



SAS[®] Visual Analytics 7.5: Administration Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2019. *SAS® Visual Analytics 7.5: Administration Guide*. Cary, NC: SAS Institute Inc.

SAS® Visual Analytics 7.5: Administration Guide

Copyright © 2019, 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

September 2021

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.

7.5-P1:vaag

Contents

| | |
|---|-----------|
| <i>About this Document</i> | v |
| <i>What's New in Administration of SAS Visual Analytics</i> | ix |
| Chapter 1 / Getting Started | 1 |
| Orientation | 1 |
| Adding Users | 3 |
| Operating Servers | 6 |
| Chapter 2 / Loading Data | 11 |
| About Loading Data | 12 |
| Get LASR Table Status | 15 |
| Administer LASR Tables | 17 |
| Administrator Load | 18 |
| Self-Service Import | 20 |
| Reload-on-Start | 22 |
| Autoload | 25 |
| Chapter 3 / Security | 41 |
| Permissions | 42 |
| Access to In-Memory Data | 46 |
| Row-Level Security | 49 |
| Key Actions Auditing | 55 |
| Locked-Down Servers | 64 |
| Mobile Device Access | 64 |
| Authentication | 68 |
| Encryption | 69 |
| Chapter 4 / Fine-Tuning | 79 |
| Administering SAS Home | 80 |
| Supporting Guest Access | 80 |
| Supporting Text Analytics | 80 |
| Supporting Stored Processes | 81 |
| Supporting Report Distribution | 82 |
| Supporting the Search Features | 83 |
| Supporting User-Defined Formats | 84 |
| Managing the Availability of Fonts | 84 |
| Managing Alerts and Notifications | 85 |
| Supporting the Monitoring Features | 86 |
| Supporting the HDFS Browse Features | 89 |
| Adjusting the Logging Configuration | 89 |
| Supporting Footers in Printed Reports | 92 |
| Using Multiple SAS Application Servers | 93 |
| Streamlining Access to Management Pages | 95 |
| Customizing the Appearance of Applications or Reports | 97 |

| | |
|--|------------|
| Chapter 5 / Geographic Maps | 99 |
| Geo Map Server | 99 |
| Geo Map Data Sets | 102 |
| Custom Polygons: Prerequisites | 102 |
| Custom Polygons: Instructions | 103 |
| Custom Polygons: Data Sets | 107 |
| Custom Polygons: Examples | 111 |
| | |
| Chapter 6 / SAS LASR Analytic Server | 123 |
| About LASR Servers | 124 |
| LASR-Related Metadata | 125 |
| In-Memory LASR Names | 126 |
| LASR Server Definition | 127 |
| LASR Library Definition | 135 |
| Limiting Space for Tables | 140 |
| Distributed Server: Monitoring | 141 |
| Distributed Server: Parallel Load | 146 |
| Distributed Server: Co-located HDFS | 150 |
| Distributed Server: High-Volume Access to Smaller Tables | 156 |
| | |
| Chapter 7 / Reports for Administrators | 159 |
| About the Predefined Reports | 159 |
| How to Provide Administrative Data | 162 |
| About auditRefresh | 166 |
| | |
| Chapter 8 / Reference | 169 |
| Software Components | 169 |
| Roles and Capabilities | 171 |
| Configuration Properties | 177 |
| High-Cardinality Constraints | 188 |
| Predefined LASR Libraries | 190 |
| Administrative Preferences | 194 |
| Sample Objects | 196 |
| | |
| Chapter 9 / Troubleshooting | 199 |
| Access Issues | 199 |
| Server Operation Issues | 201 |
| Load, Reload, and Import Issues | 203 |

About this Document

Audience

This document helps you administer SAS Visual Analytics 7.5 and SAS Visual Statistics 7.5.

This document is part of the integrated collection of [SAS 9.4 administration](#) guides.

Documentation Conventions

SAS Configuration Directory

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

| Host | Example |
|----------|-----------------------------------|
| UNIX: | <code>/opt/sas/config/Lev1</code> |
| Windows: | <code>C:\sas\Config\Lev1</code> |

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

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

Short Forms and Labels

The interface and this document use the following short-form terminology:

| Long Form | Short Form |
|----------------------------------|--------------|
| Hadoop Distributed File System | HDFS |
| SAS LASR Analytic Server | LASR server |
| SAS LASR Analytic Server library | LASR library |
| SAS LASR Analytic Server table | LASR table |

The following table documents selected navigation labels in the **SAS VISUAL ANALYTICS** section of the applications menu.

| Navigation Label | Corresponding Application |
|------------------|--|
| Administrator | SAS Environment Manager Administration |
| Data Preparation | SAS Visual Data Builder |
| Report Builder | SAS Visual Analytics |

Related Deployment Documentation

| Deployment Type | Activity-Specific Reference | |
|--|-------------------------------|-------------------------|
| SAS Visual Analytics 7.5 (Distributed LASR server) | Install (new) | Upgrade |
| SAS Visual Analytics 7.5 (Non-distributed LASR server) | Install (new) | Upgrade |

See Also

[“Determining Your Update Path for SAS Visual Analytics” in *SAS Guide to Software Updates and Product Changes*](#)

What's New in Administration of SAS Visual Analytics

General Updates

Administration of SAS Visual Analytics 7.5 differs from administration of prior releases as follows:

- The visual exploration and SAS report (2G) object types are merged into a single object type, SAS report (2G).
- The applications SAS Visual Analytics Explorer and SAS Visual Analytics Designer are merged into a single application, SAS Visual Analytics.
- All SAS Visual Analytics 7.5 interfaces are HTML5 applications. There is no longer a choice of application presentation mode (classic or modern).
- The application SAS Visual Analytics Administrator is incorporated into the general purpose administrative application, SAS Environment Manager Administration. See [“Updates for SAS Visual Analytics Administrators”](#) in *Using SAS Environment Manager Administration*.
- In deployments that use co-located Hadoop, SSH is used to connect to Hadoop when you browse HDFS from SAS Environment Manager Administration. See [“Supporting the HDFS Browse Features”](#).

In the previous release of SAS Visual Analytics, this functionality is available as a hot fix.

- SAS Visual Analytics now uses the transport service for most of its middle-tier communications. In prior releases, the precursor applications (designer and explorer) did not use the transport service as extensively. Here are the main effects of this change:
 - SAS Visual Analytics is subject to data limits that the transport service enforces.
 - SAS Visual Analytics benefits from results caching functionality that the transport service provides.

See [“Configuration Properties: Transport Services”](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

- The graphical user interface for defining row-level permission conditions (in SAS Environment Manager Administration) is not functionally identical to the corresponding interface (in SAS Visual Analytics Administrator) in prior releases.

The new interface works with all supported syntax, but it does not verify syntax or include all of the usability-oriented features of the old interface. See [“How to Set a Row-Level Permission”](#).

- New font services enable administrators to manage the list of available fonts. See [“Make More Fonts Available”](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Changes to SAS Visual Analytics Configuration Properties

Most of the following changes result from the merging of applications.

| 7.4 Property | | 7.5 Adjustment | |
|--|--|---|--|
| va.Alert.EvaluationCycleMilliseconds | | Default value: 60000 | |
| va.CheckCardinalityBeforeQuery | | Discontinued | |
| va.DistinctCountServerLimit | | Discontinued | |
| va.SortResultServerLimit | | Renamed: va.detailTableSortResultLimit | |
| vae.BoxPlotServerLimit | | Discontinued | |
| vae.DecisionTreeServerLimit | | Renamed: va.decisionTreeCardinalityLimit | |
| vae.DecisionTreeTimeout | | Discontinued | |
| vae.FetchRowsServerLimit | | Discontinued | |
| vae.FrequencyServerLimit | | Discontinued | |
| vae.modeling.ClassCardinalityLimit | | Renamed: va.modelingClassCardinalityLimit | |
| vae.modeling.DecisionTreePredictorCardinalityLimit | | Renamed: va.modelingDecisionTreePredictorCardinalityLimit | |
| vae.modeling.DecisionTreePredictorBinsCardinalityLimit | | Discontinued | |
| vae.modeling.DecisionTreeResponseCardinalityLimit | | Renamed: va.modelingDecisionTreeResponseCardinalityLimit | |
| vae.modeling.GroupByCardinalityLimit | | Renamed: va.modelingGroupByCardinalityLimit | |
| vae.PageRowCount | | Discontinued | |
| vae.PathingPathLengthLimit | | Renamed: va.sankeyMaxPathLength | |

| 7.4 Property | 7.5 Adjustment |
|---|---|
| vae.PathingTopKLimit | Renamed: va.sankeyMaxNumberOfPathDisplayed |
| vae.PathingTransactionIdsLimit | Renamed: va.sankeyMaxTransactionId |
| vae.RealScatterServerLimit | Discontinued |
| vae.ScatterPlotServerLimit | Discontinued |
| vae.SummaryServerLimit | Discontinued |
| vae.TableSortingEnabled | Discontinued |
| vav.ui.mode | Discontinued |
| viewerservices.default.max.cells.produced | Increased scope ¹ |
| viewerservices.lasr.socketTimeout.milliseconds.interactions | Default value: 300000, increased scope ¹ |
| viewerservices.validate.schema.create | Increased scope ¹ |
| viewerservices.validate.schema.read | Increased scope ¹ |

¹ This property affects SAS Visual Analytics 7.5, but did not affect SAS Visual Analytics Designer or SAS Visual Analytics Explorer in previous releases.

Changes to SAS Visual Analytics Capabilities

The following changes result from the merging of applications.

| 7.4 Capability | 7.5 Adjustment |
|--|---------------------------|
| Visual Analytics: Explore Data | Discontinued |
| Visual Analytics Explorer: Export as Image | Discontinued |
| Visual Analytics Explorer: Export as Report | Discontinued |
| Visual Analytics Explorer: Refresh Data | Discontinued |
| Visual Analytics Admin: Manage Authorization | Discontinued ¹ |
| Visual Analytics Admin: Manage LASR Analytic Server | Repurposed ² |
| Visual Analytics Admin: Monitor LASR Analytic Server | Repurposed ² |
| Visual Analytics Admin: Browse HDFS | Repurposed ² |
| Visual Analytics: Advanced: Manage Environment | Repurposed ² |

¹ The ability to set metadata-layer permissions on an object is controlled by the WriteMetadata permission for that object.

² Affects SAS Environment Manager Administration. See [“Capabilities” in Using SAS Environment Manager Administration](#).

Related Information

For changes in the underlying platform, see [“SAS 9.4 Intelligence Platform” in What’s New in SAS 9.4 and SAS Viya](#).

For changes in the general purpose administrative application, see [“What’s New in SAS Environment Manager Administration” in Using SAS Environment Manager Administration](#).

For changes in the end-user applications, see [SAS Visual Analytics: What’s New](#) (or, for prior releases, see [“SAS Visual Analytics” in What’s New in SAS 9.4 and SAS Viya](#)).

Getting Started

| | |
|--|----------|
| Orientation | 1 |
| Tasks | 1 |
| Tools | 2 |
| Adding Users | 3 |
| Preliminary Steps | 3 |
| Add a User | 3 |
| Host Account Privileges | 5 |
| See Also | 6 |
| Operating Servers | 6 |
| Get LASR Server Status | 6 |
| Start or Stop a LASR Server | 7 |
| Enable Autostart For a LASR Server | 8 |
| Operate Other Servers | 9 |
| See Also | 9 |

Orientation

Tasks

| | |
|---|--|
| Get familiar with the software architecture. | See “Software Components” . |
| Get familiar with the software functionality. | See SAS Visual Analytics: Overview . |
| Register users. | See “Adding Users” . |
| Make data available. | See “About Loading Data” . |

Make sure backups occur.


See “Setting Up Your Backups” in [SAS Intelligence Platform: System Administration Guide](#).

Tools

SAS Environment Manager Administration

Most administrative tasks can be performed in SAS Environment Manager Administration.

To access SAS Environment Manager Administration from SAS Home (or any other SAS 9 integrated application):

- 1 In the application banner, click .
- 2 In the applications menu, under **SAS VISUAL ANALYTICS**, select **Administrator**.

Note: If that menu option is not available, make sure that you have the Visual Analytics: Advanced: Manage Environment [capability](#). In a new deployment, only members of the Visual Analytics Data Administrators group have that capability.

In the current browser window, the application context switches to SAS Environment Manager Administration.

For details and alternatives, see “[Launching SAS Environment Manager Administration](#)” in [Using SAS Environment Manager Administration](#).

SAS Environment Manager

Advanced resource monitoring features are available in SAS Environment Manager, which is a separate application from SAS Environment Manager Administration. See [SAS Environment Manager: User’s Guide](#).

SAS Management Console

A few administrative tasks can be performed only in SAS Management Console. For example, you must use SAS Management Console to set configuration properties and promote objects. For details, see “[Comparison of SAS Management Console and SAS Environment Manager](#)” in [SAS Intelligence Platform: System Administration Guide](#).


To access SAS Management Console, click the Windows **Start** button, and select **SAS** ⇒ **SAS Management Console** from the menu. For details and UNIX instructions, see “Starting SAS Management Console” in *SAS Intelligence Platform: Desktop Application Administration Guide*.

TIP For tasks that can be performed in either SAS Management Console or in SAS Environment Manager Administration, this document provides instructions for the latter application. However, any task that you previously performed in SAS Management Console remains available in that application. No functionality has been removed from SAS Management Console.

Adding Users

For concepts, background, and alternate methods, see “User Administration” in *SAS Intelligence Platform: Security Administration Guide*.

Preliminary Steps

- 1 Sign in to [SAS Environment Manager Administration](#) as someone who is authorized to add users (for example, sasadm@saspw).
- 2 In the vertical navigation bar, select  **Users**.

Add a User

- 1 Identify or create an account with which the user can access the SAS Metadata Server.

In the simplest case, accounts are known to the metadata server’s host. For details and exceptions, see “Authentication to the Metadata Server” in *SAS Intelligence Platform: Security Administration Guide*.


Note: You do not have to register users who need only limited, anonymous access to SAS Visual Analytics. See “Configure Guest Access” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

- 2 Make sure the account has any necessary host-layer privileges.
 - If the user accesses servers on Windows, see “Windows Privileges” in *SAS Intelligence Platform: Security Administration Guide*.
 - If the user imports or loads data in SAS Visual Analytics or starts and stops a LASR server, see “Host Account Privileges”.

- 3 In the navigation pane on the **Users** page, click . Select **New user**.

Note: If that action is not available, you are not authorized to add users. Sign in as someone who has user administration capabilities (for example, sasadm@saspw).

- 4 In the New User window, provide information as follows:
 - a In the **Name** field, enter a unique identifier for the user.
 - b Enter a display name for the user.
 - c Click **Save**. The new user definition is displayed.
- 5 In the unusual circumstance in which the new user should have only an **internal account**, skip this step.

On the new user's **Accounts** tab, click . In the Edit Accounts window, click the **+** to add an inbound login for the user. See [“How Logins Are Used” in SAS Intelligence Platform: Security Administration Guide](#).

In the new row, provide information as follows:


- a If the **web** authentication domain is available, see [“Logons for Users Who Participate in Web Authentication” in SAS Intelligence Platform: Security Administration Guide](#).





Otherwise, select the **DefaultAuth** authentication domain.


Note: The preceding instructions assume that your deployment uses the standard authentication domains. See [“Authentication Domains” in SAS Intelligence Platform: Security Administration Guide](#).

- b Click in the **Stored User ID** field, and enter the user's account ID (as identified in step 1).
For a Windows account, you must enter the user ID in one of the following formats:

user-ID@domain.extension OR *domain\user-ID* OR *machine\user-ID*

- c Leave the **Stored Password** field blank. It is not necessary to store a password for inbound (initial authentication) purposes. Click **Save**.
- 6 On the new user's **Member Of** tab, click , and add direct memberships for the user as follows:

| Type of User | Direct Member Of |
|------------------------------------|---|
| Unrestricted user |  Metadata Server: Unrestricted |
| Platform administrator |  SAS Administrators |
| SAS Visual Analytics administrator |  Visual Analytics Data Administrators |
| SAS Visual Data Builder user |  Visual Data Builder Administrators |

| Type of User | Direct Member Of |
|---------------------------|--|
| SAS Visual Analytics user |  Visual Analytics Users |
| Users of other products | See your product-specific documentation. |

Note: The preceding instructions assume that your deployment uses the predefined groups and roles. Users of SAS Visual Analytics get their role memberships indirectly through their group memberships. See [“Initial Role Members”](#).

Note: Only an unrestricted user can add members to the Metadata Server: Unrestricted role. See [“Create an Additional Unrestricted User”](#) in *Using SAS Environment Manager Administration*.

- 7 (Optional) On the new user’s **Accounts** tab, [store additional credentials](#).
- 8 (Optional) [Store contact information](#) for the new user.
- 9 Validate the new user. Ask the user to sign in to an application and verify that the expected functionality is available.

Note: If necessary, see the troubleshooting information for accessing [SAS Environment Manager Administration](#) or [SAS Visual Analytics](#). To troubleshoot access to another product, see the documentation for that product.

Host Account Privileges

Introduction

The requirements in this section apply to SAS Visual Analytics users who import data, load data, or start and stop a LASR 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 PIDs directory that is beneath the [va.monitoringPath](#) directory, and the [va.lastActionLogPath](#) directory.

SAS LASR Analytic Server

The account must be able to authenticate to the host for the [LASR 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 LASR server. For more complex environments, see [“Authentication”](#).
- For a distributed server, give the account passwordless SSH access to all of the machines in the cluster. See [“Passwordless SSH” in 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 SAS Intelligence Platform: Security Administration Guide](#).


See Also

- [“Uniqueness Requirements” in SAS Intelligence Platform: Security Administration Guide](#)
- [“Who Can Manage Users, Groups, and Roles?” in SAS Intelligence Platform: Security Administration Guide](#)


Operating Servers


Get LASR Server Status


- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), select  **LASR**.
- 2 On the **LASR Servers** tab, select a server.
- 3 Right-click the selected server, and select **Get status**.


Note: As an alternative, you can click  to retrieve status information for all servers that have a selected (checked) check box.

The **Status** column uses the following icons:

-
-  Running




 -  Stopped

 -  Over capacity


 -  Unknown

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, reload, or change source for tables. See [“Limiting Space for Tables”](#).

Here are additional details:

- Status information is not automatically updated. To get the most current status information, repeat the **Get status** action. Or, click  to retrieve status information for all servers that have a selected (checked) check box.
- The Refresh action () refreshes the displayed server metadata. It does not update server status information.
- 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 [“Limiting Space for Tables”](#).
- For a distributed LASR server, a **Virtual Memory** column indicates how much of the total cluster memory is currently in use by each server process. The displayed values are calculated as if all LASR star schemas are output as tables. See [“Distributed Server: Monitoring”](#).
- To customize the display, see [“Tips for Interacting with Tables”](#) in *Using SAS Environment Manager Administration*.
- To clear all selected check boxes, click .

Start or Stop a LASR Server




- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), select  **LASR**.
- 2 On the **LASR Servers** tab, select the server that you want to start or stop.
- 3 Right-click the selected server, and select **Start** or **Stop**.

Note: There is no restart action. To restart a server that is currently running, you must first stop the server.

Here are key points:

- Starting a server causes tables that participate in reload-on-start to be reloaded. See “[Reload-on-Start](#)”.
- Stopping a server unloads all of its tables. See “[Server lifetime \(LIFETIME=\)](#)”.

Here are additional details:

- If you click , all stopped servers that have a selected (checked) check box are started.
- If you click , all running servers that have a selected (checked) check box are stopped.
- If you click , all selected check boxes are cleared.

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

Enable Autostart For a LASR Server


Introduction


A LASR 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 LASR 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 the following instructions.

How to Enable Autostart

- 1 From the vertical navigation bar in [SAS Environment Manager Administration](#), select  **Libraries**.
- 2 Locate the definition for a LASR library that is assigned to the LASR server.

- a In the drop-down list, select the category **High-Performance Analytics (e.g. LASR)**.
 - b (Optional) Use the filter box to find the library.
- 3 Right-click the library, and select **Open**.
 - 4 On the library's **Advanced** tab, select **Extended Attributes** from the **View** drop-down list.
 - 5 Click .
 - 6 In the Edit Extended Attributes window, add or find the **VA.AutoLoad.AutoStart** property. In the **Value** cell for that property, enter **Yes**.
-
- Note:** Although the property name includes the term **Autoload**, this property is not used exclusively for autoload.
-
- 7 Click **Save**.

Operate Other Servers

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

Here are basic instructions for restarting the platform:

| Host | Instruction |
|---------|--|
| UNIX | From your equivalent of <code>/opt/sas/config/Lev1</code> , run <code>./sas.servers restart</code> . |
| 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 *SAS Intelligence Platform: System Administration Guide*.

See Also

- [“About LASR Servers”](#)
- To troubleshoot LASR server operation, see [“Server Operation Issues”](#).

Loading Data

| | |
|---|-----------|
| About Loading Data | 12 |
| Introduction | 12 |
| Load Methods | 12 |
| Reload Methods | 13 |
| Data Compression | 14 |
| Table and Column Names | 15 |
| Get LASR Table Status | 15 |
| Administer LASR Tables | 17 |
| Unload, Reload, or Delete a Table | 17 |
| Replace a Source Table | 17 |
| Results | 17 |
| Additional Considerations | 18 |
| Administrator Load | 18 |
| Preparation | 18 |
| Load a Table | 19 |
| Self-Service Import | 20 |
| Introduction | 20 |
| Requirement: User Privileges | 20 |
| Requirement: SAS/ACCESS | 20 |
| How to Protect Imported Data | 21 |
| How to Limit Import Size | 22 |
| Reload-on-Start | 22 |
| Introduction | 22 |
| How Reload-on-Start Works | 23 |
| How to Enable Reload-on-Start | 23 |
| Additional Considerations | 24 |
| Reference | 24 |
| Autoload | 25 |
| Introduction | 25 |
| How Autoload Works | 26 |
| The Autoload Directories | 27 |
| Timing of Autoload | 27 |
| How to Start Autoload | 28 |
| How to Stop Autoload | 29 |

| | |
|------------------------------------|----|
| How to Add an Implementation | 29 |
| Additional Considerations | 34 |
| Reference | 35 |

About Loading Data

Introduction

Users can easily import data. See [SAS Visual Data Builder: Accessing Data](#). This chapter documents administrative aspects of data loading.

SAS Visual Analytics uses data that is loaded to memory in a LASR 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. |

Load Methods

Load methods vary by data source.

| Load Method | Data Source | | | | |
|--------------------|--------------------------|--------------|---------------------------|-------------|--------------------|
| | Spreadsheet or Delimited | SAS Data Set | SASHDAT File ¹ | Data Server | Other ² |
| → Interactive load | | ✓ | ✓ | ✓ | |
| Run a data query | | ✓ | ✓ | ✓ | |
| Import from server | | ✓ | | ✓ | ✓ |


| Load Method | Data Source | | | | |
|---------------------|--------------------------|--------------|---------------------------|-------------|--------------------|
| | Spreadsheet or Delimited | SAS Data Set | SASHDAT File ¹ | Data Server | Other ² |
| Import a local file | ✓ | ✓ | | | |
| Autoload | ✓ | ✓ | | | |

¹ Co-located HDFS or NFS-mounted MapR. See “[Distributed Server: Parallel Load](#)”.

² 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.¹</p> <p>Output from LASR star schemas.¹</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> |

¹ 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 autoloaded, 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 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”](#).

- 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”](#).
- 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 SAS Visual Analytics) 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”](#).
- 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 *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 LASR Table Status

- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), select  **LASR**.
- 2 On the **LASR Tables** tab, select a table.
- 3 Right-click the selected table, and select **Get status**.


Note: As an alternative, you can click to retrieve status information for all tables that have a selected (checked) check box.

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

Unknown

Here are additional details:

- Status information is not automatically updated. To get the most current status information, repeat the **Get status** action. Or, click  to retrieve status information for all tables that have a selected (checked) check box.
- The Refresh action () refreshes the displayed table metadata. It does not update table status information.
- To customize the display, see [“Tips for Interacting with Tables” in Using SAS Environment Manager Administration](#).
- To clear all selected check boxes, click .

Here are tips about the information that is displayed on the **LASR Tables** tab:

- 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.

.....

- 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 displays the user ID that started the server.

.....

- The **LASR Name** column displays table names in the in-memory format `server-tag.table-name`. See [“In-Memory LASR Names”](#).
- 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 SAS LASR Analytic Server: Reference Guide](#).

.....

Administer LASR Tables

Unload, Reload, or Delete a Table

- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), select  **LASR**.
- 2 On the **LASR Tables** tab, right-click on a table, and select an action. Here are tips:
 - If most actions are disabled, click , and then right-click on the table again.
 - You cannot delete or reload a table that is currently loaded. You must first unload 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 [“Row-Level Security”](#).
.....

Results

To view a log for the most recent interactive action for a table, access 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”](#).




Additional Considerations

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

.....

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.
 - Before you use a tab toolbar button, make sure only the appropriate check boxes are selected. To clear all check boxes, click .
 - To perform an action on multiple tables, select check boxes, and then click an icon in the tab toolbar.

Administrator Load

Preparation

Register Source Tables

Only registered tables can be interactively loaded or staged using the instructions in this section. For alternate methods for making data available, see [“Autoload”](#) and [“Self-Service Import”](#).

- 1 On the **Folders** page in [SAS Environment Manager Administration](#), right-click on a library, and select **Register and Update Tables**.
- 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: For alternate methods for registering tables, see [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 LASR server. See [“Distributed Server: Parallel Load”](#).

- 1 On the **Folders** page in [SAS Environment Manager Administration](#), right-click on a table. Select **Add to HDFS** (to stage to co-located HDFS) or **Add to Data Server** (to stage to NFS-mounted MapR).
- 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 vertical navigation bar in [SAS Environment Manager Administration](#), select  **LASR**.
- 2 On the **LASR** page, select the **LASR Tables** tab.
- 3 Click **→**.
- 4 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 **LASR Servers** tab (right-click on a server).

Self-Service Import

Introduction

Data imports that are performed in SAS Visual Analytics or SAS Visual Data Builder are referred to as self-service imports. This topic provides information to help an administrator support self-service imports.

Requirement: User Privileges

- Individual data source-specific capabilities affect the availability of all self-service import actions. In SAS Visual Analytics, 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 3.2](#).
- Self-service import actions use a workspace server and a LASR server, so users must have appropriate host-layer access. See [“Host Account Privileges”](#).

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

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](#).

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](#).

File Size Limit

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

Tables Limit

To limit the total amount of space that a LASR server can use to host tables, see [“Limiting Space for Tables”](#).

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”](#).

How Reload-on-Start Works

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



- 1 In SAS Visual Analytics, 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

- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), select  **Libraries**.
- 2 Open a LASR library.
- 3 On the library's **Options** tab, select **Storage**. In the **Data provider library** field, specify a Base SAS library. The specified library functions as the backing store for participating tables.
- 4 On the library's **Advanced** tab, select the **Extended attributes** view. Click , Add and set attributes as follows:

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

- 5 Click **Save**.
- 6 (Optional) To selectively exclude a LASR table from participation:
 - a In the LASR library's definition, select the **Tables** tab.
 - b Open a LASR table.

- c On the LASR table's **Advanced** tab, select the **Extended Attributes** view.
- d Add a row that sets the **Value** cell for the VA.ReloadOnStart.Enabled property to **No**.
- e Click **Save**.

Additional Considerations

- Not all tables can participate in reload-on-start. See [“Reload Methods”](#).
- 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”](#).
 - The identity that starts the server has host access to the table (in the associated data provider library).
- Reload-on-start occurs after the LASR server is started by autoloading, by an explicit start request in SAS Environment Manager Administration, 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”](#).

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](#).

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 autoloading include the following:

- You do not have to start the server. If a LASR server stops, the next run of autoloading 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 autoloading.

For limitations of autoloading, see [“Additional Considerations”](#).

How Autoload Works

Here is a summary of how autoloading works:

- 1 Autoloading 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, autoloading 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 autoloading 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, autoloading 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 LASR server does not trigger an immediate run of autoload.
- Stopping the associated LASR server does not stop autoload activity. If the server is down when a run of autoload begins, autoload starts the server.

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

Note: On UNIX, the interval clock starts on the hour. For example, if the interval is 15 minutes, then autoloading runs on the hour and at 15, 30, and 45 minutes after the hour.

How to Start Autoload

To start scheduled runs for an implementation of autoloading:

- 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 LASR server and load data. See [“Host Account Privileges”](#).
 - On UNIX, enable the account to run cron jobs.
 - In the [SAS configuration directory](#), give the account Read and Write access to the autoloading directories and their contents. For the public implementation of autoloading, 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.)

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

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: /Shared Data/SAS Visual Analytics/Public/LASR 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.

.....

Note: On Windows, 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`).

.....

Note: On UNIX, 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”](#).

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 LASR Table Status”](#).

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 LASR server.

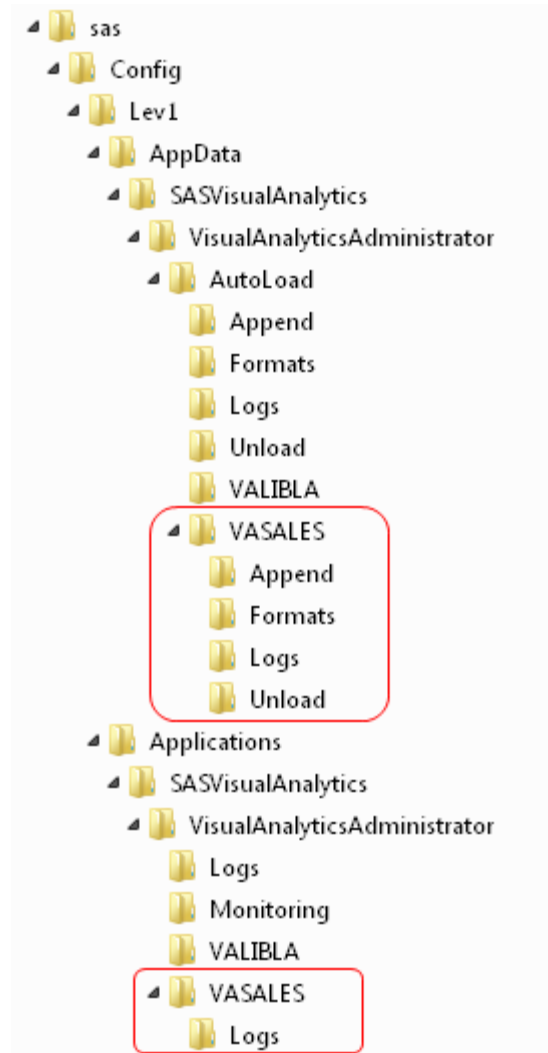
How to Add an Implementation

Introduction

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

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`.

Note: Creating the new scripts directory beneath the existing autoload 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.sh

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

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

Verify that the appropriate configuration files are referenced. See ["Configuration Files for Autoload"](#).

schedule.sh and unschedule.sh

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

```
RUNSAS_PATH="autoload-scripts-branch/VASALES/runsas.sh"
```

4 In [SAS Environment Manager Administration](#), 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 LASR 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 ¹ | 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 |

¹ 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 autoload scripts directory. For example:

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

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

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 Environment Manager Administration](#), 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 LASR 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 ¹ | 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 |

- 1 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](#)”.
- 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”](#).

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](#).

Note: On UNIX, 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”](#).
- 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 [“Connection Options”](#) in *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.

Note: On Windows, the list is in the AutoLoad.cfg file in the implementation's autoload scripts directory.

Note: On UNIX, 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 *SAS Intelligence Platform: System Administration Guide*.

User-Defined Formats for Autoload

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

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 autoloading 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 autoloading to a server that uses UTF-8 encoding. For more information, see [Avoiding Character Data Truncation By Using the CVP Engine in 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 SAS Visual Analytics, 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 autoloading 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 autoloading (but is not exclusive to autoloading):

VA.AutoLoad.AutoStart

specifies whether the associated LASR 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 autoloading. For information about how tables that are already loaded are affected, see "[How Autoloading Works](#)".

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

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 [“Importing Local Data Files” in SAS Visual Data Builder: Accessing Data](#).

The following details are specific to autoload:

- The file size limitation for interactive import is not applicable to autoload.
- You cannot autoload a ZIP file.
- Autoload always reads column names from the first row and begins data import on the second row.
- When you autoload 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.
- Autoload 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 autoload 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.

Security

| | |
|---|-----------|
| Permissions | 42 |
| Key Points | 42 |
| Permission Definitions | 42 |
| Permissions by Task | 43 |
| See Also | 45 |
| Access to In-Memory Data | 46 |
| SAS LASR Authorization Service | 46 |
| Signature Files | 48 |
| Server Tags | 49 |
| Row-Level Security | 49 |
| Introduction | 49 |
| Permission Precedence | 50 |
| How to Set a Row-Level Permission | 51 |
| Batch Approach to Row-Level Permissions | 52 |
| Syntax for a Row-Level Permission Condition | 52 |
| Key Actions Auditing | 55 |
| Introduction | 55 |
| How to Safely Enable Auditing | 57 |
| Audit Content and Coverage | 58 |
| Locked-Down Servers | 64 |
| Mobile Device Access | 64 |
| Introduction | 65 |
| About the Mobile Devices Page | 65 |
| Add a Device to a List | 66 |
| Remove a Device from a List | 67 |
| Determine When a Device Was Blacklisted | 67 |
| Advanced: Change How Devices Are Managed | 67 |
| Related Tasks | 68 |
| Authentication | 68 |
| Introduction | 68 |
| Shared Accounts for Self-Service Imports | 68 |
| Policy for Concurrent User Logins | 69 |
| Encryption | 69 |
| Introduction | 69 |

| | |
|---|----|
| On-Disk Encryption of Reload-on-Start Files | 70 |
| On-Disk Encryption of SASHDAT Files | 73 |

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, tables, libraries, servers, and folders. See [“Metadata Authorization Model” in SAS Intelligence Platform: Security Administration Guide](#).
- To set permissions, use SAS Environment Manager Administration or an alternate interface. See [“Access Management” in SAS Intelligence Platform: Security Administration Guide](#).
- 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.

- Never 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.

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 LASR server, stop the server or set a tables limit. |
| 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. |

| Permission | Affected Actions |
|---------------------------------|---|
| 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 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. |
| ManageMemberMetadata (MMM) | On a group or role, enables a non-administrator to modify membership. (Provides edit access to the Members tab.) |
| ManageCredentialsMetadata (MCM) | On a group, enables a non-administrator to manage shared logins. (Provides edit access to the Accounts tab.) |

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

Permissions by Task

Reports

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

Table 3.1 *Permissions for Working with Reports*

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

To create, update, or delete a report, access to SAS Content Server is also required. See [“Administer the SAS Content Server”](#) in *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 the following table.

LASR Tables and Servers

The following table documents metadata-layer permissions for working with LASR tables and LASR servers.

Table 3.2 *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 ¹ | 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 ² | 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 | - |
| Update a table's metadata | - | RM | RM | RM, WM |
| Delete a table from metadata | - | RM, WM | RM, WMM | RM, WM |

¹ This row applies to requests for which no corresponding LASR table object exists in metadata. Read and Write permissions on the folder support actions against the new table.

² This row applies to requests for which a corresponding LASR table object already exists in metadata.

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 ² | Folder | Query or Schema |
|---|--------|--------------------|---------|-----------------|
| Save a new query or schema ¹ | RM | RM | RM, WMM | - |
| Run a query or schema ¹ | 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 |

¹ These tasks create new LASR tables, so the permission requirements for loading a LASR table must also be met. See [Table 3.2 on page 44](#).

² 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 3.2 on page 44](#).

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.

See Also

- “Examine Access” in *Using SAS Environment Manager Administration*
- “Set an Explicit Grant or Denial” in *Using SAS Environment Manager Administration*
- “How to Set a Row-Level Permission”

Access to In-Memory Data

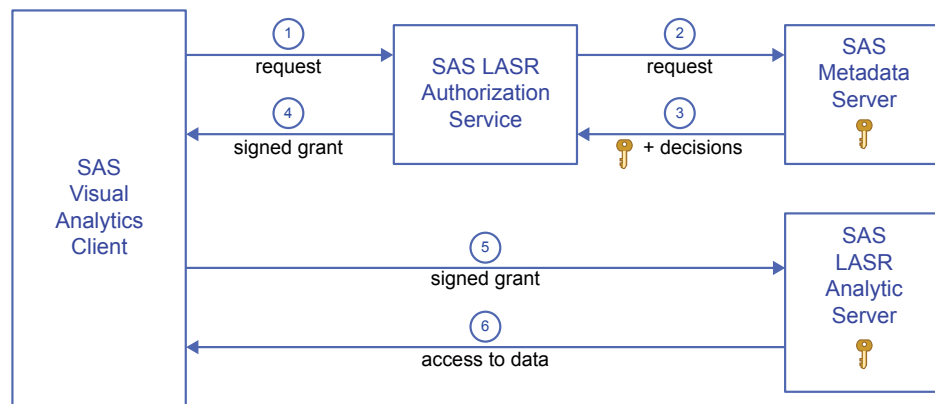
SAS LASR Authorization Service

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 LASR 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”.
 - the security key for the target LASR 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 LASR server.
- 6 The LASR 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 LASR server and the metadata server. LASR security keys are created and stored as follows:

- When a LASR server is started, a key is generated. In the LASR 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 LASR 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”](#).

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 LASR 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”](#).


Signature Files

Signature files are created when a LASR 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 tasks 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 LASR 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 the vertical navigation bar in [SAS Environment Manager Administration](#), select  **Servers**.
- 2 At the top of the navigation pane, select **SAS Servers** from the **View** drop-down list.
- 3 Select a LASR server. The server's definition is displayed.
- 4 On the **Options** tab, select **Advanced** ⇒ **Additional Options**.
- 5 In the **Additional Options** pane, the **Signature files location on server** field indicates the current path.

If necessary, click , and enter a different path.

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

Note: You cannot use a path that contains double-byte character set (DBCS) characters.

- 6 In the operating system, host-protect the directory as follows:

| | |
|---------|---|
| Windows | Limit Read and Write access as described above. |
| UNIX | 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”](#).

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 |
|---------------------------|--------------------------|
| <code>/hps</code> | <code>hps</code> |
| <code>/hps/special</code> | <code>hps.special</code> |
| <code>/sales</code> | <code>sales</code> |

- 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).
- 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. 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 table. To establish row-level security, you add constraints called permission conditions to explicit grants of the Read permission. 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 filter.

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

- ✔ Authorized
The requesting user can see all rows.
- 🕒 Row-level access
The requesting user can see only those rows that meet the specified permission condition (filter).
- ✘ Not authorized
The requesting user cannot see any rows.

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. Another condition on TableA limits Read permission for the SASUSERS group. The user is a member of both GroupA and SASUSERS. | 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. |
| If there are multiple permission conditions at the highest level of identity precedence, then any data | A condition on TableA limits Read permission for GroupA. | The user can see any row that is permitted for either GroupA or GroupB. |

| Principle | Scenario | Outcome and Explanation |
|--|---|-------------------------|
| that is allowed by any of the tied conditions is returned. | Another condition on TableA limits Read permission for GroupB. The user is a first-level member of both GroupA and GroupB. | |

How to Set a Row-Level Permission

- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), select the **Folders** page (📁).
- 2 In the navigation pane, select a LASR table.
The table's definition opens in the content pane.
- 3 On the table's **Authorization** tab, click .
The Edit Authorization window opens.
- 4 In the **Read** column, click the effective access icon for the identity whose row-level access you want to constrain.



Note: If that identity is not listed, click **+**. (An explicit grant of the ReadMetadata permission is automatically set for each identity that you add.)

- 5 In the Read pop-up window, select **Conditional grant**.
If **Conditional grant** is already selected, select **Conditional grant** again to view or edit the existing condition.
If **Conditional grant** is not listed, make sure that you are working with the Read permission for a LASR table.

TIP One way to verify that a table is a LASR table is to locate it on the **LASR Tables** tab. On the **LASR** page, select the **LASR Tables** tab. Use the **Filter by** control to search for the table by its name.


- 6 In the Edit Permission Condition window, create a filter expression for the selected identity. See [“Syntax for a Row-Level Permission Condition”](#).

Note: The Edit Permission Condition window does not validate the syntax that you enter.

- 7 Click **OK**.
In the Edit Authorization window, notice that the cell contains the conditional grant icon  with an explicit control indicator .

- 8 If you set a permission for a group, review its impact on the other listed identities. Constraints that you add for a group might affect access for members of that group.
- 9 Click **Save**.
- 10 In SAS Visual Analytics, verify that access is as expected.

TIP A LASR exception that indicates that a WHERE clause could not be parsed as written indicates a problem with a row-level permission expression.

TIP To ensure that results in SAS Visual Analytics reflect the current filter, refresh the data source after you open it. (In the **Data** pane, click , and select **Refresh table-name**.)


Batch Approach to Row-Level Permissions

As an alternative to using SAS Environment Manager Administration to create permission conditions, you can use the batch tools for metadata authorization. See [“Batch Tools for Metadata Authorization”](#) in *SAS Intelligence Platform: Security Administration Guide*.

Here is an example as submitted in the batch tools:

```
sas-set-metadata-access -profile Admin "/Shared Data/LASRtableA(Table)"
-grant sasusers:Read -condition "Age=14"
```

In SAS Visual Analytics, verify that results are as expected.

TIP To ensure that results in SAS Visual Analytics reflect the current filter, refresh the data source after you open it. (In the **Data** pane, click , and select **Refresh table-name**.)

Syntax for a Row-Level Permission Condition

Key Points

- Columns are not case sensitive.
- Operators are not case sensitive.
- Character values are case sensitive.

- Enclose character values in quotation marks.
- Do not qualify column names with a table name.
- Do not include months or dates in any expression.
- Do not include the WHERE keyword in any expression.
- Do not include a space between adjacent operators (for example, specify `<=`, not `< =`).
- Only those operators that are documented in this topic are supported.

Supported Operators

| Operator | Example |
|-------------------------------|--|
| - , + , * , / | Profit > (Sales * .5) |
| = , ^= , < , <= , > , >= , <> | Toy_Price=25 |
| And, Or, Not | Toy_Type='cars' OR Toy_Type='dolls' |
| In, Not In | Toy_Type IN ('dolls' 'cars' 'animals') |
| Like, Not Like | Toy_Type LIKE 'd%' |
| Between, Not Between | Toy_Price BETWEEN 20 AND 30 |
| Is Missing, Is Not Missing | Toy_Type IS MISSING |
| Contains (?), Not Contains | Toy_Type CONTAINS 'cars' |

Here are details:

- In the Edit Permission Condition window, you can select items or enter text directly.
- You can use abbreviations for the comparison operators (for example, NE instead of ^=).
- The BETWEEN and NOT BETWEEN operators are inclusive (they include the range boundary values).
- The LIKE operator tests for a matching pattern. The details are the same as for the LIKE condition.
- In the Edit Permission Condition window, the **Starts with** and **Ends with** items in the **Advanced Operators** list provide shortcuts to creating specific-purpose LIKE expressions.

Identity-Driven Conditions: Overview

Identity-driven properties are intended for use in conditions that are assigned to groups. All members of a group are subject to that group's condition (unless there is a relevant higher-precedence access control). The condition has a different effect on each member, based on the member's authenticated user ID, metadata name, external identity value, or memberships.

The following section 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.

Identity-Driven Conditions: Reference

| Property | Description and Examples |
|---------------------------|---|
| 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:</p> <pre>userid='SUB::SAS.Userid'</pre> |
| SUB::SAS.PersonName | <p>returns the requesting user's name (as specified in the Name field in the user's metadata definition).</p> <p>Here is an example:</p> <pre>name='SUB::SAS.PersonName'</pre> |
| 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> <p>Here is an example:</p> <pre>extid='SUB::SAS.ExternalIdentity'</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:</p> <pre>dept IN ('SUB::SAS.IdentityGroups')</pre> |

Here are additional details:

- Two additional properties (SAS.IdentityName and SAS.IdentityGroupName) are not documented here because they are less frequently useful. See [“Fine-Grained Controls for Data” in SAS Intelligence Platform: Security Administration Guide](#).

- The italics in the preceding examples reference a column in your LASR table. Your column name might be different.
- To create an identity-driven condition in the Edit Permission Condition window, enter the appropriate syntax. In the current release, there is no point-and-click interface for creating identity-driven filters.

Identity-Driven Conditions: Example

If a LASR table has an empID column with values that match the user IDs with which users authenticate, you might assign the following condition to the SASUSERS group:

```
empID= 'SUB::SAS.Userid'
```

At runtime, each user's ID is substituted into the right side of the condition. For example, the condition might resolve as follows:

For userA: empID= 'USERA'

For userB: empID= 'USERB'

As a result, each user gets only those rows where the value in the empID column matches their authenticated user ID, as follows:

For userA: USERA

For userB: USERB

TIP If the user IDs in your data contain lower case or mixed case characters, apply the upcase function on the left side of the expression (in this example: `upcase(userid)`).

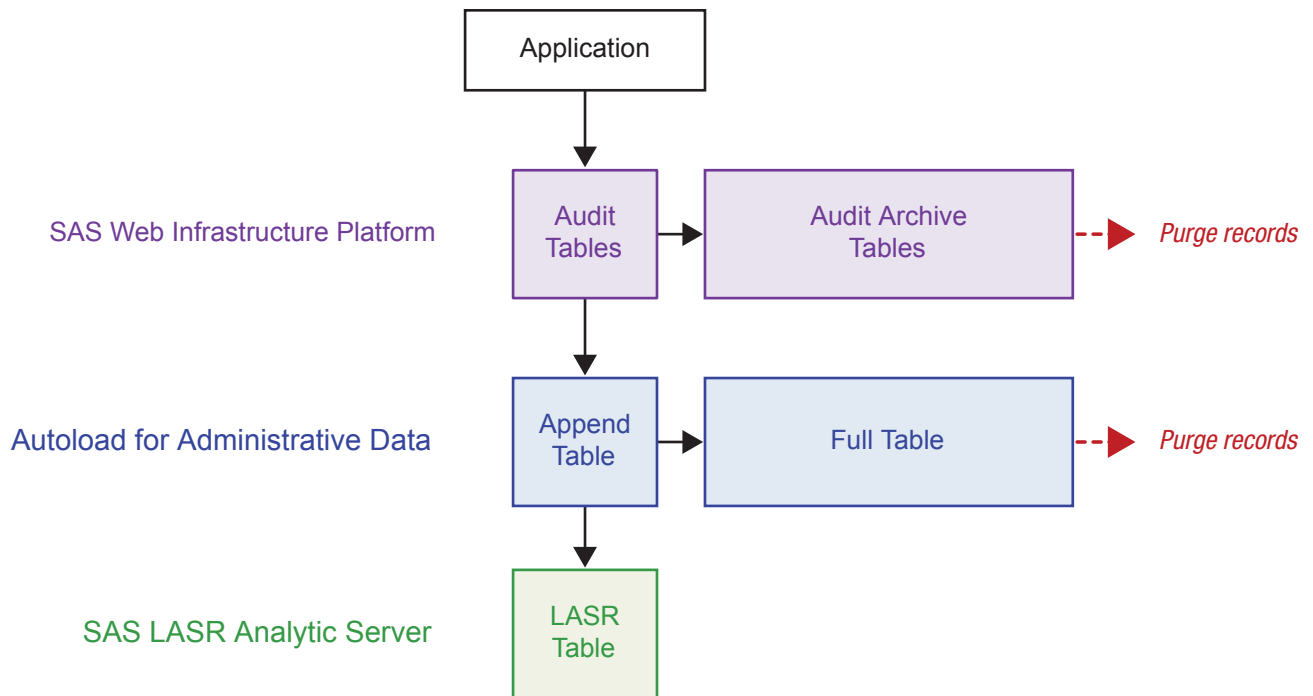
Key Actions Auditing

Introduction

This topic provides information that is specific to SAS Visual Analytics. For general information, see [“Configure Auditing for SAS Web Applications”](#) in *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 autoload data directory (drop zone). The extraction process writes the new data to the Append subdirectory of the EVDMLA autoload data directory. The scope of extraction consists of all records for a fixed set of object types. See "[Audit Content and Coverage](#)". Audit records for logon and logoff actions are not extracted.
- 3 The next time the [autoload process](#) runs, it notices that the audit data in the Append subdirectory is newer than the corresponding LASR table. The autoload 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 autoload data directory. This is standard autoload append behavior. See "[How Autoload Works](#)".

- To manage the size of the AUDIT_VISUALANALYTICS tables in the autoloader 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 archiving of audit records is configured. See ["Archive Process for Audit Records"](#) in *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 3.7 on page 63](#). 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 ["Purge Audit Records"](#) in *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"](#).
- 4 Set the property `va.AuditingEnabled` to `true`. See ["How to Set Configuration Properties"](#).
- 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 7, “Reports for Administrators”](#).
- 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 [“Limiting Space for Tables”](#).
- The `server_app` field is populated for actions that use the transport service. For example, when a user prints an object, the `executor_nm` value identifies the client (for example, `Visual Analytics Viewer 7.5`) and the `server_app` value identifies the underlying component (for example, `Visual Analytics Transport Service 7.5`).
- The `email_recipients` field is not populated for actions that are performed in SAS Visual Analytics Apps.
- For some of the specialized fields, `new` and `old` values are recorded. The `new` value reflects current information.

Table 3.5 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 Data Builder 7.5 |
| 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(Report) |
| lasr_server_name | Machine name and port of a LASR 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 7.5 |
| 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 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 3.6 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 |
| Access encrypted SASHDAT (use of passphrase). | [Library] or [Server.Hadoop] Read | client_id, library_name, or hadoop_server_name |

Table 3.7 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) |
| 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 *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](#)”.
- For a distributed server, each user’s home directory (~) to provide access to SSH keys. See “[Passwordless SSH](#)” in *SAS LASR Analytic Server: Reference Guide*.
- The directory where process IDs are written. See `va.monitoringPath`.
- The directory where last action logs are written. See `va.lastActionLogPath`.
- The directory that contains geographic data sets. See “[Geo Map Data Sets](#)”.
- The directory that contains the SAS linguistic files for text analytics. See “[Supporting Text Analytics](#)”.
- Any directory to which users export code. See the preference **Record actions as SAS statements** in [Preferences for Administrators](#).
- Any directory that serves as a data provider for reload-on-start. See “[Reload-on-Start](#)”.
- Any directory from which users import non-local data. See “[Self-Service Import](#)”.
- 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](#)”.

Mobile Device Access

This topic explains how to manage the availability of the native mobile applications for SAS Visual Analytics ([SAS Visual Analytics Apps](#)).

To get started, select  **Mobile Devices** from the vertical navigation bar in [SAS Environment Manager Administration](#).

Introduction

Use the **Mobile Devices** page to specify which devices can use mobile apps from SAS.

You can manage devices either by exclusion or by inclusion.

- If you manage by exclusion, all devices can use mobile apps from SAS except those that are on the blacklist. A blacklist is a list of mobile devices that are not authorized to use mobile apps from SAS.
- If you manage by inclusion, only devices that are on the whitelist can use mobile apps from SAS. A whitelist is a list of mobile devices that are authorized to use mobile apps from SAS.

Here are key points:

- The mobile device lists affect devices, not users. To manage what a particular user can see or do in mobile apps from SAS, use permissions and capabilities.
- A deployment enforces only one mobile device 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.

For related information, including capabilities, configuration properties, logging, and troubleshooting, see [Manage Devices](#) and “[Configuration Properties: Transport Services](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

About the Mobile Devices Page


Here are details and tips:

- By default, the **Logon History** tab displays the most recent sign-in event for each device. The following occurrences are sign-in 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)


To view previous sign-in events that are accompanied by a device change, select the **Include device history** check box.

- On the **Logon History** tab, the **Status** column provides information about a sign-in event. The **Status** column does not indicate the current status of a device connection. The **Status** column uses the following icons:

 Success

 Failure (authentication)

 Failure (device in the blacklist)

 Failure (device not in the whitelist)

 Unknown

- 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). 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.
- 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.

Add a Device to a List

- 1 Select the **Blacklist** or **Whitelist** tab.
- 2 Click **+**. Enter the device ID in the Add to Blacklist or Add to Whitelist window.
- 3 Click **Save**.

Here are additional details:

- To add multiple devices, click **+**.
- The device IDs that you enter are not validated by the software.

Note: A device ID is a unique identifier (usually a hardware device number) that is determined and communicated by the connecting mobile app.

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

Remove a Device from a List

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

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

Determine When a Device Was Blacklisted

- 1 Use the **Blacklist** tab to obtain the device ID.
- 2 On the **Management History** tab, select **Device ID** from the drop-down list.
- 3 Click in the text field, and enter the device ID.

TIP You can use the same process to determine when a device was whitelisted.

Advanced: Change How Devices Are Managed

CAUTION

These are deployment-level instructions that affect all access to mobile apps from SAS.

To switch from enforcing one list to enforcing the other:

- 1 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.
- 2 On the tab for the list that you want to enforce, click **Enable**. In the confirmation window, click **Yes**.

Note: In the standard configuration, only unrestricted users and members of the SAS Administrators group can perform this task. The ability to perform this task is controlled by WriteMetadata access to the software component **Visual Analytics Transport Service 7.5**.

Related Tasks

To require passcode protection, use a capability. See “[Lock SAS Visual Analytics App with a Passcode](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

To limit the duration of offline access, use a capability. See “[Use the Time-out Setting to Prevent Access](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

To prevent local caching of report data, use a capability. See “[Prevent Report Data from Being Cached on the Device](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

To enable your mobile apps to include SAS Visual Analytics content, see “[Mobile Software Development Kits](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

To resolve access issues, see “[Troubleshooting: SAS Visual Analytics App](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Authentication

Introduction

SAS Visual Analytics uses platform-level functionality for authentication. See “[Authentication Model](#)” in *SAS Intelligence Platform: Security Administration Guide*.

Note: For information about authentication for mobile devices, see “[Manage Devices](#)” in *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 *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 LASR server. Make sure the account has the privileges that are required to operate the server and load data. See [“Host Account Privileges”](#).
- 2 For each service account, create a SAS Application Server that contains a standard workspace server. See [“Add a New Server”](#).
- 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 *SAS Intelligence Platform: Security Administration Guide*.
- 4 For each SAS Application Server, create a corresponding LASR 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 [“LASR Server Definition”](#).
- 5 For each LASR server instance, create one or more LASR libraries. Assign each library to the SAS Application Server that corresponds to the library’s LASR server instance. See [“LASR Library Definition”](#) and [“Which Server is Used?”](#).
- 6 On the **Authorization** tab for each SAS Application Server and LASR server instance, limit ReadMetadata access. See [“Hide Server Definitions”](#) in *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

For information about the Policy.ConcurrentUserLogins property, see [“Disable Concurrent Sign-in Sessions”](#) in *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 *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*.

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 *SAS Intelligence Platform: System Administration Guide*.
- To use AES, SAS/SECURE must be installed and available. See [“Providers of Encryption”](#) 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”](#).

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](#)” in *SAS Guide to Metadata-Bound Libraries*.

- 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 the appropriate 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 *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”](#) in *SAS Guide to Metadata-Bound Libraries*. 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 LASR server to start must have the ReadMetadata and Select permissions on the corresponding secured table object.

7 To verify the results:

- In SAS Visual Data Builder or SAS Visual Analytics, 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 SAS Environment Manager Administration, stop and then start a LASR 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” in SAS Guide to Metadata-Bound Libraries](#).

Additional Information

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

For example:

- To update a passphrase, see [“Changing a Metadata-Bound Library’s Encryption Options” in SAS Guide to Metadata-Bound Libraries](#).
- To remove encryption, see [“Unbinding a Metadata-Bound Library” in SAS Guide to Metadata-Bound Libraries](#).

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 TKGrid Encryption Extension must be installed and available. See *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.
 - To use AES, SAS/SECURE must be installed and available. See “[Providers of Encryption](#)” 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 [“Establishing Connectivity to a SASHDAT Library”](#) in *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 *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”](#).
 - 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 LASR server.
 - For SASHDAT files in co-located HDFS, examine each table’s **Encryption** property on the **HDFS** tab. See [“About the HDFS Tab”](#).
 - 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 SAS Visual Data Builder to save the tables back to the target directory.

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

Fine-Tuning

| | |
|---|-----------|
| <i>Administering SAS Home</i> | 80 |
| <i>Supporting Guest Access</i> | 80 |
| <i>Supporting Text Analytics</i> | 80 |
| Introduction | 80 |
| Linguistic Files | 81 |
| Stop Lists | 81 |
| <i>Supporting Stored Processes</i> | 81 |
| <i>Supporting Report Distribution</i> | 82 |
| Introduction | 82 |
| About the Scheduling Server | 82 |
| About the Size of Email Attachments | 82 |
| <i>Supporting the Search Features</i> | 83 |
| <i>Supporting User-Defined Formats</i> | 84 |
| <i>Managing the Availability of Fonts</i> | 84 |
| <i>Managing Alerts and Notifications</i> | 85 |
| Introduction | 85 |
| Requirements | 85 |
| How to Delete Other Users' Alerts | 86 |
| <i>Supporting the Monitoring Features</i> | 86 |
| Introduction | 86 |
| Network Name Resolution | 87 |
| Managing the Monitor | 87 |
| Validating the Monitor | 88 |
| Logging for the Monitoring Server | 88 |
| <i>Supporting the HDFS Browse Features</i> | 89 |
| <i>Adjusting the Logging Configuration</i> | 89 |
| Log Directories | 89 |
| How to Change Log Levels | 90 |
| How to Log Submitted Code | 90 |
| How to Log LASR Actions | 91 |
| Logging of Access Denials | 91 |
| Logging for Platform Servers | 91 |

| | |
|--|-----------|
| Logging for SAS Visual Analytics Apps | 91 |
| See Also | 92 |
| Supporting Footers in Printed Reports | 92 |
| Using Multiple SAS Application Servers | 93 |
| Which Server is Used? | 93 |
| Add a New Server | 94 |
| Streamlining Access to Management Pages | 95 |
| Home Page Method | 95 |
| Applications Menu Method | 96 |
| See Also | 97 |
| Customizing the Appearance of Applications or Reports | 97 |

Administering SAS Home

See “The SAS Visual Analytics Home Page” in *SAS Intelligence Platform: Web Application Administration Guide*.

Supporting Guest Access

See “Configure Guest Access” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Supporting Text Analytics

Introduction

This topic provides information to help administrators support text analysis features. For user instructions, see “Working with Text Topics” in *SAS Visual Analytics: Working with Report Content*.

Linguistic Files

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

Stop Lists

To omit certain words from text analysis that is performed by a LASR server, register and load a stop list for that server. For example, you can filter out noise by omitting commonly used words.

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

Supporting Stored Processes

A stored process is a SAS program that is stored on a server and defined in metadata. For information about how to create and register a stored process, see [“Managing Stored Process Metadata”](#) in *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 LASR 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 Report Distribution

Introduction

This topic helps administrators support the report distribution feature that SAS Visual Analytics provides.

Here are the key points:

- In SAS Visual Analytics, only those users who have the Distribute Reports capability can distribute reports. See [“Distribute Reports”](#).
- Use only SAS Visual Analytics to schedule and distribute reports. (If you use the Schedule Manager plug-in to SAS Management Console, and you set an option that is not available in SAS Visual Analytics, the report job might be incompatible with that application.)
- 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”](#).
- 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`](#).

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".

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 "Administer the Search Facility" in *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.

SAS Visual Analytics and the report 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”](#).

Other clients (such as SAS Visual Data Builder) 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 [“Establishing Connectivity to a Library of SAS Data Sets”](#) in *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 *SAS Intelligence Platform: Application Server Administration Guide*.

Managing the Availability of Fonts

See [“Make More Fonts Available”](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Managing Alerts and Notifications

Introduction

This topic provides information to help administrators manage the data-driven alerts that users can add to reports in SAS Visual Analytics.

Requirements

Evaluation of data-driven alerts requires the following conditions:

- The associated LASR 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 *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 “[Set Global Properties for SAS Applications](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

.....
Note: With this approach, alert recipients should leave **Administrator default** 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, click the right-most icon in the SAS Visual Analytics banner, select **Settings**, and then select **SAS Report Viewer** ⇒ **Alert Notifications** in the navigation pane.

- 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 the va.Alert.* [configuration properties](#).

How to Delete Other Users' Alerts

To delete other users' alerts:

- 1 From the vertical navigation bar in SAS Environment Manager Administration, select **Tools**.
- 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. See "[Roles and Capabilities](#)".
- Anyone who can access the **Alerts** tab can delete all listed alerts.
- Alerts are stored in the SharedServices database in the middle tier.

Supporting the Monitoring Features

Introduction

Note: This topic is applicable to only distributed LASR servers.

On the **LASR** page in SAS Environment Manager Administration, the following features depend on the LASR Analytic Server Monitor (the monitor):

- table details on the **Monitor Processes** tab
- per-instance memory gauges on the **LASR Servers** tab

Network Name Resolution

Successful functioning of the monitor 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 LASR server.

Managing the Monitor

Commands

Use the following commands to manage the monitor:

| Host | Command |
|---------|---|
| UNIX | <code>LASRMonitor.sh</code> <code>start</code> <code>stop</code> <code>restart</code> <code>status</code> |
| Windows | <code>LASRMonitor.bat</code> <code>start</code> <code>stop</code> <code>restart</code> <code>status</code> <code>pause</code> <code>resume</code> |

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

Note: On Windows, the monitor 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 account under which the monitor runs must have passwordless SSH access to all machines on the cluster.

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

- 1 Determine which Windows account the service runs as. Here are sample instructions:
 - a Select **Start** ⇒ **Control Panel** ⇒ **Administrative Tools** ⇒ **Services**.
 - b Right-click on the service, and select **Properties**.
 - c On the **Log On** tab, next to the **This account** radio button, note the user ID.
- 2 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 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.

Validating the Monitor

See [“Validating Your SAS LASR Analytic Server Monitor Deployment” in SAS Visual Analytics: Installation and Configuration Guide \(Distributed SAS LASR\)](#).

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 monitor 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.
```

Supporting the HDFS Browse Features

This topic applies to distributed LASR servers that use co-located HDFS. See [“Distributed Server: Co-located HDFS”](#).

On the **Tools** page in SAS Environment Manager Administration, the **HDFS** tab provides a host-layer view of folders and files in the Hadoop file system. This feature is an alternative to using a third-party HDFS console.

To communicate with Hadoop, SAS Environment Manager Administration uses the client-side helper functionality of the SAS Plug-ins for Hadoop (the plug-ins). To launch the plug-ins, SAS Environment Manager Administration depends on SAS LASR Analytic Server monitor (the monitor). For configuration instructions, see [“Post-Deployment Tasks for Co-located HDFS” in SAS Visual Analytics: Installation and Configuration Guide \(Distributed SAS LASR\)](#).

To support the HDFS browse features:

- Make sure the monitor is running. See [“Managing the Monitor”](#).
- Make sure the process owner for the web application server that hosts SAS Environment Manager Administration can use passwordless SSH to access the Hadoop namenode.
- Make sure appropriate access to Hadoop is available.

Note: On the **HDFS** tab in SAS Environment Manager Administration, all users see all information that is available to one designated account. The displayed information is not adjusted to reflect each individual requesting user’s metadata or host permissions. The **HDFS** tab should be available only to administrators.

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).

| Host | Example Location |
|---------|---|
| UNIX | <code>/opt/sas/config/Lev1/Documents/Instructions.html</code> |
| Windows | <code>C:\sas\Config\Lev1\Documents\Instructions.html</code> |

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

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 [“Administer Logging for SAS Web Applications”](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

.....
Note: As an alternative to making a dynamic change, you can directly edit an `application-log4j.xml` file. See [“Use log4j Files”](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

How to Log Submitted Code

To include SAS logs in the logs for SAS Visual Data Builder and SAS Environment Manager Administration:

- 1 In the `SASVisualDataBuilder-log4j.xml` file and the `SASEnvironmentMgrMidTier-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 the preference **Record actions as SAS statements** in [“Administrative Preferences”](#).

How to Log LASR Actions

To log commands that are issued to a LASR server from SAS Visual Analytics or SAS Visual Analytics Report 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 SAS Environment Manager Administration and SAS Visual 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.

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

Logging for Platform Servers

See “Administering Logging for SAS Servers” in *SAS Intelligence Platform: System Administration Guide*.

Logging for SAS Visual Analytics Apps

The logging configuration for SAS Visual Analytics Apps 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 Visual Analytics Apps. See “[Configuration Properties: Transport Services](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

See Also

- Information about the Last Action Log in “[Administer LASR Tables](#)”
- “Logs” in *Using SAS Environment Manager Administration*

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 Sign in to SAS Management Console as an administrator (for example, `sasadm@saspw`).
- 2 Expand the **Application Management** folder, and select **SAS Application Infrastructure** ⇒ **Visual Analytics 7.5** ⇒ **Visual Analytics Services 7.5** ⇒ **Visual Analytics Transport Service 7.5**.
- 3 Right-click **Visual Analytics Transport Service 7.5**, and select **Properties**.
- 4 In the Visual Analytics Transport Service 7.5 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 hexadecimal 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>
```

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 LASR server, follow the associations from the target LASR 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](#).
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 SAS Environment Manager Administration and SAS Visual Data Builder, users can opt out of automatic selection and instead force the use of a particular server. See the preference **Application server** in [Preferences for Administrators](#).

Note: For example, if you schedule data queries in SAS Visual 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 [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 [SAS Intelligence Platform: Middle-Tier Administration Guide](#).
- To support geo maps, 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 SAS Visual Analytics, or SAS Visual Data Builder. (The pooled workspace server pre-screens available SAS/ACCESS licenses and then populates the **Import Data** panel.)
- To support scheduled queries in SAS Visual Data Builder, include a SAS DATA Step Batch Server within the SAS Application Server.
- To support scheduled report distribution in SAS Visual Analytics, include a Java batch server within the SAS Application Server.
- To support customized formats for data, see “[Supporting User-Defined Formats](#)”.
- To support interactions with a LASR 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 LASR server to specify an alternate location. See [va.monitoringPath](#) on page 178.

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](#)”.



Streamlining Access to Management Pages

In the initial configuration, non-administrators have access to the **Folders**, **Users**, and **Libraries** pages in SAS Environment Manager Administration. However, only administrators can access SAS Environment Manager Administration from SAS Visual Analytics. If you want to provide seamless partial access for non-administrators, use either of the following methods.

Home Page Method

The easiest way to give non-administrators seamless access to SAS Environment Manager Administration is to add a shared link on the home page.

- 1 Verify which pages are currently available to non-administrators.
 - a On the **Users** page in SAS Environment Manager Administration, create a test user that has only an [internal account](#), such as test@saspw. Make the test user a member of the Visual Analytics Users group.
 - b At the direct URL for SAS Environment Manager Administration (see [Step 2d](#)), sign in as the test user.

Make sure only the expected pages are available in the vertical navigation bar.
- 2 Add a shared link to the home page.
 - a Sign in to the home page as an administrator (for example, sasadm@saspw).
 - b In the **Links** tile, click , and select **Edit Shared Links**.
 - c In the Edit Shared Links window, click .
 - d In the **Details** column, enter a label and your equivalent of the following URL: *machine.company.com/SASEnvironmentMgrMidTier/*.

TIP For the exact URL, see “Direct Access” in *Using SAS Environment Manager Administration*.

- e Click **Save**.
- 3 Verify that results are as expected for the test user.
 - 4 On the **Users** page in SAS Environment Manager Administration, delete the test user.


This method has the following limitations:

- This method is not selective. The shared link is visible to all users.
- Each time you click the link, a new browser tab or window opens.
- This method does not provide return access through the applications menu. For example, after you navigate from SAS Environment Manager Administration to the home page, you cannot use the applications menu to return to SAS Environment Manager Administration.

Applications Menu Method

To provide access to SAS Environment Manager Administration from the applications menu, the Manage Environment capability is required.

For example, if you want all users of SAS Visual Analytics to be able to open SAS Environment Manager Administration from the applications menu, complete the following steps:

- 1 Preview the effects of providing the Manage Environment capability.
 - a On the **Users** page in SAS Environment Manager Administration, create a test [role](#) that provides only the **Visual Analytics** ⇒ **Advanced** ⇒ **Manage Environment** capability.
 - b Create a test user that has only an [internal account](#), such as test@saspw. Make the account a member of the Visual Analytics Users group and the test role.
 - c At the direct URL for SAS Environment Manager Administration, sign in as the test user. Verify that only the expected pages are available.
 - d In the applications menu (☰), verify that the **Administrator** item is present. Select that item, and verify that SAS Environment Manager Administration opens to the **Folders** page.
- 2 On the **Users** page in SAS Environment Manager Administration, delete the test role.
- 3 Open the Visual Analytics Report Viewing role. On the **Capabilities** tab, click . In the Edit Capabilities window, select the check box for the **Visual Analytics** ⇒ **Advanced** ⇒ **Manage Environment** capability.
- 4 Sign in as the test user. Verify that results are as expected.

- 5 On the **Users** page in SAS Environment Manager Administration, delete the test user.

CAUTION

The Manage Environment capability provides access to the Alerts tab. You cannot selectively hide the **Alerts** tab (on the **Tools** page of SAS Environment Manager Administration). Anyone who can access the **Alerts** tab can delete all listed alert definitions.

See Also

- [“Capabilities” in *Using SAS Environment Manager Administration*](#)
- [“Create an Internal Account” in *Using SAS Environment Manager Administration*](#)
- [“Assign Capabilities to a Role” in *Using SAS Environment Manager Administration*](#)
- [“Revoke Access to Management Pages” in *Using SAS Environment Manager Administration*](#)
- [“The SAS Visual Analytics Home Page ” in *SAS Intelligence Platform: Web Application Administration Guide*](#)
- [“Who Can Launch SAS Environment Manager Administration?” in *Using SAS Environment Manager Administration*](#)

Customizing the Appearance of Applications or Reports

See [SAS Theme Designer: User's Guide](#).

Geographic Maps

| | |
|---|------------|
| Geo Map Server | 99 |
| Introduction | 99 |
| OpenStreetMap Server | 100 |
| Esri Server | 101 |
| Geo Map Data Sets | 102 |
| Custom Polygons: Prerequisites | 102 |
| Introduction | 102 |
| General Prerequisites | 103 |
| Considerations for Non-English Character Data | 103 |
| Custom Polygons: Instructions | 103 |
| Overview | 103 |
| MAPSCSTM Library | 104 |
| Deployment Program | 104 |
| Output Data Sets | 106 |
| Verification | 106 |
| Custom Polygons: Data Sets | 107 |
| ATTRLOOKUP Data Set | 107 |
| CENTLOOKUP Data Set | 109 |
| Custom Polygon Data Set | 109 |
| Custom Polygons: Examples | 111 |
| Example: German Postal Codes | 111 |
| Example: US Counties | 116 |

Geo Map Server

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 "Configure the Cross Domain Proxy Servlet through a Whitelist" in *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.

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.

To change the protocol that SAS Visual Analytics uses to connect to the OpenStreetMap servers that SAS hosts, set the property [va.SASGeomapCommunicationProtocol](#).

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.GeoMapServerUrl](#).
 - 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, SAS Visual Analytics provides user preferences and per-object settings that determine which server (OpenStreetMap or Esri) is used.

To reference an Esri server from SAS Visual Analytics:

- 1 Set the property `va.SASGeomapEsriURL`.
 - 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.

Geo Map 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.

See “Working with Geo Maps” in *SAS Visual Analytics: Working with Report Content*.

Custom Polygons: Prerequisites

Introduction

To add custom polygon data into an existing SAS Visual Analytics environment, there are certain requirements. By default, SAS Visual Analytics supports country- and state-level polygons for regional overlays in geographic objects. 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.

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 104](#).

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 LASR 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 LASR 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*.

Custom Polygons: Instructions

Overview

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.

To create custom polygons:

- 1 Make sure you have a current backup of the [ATTRLOOKUP](#) and [CENTLOOKUP](#) tables. These instructions modify those tables.

- 2 Create the library that contains your custom polygon data sets. See “[MAPSCSTM Library](#)”.
- 3 Obtain the polygon data that you want to add. The data must be in Esri shapefile file format. See “[Custom Polygon Data Set](#)”.
- 4 Download the custom polygon deployment program at support.sas.com/rnd/datavisualization/vageo/va74polygons.sas.
- 5 In the custom polygon deployment program, define the library references and macro variables for your custom polygons.
- 6 If necessary, modify the PROC SQL statement in step 2 of the custom polygon deployment program to ensure that the lookup values are unique.
- 7 Run the custom polygon deployment program.
- 8 To make the custom polygons available in SAS Visual Analytics, restart the SAS Web Application Server. See “[Understanding SAS Web Application Server Management](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.
- 9 Verify the results by creating a geo map object that uses the custom polygons. See “[Verification](#)”.

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 uses multiple server contexts, then add the LIBNAME statement to the file for each server context, including the default SASApp 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.

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.

Note: The `va74polygons.sas` program is appropriate for use in SAS Visual Analytics 7.5.

Note: The [example programs](#) in this chapter also feature the custom polygon data.

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 must be manually created. For more information, see [“MAPSCSTM Library”](#).

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: US Counties”](#) 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.

Verification

After the program has been successfully run, and the web application server has been restarted, SAS Visual Analytics has access to the new polygons. The program has output a data set in the MAPSCSTM library that contains a single record for each region generated by the program. The following steps enable the testing of individual or multiple polygons based on assigned filters.

- 1 Sign in to SAS Visual Analytics as a user who has data upload capabilities.
- 2 Create a new report.
- 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**.
- 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 object, 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 object 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.

Custom Polygons: 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 7.5.

ID3NAME

identifies the region that level 3 subdivisions belong to. This variable is not used in SAS Visual Analytics 7.5.

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 7.5.

ID3

specifies the ID value of the region that a level 3 subregion belongs to. This variable is not used in SAS Visual Analytics 7.5.

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 objects 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 must 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 must 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 objects 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.

Custom Polygons: Examples

Example: 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\Levl\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=note; /* 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 object */
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: 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\Levl\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 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 object */
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);
        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;
```


SAS LASR Analytic Server

| | |
|--|------------|
| About LASR Servers | 124 |
| Introduction | 124 |
| Distributed or Non-distributed? | 124 |
| Non-distributed Server Dependencies | 125 |
| LASR-Related Metadata | 125 |
| In-Memory LASR Names | 126 |
| LASR Server Definition | 127 |
| Introduction | 127 |
| Add a LASR Server | 127 |
| Set LASR Server Options | 129 |
| Set LASR Server Extended Attributes | 129 |
| LASR Server: Options Tab | 129 |
| LASR Server: Connections Tab | 133 |
| LASR Server: Extended Attributes | 134 |
| See Also | 134 |
| LASR Library Definition | 135 |
| Introduction | 135 |
| Add a LASR Library | 135 |
| Set LASR Library Options | 137 |
| Set LASR Library Extended Attributes | 137 |
| LASR Library: Options Tab | 138 |
| LASR Library: Data Server Tab | 139 |
| See Also | 140 |
| Limiting Space for Tables | 140 |
| Introduction | 140 |
| Over Capacity | 140 |
| How to Set a Tables Limit | 141 |
| Distributed Server: Monitoring | 141 |
| Memory Gauges | 141 |
| Resource Monitor | 142 |
| Process Monitor | 144 |
| Memory Usage: A Closer Look | 145 |
| Distributed Server: Parallel Load | 146 |
| Introduction | 146 |

| | |
|---|------------|
| Method: Co-located Storage | 147 |
| Method: NFS-Mounted Storage | 147 |
| Method: SAS Embedded Process | 147 |
| Example Depictions | 148 |
| Distributed Server: Co-located HDFS | 150 |
| Introduction | 150 |
| About the HDFS Tab | 150 |
| How to Introduce an Additional Directory | 155 |
| How to Delete an HDFS Table | 156 |
| See Also | 156 |
| Distributed Server: High-Volume Access to Smaller Tables | 156 |
| Introduction | 156 |
| Instructions | 157 |
| Extended Attribute | 157 |
| Example | 158 |

About LASR Servers

Introduction

SAS LASR Analytic Server (LASR 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 [SAS LASR Analytic Server: Reference Guide](#).

Distributed or Non-distributed?

A LASR server can be distributed or non-distributed.

- A distributed LASR server runs on multiple blades in a chassis.
- A non-distributed LASR 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 **Monitor Resources** tab, or the **Monitor Processes** tab.

Non-distributed Server Dependencies

When a non-distributed server is started interactively (in SAS Environment Manager Administration) 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 LASR 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 (SASServer_{n_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 LASR server include the following:

LASR Analytic Server

a metadata definition for a LASR server process

LASR Analytic Server connection

a metadata representation of one instance of a LASR server

LASR library

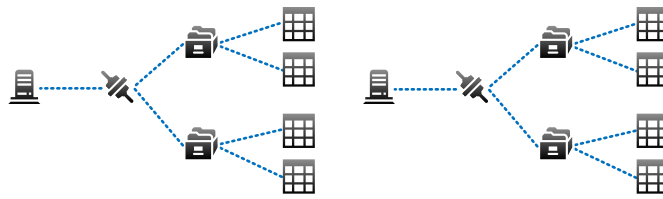
a metadata representation of a data library that is associated with a LASR server connection

LASR table

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

The following figure depicts the relationships among these metadata objects.

Figure 6.1 Server, Connection, Libraries, and Tables



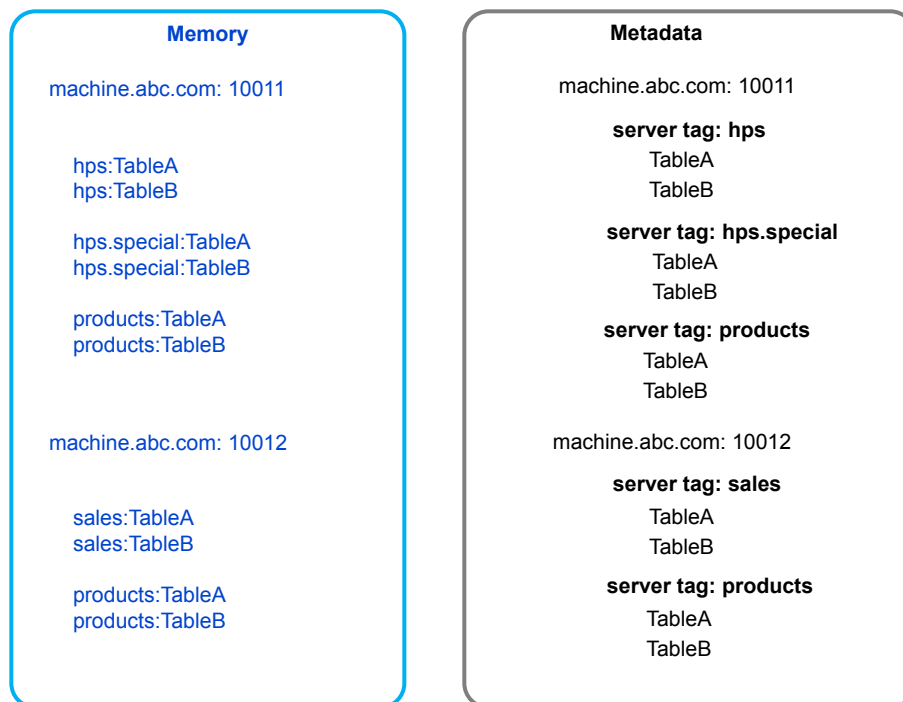
The preceding figure illustrates these points:

- 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 6.2 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](#)”.
- 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 LASR server. For example, when you unload a table from memory, the corresponding table object is not deleted from metadata.

LASR Server Definition

Introduction



This topic documents the metadata definition for a LASR server.

Defining an additional instance of a LASR server in metadata facilitates separation for management and ease-of-use purposes.


- Each server instance has a unique, multi-user connection to the hardware on which the LASR 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.

Note: This topic assumes that you are using SAS Environment Manager Administration. For SAS Management Console instructions, see “Establishing Connectivity to a SAS LASR Analytic Server” in “[Connecting to Common Data Sources](#)” in *SAS Intelligence Platform: Data Administration Guide*.

Add a LASR Server

- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), click  **Servers**.
- 2 In the navigation pane on the **Servers** page, click .
- 3 In the New Server window, complete these steps:
 - a Enter a name for the new server.
 - b In the **OPTIONS** section, enter information as follows:

| | |
|--|---|
| High-Performance Analytics environment host | Enter the fully qualified machine name of the host (for example, <code>va.abc.com</code>). |
| Port number | Enter the number of the port on which this instance of the server listens. Note: Uniqueness requirements apply. See “In-Memory LASR Names” . |
| Single machine server | If the LASR server is contained on a single machine, select Yes . For a distributed server, leave the default setting (No) in place. |
| High-Performance Analytics environment install location | Enter the host path where the files that define the cluster are located (for example, <code>/opt/TKGrid</code>). Note: For a distributed server, this field is required. For a non-distributed server, this field is not used. |
| Associated machine | Select the machine on which the LASR server runs. If the machine is not already listed, click New Machine to register a machine. |


- c Click **Save**. The new server definition is displayed.
- 4 On the new server’s **Options** tab, click . Make changes as needed. See [“LASR Server: Options Tab”](#).

IMPORTANT In the **Options: Advanced: Additional Options** pane, make sure the **Signature files location on server** field references an appropriate directory. See [“Signature files location on server”](#).


- 5 On the new server’s **Authorization** tab, review access to the new server. If necessary, adjust access. See the **Server** column in the [“Permissions by Task”](#) tables.

IMPORTANT Only users who have the Administer permission for a server can stop that server or set that server’s tables limit. Consider adding a grant of the Administer permission for the Visual Analytics Data Administrators group. See [“Set an Explicit Grant or Denial”](#) in *Using SAS Environment Manager Administration*.

Set LASR Server Options

- 1 On the [Servers](#) page in [SAS Environment Manager Administration](#), select a LASR server.
- 2 On the server's **Options** tab, click .
- 3 In the Edit Server Options window, make changes as needed. See [“LASR Server: Options Tab”](#).

Set LASR Server Extended Attributes

- 1 On the [Servers](#) page in [SAS Environment Manager Administration](#), select a LASR server.
- 2 On the server's **Advanced** tab, select **Extended attributes** from the drop-down list.
- 3 Click .
- 4 In the Edit Extended Attributes window, make changes as needed. See [“LASR Server: Extended Attributes”](#).

LASR Server: Options Tab

Where applicable, the corresponding PROC LASR statement option is provided in parentheses following the field name.

Options

Single machine server

specifies the type of LASR server. If the LASR server is contained on a single machine, select **Yes**. For a distributed server, select **No**.

High-Performance Analytics install location

for a distributed server, specifies the host path where files that define the cluster are located (for example, `/opt/TKGrid`).

.....
Note: This field is not applicable to a non-distributed server.

.....
Note: This setting is in the **Options: Distributed Server Options** pane.

Number of machines to use

for a distributed server, specifies the number of machines in the cluster to use for the server. To use all of the machines in the cluster, specify `ALL`.

.....
Note: This field is not applicable to a non-distributed server.

.....
Note: This setting is in the **Options: Distributed Server Options** pane.

Options: Advanced

Version Information

These fields provide descriptive metadata. It is not necessary to adjust the values in these fields.

Memory Limits

These settings affect the circumstances in which a distributed LASR server rejects certain tasks. For details and guidance, see [“Memory Management” in SAS LASR Analytic Server: Reference Guide](#).

Data loading (%) (TABLEMEM=)

specifies a threshold for physical memory utilization (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.

The initial value of 80 prevents tables from being loaded when memory use exceeds 80%. If this field is blank, the server-level default value (75) is used.

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

.....
Note: This field is not applicable to a non-distributed server.

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

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.

This is a memory utilization threshold for SAS High-Performance Analytics procedures and other external processes. The initial value of **80** prevents SAS High-Performance Analytics procedures from retrieving data when memory use exceeds 80%. If this field is blank, the server-level default value (75) is used.

Note: This field is not applicable to a non-distributed server.

Logging Options

For background information, see “[Server Logging](#)” in *SAS LASR Analytic Server: Reference Guide*.

Enable logging

specifies whether logging is enabled for the LASR server.

Path to log files

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

Note: This field is not applicable to a non-distributed server. For a non-distributed server, log files are always written to the signature files directory.

Maximum file size (MB)

specifies the maximum size of the log file (in megabytes). When the log file reaches the size limit, the contents are moved to a rollover file. The default value is 100.

Note: This field is not applicable to a non-distributed server.

Maximum rollover files

specifies the maximum number of rollover files that are retained. When the limit is reached, the oldest file is overwritten. The default value is 10.

Note: This field is not applicable to a non-distributed server.

Keep log files when the server terminates

specifies whether to keep server log files if the server terminates. To leave log files in the file system when the server terminates, select **Yes**. The default value is **No** (by default, the files are removed).

Note: This field is not applicable to a non-distributed server.

Additional logging parameters

specifies any additional logging parameters for the server.

Note: This field is not applicable to a non-distributed server.

Additional Options

Vendor

provides optional descriptive information.

Associated machine

identifies the machine on which the LASR server runs. You cannot edit this value in an existing server definition.

Note: If the machine is not already listed, click **New** to register a machine.

Force overwrite of server description file

specifies that the server should be started even if the server description file already exists. The server process that is in the existing server description file is stopped, and then the file is overwritten with the details for the new server.

Note: If the server is for SAS Visual Analytics, this setting is not used.

Note: This field is not applicable to a non-distributed server.

Signature files location on server

specifies the location where the SAS High-Performance Analytics environment writes signature files. The location is set during installation. Here are key points:

- It is important to protect the specified directory. See [“Signature Files”](#).
- Do not use a signature files path that contains double-byte character set (DBCS) characters.
- For a distributed server, the specified path must be located on the SAS High-Performance Analytics environment root node.
- For a non-distributed server, the path is located on the local machine.

For more information, see [“Signature Files” in SAS LASR Analytic Server: Reference Guide](#).

Server lifetime (LIFETIME=)

specifies the duration of the server process. If this field is blank, the server runs indefinitely, which 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.

If you specify a single value, the server runs for the specified length of time and stops. For example, if you specify a value of 7200 (seconds), the server runs for two hours and then stops.

If you specify a value of *run_time (inactivity_time)*, the server runs for the specified *run_time*, and continues until it has been inactive for the specified

inactivity_time. For example, if you specify a value of 3600 (1200), the server runs for one hour, and continues to run until it has been inactive for 30 minutes.


Note: The (*inactivity_time*) feature is applicable to only a distributed server.

Display detailed diagnostics (VERBOSE)

specifies whether all of the information that is returned by the grid host is displayed. Use detailed information to diagnose setup problems.

Note: This field is not applicable to a non-distributed server.

LASR Server: Connections Tab

Each server instance has one connection object, which defines a multi-user connection. If you need to make changes to the connection definition, right-click the connection object and select **Open**. On the connection's **Options** tab, click .

The **Options** tab for a LASR server's connection object includes the following fields:

Port number

specifies the number of the port on which this instance of the server listens.

Note: Uniqueness requirements apply. See [“In-Memory LASR Names” on page 126](#).

Note: This setting is in the **Options** pane.

High-Performance Analytics Environment host

specifies the fully qualified name of the host machine (for example, `va.abc.com`).

Note: This setting is in the **Options** pane.

LASR authorization service location

specifies whether the server requires the LASR authorization service, and, if so, provides that service's URL. If the server is for SAS Visual Analytics, make sure the check box is selected. See [“SAS LASR Authorization Service” on page 46](#).

Note: This setting is in the **Options: LASR Authorization Service Location** pane.

Server description file (CREATE=)

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

Note: This setting is in the **Options: Advanced: Additional Options** pane.

LASR Server: Extended Attributes

VA.MonitoringPath

specifies a custom directory for monitoring artifacts for this LASR 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 SAS Environment Manager Administration. See ["Limiting Space for Tables"](#).

VA.TextAnalyticsBinaryLocation

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

| Host | LASR Server | Examples |
|---------|-----------------|--|
| UNIX | non-distributed | <code>/SASFoundation/Version/misc/tktg</code> |
| | distributed | <code>/opt/TKTGDat</code> |
| Windows | non-distributed | <code>\SASFoundation\Version\tktg\sasmisc</code> |

VA.TextAnalyticsStopList

specifies the location of an English language stop list that SAS provides. See ["Supporting Text Analytics"](#).

VA.TextAnalyticsStopList.de

specifies the location of a German language stop list that SAS provides. See ["Supporting Text Analytics"](#).

TIP Extended attributes for autostart, autoload, and reload-on-start are at the library level, not the server level. See [Table 2.1](#).

See Also

- ["Limiting Space for Tables"](#)
- ["Supporting Text Analytics"](#)
- ["LASR Page" in Using SAS Environment Manager Administration](#)

- “Servers Page” in *Using SAS Environment Manager Administration*

LASR Library Definition

Introduction




This topic documents the metadata definition for a LASR library.

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.

Note: This topic assumes that you are using SAS Environment Manager Administration. For SAS Management Console instructions, see “Establishing Connectivity to a SAS LASR Analytic Server” in “[Connecting to Common Data Sources](#)” in *SAS Intelligence Platform: Data Administration Guide*.

Add a LASR Library

- 1 In the vertical navigation bar in [SAS Environment Manager Administration](#), click  **Libraries**.
- 2 On the **Libraries** page, select **High-Performance Analytics libraries** from the drop-down list. If an existing LASR library meets the current need, do not add a new library. Otherwise, proceed to the next step.
- 3 On the **Libraries** page, click .
- 4 In the New Library window, complete these steps:
 - a Enter a name for the new library (for example, `sales LASR`).
 - b To specify the metadata location for the new library definition, click , and select a folder.
 - c Select the type **SAS LASR Analytic Server Library**.
 - d In the **OPTIONS** section, enter information as follows:

| | |
|---------------|--|
| Libref | Specify a one- to eight-character identifier of your choice (for example, <code>saleslib</code>). |
|---------------|--|

Note: The first character must be a letter or an underscore. Other characters can be letters, numbers, or underscores.

Server tag

Specify a string that is unique within the server instance (the server's host-port combination). See [“In-Memory LASR Names”](#).

In most cases, you can simply use the libref value as the server tag.

If the library's data is loaded using SAS Embedded Process or is loaded from a co-located provider, additional requirements apply. See [Server Tags](#).

Note: If you do not enter a value, the value `WORK` is used.

Note: You cannot edit the **Engine** value (`SASIOIA`), which is the engine for all LASR libraries.


- e In the **DATA SERVER** section, assign the new library to a LASR server.

Note: If there is no appropriate server definition, click **New Server**, and create a new LASR server definition. See [Add a LASR Server](#).

- f Click **Save**. The new library definition is displayed.

- 5 (Optional) On the new library's **Assigned SAS Servers** tab, click . Assign the library to one or more SAS Application Servers.

Note: Assigning a LASR library to a SAS Application Server facilitates interactions from clients such as SAS Enterprise Guide. Assignments can affect which SAS Application Server is used for interactions with this LASR library and its associated LASR server. See [“Using Multiple SAS Application Servers”](#).

- 6 On the new library's **Options** tab, review the settings. If necessary, click , and make changes. See [“LASR Library: Options Tab”](#).



Note: If you specified a libref that has already been assigned, you must set options to indicate the library is pre-assigned.

- 7 On the new library's **Authorization** tab, review access to the new library. If necessary, adjust access. See the **Library** column in the [Permissions by Task](#) tables.



IMPORTANT Only users who have the Administer permission for a library can load new tables to that library. Consider adding a grant of the Administer permission for an appropriate group. See [“Set an Explicit Grant or Denial”](#) in *Using SAS Environment Manager Administration*.

- 8 (Optional) Review the following convenience and performance features, and configure any that are appropriate for the new library.
 - If you want the associated server to start on demand for data load and import requests for the new library, enable autostart for the library. See [“Enable Autostart For a LASR Server”](#).
 - 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”](#).
 - If you want to automatically synchronize the library’s in-memory data with source tables in a host directory, set up a corresponding implementation of autoloading. See [“Autoload”](#).
 - 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”](#).

Set LASR Library Options

- 1 On the  **Libraries** page in [SAS Environment Manager Administration](#), select a LASR library.
- 2 On the library’s **Options** tab, click .
- 3 In the Edit Library Options window, make changes as needed. See [“LASR Library: Options Tab”](#).

Set LASR Library Extended Attributes

- 1 On the  **Libraries** page in [SAS Environment Manager Administration](#), select a LASR library.
- 2 On the library’s **Advanced** tab, select **Extended attributes** from the drop-down list.
- 3 Click .
- 4 In the Edit Extended Attributes window, make changes as needed. Here are links to reference information:
 - [“Library-Level Attributes for Autoload”](#)
 - [“Library-Level Attributes for Reload-on-Start”](#)
 - [“Protect Encryption Settings”](#)

LASR Library: Options Tab

Where applicable, the corresponding LIBNAME option is provided in parentheses following the field name.

Options

Libref

enter a one- to eight-character name of the SAS library. The first character must be a letter, and all other characters can be either letters or numbers.

.....
Note: If you specify a libref that has already been assigned, you must also specify that the library is pre-assigned. Select **Options** ⇒ **Advanced** ⇒ **Pre-assign**. If you do not select this option, any queries that use this libref fail.
.....

Engine

the value is set to `SASIOIA` and cannot be changed.

Server tag (TAG=)

enter a string that is unique within the server instance (the server's host-port combination). See ["Understanding Server Tags" in SAS LASR Analytic Server: Reference Guide](#).

Here are tips:

- In most cases, you can simply use the libref value as the server tag.
- If the library's data is loaded using SAS Embedded Process, or is loaded from a co-located provider, additional requirements apply. See ["Server Tags"](#).
- If you do not specify a value, a tag of `WORK` is used.

Options: Storage

Data provider library

if you want the library to support reload-on-start, select a Base SAS library. See ["Reload-on-Start"](#).

Options: Advanced

Pre-Assign

Library is pre-assigned

if the library is pre-assigned, select this check box and choose a pre-assignment type.

Pre-Assignment Type

specify the type of pre-assignment used for the library, if applicable. Choices are as follows:

By native library engine

The library is assigned through METAATORESOURCES options, using the library engine defined for that library.

By metadata library engine

The library is assigned through METAATORESOURCES options, using the Metadata Library Engine (MLE). Using MLE ensures that access controls that are placed on the library and its tables and columns are enforced.

By external configuration

The library is assigned through an external definition or by an autoexec file.

For background information, see [“Understanding Pre-Assigned Libraries”](#) in *SAS Intelligence Platform: Data Administration Guide*.

Other Options

Option(s) to be appended

enter any other valid LIBNAME options in this field. See [“LIBNAME Statement Syntax”](#) in *SAS LASR Analytic Server: Reference Guide*.

.....
Note: Be sure to use the correct syntax. Text that you enter in this field is added to the generated LIBNAME statement without any error checking.

LASR Library: Data Server Tab

Database server

specifies the associated LASR server. If you need to add a new LASR server, click **New**. See [“Add a LASR Server”](#).

Connection

Use the pre-selected value (which prepends the selected server name with the string `Connection:`).

Default login

This field is not editable. The value is None.

See Also

- [“LASR Page” in *Using SAS Environment Manager Administration*](#)
- [“Libraries Page” in *Using SAS Environment Manager Administration*](#)

Limiting Space for Tables

Introduction

To limit the amount of space that a LASR 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, reload, or change source for 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 LASR Server Status](#)” on page 6.

How to Set a Tables Limit

To set a server’s tables limit:

- 1 On the  **LASR** page in [SAS Environment Manager Administration](#), select the **LASR Servers** tab.
- 2 In the **Tables Limit** column, click a cell, and enter a number.


Note: 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).

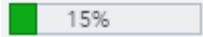
Note: These instructions affect the [VA.MaxTotalMemoryForTables](#) extended attribute.

Distributed Server: Monitoring

Memory Gauges

For a distributed LASR server, an overall memory gauge is displayed on the  **LASR** page in [SAS Environment Manager Administration](#). 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”](#).
- The gauge provides information for only the distributed LASR server that is referenced in the `service.properties` file in the [SAS configuration directory](#) (at `/Applications/SASVisualAnalytics/HighPerformanceConfiguration`).

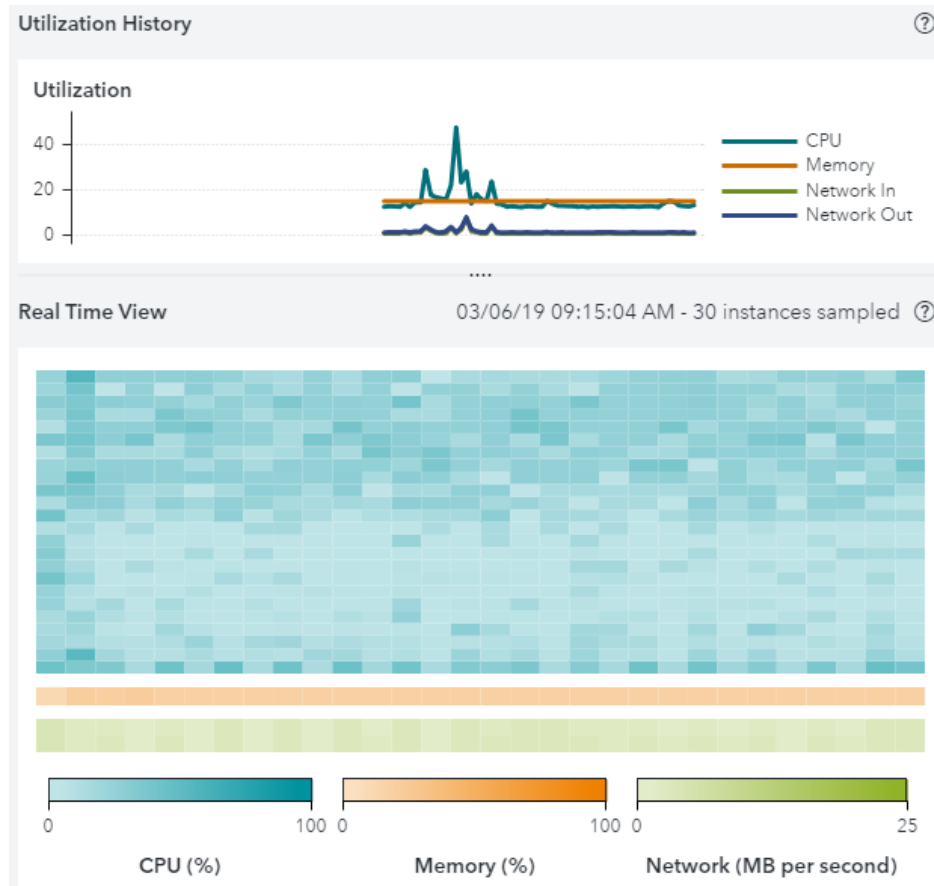
Note: This constraint also applies to memory usage information on the **Resource Monitor** and **Process Monitor** tabs.

- For a distributed LASR 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”](#).

Resource Monitor

For a distributed LASR server, you can monitor resource utilization on the **Monitor Resources** tab, which is on the  **LASR** page in SAS Environment Manager Administration.



In the upper half of the **Monitor Resources** 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.

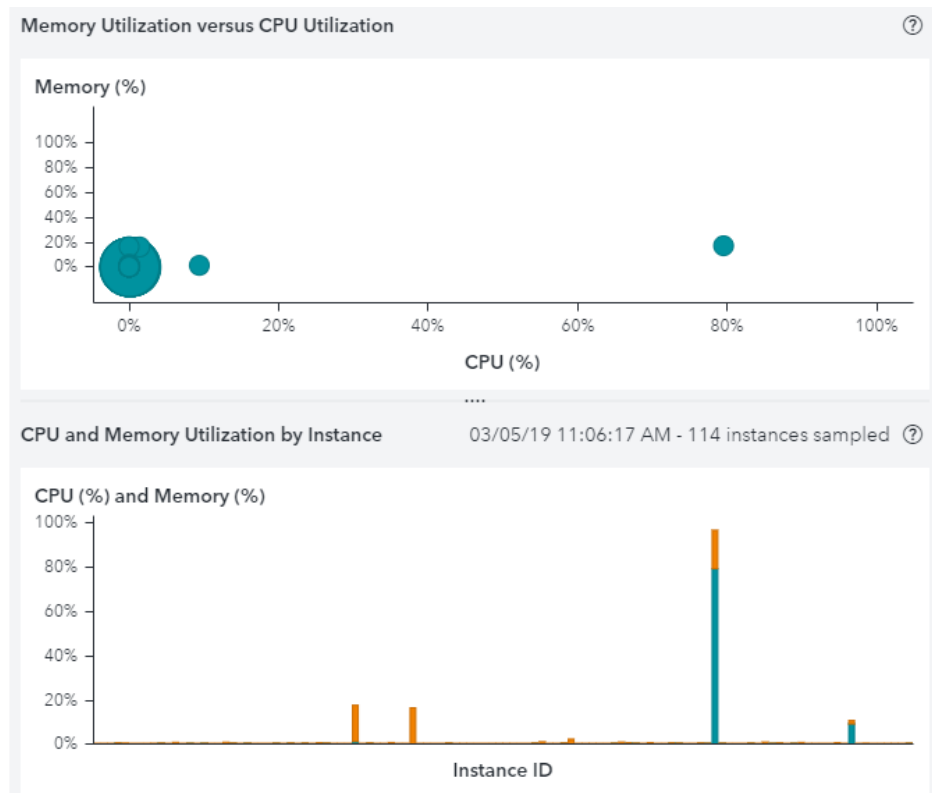
To view resource utilization for a particular sampling period, place your pointer over a line. To select that sampling period in the entire display, click a line. You can then place your pointer over the sampling period on each line to view details.

In the lower half of the **Monitor Resources** 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 LASR server, you can monitor per-process utilization on the **Monitor Processes** tab, which is on the  LASR page in SAS Environment Manager Administration.



Controls at the top of the **Monitor Processes** tab enable you to specify which server instances to display. Here are details:

- The drop-down list is for viewing top resource consumers. If you select a Top 5 or Top 10 filter and no application instances have a resource utilization greater than zero for the filter criteria, then no application instances appear in the graphs.
- The search field is for filtering application instances. Specify a value to use as a filter. Only those application instances that have an ID that contains the specified value appear in the graphs.

The **Monitor Processes** 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 LASR server, SAS Environment Manager Administration provides two distinct expressions of memory usage:

At the top of the page, the overall memory gauge provides a practical estimate of effective capacity.

The value for the overall memory gauge is calculated as follows: $(\text{total allocations} - \text{SASHDAT allocations}) / (\text{total memory for the cluster})$.

| | |
|---------------------|---|
| 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: $(\text{total allocations for this instance}) / (\text{total memory for the cluster})$.

Note: The per-instance information on the **Monitor Processes** 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.

For more information, see [“Supporting the Monitoring Features”](#).

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.

Method: Co-located Storage

| | |
|--------------------|--|
| Topology: | The storage and analytics nodes must be on the same machines. |
| Provider: | Co-located HDFS . |
| 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” or use SAS Visual Data Builder. |

Method: NFS-Mounted Storage

| | |
|--------------------|---|
| Topology: | The storage cluster can be separate from the analytics cluster. ¹ |
| Provider: | MapR. See “MapR Distribution for Apache Hadoop” in <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” or use SAS Visual Data Builder. |

¹ Regardless of topology, the LASR 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. ² |
| 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”, use SAS Visual Data Builder, or use an import action.¹

- 1 Load is parallel if embedded processing is available, LASR table name matches source table name, and server tag is valid as a SAS libref.
- 2 See [SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide](#).

Example Depictions

The following figures depict staging to and loading from co-located HDFS:

Figure 6.3 Stage to Co-located Storage

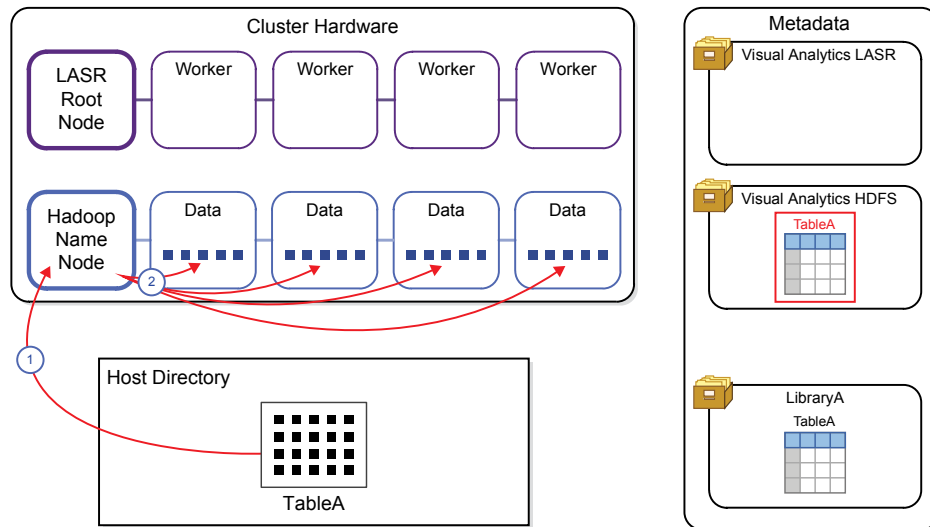
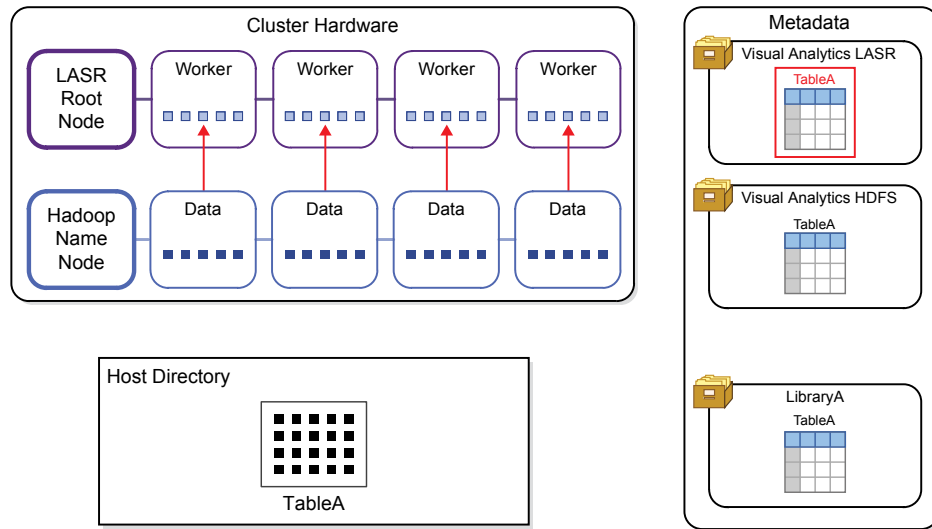


Figure 6.4 Load from Co-located Storage

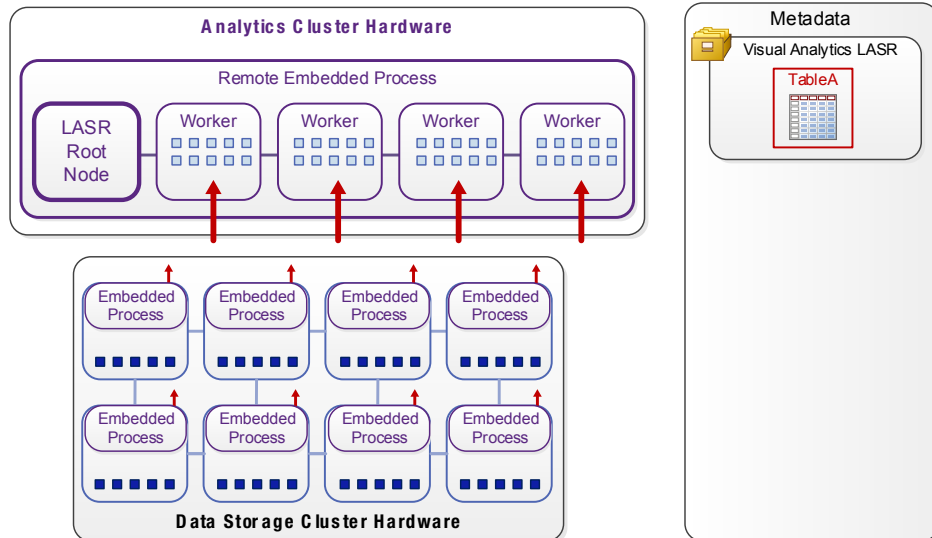


For NFS-mounted MapR, the stage and load processes are similar to the preceding example, except as follows:

- 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 6.5 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 LASR 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 LASR 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 LASR server to use HDFS to read SASHDAT tables in parallel very efficiently.

About the HDFS Tab

Introduction

The **HDFS** tab is on the  **Tools** page in SAS Environment Manager Administration.

Note: The **HDFS** tab is available in deployments that use co-located HDFS. Only users who have the Browse HDFS capability can access the **HDFS** tab.

The **HDFS** tab provides a host-layer view of HDFS folders and tables. The **HDFS** tab supports the following tasks:

- Browse HDFS folders and tables (source files).
- View properties, row count, 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. (There is no row count or column information for files that are not SASHDAT files.)

- Delete HDFS tables that are stored in SASHDAT format. (Files that are not SASHDAT files are listed, but they cannot be deleted.)

HDFS System Properties

To view HDFS system properties, click  in the navigation pane of the **HDFS** tab.

The following information is provided:

Table 6.1 HDFS System Properties

| Property | Description |
|-------------------------|---|
| 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 available. |
| Dead data nodes | Specifies the number of Hadoop DataNodes that are not available. |

Tables: Basic Information


To view a table's basic file information:

- 1 In the navigation pane on the **HDFS** tab, select a folder.

In the content pane, source files are listed. The following information is displayed for each source file:

Table 6.2 Basic Information: Files List

| 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 the metadata about the file. |
| Date Modified | Specifies the date on which the file was created or replaced. |

- 2 For additional details, select the table's source file, and click .


The following information is provided:

Table 6.3 Basic Information: Properties Pane

| Field | Description |
|----------------------------|--|
| Path | Specifies the HDFS directory. |
| Description | Specifies the description that is stored with 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 copies of the data. |
| File size | Specifies the size of the file in bytes. |
| Blocks | Specifies the number of blocks that are used to store the original copy 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 that stored the data. |
| 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. <code>True</code> indicates that the file is in the SASHDAT format. |
| Compression | Specifies whether the file is compressed. <code>True</code> indicates that the file is compressed. |
| Encryption | Specifies whether the file is encrypted. <code>True</code> indicates that the file is encrypted. |

Note: The **HDFS** tab might display multiple source files for a table while the table is being added to HDFS. After the table is added, the multiple source files disappear.

Tables: Column Information


To view a table's column information, select its source file, and click  in the table toolbar.

The following information is provided:

Table 6.4 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, Character, or String. 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 length of the format of 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. |

Tables: Row Count Information


To view a table's row information, select its source file, and click .

The following information is provided:

Table 6.5 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. |
| Utilization (%) | Specifies the percentage of allocated space that is used for storing rows of data. |

Tables: Block Detail Information

To view a table's block details, select a file, and click  in the table toolbar. In the Block Details and Distribution window, select the **Details** tab.

The following information is provided:

Table 6.6 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. |

| Field | Description |
|--------------------|--|
| 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 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.

Tables: Block Distribution Information

Files that are 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.

To view a table's block distribution, select its source file, and click  in the table toolbar. The Block Details and Distribution window is displayed. The window provides two views of distribution information.

- The **Block Distribution** tab enables you to select a block number and view the host names that store the original or redundant blocks. This enables you to see how many copies of the block exist and to identify 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.

- The **Host Distribution** tab enables you to select a host name and view the block numbers that are stored on that machine. You can expand a host name, and then view the block numbers that are stored on that machine. When you select a 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 [“Establishing Connectivity to a SASHDAT Library” in 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](#).
- The server tag for the corresponding LASR library must be the source path in dot-delimited format. See [Server Tags](#).

- 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 On the **Folders** page in SAS Environment Manager Administration, right-click the table, 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 on the **Tools** page. Select the table, and click  in the tab's toolbar.

See Also

[“Supporting the HDFS Browse Features”](#)

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 LASR 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 *SAS LASR Analytic Server: Reference Guide*.

Instructions

To optimize high-volume access to smaller tables in a distributed LASR 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 LASR 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 Administration 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 LASR Table Status](#)”.

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 Administrator 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 LASR 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 LASR 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.

Reports for Administrators

| | |
|--|------------|
| <i>About the Predefined Reports</i> | 159 |
| Location | 159 |
| Access | 160 |
| Data Currency | 161 |
| Interacting with Reports | 161 |
| About Custom Reports | 161 |
| <i>How to Provide Administrative Data</i> | 162 |
| Orientation | 162 |
| 1. Start Autoload | 163 |
| 2. Feed Data to Autoload | 164 |
| <i>About auditRefresh</i> | 166 |
| Introduction | 166 |
| Start auditRefresh | 167 |
| Stop auditRefresh | 167 |
| Reduce the Task Frequency | 167 |
| Change the Purge Boundary | 168 |
| Logs and Process IDs for auditRefresh | 168 |

About the Predefined Reports

Location

The following predefined administrative reports are in the folder `/Products/SAS Visual Analytics Administrator/Reports/Usage`:

| | |
|-------------------------------|--|
| 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”](#).

TIP You can open reports from the navigation pane of the **Folders** page in SAS Environment Manager Administration. At the top of the navigation pane, the **Usage Reports** button provides a shortcut to the folder that contains the predefined administrative reports. See [“Open a Report” in Using SAS Environment Manager Administration](#).

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 LASR 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 [SAS Visual Analytics: Viewer Documentation](#).

About Custom Reports

Do not modify the predefined reports. Instead, use SAS Visual Analytics 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 SAS Visual Analytics, 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 On the **LASR** page in SAS Environment Manager Administration, select the **LASR Tables** tab.
- 2 At the top of the tab, select **LASR Name** from the drop-down list, and enter *EVDM* in the search field. To interpret the display, see [“Get LASR Table Status”](#).

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.

Note: On Windows, share the directory. Change the EVDMLA library’s VA.AutoLoad.Location extended attribute to be a UNC path.

Note: On UNIX, 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”](#).

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 LASR server and load data. See [“Host Account Privileges”](#).
 - 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”](#).

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

Give the scheduler account's metadata identity the required [metadata-layer permissions](#) 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: /Shared Data/SAS Visual Analytics/Autoload/EVDMLA 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".

- 4 Verify that the scheduled task is running.

.....

Note: 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**.

.....

Note: 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".

.....

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 autoload 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.
- 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](#)”.
- Extraction occurs only if SAS Visual Data Builder is running.
- Extraction uses the pooled workspace server in the suite-level default SAS Application Server. See [va.defaultWorkspaceServer](#).
- The extraction process must have Read and Write access to the autoload data directory and that directory’s Append subdirectory. The extraction process runs under the same account as the web application server on which SAS Visual Data Builder resides.

Note: Autoload’s append action is used, so the extracted table is written to the Append subdirectory.

- If autoload 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](#)”.

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 *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 *SAS Intelligence Platform: System Administration Guide*.
- Relationship data is extracted for the following object types:
 - 2G reports (Report.BI 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 autoload data directory. The extraction process runs under the same account as the web application server on which SAS Visual Data Builder resides.

Note: Autoload's refresh action is used, so the extracted table is written to the autoload data directory.

Agent-Collected Metrics

Note: Agent-collected metrics do not provide information about distributed LASR servers.

To feed agent-collected metrics (ACM) to the drop zone:

- 1 Enable data collection. See [Initializing and Enabling the Service Architecture](#) in *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 *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"](#).

- 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 *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 autoload 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](#)”.

Start auditRefresh

- 1 Log on to the autoloading host as the scheduler account that you identified or created when you started the administrative data implementation of autoloading.
- 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.

Note: 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**.

Note: 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, autoloading, or the associated LASR 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.

Note: On Windows, in the `auditRefreshSchedule.bat` script, change the value for `TIME_INTERVAL_DAYS=` . The initial value is 1.

Note: On UNIX, 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”](#).

Reference

| | |
|--|------------|
| Software Components | 169 |
| Roles and Capabilities | 171 |
| About Capabilities | 171 |
| Predefined Roles | 171 |
| Initial Role Members | 175 |
| Capability Definitions | 175 |
| See Also | 177 |
| Configuration Properties | 177 |
| How to Set Configuration Properties | 177 |
| Properties Reference | 178 |
| High-Cardinality Constraints | 188 |
| Introduction | 188 |
| Client-Side Thresholds for High-Cardinality Data | 188 |
| Middle-Tier Thresholds for High-Cardinality Data | 188 |
| Predefined LASR Libraries | 190 |
| Administrative Preferences | 194 |
| Introduction | 194 |
| Preferences: Application Server | 194 |
| Preferences: Clear Credentials Cache | 194 |
| Preferences: Manage Environment | 195 |
| Sample Objects | 196 |
| Configured Samples | 196 |
| Tips and Guidelines | 196 |
| Manual Configuration | 197 |

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 Visual Analytics Apps](#) 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 Visual Analytics Apps, 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 6, “SAS LASR Analytic Server”](#).

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”](#).

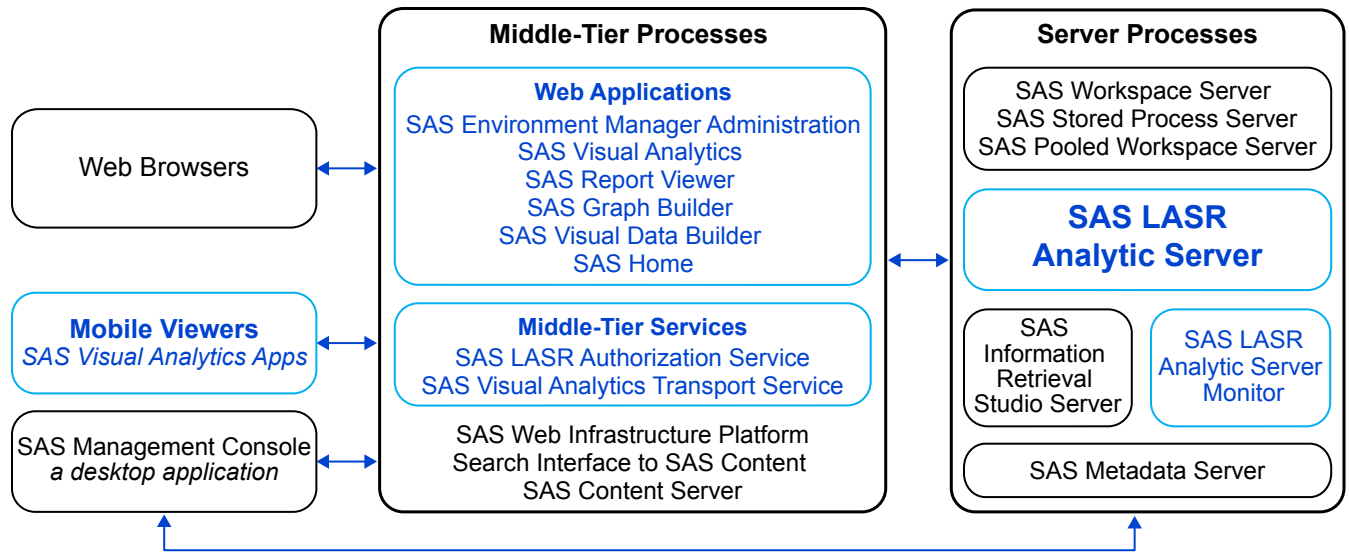
[SAS Intelligence Platform](#)

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.
- 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 LASR server.
- For more examples, see [“Add a New Server” on page 94](#).

Here is a conceptual view of selected components:

Figure 8.1 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.

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 [“Configure Guest Access”](#) in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Visual Analytics: Report Viewing

provides commenting and personalization features, in addition to basic functionality.

Visual Analytics: Analysis

provides the ability to create and distribute reports, 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 administrative tasks for SAS Visual Analytics, in addition to most other SAS Visual Analytics capabilities.

Table 8.1 Capabilities by Role











| Capability | Basic | Report Viewing | Analysis | Data Building | Administration |
|---------------------------------------|-------|----------------|----------|---------------|----------------|
| Visual Analytics | | | | | |
| View Report and Stored Process | ✓ | ✓ | ✓ | ✓ | ✓ |
| Create Report | | | ✓ | ✓ | ✓ |
| 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 | | | | ✓ | |
| Manage Environment | | | | | ✓ |

| Capability | Basic | Report Viewing | Analysis | Data Building | Administration |
|------------------------------------|-------|----------------|----------|---------------|----------------|
| 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 | | | | | |
| Build Analytical Model | | | ✓ | | |
| Visual Analytics Admin | | | | | |
| Manage LASR Analytic Server | | | | | ✓ |
| Monitor LASR Analytic Server | | | | | ✓ |
| Browse HDFS | | | | | ✓ |

To manage roles, see [Users Page](#) in *Using SAS Environment Manager Administration*.

Initial Role Members

The following table depicts the initial memberships of the predefined roles:

| Group | Direct Member Of (Role) |
|--|--|
|  SASUSERS |  Visual Analytics: Basic |
|  Visual Analytics Users |  Visual Analytics: Report Viewing |
|  Visual Analytics Users |  Visual Analytics: Analysis |
|  Visual Data Builder Administrators |  Visual Analytics: Data Building |
|  Visual Analytics Data Administrators |  Visual Analytics: Administration |

Capability Definitions

Here are descriptions of the SAS Visual Analytics capabilities:

Visual Analytics

View Report and Stored Process

Access web and mobile report viewers. View reports and stored process output. (Access to SAS Visual Analytics Apps is also affected by device-level constraints. See [Mobile Device Access](#).)

Create Report

Access SAS Visual Analytics. Create and modify reports.

Build Custom Graph

Access SAS Visual Analytics Graph Builder. Create and modify graph template objects for use in SAS Visual Analytics.

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 as PDF files.

Email

Send a link to a report via email.

Personalization

Use individualized features such as setting preferences, accessing recently viewed objects, and managing favorites.

Build Analytical Model

Create and modify analytical models using SAS Visual Statistics (a separately licensed add-on).

Visual Analytics: Self-Service Import

Import and Load Data

A prerequisite for access to self-service import functionality in SAS Visual Analytics. See [“Self-Service Import”](#).

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 SAS Visual Data Builder. Set advanced options when loading data in SAS Visual Analytics.

Manage Environment

Access the **LASR** and **Tools** pages in SAS Environment Manager Administration. Launch SAS Environment Manager Administration from the applications menu (under **SAS VISUAL ANALYTICS** ⇨ **Administrator**). See also [“Capabilities” in Using SAS Environment Manager Administration](#).

Manage Mobile Devices

Access the **Mobile Devices** page in SAS Environment Manager Administration.

Distribute Reports

Schedule and manage the distribution of reports in SAS Visual Analytics.

Visual Analytics Transport Service

Purge Mobile Report Data

Causes cached data in SAS Visual Analytics Apps 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.

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 Visual Analytics Apps. 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.

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 Visual Analytics Apps 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

Build Analytical Model

Create and modify analytical models using SAS Visual Statistics (a separately licensed add-on).

Visual Analytics Admin

Manage LASR Analytic Server

Access the **LASR Servers** and **LASR Tables** tabs on the **LASR** page.

Monitor LASR Analytic Server

Access the **Monitor Processes** and **Monitor Resources** tabs on the **LASR** page. This capability is applicable to deployments that use a distributed server.

Browse HDFS

Access the **HDFS** tab on the **Tools** page. This capability is applicable to deployments that use co-located HDFS.

Note: For conciseness, version numbers are omitted in this topic.

See Also

- [“Streamlining Access to Management Pages”](#)
- [“Roles and Capabilities” in *Using SAS Environment Manager Administration*](#)

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 alert properties (**va.Alert.***) on the **Visual Analytics Hyperlink Service** node.
- Set transport service properties (**viewerservices.***) on the **Visual Analytics Transport Service** node.

Note: This topic includes only selected transport service properties. For a comprehensive list of transport services properties, see “[Configuration Properties: Transport Services](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

- Set other properties on the **Visual Analytics** node, except where otherwise noted in the documentation for a specific property.

Note: For conciseness, version numbers are omitted 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/`.

| Host | Command |
|---------|---------------------------------------|
| UNIX | <code>appsrvconfig.sh restart</code> |
| Windows | <code>appsrvconfig.cmd restart</code> |

For details and alternatives, see “[Understanding SAS Web Application Server Management](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide* and “[Use Configuration Manager](#)” in *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Properties Reference

CAUTION

Increasing a data limit can affect performance and stability. In the following table, a diamond (◆) indicates a property that affects data limits. The default settings are appropriate in most environments. Do not set excessively high data limits. If you have questions about adjusting configuration properties, contact SAS Technical Support.

| Property | Description |
|-----------------------------------|--|
| App.AllowGuest | enables or disables guest access. Valid values are <code>true</code> and <code>false</code> . See "Configure Guest Access" in <i>SAS Intelligence Platform: Middle-Tier Administration Guide</i> . |
| las.caching.key.lifetime | In deployments that enable guest access, this property is usually set on the Visual Analytics Transport Service , Visual Analytics Viewer , and Visual Analytics Hub nodes. |
| las.caching.permission.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.user.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. |
| lasrnmgt.server.monitor.refresh | 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. |
| Printing.Footer.Content.Formatted | 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. 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. |
| Printing.Timeout | enables standard footer text to be added to the bottom of all printed reports. The footer definition can include basic formatting options such as font selection and size. It can also include attributes such as bold, italic, or underline. Footers cannot contain images. See "Supporting Footers in Printed Reports". sets a maximum wait time for printing a report. The default is 90000 milliseconds (15 minutes). To disable this property, set its value to 0. 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. |

| Property | Description |
|--|---|
| va.Alert.DefaultEvaluationIntervalMilliseconds | <p>specifies the evaluation interval (how frequently the system makes a determination about whether the alert's conditions have been met). The default is 60000 milliseconds (10 minutes).</p> <p>This property affects only alerts that do not use a custom interval. In the Edit Alert window in SAS Visual Analytics, selecting the Use the system default setting causes the value for this property to be used.</p> <p>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 performance.</p> |
| va.Alert.DefaultMaxEvaluationTimeMilliseconds | <p>specifies how long an individual evaluation can run before it terminates and restarts. The default is 180000 milliseconds (30 minutes).</p> |
| va.Alert.EvaluationCycleMilliseconds | <p>specifies how frequently the system verifies that alerts are running. The default is 60000 milliseconds (1 minute). If a large number of alerts are registered, consider increasing the value to reduce the use of resources.</p> |
| va.Alert.Eventgen.disabled | <p>specifies whether alerts generate notifications. The default is <code>false</code>. To disable notifications, set this property to <code>true</code>.</p> |
| va.Alert.SMS.showServerName | <p>specifies whether to append <code>Server: server-name</code> to the end of an SMS text message that is generated by an alert. The default is <code>true</code>. 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 <code>false</code>.</p> |
| va.AlertThreadPool.CoreSize | <p>specifies the number of threads that are available in normal circumstances (for concurrent evaluation of alerts). The default is 3.</p> |
| va.AlertThreadPool.IdleTimeoutSeconds | <p>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 <code>CoreSize</code>. The default is 1800 seconds (30 minutes).</p> <p>This property is applicable only if the <code>MaxSize</code> is greater than the core size.</p> |
| va.AlertThreadPool.MaxSize | <p>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 <code>CoreSize</code> (up to the value that is set for this property). The default is 3.</p> |
| va.AlertThreadPool.QueueSize | <p>specifies the maximum number of tasks that can be queued. The default is 100000.</p> |
| va.AuditingEnabled | <p>specifies whether applications write audit records. Valid values are <code>true</code> and <code>false</code>. The default is <code>false</code>.</p> |

| Property | Description |
|---|---|
| va.baseSchedulingFolder | <p>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".</p> <p>specifies the name of the parent folder for jobs and flows that are used in report distribution. The default is <code>/System/Applications/SAS Visual Analytics/ScheduledDistribution</code>. See "Supporting Report Distribution".</p> <p>To schedule a report, users must have WriteMemberMetadata access to the specified base folder.</p> |
| ◆ va.CardinalityLimitForGroupByCountDistinctTempTable | <p>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 LASR server to hang. (This property affects only distinct count requests, providing a narrower constraint than the <code>va.CardinalityLimitForGroupByTempTable</code> property.) If you choose to set this property, the suggested value is 1000000.</p> |
| ◆ va.CardinalityLimitForGroupByTempTable | <p>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 LASR 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.</p> |
| ◆ va.CategoryCardinalityServerLimit | <p>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. The default value is 50000.</p> |
| ◆ va.CheckCardinalityWithinQuery | <p>controls whether the LASR server enforces cardinality limits. By default, these checks do occur.</p> |
| va.ComparisonEpsilon | <p>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:</p> $ABS(a-b) \leq \epsilon * MAX(ABS(a), ABS(b))$ |

| Property | Description |
|--|--|
| 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.decisionTreeCardinalityLimit | sets the maximum number of distinct levels for a variable in a SAS Visual Analytics decision tree. The initial value is 10000. |
| 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 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?" . |
| ◆ va.detailTableSortResultLimit | 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. The default value is 50000. |
| ◆ 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 value is 50000. |
| 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" . |
| 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" . |
| va.extractRelationshipData | enables periodic extraction of data about relationships among SAS Visual Analytics objects. Relationship data is extracted from the relationship service to the autoload data directory (drop zone) for administrative reporting. Valid values are true and false. The default is false. See Chapter 7, "Reports for Administrators," on page 159. |
| 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. |

| Property | Description |
|--|--|
| <code>va.GeoMapNumResolutions</code> | for an alternate OpenStreetMap server (specified in the property <code>va.GeoMapServerUrl</code>), sets the number of levels in each geo map. The default is 18. 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. |
| <code>va.GeoMapServerUrl</code> | specifies a comma-delimited list of URL addresses that reference alternate OpenStreetMap servers (for example, <code>http://serverA.org, http://serverB.org, http://serverC.org</code>). To use the OpenStreetMap servers that SAS hosts, leave the value for this property blank. See “OpenStreetMap Server” on page 100 . |
| <code>va.IgnoreMissingInCountDistinct</code> | determines whether missing values are ignored in counts when the Distinct operator is used. Valid values are <code>true</code> and <code>false</code> . The default is <code>false</code> (missing values are included in distinct counts). |
| <code>va.LASRMonitor.HostPort</code> | specifies the machine name and port for the process that monitors a distributed LASR server. The value is in the format <code>host:port</code> (for example, <code>machine.company.com:9971</code>). |
| <code>va.lastActionLogPath</code> | specifies the location of last action logs. See “Get LASR Server Status” and “Get LASR Table Status” . The standard location is within the SAS configuration directory at <code>/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/Monitoring/Logs</code> . In a multi-machine deployment, the specified location exists on the middle-tier host. |
| ◆ <code>va.maxPeriodCalculations</code> | 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. 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). |
| ◆ <code>va.MaxSparkTables</code> | sets the maximum number of spark tables. The default is 300. |
| ◆ <code>va.MaxTiesToIncludeOnRank</code> | sets the maximum number of identically ranked values that can be returned in a rank operation. Valid values are integers. The default is 100. |

| Property | Description |
|--|---|
| va.modelingClassCardinalityLimit | sets the maximum number of distinct levels in a SAS Visual Statistics 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. |
| ◆ va.modelingDecisionTreePredictorCardinalityLimit | sets the maximum number of distinct levels for a category variable in a SAS Visual Statistics decision tree. The initial value is 1024. |
| ◆ va.modelingDecisionTreeResponseCardinalityLimit | sets the maximum number of distinct levels for the response category variable in a SAS Visual Statistics decision tree. In the initial configuration, this property is not set, so the default value (100) is in effect. |
| ◆ va.modelingGroupByCardinalityLimit | sets the maximum number of distinct levels for the group-by variables in a SAS Visual Statistics 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. |
| va.monitoringPath | specifies the location for certain process ID files and logs. The standard location is within the SAS configuration directory at <code>/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/Monitoring</code> . The specified location must exist on a workspace server host. |
| | To specify a custom directory for monitoring artifacts for a particular LASR server, set an extended attribute for that server. See " VA.MonitoringPath ". |
| va.publicLASRLibrary | identifies the general purpose library for data import and load activities (for example, <code>Visual Analytics Public LASR</code>). 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 <code>va.publicLASRLibrary</code> (for example, <code>Public LASR Analytic Server</code>). |
| ◆ va.sankeyMaxPathLength | 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, a message indicates that the Sankey diagram excludes paths longer than the specified value. If this property is not defined, the default value (2000) is in effect. Client-side controls in the user interface provide additional, more stringent limits (to facilitate quick display of Sankey diagrams). |

| Property | Description |
|--|--|
| | 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 <code>va.sankeyMaxPathLength</code> . A numeric value uses a width of 40 for this calculation. |
| ◆ <code>va.sankeyMaxNumberOfPathsDisplayed</code> | 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 (2000000000) is in effect. |
| ◆ <code>va.sankeyMaxTransactionId</code> | 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. |
| <code>va.SASGeomapCommunicationProtocol</code> | sets the protocol for connections between SAS Visual Analytics and the OpenStreetMap servers that SAS hosts. Valid values are <code>http</code> and <code>https</code> . |
| <code>va.SASGeomapEsriURL</code> | 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 101. |
| <code>va.SelfServe.MaxUploadSizeInMegabytes</code> | sets the maximum file size (in megabytes) that a user can import. This property affects importing local files in SAS Visual Data Builder and SAS Visual Analytics. 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. |
| <code>va.SelfService.ImportGoogleRowLimit</code> | sets a maximum number of rows for an import from Google Analytics. The default is 100000. |
| <code>va.SelfService.ImportRowsHardCap</code> | 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). 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. |
| <code>va.SelfService.ImportRowsSoftCap</code> | 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). 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. |

| Property | Description |
|--|---|
| <ul style="list-style-type: none"> ◆ <code>va.SummaryServerRowLimit</code> | <p>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. The default value is 50000.</p> |
| <ul style="list-style-type: none"> <code>va.supportSharedThumbnails</code> | <p>determines whether SAS Visual Analytics generates thumbnails (specific preview images). Valid values are <code>true</code> and <code>false</code>. The default is <code>false</code> (specific preview images are not generated).</p> <p>Generated thumbnails are displayed only if this property is also set to <code>true</code> on the Visual Analytics Hub software component.</p> |
| <ul style="list-style-type: none"> ◆ <code>viewerservices.default.max.cells.produced</code> | <p>sets the maximum number of data cells to include in the results for a single data query. The default is 250000 data cells, which is sufficient for most environments. In very rare scenarios, you might need to modify the value.</p> |
| <ul style="list-style-type: none"> <code>viewerservices.image.default.max.bytes</code> | <p>sets the maximum size of images (PNG, BMP, JPEG, or GIF) in a rendered report. Larger images are resized on the server side before delivery. The default is 300 KB, which is sufficient for most environments.</p> <p>To ensure faster download times and smaller memory footprints, do not set the value to 0 (which prevents resizing of images in the middle tier) or increase the value.</p> |
| <ul style="list-style-type: none"> <code>viewerservices.lasr.socket.timeout.milliseconds.interaction</code> | <p>sets the maximum wait time for attempts to contact SAS LASR Analytic Server. This property is applicable to live requests for tasks such as filtering, brushing, and drilling. The default is 300000 milliseconds (5 minutes), which is sufficient for most environments. If connections to the LASR server are timing out, consider modifying the value.</p> |
| <ul style="list-style-type: none"> <code>viewerservices.lasr.socket.timeout.milliseconds.subscribe</code> | <p>sets the maximum wait time for a response to a query in a report after contacting 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.</p> |
| <ul style="list-style-type: none"> <code>viewerservices.validate.schema.create</code> | <p>enables XML schema validation when reports are rendered. When this property is set to <code>true</code>, all actions that apply to the writing of reports are captured in the <code>SASVisualAnalyticsTransport-log4j</code> file. The default is <code>false</code>. Set this property only if SAS Technical Support instructs you to do so.</p> |
| <ul style="list-style-type: none"> <code>viewerservices.validate.schema.read</code> | <p>enables XML schema validation when reports are rendered. Also, this property checks for schema validation errors when reports are created. When this property is set to <code>true</code>, all actions that apply to opening and viewing reports are captured in the <code>SASVisualAnalyticsTransport-log4j</code> file. The default is <code>false</code>. Set this property only if SAS Technical Support instructs you to do so.</p> |

| Property | Description |
|--------------------------------------|---|
| viewerservices.validate.schema.write | enables XML schema validation when reports are rendered. When this property is set to <code>true</code> , all actions that apply to the writing of reports are captured in the <code>SAS\visualAnalytics\Transport-log4j</code> file. The default is <code>false</code> . Set this property only if SAS Technical Support instructs you to do so. |

See Also

[“High-Cardinality Constraints”](#)

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 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, or to a group of applications. 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.

Client-side thresholds for are documented in [“High-Cardinality Thresholds for Objects”](#) in *SAS Visual Analytics: Reference*. 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 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.

The following table provides some examples. For each listed object type or set of object types, the table identifies the maximum number of unique values (not the maximum volume of data), as defined by a particular configuration property.

Table 8.2 Examples of Middle-Tier Thresholds

| Type of Object | Rows | Property |
|--|--------|--|
| Summary table | 50,000 | va.SummaryServerRowLimit |
| Bar chart: single category | | |
| Heat map: single category | | |
| Box plot: at least one measure, no categories | | |
| Bubble plot: three measures, grouped | | |
| Bubble plot: three measures, grouped with animation category | | |
| Bubble plot: three measures, not grouped, horizontal or vertical series (or both) | | |
| Line chart: at least one measure, single category (numeric, date, time, or string) | | |
| Decision tree (in SAS Visual Analytics) | 10,000 | va.decisionTreeCardinalityLimit |
| Decision tree (in SAS Visual Statistics) - predictor | 1024 | va.modelingDecisionTreePredictorCardinalityLimit |
| Decision tree (in SAS Visual Statistics) - response | 100 | va.modelingDecisionTreeResponseCardinalityLimit |

IMPORTANT The preceding table lists only one configuration property for each object type (or set of object types). Most objects are subject to additional data limits that are defined by additional configuration properties. See [“Properties Reference”](#).

CAUTION

Increasing a 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 configuration properties, contact SAS Technical Support.

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 8.3 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 . |
| Location | <code>/Shared Data/SAS Visual Analytics/Public</code> The location must match the value of a configuration property. See va.defaultPublicFolder . |
| Data server | Public LASR Analytic Server The server name must match the value of a configuration property. See va.publicLASRServer . |
| 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”.) Data: <code>/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/Autoload</code> Scripts: <code>/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator</code> LASR table objects: <code>/Shared Data/SAS Visual Analytics/Public/LASR</code> |
| Reload-on-start | Enabled Library: Visual Analytics Public Data Provider (libref: DPPUBLIC) Directory: <code>/AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/PublicDataProvider</code> LASR table objects: <code>/Shared Data/SAS Visual Analytics/Public/LASR</code> |

Table 8.4 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”.) 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 8.5 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 7, “Reports for Administrators”. |
| Autostart | Enabled |
| Autoload | Enabled (To use autoload, start the scheduled task. See “Autoload”.) 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 |

Administrative Preferences

Introduction

To view or modify your preferences, use the Settings window. To access the Settings window, click the right-most icon in the SAS Environment Manager Administration banner, and select **Settings**.

This topic documents user preferences that are specific to SAS Environment Manager Administration. For other preferences, see [“Settings” in SAS 9.4 Web Applications: General Usage Help](#).

Preferences: Application Server

The value in the **Application server** drop-down list specifies how a SAS Application Server is selected for requests on the **LASR** page.

(auto-select)

causes an appropriate server to be automatically selected for each request. See [Which Server is Used? on page 93](#) in *SAS Visual Analytics: Administration Guide*.

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.

Preferences: Clear Credentials Cache

To remove cached credentials from the current session:

- 1 In the Settings window, select **SAS Environment Manager Administration** ⇒ **Clear Credentials Cache**.
- 2 In the right pane, click **Clear**.
- 3 In the confirmation window, click **Yes**.

Preferences: Manage Environment

Resource monitor sample rate

Specifies, in milliseconds, the sampling rate that the resource monitor uses for polling the machines in the cluster. This setting is neither applicable to nor displayed for a non-distributed server.

Process monitor sample rate

Specifies, in milliseconds, the sampling rate that the performance monitor uses for polling application instances. This setting is neither applicable to nor displayed for 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 is generated when you perform certain tasks in SAS Environment Manager Administration. You can save all recorded code in a single file or you can save the recorded code for each task in its own file.

Note: The **File** field is required to save all recorded code in a single file. Provide the fully qualified path and name of a file that already exists on your default workspace server. Your default workspace server is the SAS Workspace Server that is part of your selected SAS Application Server. You must have Read and Write access to the specified file. The Settings window does not validate the file that you specify.

Note: The **Path** field is required to save the recorded code for each task in its own file. Specify a location that already exists on your default workspace server. Your default workspace server is the SAS Workspace Server that is part of your selected SAS Application Server. You must have Read and Write access to the specified location. The Settings window does not validate the path that you specify.

If you enable recording, the following actions are recorded:

- Start or stop a LASR 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 physical table from co-located HDFS or NFS-mounted MapR.
-

Note: You can modify and schedule recorded statements. However, metadata server connection information is not recorded. See [Connection Options](#) in *SAS Language Interfaces to Metadata*.

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 sample tables and sample reports.
- The metadata favorites folder `/Products/SAS Visual Analytics/Visual Analytics Samples`, which provides access to the sample reports.
- 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 LASR Server”](#).

.....

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 SAS Visual Analytics and SAS Visual 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.

Note: Each user can hide the sample collection. An administrator can unpublish the sample collection.

- 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 8.6 SPK Locations

| | |
|----------------------|---|
| UNIX (example) | <code>/opt/sas/SASHome/SASVisualAnalyticsAdministrator/7.5/Config/Deployment/ Packages</code> |
| Windows (example) | <code>C:\sas\SASHome\SASVisualAnalyticsAdministrator\7.5\Config\Deployment \Packages</code> |

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 VAAdditionalSampleTables.spk. Then, import VASampleReport.spk and VAAdditionalSampleReports.spk.
 - ii If your deployment 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 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.

Troubleshooting

| | |
|--|-----|
| <i>Access Issues</i> | 199 |
| <i>Server Operation Issues</i> | 201 |
| <i>Load, Reload, and Import Issues</i> | 203 |

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. See [“Adding Users” on page 3](#).
- Make sure that the metadata server and the middle tier are running. See [“Operate Other Servers” on page 9](#).

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 171](#).
- Make sure that users are not inadvertently connecting as guests. See [“Configure Guest Access” in SAS Intelligence Platform: Middle-Tier Administration Guide](#).
- Make sure that the appropriate applications are licensed and installed.

Issue: Users cannot access any LASR tables in SAS Visual Analytics.


Resolution:

- Make sure that the LASR server is running and that tables are loaded.

- In metadata, make sure that the LASR authorization service is enabled. On the **Options** tab for the LASR 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 126](#).

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 42](#).
- Make sure the table does not have any invalid permission conditions. On the table's **Authorization** tab, 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. (A copy-and-paste action in SAS Visual 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 17](#).
- If the table was autoloaded, 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:

- Clear the credentials cache. Then, attempt access again. If you are prompted for a user ID and password, enter DBMS credentials. See [“Preferences: Clear Credentials Cache”](#).
- 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 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 43](#).

- 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 [“Connection Options” in SAS Language Interfaces to Metadata](#).

Server Operation Issues

Issue: Inability to start a LASR 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 Environment Manager Administration 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 [“LASR Server Definition” on page 127](#).

.....

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 93](#).
- 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 64](#).
- Make sure that the host name is specified correctly in the connection object for the LASR 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 LASR 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.
- If the SAS middle tier was recently restarted, the server can become unresponsive. See [“Non-distributed Server Dependencies” on page 125](#).

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 178](#).

Issue: The SAS LASR Analytic Server Monitor graphs are blank.

Resolution:

- Make sure that the LASR server is running.
- Make sure that the LASR server is distributed. The **Monitor** tabs are not supported for non-distributed servers.
- Make sure that the TKGid 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 Monitor” on page 87](#).

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 87](#).

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 87](#).
 - Make sure that the monitoring process is running. See [“Supporting the Monitoring Features” on page 86](#).

Load, Reload, and Import Issues

Issue: Inability to load, reload, or import tables.

Resolution:

- Make sure that you can access the LASR 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 42](#).
- 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 73](#).
- 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 22](#).

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 70](#).

- 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 177](#).
- 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 13](#).
- 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 [“Limiting Space for Tables” on page 140](#).

- 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 130](#).
- 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 [“LASR Library Definition” on page 135](#).
 - Make sure that the Hadoop server and the LASR 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 LASR 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 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 178](#).

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 87](#).
- 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 18.

Issue: Autoload issues.

Resolution:

- Review the [additional considerations](#) for autoload.
- Examine the [autoload logs](#). Here are example locations:

| Host | Example Location |
|---------|--|
| Windows | C:\sas\Config\Levl\Applications\SASVisualAnalytics\ VisualAnalyticsAdministrator\Logs |
| UNIX | /opt/sas/config/Levl/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.

