. then
        _SID_ = _SID_ + 1;
        _TMP_ = &REGION_SHP_ID_COL.;
    retain _TMP_ _SID_;
    drop _TMP_ _SID_;
    _RID_ = compress("&REGION_PREFIX.-" || put(_SID_,8.));
    IDNAME = put(&REGION_SHP_ID_COL., idname.);
    LONG = X;
    LAT = Y;
    ISO = "&REGION_ISO.";
    RESOLUTION = 1;
    LAKE = 0;
    ISOALPHA2 = "&REGION_PREFIX.";
    AdminType = "regions";
    where density<&REGION_SHP_DENSITY.;
    keep _RID_ SEGMENT IDNAME LONG LAT X Y ISO DENSITY RESOLUTION LAKE ISOALPHA2 AdminType;

```

```

rename _RID_=ID;
run;

/* Step 4: Add a custom region (country) to ATTRLOOKUP - a new entry will appear in the country selection list */
proc sql;
  insert into valib.attrlookup
  values (
    "&REGION_LABEL.",          /* IDLABEL=State/Province Label */
    "&REGION_PREFIX.",        /* ID=SAS Map ID Value */
    "&REGION_LABEL.",          /* IDNAME=State/Province Name */
    "",                        /* ID1NAME=Country Name */
    "",                        /* ID2NAME */
    "&REGION_ISO.",           /* ISO=Country ISO Numeric Code */
    "&REGION_LABEL.",          /* ISONAME */
    "&REGION_LABEL.",          /* KEY */
    "",                        /* ID1=Country ISO 2-Letter Code */
    "",                        /* ID2 */
    "",                        /* ID3 */
    "",                        /* ID3NAME */
    0                          /* LEVEL (0=country level, 1=state level) */
  );
quit;

/* Register each single region in the lookup data set */
proc sql;
  insert into valib.attrlookup
  select distinct
    IDNAME,                    /* IDLABEL=State/Province Label */
    ID,                        /* ID=SAS Map ID Value */
    IDNAME,                    /* IDNAME=State/Province Name */

    "&REGION_LABEL.",          /* ID1NAME=Country Name */
    "",                        /* ID2NAME */
    "&REGION_ISO.",           /* ISO=Country ISO Numeric Code */
    "&REGION_LABEL.",          /* ISONAME */
    trim(IDNAME) || "&REGION_LABEL.", /* KEY */

    "&REGION_PREFIX.",        /* ID1=Country ISO 2-Letter Code */
    "",                        /* ID2 */
    "",                        /* ID3 */
    "",                        /* ID3NAME */
    1                          /* LEVEL (1=state level) */
  from &REGION_DATASET.;
quit;

/* Register custom region in CENTLOOKUP */
proc sql;
  /* Add custom region */
  insert into valib.centlookup
  select distinct
    "&REGION_DATASET." as mapname,
    "&REGION_PREFIX." as ID,
    avg(x) as x,
    avg(y) as y
  from &REGION_DATASET.;

```

```
/* Add custom provinces */
insert into valib.centlookup
  select distinct
    "&REGION_DATASET." as mapname,
    ID as ID,
    avg(x) as x,
    avg(y) as y
  from &REGION_DATASET.
  group by id;
quit;

/* Step 5: (optional) Creates a validation data set which can be used to validate the newly added regions in SAS Visual Analytics */
proc sql;
  create table &REGION_DATASET._VALIDATE as
    select distinct ID as ID,
      IDNAME as NAME
    from &REGION_DATASET.;
  create table &REGION_DATASET._VALIDATE as
    select *,
      round(ranuni(1) * 10000) as measure1,
      round(ranuni(1) * 100000) as measure2 format=dollar20.0
    from &REGION_DATASET._VALIDATE
    group by ID, NAME
    order by ID, NAME;
quit;
```

Appendix 3

Troubleshooting

Troubleshooting: SAS Visual Analytics	181
Access Issues	181
Server Operation Issues	183
Load, Reload, and Import Issues	185
Troubleshooting: SAS Mobile BI	187
Permission Origins	188
Introduction	188
Simple Permission Origins	188
Inherited Permission Origins	189

Troubleshooting: SAS Visual Analytics

Access Issues

Issue: Inability to sign in.

Resolution:

- If the error message is `Public access denied`, make sure that the user has a well-formed definition in metadata. In a user's metadata definition, this problem can be caused by a user ID that is not in a qualified format. This problem is not caused by passwords or authentication domain assignments on a user's **Accounts** tab. See ["Adding Users" on page 3](#).
- Make sure that the metadata server and the middle tier are running. See ["Operate Other Servers" on page 8](#).

Issue: Missing or inaccessible applications or features.

Resolution:

- Make sure that each user's memberships provide the appropriate capabilities. See ["Roles and Capabilities" on page 133](#).
- Make sure that users are not inadvertently connecting as guests. See ["Supporting Guest Access" on page 73](#).
- Make sure that the appropriate applications are licensed and installed.


Issue: Users cannot access any LASR tables in the explorer or the designer.

Resolution:

- Make sure that the SAS LASR Analytic Server is running and that tables are loaded.
- In metadata, make sure that the LASR authorization service is enabled. On the **Options** tab in the Properties window for the SAS LASR Analytic Server's connection object, verify that the **Use LASR authorization service** check box is selected.
- Make sure that uniqueness requirements are met. See [“In-Memory LASR Names” on page 96](#).

Issue: Users cannot access a particular LASR table.

Resolution:

- Make sure that users have the ReadMetadata and Read permissions for the LASR table. Also, make sure that the SAS Trusted User's ReadMetadata access is not blocked. See [“Permissions” on page 36](#).
- Make sure the table does not have any invalid permission conditions. On the table's **Authorization** page, look for any conditional grants . To restore access, remove any permission conditions that are no longer valid. If appropriate, set new conditions.

Note: A table that has a conditional grant becomes inaccessible if its metadata is updated with information that renders the permission condition invalid. For example, a permission condition might reference a column that is no longer part of the table.

- Make sure that each LASR table for the target LASR library has a unique name. For example, a copy-and-paste action in the data builder can result in multiple tables that have the same name within a particular library. To restore access, delete one of the tables. See [“Unload, Reload, or Delete a Table” on page 14](#).
- Make sure that concurrent user logins are allowed. See [“Policy for Concurrent User Logins” on page 61](#).
- If the table was autoloading, try again. (Data availability might be interrupted during autoloading processing.)

Issue: Inability to take a capability away from a user.

Resolution:

- Make sure that the user is not assigned to any role that provides that capability. Consider indirect and implicit memberships, as well as direct memberships. Remember that all registered users are automatically members of the PUBLIC and SASUSERS groups.
- Make sure that the user is not assigned to the **Metadata Server: Unrestricted** role.

Issue: Inability to access a third-party DBMS table.

Resolution:

- From the main menu, select **File** ► **Clear Credentials Cache**. Then, attempt access again. If you are prompted for a user ID and password, enter DBMS credentials.

- If the third-party DBMS uses proprietary authentication and is pre-assigned, you must store a DBMS user ID and password. See [How to Store Passwords for a Third-Party Server](#) in the *SAS Intelligence Platform: Security Administration Guide*.

Issue: Inability to register tables.

Resolution:

- Make sure that you have the necessary metadata-layer permissions. See [“Permissions by Task” on page 37](#).
- Make sure that you have Read access to the physical source tables (host-layer permissions).
- On Windows, make sure that the account that you are using has the **Log on as a batch job** Windows privilege. See [“Host Account Privileges” on page 5](#).
- If you are prompted for a user ID and password, enter host credentials for the workspace server.

Issue: Problem running exported code (inability to connect to the metadata server).

Resolution:

- Metadata server connection information is not included in exported code. Either supply connection information or use a SAS session that already includes connection information (for example, the SAS DATA Step Batch Server). For information about metadata server connection options, see [SAS Language Interfaces to Metadata](#).

Server Operation Issues

Issue: Inability to start a SAS LASR Analytic Server.

Resolution:

- Make sure that any host-layer requirements are met. See [“Host Account Privileges” on page 5](#).

Note: If adjustments to host-layer privileges do not have an immediate effect, try signing out, closing your browser, and then accessing SAS Visual Analytics again.
- Make sure that the server’s metadata definition is complete. In particular, valid values for the install path, signature files location, and number of machines to use are required. See [“Add a SAS LASR Analytic Server” on page 97](#).

Note: The associated error message specifies: Failed to create a LASR Analytic Server signature file .
- Make sure that each server on a particular host uses a unique port number.
- If your deployment has multiple SAS Application Servers, make sure an appropriate server is being used. See [“Using Multiple SAS Application Servers” on page 90](#).
- If the error indicates that the LASR procedure is not found, make sure that the workspace server that is being used has a valid license for SAS Visual Analytics software.

- If the error indicates that a path is not in the list of accessible paths, see [“Locked-Down Servers” on page 56](#).
- Make sure that the host name is specified correctly in the connection object for the SAS LASR Analytic Server.

Note: The associated error message specifies: You cannot combine the `STARTSERVER` or `SERVEROPTS=` syntax with the specification of a host other than the local machine.

Issue: Inability to stop a SAS LASR Analytic Server.

Resolution:

- Make sure that you have the Administer permission for the server.
- Make sure that any host-layer requirements are met. See [“Host Account Privileges” on page 5](#).
- If the error is `Procedure LASR not found`, make sure that the workspace server that is being used has a valid license for SAS Visual Analytics software.
- Make sure that concurrent user logins are allowed. See [“Policy for Concurrent User Logins” on page 61](#).
- If the SAS middle tier was recently restarted, the server can become unresponsive. See [“Non-distributed Server Dependencies” on page 94](#).

Issue: No last action log is available for a server.

Resolution:

- If no action on a server has been initiated from the **LASR Servers** tab, no last action log exists for that server.
- Not all actions generate a last action log. In most cases, `success` and `failure` results generate last action logs; `not processed` results do not.
- If the last action log file for a server has been deleted from its file system location, no last action log is available for that server. Last action logs are written to the directory that is specified by a suite-level configuration property. See [“`va.lastActionLogPath`” on page 142](#).

Issue: The SAS LASR Analytic Server Monitor graphs are blank.

Resolution:

- Make sure that the SAS LASR Analytic Server is running.
- Make sure that the SAS LASR Analytic Server is distributed. The **Monitor** tabs are not supported for non-distributed servers.
- Make sure that the TGrid location in the `service.properties` file is correct. The file is located in the [SAS configuration directory](#) under `/Applications/SASVisualAnalytics/HighPerformanceConfiguration`.

Note: Any changes that you make to the **High-Performance Analytics environment install location** field in a server definition must also be manually made in the monitoring server’s properties file.

- Restart the monitoring server. See [“Managing the Monitoring Server” on page 86](#).

Issue: On the Process Monitor tab, table details are not provided.

Resolution:

- Make sure that the middle-tier machine has the necessary network name resolution. See [“Network Name Resolution” on page 86](#).

Issue: On the LASR Servers tab, per-instance memory gauges are not available.

Resolution:

- If the **Virtual Memory** column is not displayed, you have a non-distributed server. Per-instance memory gauges are not supported for a non-distributed server.
- If the **Virtual Memory** column is empty:
 - Make sure that the middle-tier machine has the necessary network name resolution. See [“Network Name Resolution” on page 86](#).
 - Make sure that the monitoring process is running. See [“Supporting the Monitoring Features” on page 85](#).

Load, Reload, and Import Issues

Issue: Inability to load, reload, or import tables.

Resolution:

- Make sure that you can access the SAS LASR Analytic Server using an account that has the necessary privileges. See [“Host Account Privileges” on page 5](#).
- Make sure that you have the necessary metadata-layer permissions for the output folder, LASR library, and LASR table (if applicable). See [“Permissions” on page 36](#).
- For actions against an encrypted SASHDAT library, make sure that you have metadata-layer Read access to the library. See [“On-Disk Encryption of SASHDAT Files” on page 64](#).
- For actions against a library that supports reload-on-start, make sure that you have host access to the associated data provider library. See [“Reload-on-Start” on page 18](#).

Note: If the library is encrypted, make sure that you have the necessary metadata-layer permissions on the corresponding secured folder, secured library, and secured table objects. See [“On-Disk Encryption of Reload-on-Start Files” on page 61](#).

- For loads to the public area, make sure that the library, server, and folder that are referenced by the `va.publicLASRLibrary`, `va.publicLASRServer`, and `va.defaultPublicFolder` configuration properties exist. See [“Configuration Properties” on page 139](#).
- If the **OK** button in an Import window remains disabled after a user populates the required fields, and the **Advanced** panel is not displayed, make sure that the user has ReadMetadata access to the library that is specified in the `va.publicLASRLibrary` property.
- If a message indicates that a table is not reloadable, use a different technique to make the table available again. See [“Reload Methods” on page 11](#).

- Determine whether a memory limit is preventing the actions:
 - In the **Status** column on the **LASR Servers** tab, make sure the target server is not over capacity. See [“Limit Space for Tables” on page 100](#).
 - For a distributed server, make sure that total memory usage (by all processes) does not meet or exceed the configured limit. See [“Memory Limits” on page 103](#).
- Make sure that concurrent user logins are allowed. See [“Policy for Concurrent User Logins” on page 61](#).
- For co-located HDFS or NFS-mounted MapR:
 - Make sure that the source library is paired with a LASR library through a match between the source path and the server tag. For example, tables in the directory `/users/sasdemo` must be loaded to a LASR library that has `users.sasdemo` as its server tag. See [“Add a LASR Library” on page 98](#).
 - Make sure that the Hadoop server and the SAS LASR Analytic Server have identical, fully qualified host names in the **Associated Machine** field in their metadata definitions.
 - For the Hadoop server, select the **Options** tab.
 - For the SAS LASR Analytic Server, select the **Options** tab, click the **Advanced Options** button, and select the **Additional Options** tab.

Note: The message for this issue describes the target library as `unidirectional`.
- If the tables were promoted, review the considerations for exporting and importing data. See [Reference Information for Promotion Tools](#) in the *SAS Intelligence Platform: System Administration Guide*.

Issue: No last action log is available for a table.

Resolution:

- If no action on a table has been initiated from the **LASR Tables** tab, no last action log exists for that table.
- Not all actions generate a last action log. In most cases, `success` and `failure` results generate last action logs; `not processed` results do not.
- If the last action log file for a table has been deleted from its file system location, no last action log is available for that table. Last action logs are written to the directory that is specified by a suite-level configuration property. See [“va.lastActionLogPath” on page 142](#).

Issue: On the LASR Tables tab, tables are not listed.

Resolution:

- Make sure that the middle-tier machine has the necessary network name resolution. See [“Network Name Resolution” on page 86](#).
- Make sure that the filter (in the tab’s toolbar) is not hiding tables that you expect to see.

Issue: In the Load a Table window, the OK button is disabled.

Resolution:

- In the **LASR Table** section, enter a name. Click in one of the other fields in the window, and then click **OK**.

Issue: Inability to change the name of the output table when loading data from co-located HDFS or NFS-mounted MapR.

Resolution:

- Add the table to the data provider again. In that transaction, assign a different name to the output table. When you load data from co-located HDFS or NFS-mounted MapR, you cannot choose a different name for the output table. See [“Administrator Load” on page 15](#).

Issue: Autoload issues.

Resolution:

- Review the [additional considerations](#) for autoload.
- Examine the [autoload logs](#). Here are example locations:

Windows	C:\sas\Config\Lev1\Applications\SASVisualAnalytics \VisualAnalyticsAdministrator\Logs
UNIX	/opt/sas/config/Lev1/Applications/SASVisualAnalytics/ VisualAnalyticsAdministrator/Logs

- In the associated AutoLoad.sas file, make sure the AL_META_LASRLIB= value references the metadata name of the appropriate LASR library. (The AL_META_LASRLIB= value should not reference an operating system directory.)

Note: The associated error messages specify: LASR Auto Load directory validation failed and Ensure that connection to metadata server is configured properly and that the proper autoload folder structure exists.

- On Windows, make sure that you are using backward slashes (\) rather than forward slashes (/) when you specify Windows host directory locations.

Note: In the Windows task scheduler, the associated Last Run Result specifies: The system cannot move the file to a different disk drive.

Troubleshooting: SAS Mobile BI

Issue: User cannot open reports in an offline device.

Resolution:

- Make sure that the user is not in any role that provides the capability that prevents this action. See [“Purge Mobile Report Data” on page 137](#).

Issue: User is prompted for an application passcode.

Resolution:

- Make sure that the user is not in any role that provides the capability that introduces this requirement. See [“Require Passcode On Mobile Devices”](#) on page 137.

Issue: SAS Visual Analytics users cannot connect to SAS Mobile BI

- Make sure your SAS Visual Analytics license is valid and correctly applied. See [Usage Note 51856](#) at support.sas.com.

Issue: On the Mobile Devices tab, a message indicates that a list is not currently in use.

Resolution:

- By design, only one list (either the blacklist or the whitelist) is in use. See [“About Mobile Device Management”](#) on page 56.

Permission Origins

Introduction

Permission origins identify the source of each effective permission in the metadata authorization layer. This information can be useful in troubleshooting. It answers the question: Why is this identity granted (or denied) this permission?



In the origins answer, only the controlling (winning, highest precedence) access control is shown. If there are multiple tied winning controls, they are all shown. Other, lower precedence controls are not shown in the answer.










Origins information is available on an object’s **Authorization** page. See [“View Authorization Information”](#) on page 40.

Simple Permission Origins

The following table provides simple examples of permission origins answers. In each example, we are interested in why UserA has an effective grant on FolderA. In each example, UserA is a direct member of both GroupA and GroupB. Each row in the table is for a different (independent) permissions scenario. In the table, the first column depicts the contents of the Origins window. The second column interprets the information.

Table A3.1 Origins: Simple Examples

Origins Information	Source of UserA's Effective Grant on FolderA
 UserA [Explicit]	On FolderA, an explicit grant for UserA
 GroupA [Explicit]	On FolderA, an explicit grant for GroupA

Origins Information	Source of UserA's Effective Grant on FolderA
 GroupA [Explicit]  GroupB [Explicit]	On FolderA, explicit grants for GroupA and GroupB Note: Two settings are shown because they are tied and they both win (UserA is a direct member of GroupA and GroupB).
 GroupA [ACT: GroupARead]	On FolderA, an ACT pattern grant for GroupA (from a directly applied ACT)
 SASUSERS [ACT: GenRead]	On FolderA, an ACT pattern grant for SASUSERS (from a directly applied ACT)
 GroupA [ACT: GroupARead]  GroupB [ACT: GroupBRead]	On FolderA, ACT pattern grants for GroupA and GroupB (from two different directly applied ACTs) Note: Two settings are shown because they are tied and they both win (UserA is a direct member of GroupA and GroupB).
 GroupA [ACT: GroupABRead]  GroupB [ACT: GroupABRead]	On FolderA, ACT pattern grants for GroupA and GroupB (from the same directly applied ACT) Note: Two settings are shown because they are tied and they both win (UserA is a direct member of GroupA and GroupB).
 UserA is unrestricted.	UserA's status as an unrestricted user (someone who is unrestricted is always granted all permissions)


















Inherited Permission Origins

In many cases, the controlling setting is not on the current object. Instead, the controlling setting is defined on a parent object and inherited by the current object.

The following table provides examples in which the controlling setting comes from a parent object. Because the source of the effective permission is a parent object, the answer must identify which parent object has the controlling setting. For this reason, the answers in the following examples identify both a parent object (the object that has the controlling setting) and the controlling setting, itself.

In each example, we are interested in why UserA has an effective grant on FolderA. In each example, UserA is a direct member of both GroupA and GroupB. Each row in the table is for a different (independent) permissions scenario. In the table, the first column depicts the contents of the Origins window. The second column interprets the information.

Table A3.2 Origins: Inheritance Examples

Origins Information	Source of UserA's Effective Grant on FolderA
 ParentFolderA  UserA [Explicit]	On ParentFolderA, an explicit grant for UserA
 ParentFolderA  GroupA [Explicit]	On ParentFolderA, an explicit grant for GroupA
 ParentFolderA  GroupA [Explicit]  GroupB [Explicit]	On ParentFolderA, explicit grants for GroupA and GroupB
 ParentFolderA  GroupA [ACT: GroupARead]	On ParentFolderA, an ACT pattern grant for GroupA (from a directly applied ACT)
 GreatGrandParentFolderA  SASUSERS [ACT: GenRead]	On GreatGrandParentFolderA, an ACT pattern grant for SASUSERS (from a directly applied ACT)
 ParentFolderA  GroupA [ACT: GroupARead]  GroupB [ACT: GroupBRead]	On ParentFolderA, ACT pattern grants for GroupA and GroupB (from two different directly applied ACTs)
 GrandParentFolderA  GroupA [ACT: GroupABRead]  GroupB [ACT: GroupABRead]	On GrandParentFolderA, ACT pattern grants for GroupA and GroupB (from the same directly applied ACT)

Glossary

access control template (ACT)

a reusable named authorization pattern that can be applied to multiple resources. An access control template consists of a list of users and groups and indicates, for each user or group, whether permissions are granted or denied.

authorization

the process of determining the permissions that particular users have for particular resources. Authorization either permits or denies a specific action on a specific resource, based on the user's identity and on group memberships.

capability

an application feature that is under role-based management. Typically, a capability corresponds to a menu item or button. For example, a Report Creation capability might correspond to a New Report menu item in a reporting application. Capabilities are assigned to roles.

credential

evidence that is submitted to support a claim of identity (for example, a user ID and password) or privilege (for example, a passphrase or encryption key). Credentials are used to authenticate a user.

grid host

the machine to which the SAS client makes an initial connection in a SAS High-Performance Analytics application.

group

a collection of users who are registered in a SAS metadata environment. A group can contain other groups as well as individual users.

Hadoop Distributed File System (HDFS)

a portable, scalable framework, written in Java, for managing large files as blocks of equal size. The files are replicated across multiple host machines in a Hadoop cluster in order to provide fault tolerance.

libref (library reference)

a SAS name that is associated with the location of a SAS library. For example, in the name MYLIB.MYFILE, MYLIB is the libref, and MYFILE is a file in the SAS library.

metadata identity (identity)

a metadata object that represents an individual user or a group of users in a SAS metadata environment. Each individual and group that accesses secured resources on a SAS Metadata Server should have a unique metadata identity within that server.

role (user role)

a set of capabilities within an application that are targeted to a particular group of users.

SAS authentication

a form of authentication in which the target SAS server is responsible for requesting or performing the authentication check. SAS servers usually meet this responsibility by asking another component (such as the server's host operating system, an LDAP provider, or the SAS Metadata Server) to perform the check. In a few cases (such as SAS internal authentication to the metadata server), the SAS server performs the check for itself. A configuration in which a SAS server trusts that another component has pre-authenticated users (for example, web authentication) is not part of SAS authentication.

SAS data set (data set)

a file whose contents are in one of the native SAS file formats. There are two types of SAS data sets: SAS data files and SAS data views.

SAS Stored Process (stored process)

a SAS program that is stored on a server and defined in metadata, and which can be executed by client applications.

SAS table

the visual rendering of a SAS data set in tabular format. *See also* [SAS data set](#).

SAS Workspace Server

a SAS server that provides access to SAS Foundation features such as the SAS programming language and SAS libraries.

SASHDAT file format

a SAS proprietary data format that is optimized for high performance and computing efficiency. For distributed servers, SASHDAT files are read in parallel. When used with the Hadoop Distributed File System (HDFS), the file takes advantage of data replication for fault-tolerant data access.

stored process

See [SAS Stored Process](#).

theme

a collection of specifications (for example, colors, fonts, and font styles) and graphics that control the appearance of an application.

unrestricted identity

a user or group that has all capabilities and permissions in the metadata environment due to membership in the META: Unrestricted Users Role (or listing in the adminUsers.txt file with a preceding asterisk).

user role

See [role](#).

web authentication

a configuration in which users of web applications and web services are verified at the web perimeter, and the metadata server trusts that verification.

Index

A

ACM data 127
 adding users 3
 alerts 84
 audit data 125
 auditing 48
 authentication 60
 autoload 22
 autostart 7

C

capabilities 133
 descriptions 136
 co-located HDFS 111

D

data loading 10
 autoload 22
 interactive load 16
 reload-on-start 19
 self-service import 16

E

encryption
 backing store 61
 SASHDAT 64
 extended attributes
 autoload 32
 autostart 7
 linguistic files 77
 reload-on-start 21
 SAS LASR Analytic Server 101
 stop lists 77

G

geo maps 78

Esri 79
 OpenStreetMap 78
 guest access
 home page 73

H

high-cardinality data 148
 home page 70
 hub
 See [home page](#)

L

LASR authorization service 41
 libraries
 HDFS 116
 LASR 98
 locked-down servers 56
 logging 87
 for reload-on-start 21
 permissions 89
 SAS LASR Analytic Server 104
 SAS LASR Analytic Server Monitor 87
 schema validation errors 89

P

parallel load
 co-located storage 109
 embedded process 109
 NFS-mounted storage 109
 permissions 36
 by task 37
 inherited 189
 origins 188
 row-level 39
 printing
 report footers 80

R

relationship data 126
 reload-on-start 19
 reports
 defining footers 80
 predefined 121
 roles 133
 custom 133
 predefined 133
 row-level security 44
 identity-driven 47
 precedence 45
 set interactively 39
 syntax (basic, batch) 46
 syntax (Text tab) 46

S

SAS LASR Analytic Server 132
 adding 97
 advanced options 102
 distributed 94
 extended attributes 101
 lifetime 105
 logging 104
 managing 6
 memory limits 103
 metadata objects 95
 monitoring 106
 non-distributed 94
 SAS LASR Analytic Server Monitor
 85
 SAS Mobile BI 56
 capabilities 137

configuration properties 145
 logging 89
 security 60
 troubleshooting 187
 SAS Visual Analytics Administrator
 Authorization page 40
 HDFS tab 112
 LASR Servers tab 108
 LASR Tables tab 12, 14
 Mobile Devices tab 59
 Process Monitor tab 107
 Resource Monitor tab 106
 troubleshooting 181
 user preferences 91
 security
 auditing 48
 data 41
 locked-down servers 56
 mobile 60
 row-level 44
 SAS token authentication 60
 server tags 44
 signature files 43

T

tables
 deleting from HDFS 116
 registering 15
 reloading 14
 staging 15
 unloading 14
 valid names 12
 transport service 132
 device management 56



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

