Paper SAS347-2014

Big Data Everywhere! Easily Loading and Managing Your Data in the SAS® LASR™ Analytic Server

Gary Mehler and Donna Bennett, SAS Institute Inc., Cary NC

ABSTRACT

SAS® Visual Analytics and the SAS® LASR™ Analytic Server provide many capabilities to analyze data fast. Depending on your organization, data can be loaded as a self-service operation. Or your data can be large and shared with many people. As the data gets larger, effectively loading it and keeping it updated become important. This paper discusses a range of data scenarios from self-service spreadsheets to very large databases, from single-subject data to large star schema topologies, and from single-use data to continually updated data that requires high levels of resilience and monitoring. Fast and easy access to big data is important to empower your organization to make better business decisions. Understanding how to have a responsive and reliable data tier on which to make these decisions is within your reach.

INTRODUCTION

Before you can perform an analysis, the data must be loaded into the SAS LASR Analytic Server that SAS Visual Analytics uses. In-memory processing makes the results available very quickly. Using the drag and drop functionality, you can quickly get results when exploring the data or designing reports. Acquiring the right data in the right format is the critical first step. Processes will vary depending on the size of the deployment and on the source, scope, and size of the data.

- ***Size of deployment*** – SAS Visual Analytics deployments can use a single machine or can span multiple machines. Deployments on a single machine are called symmetric multi-processing (SMP) configurations. Deployments that span multiple machines support massively parallel processing (MPP) and leverage grid technologies for distributing and coordinating the workload. The type of deployment (either SMP or MPP) influences where the data should be loaded into memory.

- ***Source of data*** – Some data is “pre-processed” and made available to SAS Visual Analytics users across the enterprise. This availability makes it easy for individual users to leverage corporate data. However, sometimes users might want to include “personal” or “ad hoc” data in their analysis. A user might want to load data from a spreadsheet or a personal data file to use as a baseline for analysis or to merge with corporate data. Users might need to access data directly from a database management system (DBMS) or the Internet. When users acquire and load data directly, this paper calls that self-service.

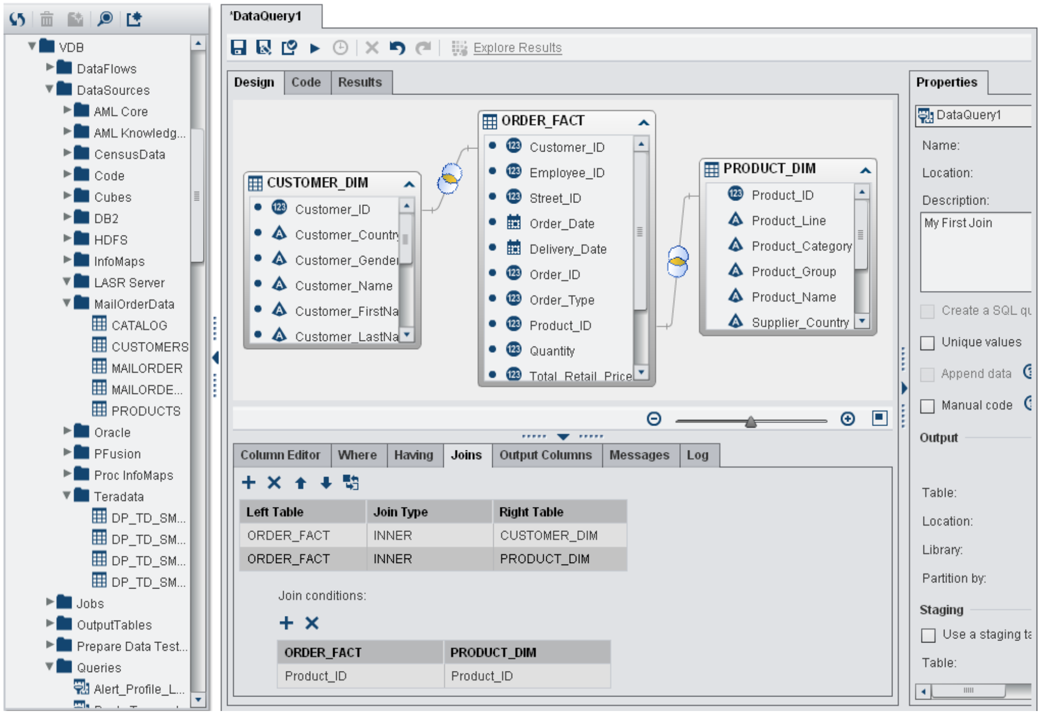
- ***Scope of data –*** The visibility and accessibility of the data is determined by two things: 1) user permissions on the data that is loaded and 2) the accessibility of the library associated with the LASR Analytic Server instance where the data is loaded. The default library is the Public library, and the system administrator specifies options that enable users to change which processing location they use.

- ***Size of data* –** The size of data influences processing choices. When you have small to medium data sets, the overhead of adding, deleting, or transforming the data is relatively low. However, as data sizes get larger, reducing data movement becomes more important. The in-memory transformations of data enabled by the SAS LASR Analytic Server ensure that large corporate data resources can be managed and updated in a timely way while avoiding unloading, modifying, and then re-loading the data.

In SAS Visual Analytics, the central application for data manipulation is SAS Visual Data Builder. (The examples in this paper often use SAS Visual Data Builder.). In addition to being the main place for data manipulation, some pieces of SAS Visual Data Builder are available in other applications of SAS Visual Analytics, including SAS Visual Analytics Explorer and SAS Visual Analytics Designer. This integration enables business users of SAS Visual Analytics Explorer and SAS Visual Analytics Designer to acquire personal data using the self-service options. For example, a user who needs to write a report might want to upload or access the data without waiting for administrative or IT support to make the data available.

The main functions of SAS Visual Data Builder are to load data, perform joins and filtering, and perform actions specific to the SAS LASR Analytic Server, so the data can be prepared for various types of analysis. Users must be granted permission to use SAS Visual Data Builder. Access to this application is typically restricted to users who have a broad understanding of the data and who can prepare the data for use across the organization. Users who have SAS Visual Data Builder permission also usually have administrative privileges.

The remainder of this paper discusses ways that data can be loaded into memory so that it is available to support data exploration, reports, and analysis using powerful SAS tools. This paper also discusses the ways to consume (or manipulate) data so the data is ready for analysis. This paper highlights a few ways to monitor the data in the SAS Visual Analytics environment and closes with a few example scenarios.



Description: My First Join

Left Table	Join Type	Right Table
ORDER_FACT	INNER	CUSTOMER_DIM
ORDER_FACT	INNER	PRODUCT_DIM

Join conditions:

ORDER_FACT	PRODUCT_DIM
Product_ID	Product_ID

Figure 1: SAS Visual Data Builder

Figure 1 shows the SAS Visual Data Builder. A navigation area is available on the left. The middle of the figure shows the main area for creating SQL joins. The right pane contains a list of detailed properties.

SAS Visual Data Builder can perform a full range of SQL functions, perform WHERE clause filtering, generate calculated columns, insert key column variables, and help with partitioning data on a multi-machine grid. After loading data, results can be previewed or explored in SAS Visual Analytics Explorer.

WAYS TO LOAD DATA

SAS Visual Analytics works with data that is loaded into memory. Individual users might want to perform their own loads (we will call that *self-service*) or leverage data that has already been processed by an automated process. Or they might need to access data directly from a DBMS or the Internet.

- Automated, repeatable processes
    - scheduled ETL jobs
    - autoload processing (for spreadsheets, delimited files, and SAS data sets)
    - access to SAS BI relational data defined in a SAS Information Map

- Self-service options that enable users to access additional data
    - Import data from a variety of sources
        - local - spreadsheets, delimited files, SAS data sets
        - servers – databases and SAS data sets
        - social media – data from Internet sources, such as Twitter

- Load on Start -- keep user-loaded data accessible even after administrative downtime to perform tasks such as backups.

First, this paper discusses the self-service operation.

Self-Service User Interface: The simplest way to start analyzing data is to load the data yourself and take a cursory look around. SAS Visual Analytics has a number of ways to facilitate that operation. When getting started on your own, consider where the data is currently and where you need to load the data for initial analysis.

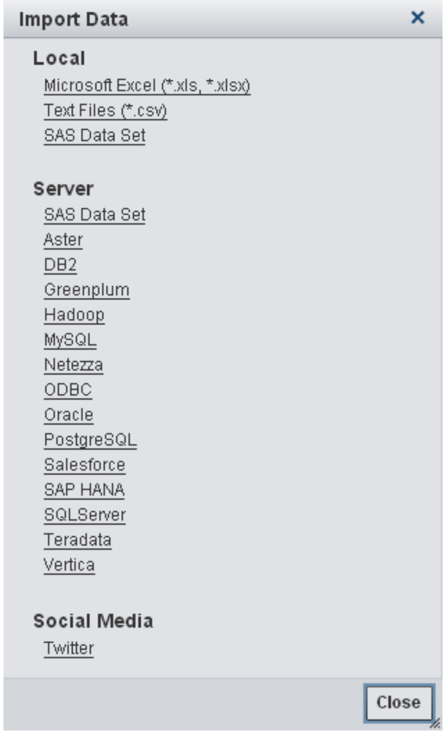
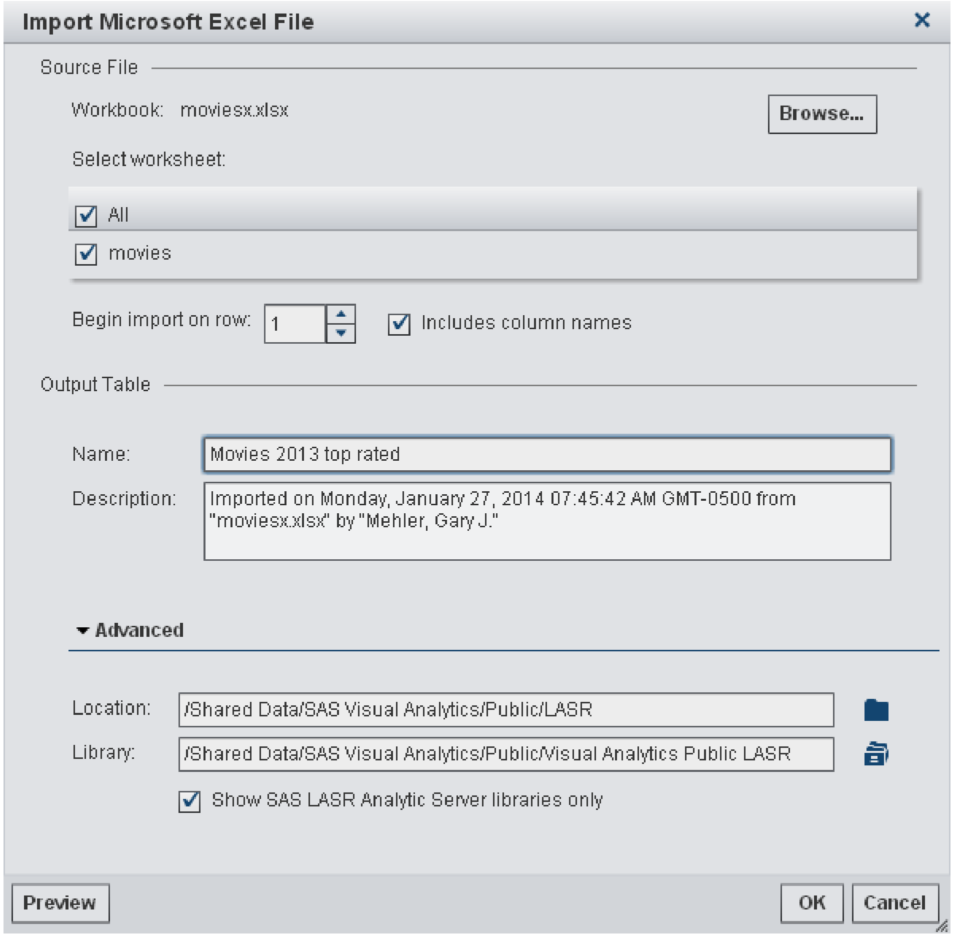


Figure 2: Available Self-service Sources in SAS Visual Analytics 6.4

Many people use Microsoft Excel for basic data manipulation, and Excel provides a good source for loading initial data that has less than 1 million rows. SAS Visual Analytics enables you to import Excel files (XLS or XLSX format) through a simple user interface. Related formats, such as delimited files, are supported as well up to an upload limit of 4GB. This load takes place through the browser-based interface of SAS Visual Analytics.

Next consider, where to load the data. For any non-sensitive data, the Public library is a good location. The Public library is accessible to anyone who has login permission to SAS Visual Analytics.



Workbook: moviesx.xlsx

Select worksheet:

☑ All

☑ movies

Begin import on row: 1

☑ Includes column names

Name: Movies 2013 top rated

Description: Imported on Monday, January 27, 2014 07:45:42 AM GMT-0500 from "moviesx.xlsx" by "Mehler, Gary J."

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

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

☑ Show SAS LASR Analytic Server libraries only

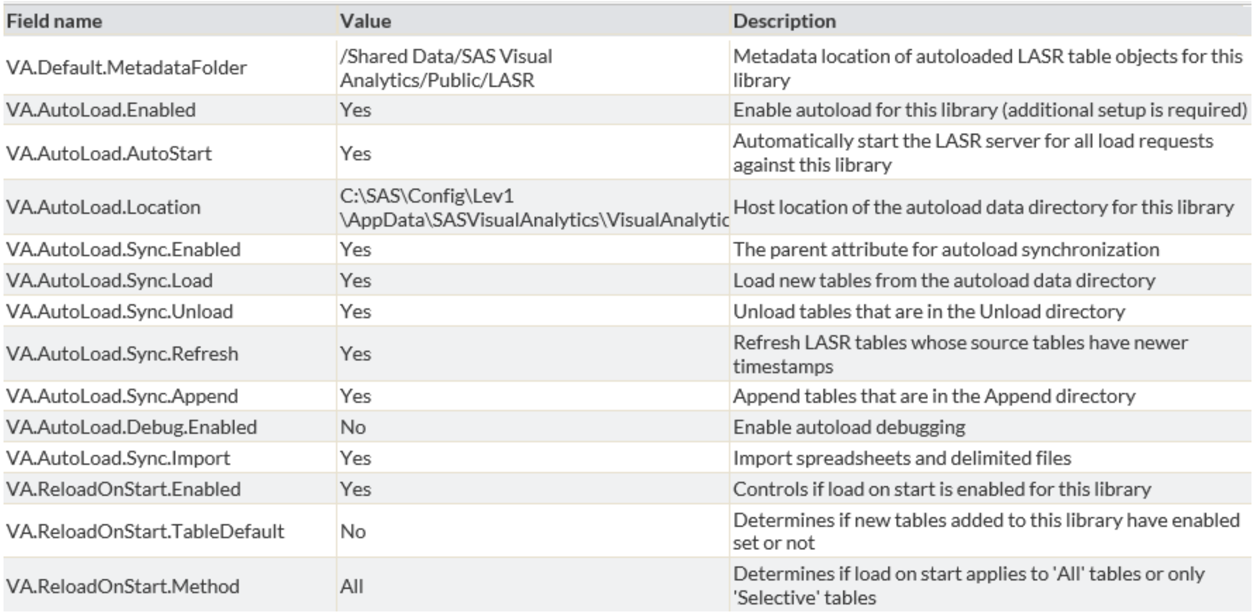
Figure 3. Example of Importing a Microsoft Excel File Using SAS Visual Analytics

Figure 3 shows the advanced options in the Import Microsoft Excel File window. Advanced users can change where the data is loaded. (In this example, it is loaded to the Public library.) A basic user has only the option to select between the default public and private locations.

As a shared library, the Public library works best if users know that the data items can be seen by others and that the user's data co-exists with other users' data. For example, a self-service user adds a file called "Sales Data" to the Public library. No descriptive comment is included, so it is likely that the self-service user could get confused if another user load data to the Public library with a similar name. As a good practice, self-loaded data should have an explanatory name and a robust description to help users understand which data is what. For the "Sales Data" example, loading the data with a name of "1Q2014 Divisional Sales Data" and a description of "Data loaded by My User Name on Tuesday" helps to identify the data. The import tools in SAS Visual Analytics help resolve naming conflicts and automatically generate default descriptions.

The import process also performs some basic housekeeping on imported data. If the data contains names with embedded spaces or other unusual characters for the resulting table or included columns, SAS Visual Analytics prompts the user with a suggested correction. If the user does not like the suggested correction, the user can specify a different name. This helps ensure complete processing of imported data when used later.

Auto-start helps self-service users remain self-sufficient. As described in the section on in-memory data, the analysis is performed on server processes that provide access to the loaded data. As long as the server processes are running, all works well. But what if a self-service operation is attempted immediately following regular system maintenance or a power outage? This is not a problem because the Public library is configured as an auto-starting server process, which means that if a self-service user attempts to load data on their own the server starts when the load starts. Auto-starting is a configurable setting that can be changed by system administrators, but the default operation is to automatically start the server if needed.



Field name	Value	Description
VA.Default.MetadataFolder	/Shared Data/SAS Visual Analytics/Public/LASR	Metadata location of autoloaded LASR table objects for this library
VA.AutoLoad.Enabled	Yes	Enable autoload for this library (additional setup is required)
VA.AutoLoad.AutoStart	Yes	Automatically start the LASR server for all load requests against this library
VA.AutoLoad.Location	C:\SAS\Config\Lev1 \AppData\SASVisualAnalytics\VisualAnalytic	Host location of the autoload data directory for this library
VA.AutoLoad.Sync.Enabled	Yes	The parent attribute for autoload synchronization
VA.AutoLoad.Sync.Load	Yes	Load new tables from the autoload data directory
VA.AutoLoad.Sync.Unload	Yes	Unload tables that are in the Unload directory
VA.AutoLoad.Sync.Refresh	Yes	Refresh LASR tables whose source tables have newer timestamps
VA.AutoLoad.Sync.Append	Yes	Append tables that are in the Append directory
VA.AutoLoad.Debug.Enabled	No	Enable autoload debugging
VA.AutoLoad.Sync.Import	Yes	Import spreadsheets and delimited files
VA.ReloadOnStart.Enabled	Yes	Controls if load on start is enabled for this library
VA.ReloadOnStart.TableDefault	No	Determines if new tables added to this library have enabled set or not
VA.ReloadOnStart.Method	All	Determines if load on start applies to 'All' tables or only 'Selective' tables

Figure 4. Default Values for the Public Library

A wide range of options can be set on in-memory libraries to control start-up, load, and refresh behavior. Figure 4 shows the default values for the standard Public Library. You can view these options in the SAS Environment Manager.

Load on Start keeps user-loaded data accessible. If a server outage has occurred, you do not want to require that self-service users who loaded spreadsheets from their personal computer reload after a server is brought up. To avoid that situation, the Public library supports a default configuration called Load on Start. This capability uses a permanent data provider area where a copy of the loaded self-service data is kept for use as needed. So, when a server restarts, any self-service data that was previously loaded is restored to the SAS LASR Analytic Server through this data provider area.

To ensure that self-service users can load data and have confidence that this data will be available for their reporting needs, the default configuration of the Public library enables both Auto Start and Load on Start. If a site administrator does not want to provide this support or has space limitations for their user community, this functionality can be disabled.

To summarize the self-service operation, users can upload their own data without an administrator. The Public library is always available for data uploading. Options for auto-starting a server session and reloading data at start-up can help keep data available for reporting and analytic exploration.

Auto Loader for batch operations: The next case to consider is that of automated loading of data. Although a user interface is an easy way for first-time users to load their own data, it can become tedious for users operating on a large number of data sources. In addition, data might be automatically provided by regularly scheduled processes. In the automatic case, it is more straightforward to provide a non-UI method for loading collections of data items on a regular basis.

The SAS Visual Analytics Auto Loader is a file system way to perform automatic loads and updates of data. By default, it checks for updates in its file system every 15 minutes and handles data updates appropriately. The Auto Loader must be enabled by the SAS Visual Analytics administrator, but after this initial setup, the Auto Loader runs unattended. The Auto Loader automatically adds a description to the data it handles. This description indicates the date and time that the data was processed.

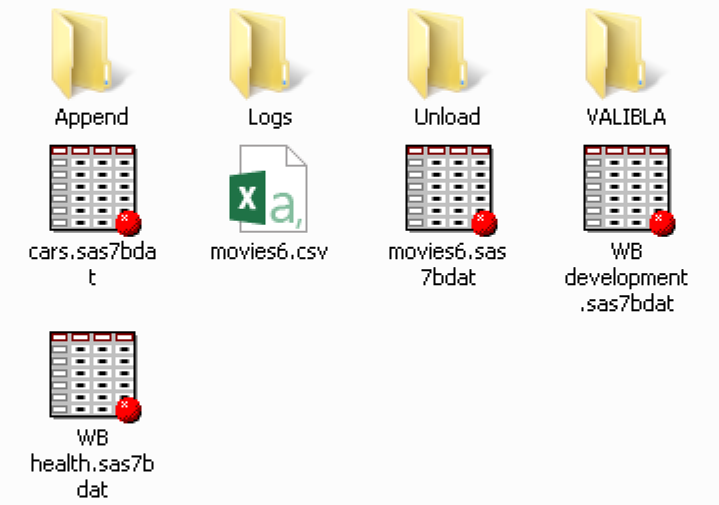


Figure 5. The Files and Folders of an Auto Loader Directory on a Microsoft Windows Operating System

For true unattended operation, the Auto Loader supports several types of operations, which are all controlled by file system directory locations:

- Initial loads are handled by placing files in the main Auto Loader folder. Spreadsheets, CSV files, and SAS data sets can be placed in this location and are appropriately converted and loaded for in-memory analytics.

  In Figure 5, note that there are two items with the name “movies6”. One is a CSV file and the other is a SAS data set. Both files exist because the Auto Loader first converts the CSV file to a SAS data set that can be quickly loaded (and re-loaded) as needed. A user update to either file causes a reload of the in-memory data for “movies6” to reflect the update.
- Updates of loaded data are also handled by updating files in the main Auto Loader folder. When a spreadsheet, CSV file, or SAS data set is updated, the in-memory data for that file is also updated.
- Appending new data is handled by a specific Append subfolder. Placing a spreadsheet, CSV file, or SAS data set in the Append subfolder causes the new content to be appended to data previously loaded into memory. For example, if the initial load of 1Q2014Sales had 1 million rows and an update of 1Q2014Sales added 100,000 rows, the in-memory data is updated to contain 1.1 million rows.

- Dropping a data item from the SAS LASR Analytic Server is accomplished by moving the file from the original directory to a subfolder that specifies to delete the data item.
- A Logs folder collects status information to help track down issues if errors occur during the automated processing performed by Auto Loader.

In summary, for organizations that receive updates via automated processes or use tools like FTP to move data, the Auto Loader area provides a convenient way for processes to load and maintain their data in the SAS LASR Analytic Server. As described for self-service UI operations, the Auto Loader will Auto Start its SAS LASR Analytic Server if needed, further providing for unattended operation. For organizations that deploy SAS Visual Analytics in a cloud environment, the Auto Loader works well with FTP tools that can automatically move data as needed.

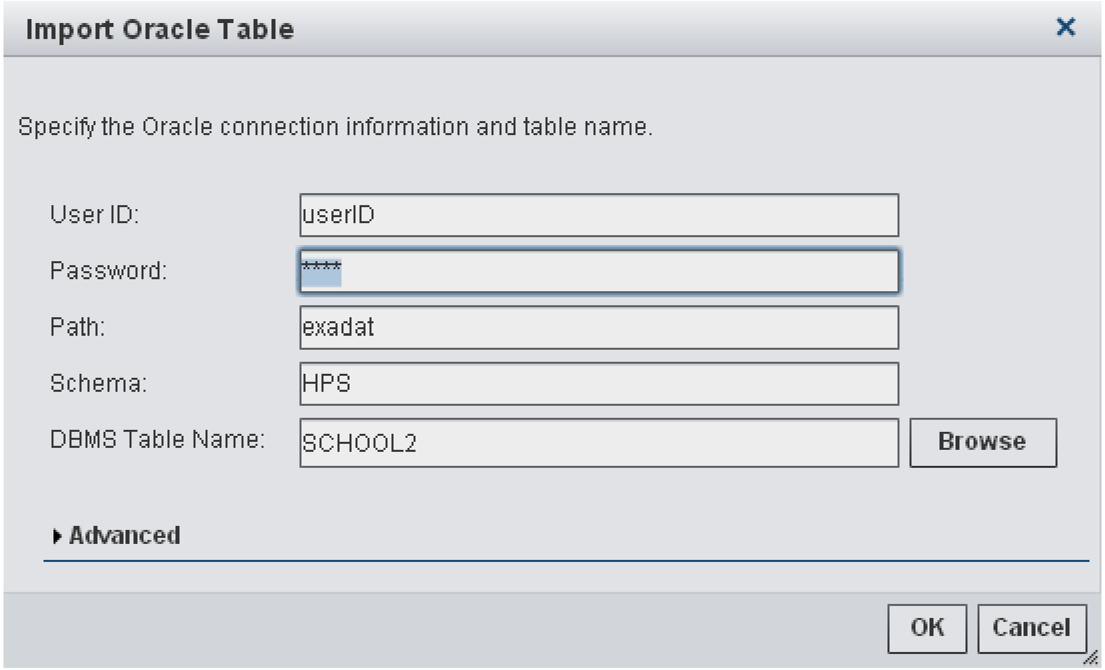
By default, SAS Visual Analytics is configured with an Auto Loader for the Public library. If other Auto Loaders are required for different servers or security contexts, more Auto Loaders can be configured on-site. There is no size limit for the amount of data that can be handled by the Auto Loader.

To summarize the use of an auto-loader, file system data files are loaded and updated on a regular basis. Using the file system can be useful for data that is too large to upload through the user interface of a web browser or as a way to interface with automatically running processes that provide data for analysis.

ADDITIONAL DATA SOURCES

Direct access to DBMS data. Although personal spreadsheets and CSV files are prevalent in many organizations, they can have size limitations and might not be the standard places where shared corporate information is stored. For this reason, SAS Visual Analytics users can also directly load from a wide range of databases, such as Oracle, DB2, Teradata, or other databases using standard protocols like ODBC. A system administrator can pre-define access to some sources and make them available in the standard navigation pane. However, a user who knows the connection information and who has database credentials can retrieve tables using the self-service interface that is available in SAS Visual Analytics Explorer, SAS Visual Analytics Designer, and SAS Visual Data Builder.

Details about a database connection are usually told to users by their database administrator and are entered using the self-service menu. For example, connection to an Oracle Server requires an Oracle user ID, password, data source name, and an optional schema. With this information, a connection is made, and the user can select the data table of interest to load to the SAS LASR Analytic Server. SAS Visual Data Builder stores this connection information (except for the passwords) for future connections. By providing easy access to a range of databases, the self-service user can access a broad range of data sources that previously required administrator configuration.



Import Oracle Table

Specify the Oracle connection information and table name.

User ID: userID

Password: ****

Path: exadat

Schema: HPS

DBMS Table Name: SCHOOL2

Figure 6. An Example of Connecting to an Oracle Table

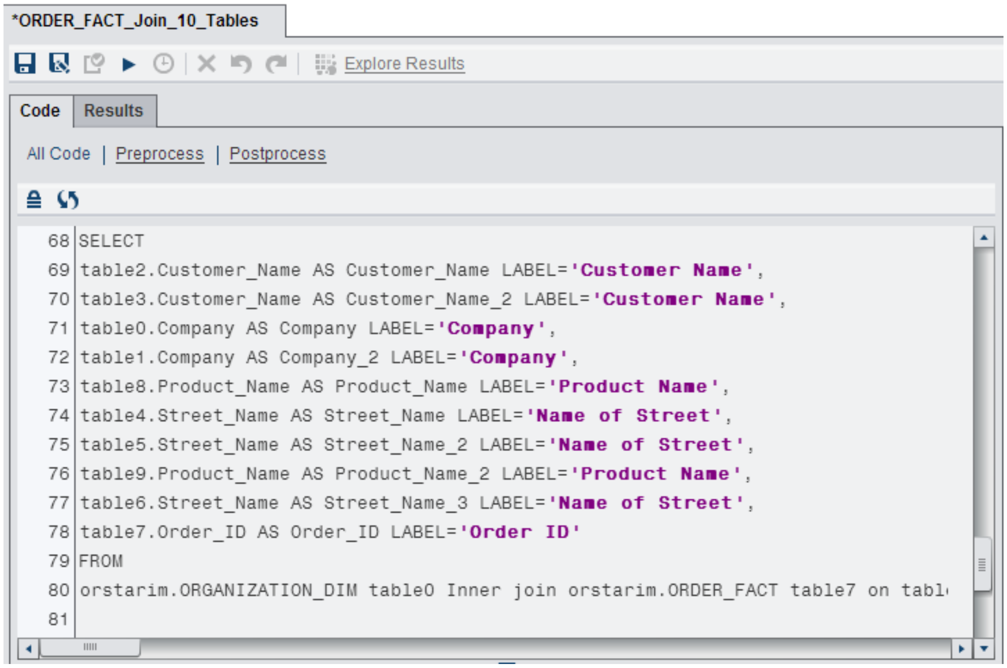
When the source database is set up to use parallel clusters, direct access can be optimized. Because many organizations store large amounts of data in DBMSs, large databases can be configured in parallel clusters. The SAS LASR Analytic Server can perform parallel data extractions to speed the data transfer from such a system.   If possible, parallel data transfers are performed to the SAS LASR Analytic Server. For example, an Oracle DBMS can be configured as an Exadata Database Machine, which can scale too many data nodes.   Other examples of databases include Hadoop, Teradata, or Pivotal Greenplum. Because parallel extracts can impact system and network performance, this mode of operation can be disabled by a site administrator or enabled to facilitate the fastest parallel access possible.

Direct access to Social Media data. The Internet provides access to a wide array of information sources including up-to-the-minute micro-blogging services like Twitter. SAS Visual Analytics provides a direct ability to search for specific strings or hashtags at twitter.com and brings that data into the SAS LASR Analytic Server. Here are some examples of when this would be helpful:

- Looking for mentions of your organization, products, or notable people. Along with the basic information of what is being said, metadata (such as retweet count, tweet location, and frequency) provide timely information about real-time trends.
- Applying text analytics to the Twitter messages themselves can provide insights about the topics being discussed or questions being asked.

For Twitter access, the user must have a valid Twitter account, and the user must authorize SAS Visual Analytics to use their credentials to perform and see search results. Authorization is completed using the sign-in processes through the self-service import menus. The user can search Twitter streams for any topic relevant to their analysis and get current data into their reports or analytic explorations. Use of information from twitter.com is subject to the account agreement the user has with twitter.com. The rate at which information can be extracted might also be limited by twitter.com.

Access to SAS BI information. SAS® Enterprise BI Server uses relational data represented as a SAS Information Map. These maps can have complex structures that represent applied filters, joins, and join orders, as well as column attributes like readable labels. To make this information easy-to-use, SAS Visual Data Builder provides a way to import such information into the in-memory environment for SAS Visual Analytics.


*ORDER_FACT_Join_10_Tables
Explore Results
Code
Results
All Code | Preprocess | Postprocess

```
SELECT
table2.Customer_Name AS Customer_Name LABEL='Customer Name',
table3.Customer_Name AS Customer_Name_2 LABEL='Customer Name',
table0.Company AS Company LABEL='Company',
table1.Company AS Company_2 LABEL='Company',
table8.Product_Name AS Product_Name LABEL='Product Name',
table4.Street_Name AS Street_Name LABEL='Name of Street',
table5.Street_Name AS Street_Name_2 LABEL='Name of Street',
table9.Product_Name AS Product_Name_2 LABEL='Product Name',
table6.Street_Name AS Street_Name_3 LABEL='Name of Street',
table7.Order_ID AS Order_ID LABEL='Order ID'
FROM
orstarim.ORGANIZATION_DIM table0 Inner join orstarim.ORDER_FACT table7 on tabl
```

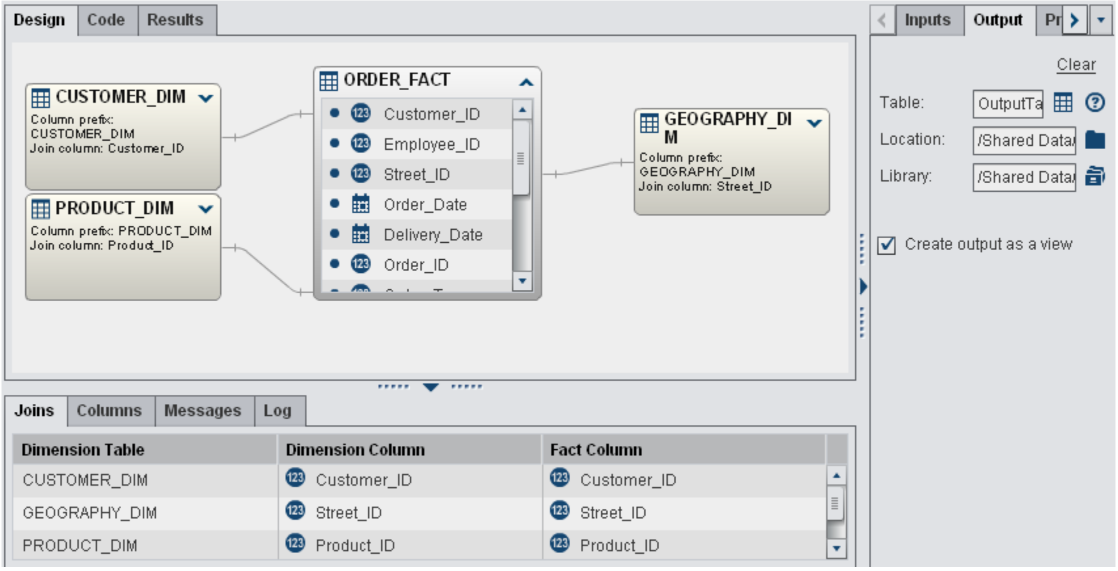
Figure 7. Data Manipulation Information in an SQL Step

CONSUMING THE DATA

Getting data into an environment like the SAS LASR Analytics Server is a necessary first step to performing analytics. However, the external data might not be in a form that is directly consumable for initial exploration or reporting. For this reason, the data might need to be joined or otherwise restructured before it can be used effectively. The most common of these transformations is performing an SQL join on operational tables to construct a wider single table that contains the variables to use in the analysis.

Joining and star schemas. SAS Visual Data Builder provides a user interface to design and perform SQL joins. These joins can be performed inside a database system or in the SAS environment. Using the built-in scheduling user interface, regular data loads can joined their tables as needed.

If the data is very large, performing these joins can be very time-consuming and generate very large output tables that are difficult to transfer and otherwise manage. The SAS LASR Analytics Server supports the in-memory construction of star schema structures that can be processed far more quickly than traditional SQL joins. In addition, an in-memory star schema can use space more efficiently for the resulting denormalization, so this approach should be considered when you need to combine data from multiple tables in a star schema configuration. This situation occurs when the result of generating an in-memory star schema is accessed as a view that references a source fact and dimension tables. You can construct a star schema table in SAS Visual Data Builder.



CUSTOMER_DIM
Column prefix: CUSTOMER_DIM
Join column: Customer_ID

PRODUCT_DIM
Column prefix: PRODUCT_DIM
Join column: Product_ID

ORDER_FACT
Customer_ID
Employee_ID
Street_ID
Order_Date
Delivery_Date
Order_ID

GEOGRAPHY_DIM
Column prefix: GEOGRAPHY_DIM
Join column: Street_ID

Table: OutputTa

Location: /Shared Data/

Library: /Shared Data/

☑ Create output as a view

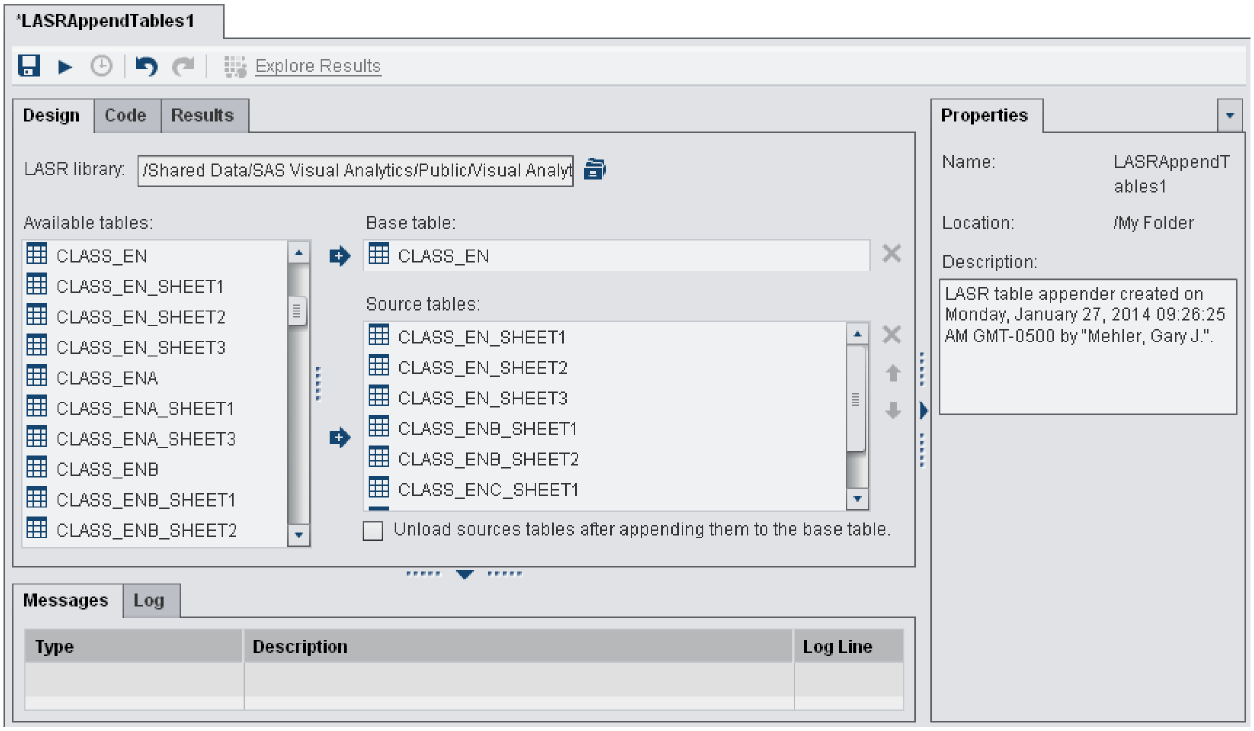
Dimension Table	Dimension Column	Fact Column
CUSTOMER_DIM	Customer_ID	Customer_ID
GEOGRAPHY_DIM	Street_ID	Street_ID
PRODUCT_DIM	Product_ID	Product_ID

Figure 8. Construction of an In-memory Star Schema Using SAS Visual Data Builder

Here are some differences between joined denormalized starts and in-memory star schemas:

- Star schemas are created in-memory so they can be generated quickly.
- There needs to be sufficient memory for the fact and dimension tables, as well as linkage tables that map dimension rows to fact entities.
- The star schema is formed by single-column keys that map fact columns to dimensions.
- Star schemas can be created as views or as full tables. For the quickest creation time, a view is preferable.
- Star schema creation happens in-memory, so fact and dimension tables need to be available. SAS Visual Data Builder loads these tables if needed, or these tables might already be available as a result of some other process.
- If row-level security has been applied to component tables (such as to restrict sales transaction facts to the region of the current user), these row-level conditions need to be applied again to the resulting star schema. The row-level conditions of source tables do not propagate to the resulting star schema, even if the output is a view.
- Creating an in-memory star schema can help solve problems with memory use, particularly if dimensions have a lot of wide variables. Creating an in-memory view is a very memory-economical way to create and utilize a star schema that would otherwise be too large to fit into memory if constructed and denormalized with classic SQL code.
- Keep in mind that tables loaded from distributed storage (like HDFS: Hadoop Distributed Filesystem) on a grid-based system work best with memory utilization in an SAS LASR Analytic Server. These tables can be loaded and accessed very quickly, so they might not be in memory and still function correctly. Building a star schema on these types of tables makes the best use of memory resources on a grid-based system.

Appending new rows. After loaded in memory, data might change as new events occur. To make it easier to keep the data as fresh as possible, new rows can be appended as needed by using a very quick in-memory Append operation. For example, consider an initial load of sales data that needs daily updates. The initial load of 1 million rows needs to have about 10,000 rows added each hour or day to keep analysis results current. In-memory append operations allow this update to occur as often as needed, very quickly. To perform this update, add a table of the new rows to the SAS LASR Analytic Server and perform an Append operation.



LASR library: /Shared Data/SAS Visual Analytics/Public/Visual Analyt

Available tables:

- CLASS_EN
- CLASS_EN_SHEET1
- CLASS_EN_SHEET2
- CLASS_EN_SHEET3
- CLASS_ENA
- CLASS_ENA_SHEET1
- CLASS_ENA_SHEET3
- CLASS_ENB
- CLASS_ENB_SHEET1
- CLASS_ENB_SHEET2

Base table:

- CLASS_EN

Source tables:

- CLASS_EN_SHEET1
- CLASS_EN_SHEET2
- CLASS_EN_SHEET3
- CLASS_ENB_SHEET1
- CLASS_ENB_SHEET2
- CLASS_ENC_SHEET1

☐ Unload sources tables after appending them to the base table.

Name: LASRAppendTables1

Location: /My Folder

Description:

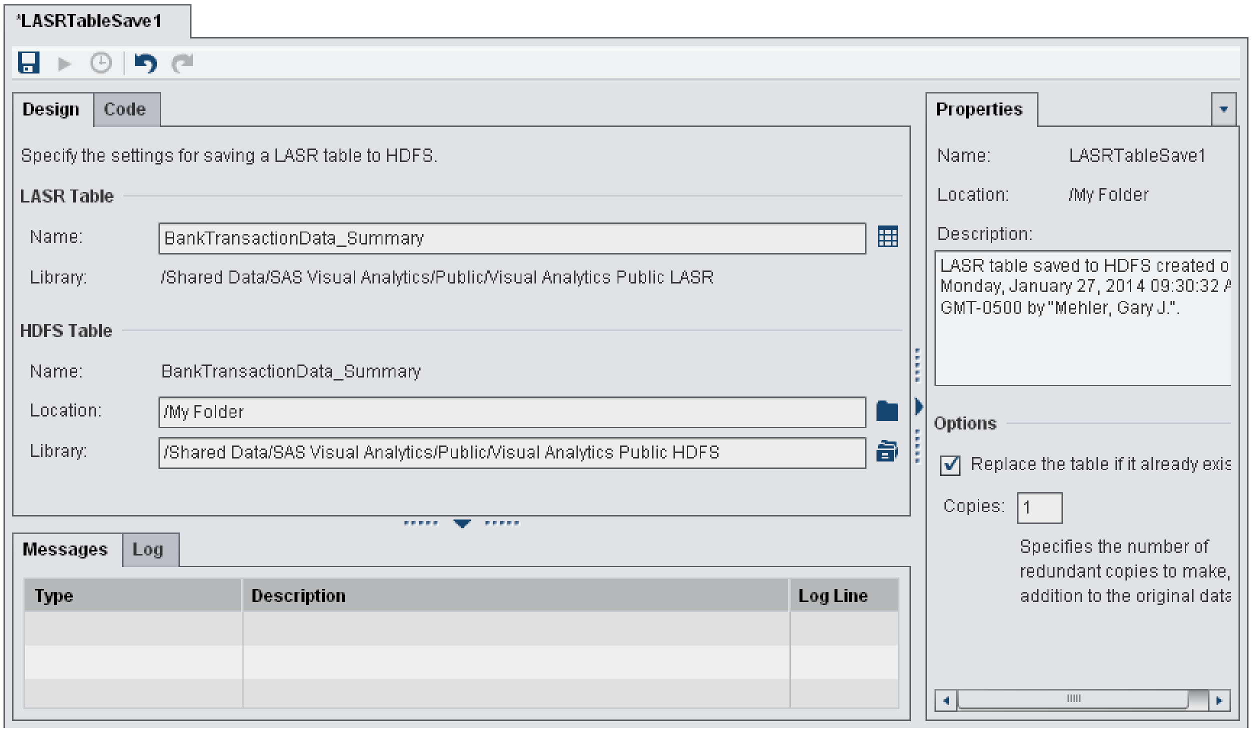
LASR table appender created on Monday, January 27, 2014 09:26:25 AM GMT-0500 by "Mehler, Gary J.".

Type	Description	Log Line

Figure 9. Appending Several Tables to a Base Table

Append operations can also work with star schema configurations for maximum benefit. Assuming that dimensions do not change very frequently, new rows can be appended to the fact table of the star schema, and the star schema quickly re-created to provide access to the newly added rows. By performing both steps in-memory, new information is made available for analysis as quickly as possible, which provides the maximum timely benefit. Figure 9 shows this type of operation in SAS Visual Data Builder.

Saving updates to permanent storage. If in-memory data is updated with star schema operations or append operations, an update in permanent storage could be helpful. This is particularly true if the data has been appended multiple times since the initial in-memory load. In this case, writing the information out is a good safety net for fast data recoverability. Figure 10 shows saving to permanent storage in SAS Visual Data Builder.



Specify the settings for saving a LASR table to HDFS.

LASR Table

Name: BankTransactionData_Summary

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

HDFS Table

Name: BankTransactionData_Summary

Location: /My Folder

Library: /Shared Data/SAS Visual Analytics/Public/Visual Analytics Public HDFS

Name: LASRTableSave1

Location: /My Folder

Description: LASR table saved to HDFS created o Monday, January 27, 2014 09:30:32 A GMT-0500 by "Mehler, Gary J.".

Options

☑ Replace the table if it already exis

Copies: 1

Specifies the number of redundant copies to make, addition to the original data

Figure 10. Saving an In-memory Table to Permanent Storage

This fast save interface is available on grid-based, in-memory servers that have use Hadoop for parallel HDFS storage. Parallel storage means the data can be saved quickly and re-loaded quickly when needed. By default, one extra copy is the data is stored, so each block of data is stored on two different nodes of a grid. As a result, one node could go down or be missing, and you could use the other node to access the complete data.

If the data is being saved transiently (to facilitate an immediate reload), this redundancy might not be needed. You might want an extra copy to ensure that the data is not damaged in case of a node failure.

UNDERSTANDING THE END-TO-END DATA PROCESS

In the real world, data comes from a range of sources and with a range of update frequencies. For analytic accuracy, the data might need to remain as current as possible. Imagine the following scenario:

- detailed data from spreadsheets or files
- large organizational data from databases
- up-to-the-minute real-world demographics from social media sites
- reference data that is delivered by automated external processes
- data that needs to be joined together to perform an analytic view
- data that needs to receive regular updates as new data rows are delivered
- for persistence, a permanent copy needs to be regularly saved for recoverability
- data size needs to be monitored to ensure that the in-memory analytics environment remains fast and reliable

This diagram explains how all the pieces fit together.

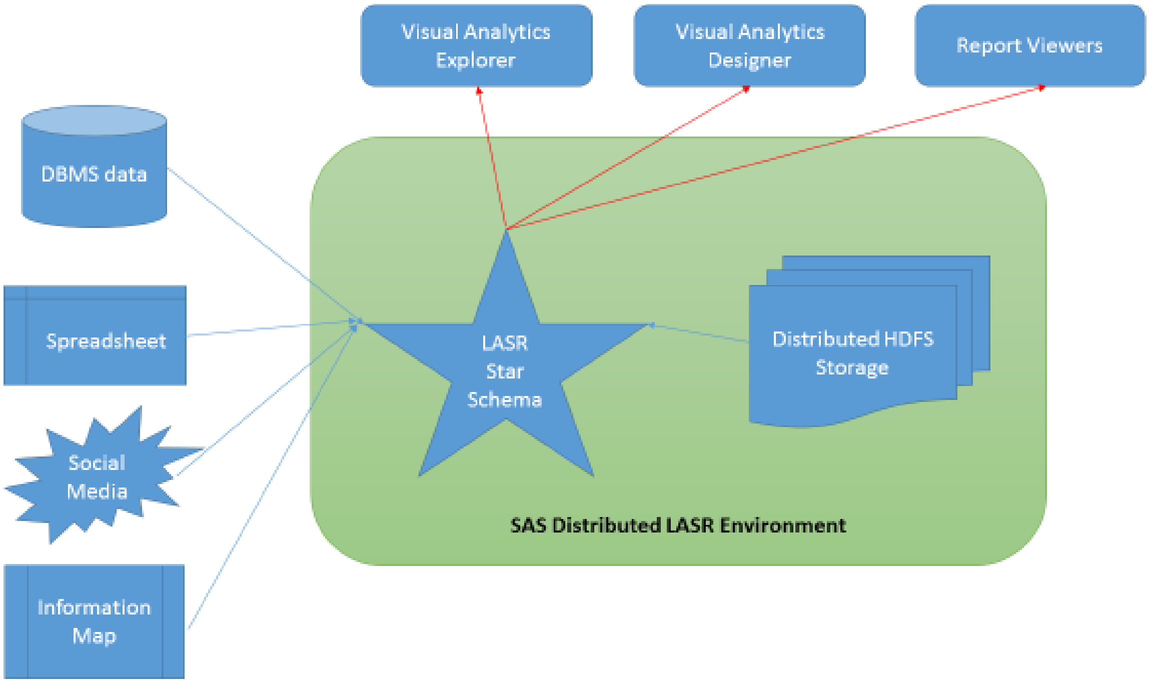


Figure 11. Initial Load and Data Update (Append) operation for SAS Visual Analytics Explorer and Designer

To get started, bring all the data pieces together and construct the in-memory structure. This example assumes a LASR star schema. The initial load places data into memory and then constructs the LASR star schema view. At that point, the data is available for consumption by the SAS Visual Analytics applications.

The flow is repeated as updates are appended to component tables that are still in memory. During the update period, queries against the LASR star schema still return results as constructed originally. When the updated data is ready and the view ready to be refreshed, a re-run of the star schema creation step quickly refreshes the contents, so these contents are now available for subsequent queries. The refresh is a very fast operation that results in minimal downtime for data consumers.

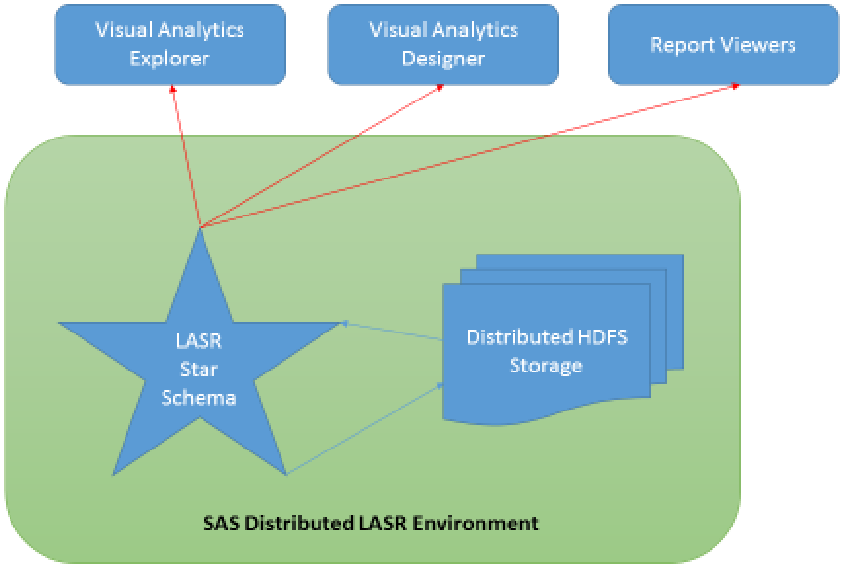

Visual Analytics Explorer
Visual Analytics Designer
Report Viewers
LASR Star Schema
Distributed HDFS Storage
SAS Distributed LASR Environment

Figure 12. Saving and Restoring Updated Data to Distributed Storage

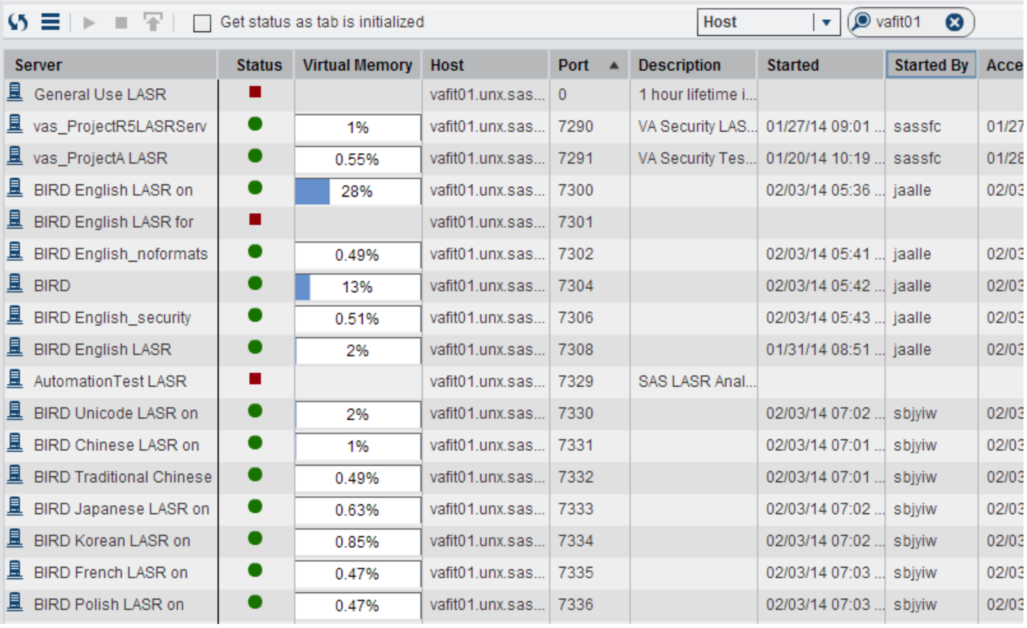
As a final step, constructed data can be saved to distributed storage and reloaded. This step ensures that the updated star can be re-created quickly in the event of a service interruption or periodic system maintenance. Writing out occurs in parallel in order to take the minimal amount of time, as does the eventual reload. The reload can be scheduled to run during a period of relative low activity or after periodic maintenance is performed.

MONITORING AN ENVIRONMENT

After the data has been loaded and consumed, it is helpful to be able to monitor the state of LASR processes and the accessible relational tables. SAS Visual Analytics Administrator has capabilities that can assist with these tasks.

Different collections of data can be served by different LASR Analytic Server instances. Figure 13 shows an internal testing system with a large number of different servers. This system is being used for development and testing purposes. Status and memory consumption are shown for easy status checking. A global filter can focus the view to a specified collection of server processes.

This view also shows information about when each server process was started and by whom. If a server process is not running, you can start it from this interface as well. Alternatively, you can shut down server processes. These operations are helpful during recurring system maintenance windows, when hardware is restarted, or when other systems administration is performed.

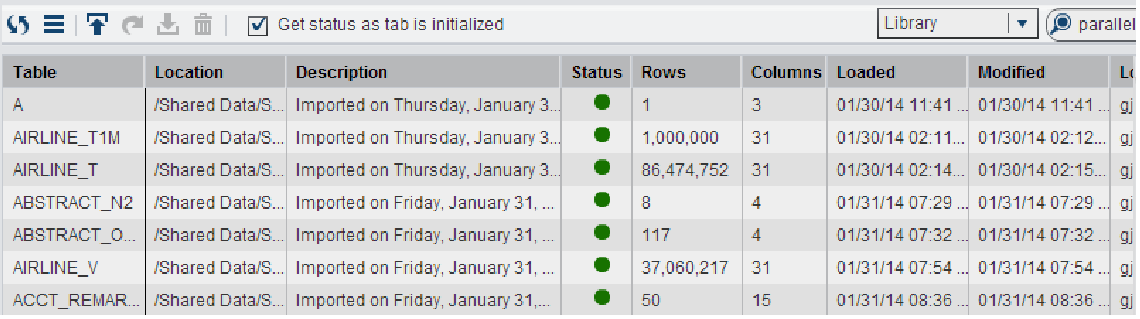


□ Get status as tab is initialized | Host ▾ | vafit01

Server	Status	Virtual Memory	Host	Port ▲	Description	Started	Started By	Acce
General Use LASR	■		vafit01.unx.sas...	0	1 hour lifetime i...			
vas_ProjectR5LASRServ	●	1%	vafit01.unx.sas...	7290	VA Security LAS...	01/27/14 09:01 ...	sassfc	01/27
vas_ProjectA LASR	●	0.55%	vafit01.unx.sas...	7291	VA Security Tes...	01/20/14 10:19 ...	sassfc	01/28
BIRD English LASR on	●	28%	vafit01.unx.sas...	7300		02/03/14 05:36 ...	jaalle	02/03
BIRD English LASR for	■		vafit01.unx.sas...	7301				
BIRD English_noformats	●	0.49%	vafit01.unx.sas...	7302		02/03/14 05:41 ...	jaalle	02/03
BIRD	●	13%	vafit01.unx.sas...	7304		02/03/14 05:42 ...	jaalle	02/03
BIRD English_security	●	0.51%	vafit01.unx.sas...	7306		02/03/14 05:43 ...	jaalle	02/03
BIRD English LASR	●	2%	vafit01.unx.sas...	7308		01/31/14 08:51 ...	jaalle	02/03
AutomationTest LASR	■		vafit01.unx.sas...	7329	SAS LASR Anal...			
BIRD Unicode LASR on	●	2%	vafit01.unx.sas...	7330		02/03/14 07:02 ...	sbjyiw	02/03
BIRD Chinese LASR on	●	1%	vafit01.unx.sas...	7331		02/03/14 07:01 ...	sbjyiw	02/03
BIRD Traditional Chinese	●	0.49%	vafit01.unx.sas...	7332		02/03/14 07:01 ...	sbjyiw	02/03
BIRD Japanese LASR on	●	0.63%	vafit01.unx.sas...	7333		02/03/14 07:02 ...	sbjyiw	02/03
BIRD Korean LASR on	●	0.85%	vafit01.unx.sas...	7334		02/03/14 07:02 ...	sbjyiw	02/03
BIRD French LASR on	●	0.47%	vafit01.unx.sas...	7335		02/03/14 07:03 ...	sbjyiw	02/03
BIRD Polish LASR on	●	0.47%	vafit01.unx.sas...	7336		02/03/14 07:03 ...	sbjyiw	02/03

Figure 13. Status of Individual LASR Analytic Server Processes

In addition to tracking the LASR Analytic Server process, you can track the status of individual relational tables. Figure 14 shows a view of individual tables and their status. If a table is loaded and available for reporting and analysis, the table has a good status. This view also shows some information about the data (including row and column count), when it was loaded, and who performed that activity.



☑ Get status as tab is initialized | Library ▾ | parallel

Table	Location	Description	Status	Rows	Columns	Loaded	Modified	Lo
A	/Shared Data/S...	Imported on Thursday, January 3...	●	1	3	01/30/14 11:41 ...	01/30/14 11:41 ...	gj
AIRLINE_T1M	/Shared Data/S...	Imported on Thursday, January 3...	●	1,000,000	31	01/30/14 02:11...	01/30/14 02:12...	gj
AIRLINE_T	/Shared Data/S...	Imported on Thursday, January 3...	●	86,474,752	31	01/30/14 02:14...	01/30/14 02:15...	gj
ABSTRACT_N2	/Shared Data/S...	Imported on Friday, January 31, ...	●	8	4	01/31/14 07:29 ...	01/31/14 07:29 ...	gj
ABSTRACT_O...	/Shared Data/S...	Imported on Friday, January 31, ...	●	117	4	01/31/14 07:32 ...	01/31/14 07:32 ...	gj
AIRLINE_V	/Shared Data/S...	Imported on Friday, January 31, ...	●	37,060,217	31	01/31/14 07:54 ...	01/31/14 07:54 ...	gj
ACCT_REMAR...	/Shared Data/S...	Imported on Friday, January 31,...	●	50	15	01/31/14 08:36 ...	01/31/14 08:36 ...	gj

Figure 14. Status of Individual LASR Analytic Server Tables

As data is loaded, updated, and consumed, these views help an administrator understand the state of their environment as well as the condition of data.

CONCLUSION

With all of the data out there, it can be difficult to keep all data channels open for self-service and IT-based sources while also keeping the data as up-to-date and available. Thankfully, the SAS LASR Analytic Server and SAS Visual Analytics make this possible and straightforward through a number of ways to perform data loads and updates. These products provide as much capability as possible without involving IT in regular operations. These products also keep data load tasks and analytics operating quickly so that processing does not get bogged down as data gets larger and larger. Being able to perform key operations in-memory means maximum efficiency and throughput. Using SAS Visual Analytics means obtaining current results from a wide range of data and update patterns.

REFERENCES

Alexander, Malcom and Sam Atassi. 2013. “Self-Service Data Management: Visual Data Builder.” *Proceedings of the SAS Global Forum 2013 Conference.* Cary, NC. SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings13/051-2013.pdf.

SAS Institute Inc. 2014. *SAS Visual Analytics User’s Guide*. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Gary Mehler
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Gary.Mehler@sas.com
http://www.sas.com

Donna Bennett
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Donna.Bennett@sas.com
http://www.sas.com