Paper SAS1877 – 2015

Access, Modify, Enhance: Self-Service Data Management in SAS® Visual Analytics

Gregor Herrmann and Anand Chitale, SAS Institute Inc., Cary, NC

ABSTRACT

SAS® Visual Analytics provides self-service capabilities for users to analyze, explore, and report on their own data. As users explore their data, there is always a need to bring in more data sources, create new variables, combine data from multiple sources, and even periodically update data. SAS Visual Analytics provides targeted user capabilities to access, modify, and enhance data that is suitable for specific business needs. This paper provides a clear explanation of these capabilities and suggests best practices for self-service data management in SAS Visual Analytics including essential performance considerations.

INTRODUCTION

Sources of data have now gone beyond the boundaries of IT managed enterprise data warehouses, and organizations are facing the flux of ad hoc sources that business users need to make more informed decisions.

One of the key criteria for successful use of a BI application for business is having the ability to bring users' ad hoc data sources in a self-service manner for data analysis without depending on the IT resources. Apart from just access to ad hoc data source, there is also an increasing need to enhance data suitable for the needs of analysis and without the need to request that the IT department make changes to the centralized data source, which can often take a long time.

Business solutions that allow self-service data access and data manipulation for business analysts are gaining more and more traction. SAS Visual Analytics comes with capabilities that can empower specifically enabled end users to bring their own data into the environment and to further refine it by modifying existing and/or adding new data items. The goal of this self-service data management capability is to provide a managed, yet self-service way, for users to provision and prepare their own data without always having to rely on IT. The subsequent sections of this document dive into more details about these self-service activities.

Please note that SAS Visual Analytics provides data management capabilities for IT as well. However, those capabilities are not the primary focus of this paper. Please refer to the recommended reading for more information.

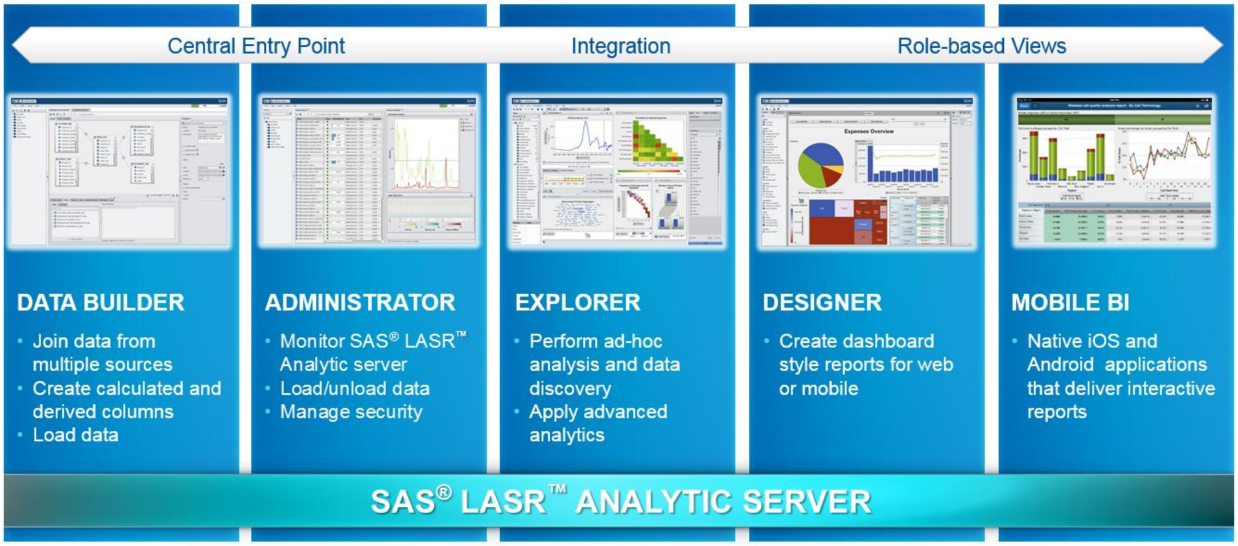
SAS VISUAL ANALYTICS TECHNOLOGY OVERVIEW

SAS Visual Analytics is a BI solution that uses intelligent ways to help business analysts and nontechnical users visualize data, see patterns and trends, and identify opportunities for further analysis. SAS Visual Analytics is backed by the power of SAS® Analytics, which is made available to the users in a self-service and approachable manner. SAS Visual Analytics enables the creation and dissemination of dashboards, reports, and the results of investigative exploration either to the web or to native mobile applications.

Exploration and Visualization
Power of Analytics
Rapid Delivery of Mobile Insights

Figure 1. Business Visualization Driven by Analytics

SAS Visual Analytics delivers a number of capabilities that run on an enterprise class SAS® 9.4 platform and that deliver an end-to-end solution for all user needs. SAS Visual Analytics includes the capability to prepare data prior to making it available to users, an interface for exploring your data (often known as data discovery), and an interface for building highly interactive and visual reports and dashboards.



Central Entry Point

Integration

Role-based Views

DATA BUILDER

- Join data from multiple sources
- Create calculated and derived columns
- Load data

ADMINISTRATOR

- Monitor SAS® LASR™ Analytic server
- Load/unload data
- Manage security

EXPLORER

- Perform ad-hoc analysis and data discovery
- Apply advanced analytics

DESIGNER

- Create dashboard style reports for web or mobile

MOBILE BI

- Native iOS and Android applications that deliver interactive reports

SAS® LASR™ ANALYTIC SERVER

Figure 2. SAS Visual Analytics Capabilities Delivered in a Single Solution for Faster and Smarter Decisions

SAS Visual Analytics also includes native applications that run on Android devices or the iPad for the delivery of information to augment the web. All of this is enabled by the revolutionary and unique SAS Analytics in-memory engine, SAS® LASR™ Analytic Server, which sits under the surface and can scale from handling a few hundred megabytes of data to many terabytes. In other words, this is a solution that can start small and grow with you.

In the sections below, we discuss how SAS Visual Analytics provides self-service approaches for accessing, modifying, and enhancing data.

SELF-SERVICE DATA LOADING

KEY DRIVERS FOR SELF-SERVICE ACCESS TO DATA

As users explore their data and create their own reports, there is always a need to bring in more data sources, combine data from multiple sources, update data periodically, and in many cases generate new data variables that are not available in the data source. Over the years, these self-service needs have become a hot topic and a default expectation by business users using any business intelligence application. There are two cases where this is particularly important:

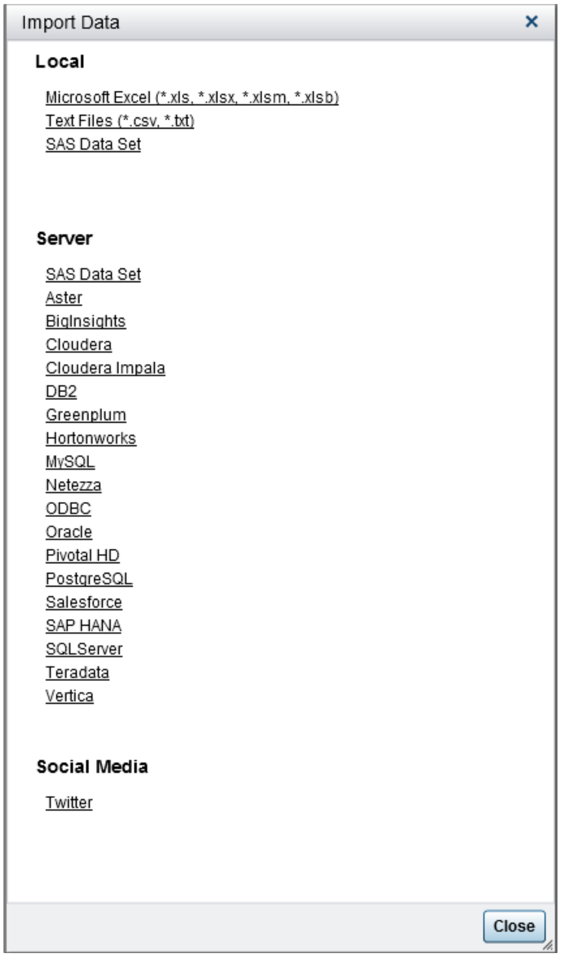
- Exploring local data files: It is no surprise that business users tend to store valuable information on their personal PC and would like to incorporate this information wherever it makes sense. This could be specific lookup information for reference or specific numbers such as the latest currency conversion numbers or ratio applications that are required to enhance the information appropriate for business needs.
- Adding data sources to the existing data warehouse: There is always a need to bring in additional information to an existing report. An example could be data that can be accessed via the Internet, a statistic that has just been released, or social media data, and so on—essentially, anything that needs to be part of analysis or simply a spreadsheet that you just received from a colleague. It might also be the case that tables stored in a database are not permanently available for exploration or report creation.

In both cases, it is of vital importance that the integration of these sources is as easy as possible and is seamlessly embedded into the workflow of the analyst. SAS Visual Analytics supports this requirement by enabling you to add data sources at any time during data exploration or report creation within the same interface.

SUPPORTED FILE FORMATS

When it comes to self-service access to data, it is expected that not all the data will always be coming from the same source and in the same file format. SAS Visual Analytics supports all file formats of a regular SAS session. Access to data in SAS is governed by licensed SAS/ACCESS® software engines that are configured for the system. After access is initiated, only the available SAS/ACCESS engines will be presented to a user as a choice for available file formats.

Apart from the structured data that is accessible from database systems and local data file formats, SAS Visual Analytics also supports analyzing unstructured data. You can use the data streams coming from Twitter with special built-in access to Twitter.com. The only prerequisite is to have a valid Twitter ID. Figure 3 shows all the accessible data sources and their supported file formats.



Import Data ✕

Local

- Microsoft Excel (*.xls, *.xlsx, *.xlsm, *.xlsb)
- Text Files (*.csv, *.txt)
- SAS Data Set

Server

- SAS Data Set
- Aster
- BigInsights
- Cloudera
- Cloudera Impala
- DB2
- Greenplum
- Hortonworks
- MySQL
- Netezza
- ODBC
- Oracle
- Pivotal HD
- PostgreSQL
- Salesforce
- SAP HANA
- SQLServer
- Teradata
- Vertica

Social Media

- Twitter

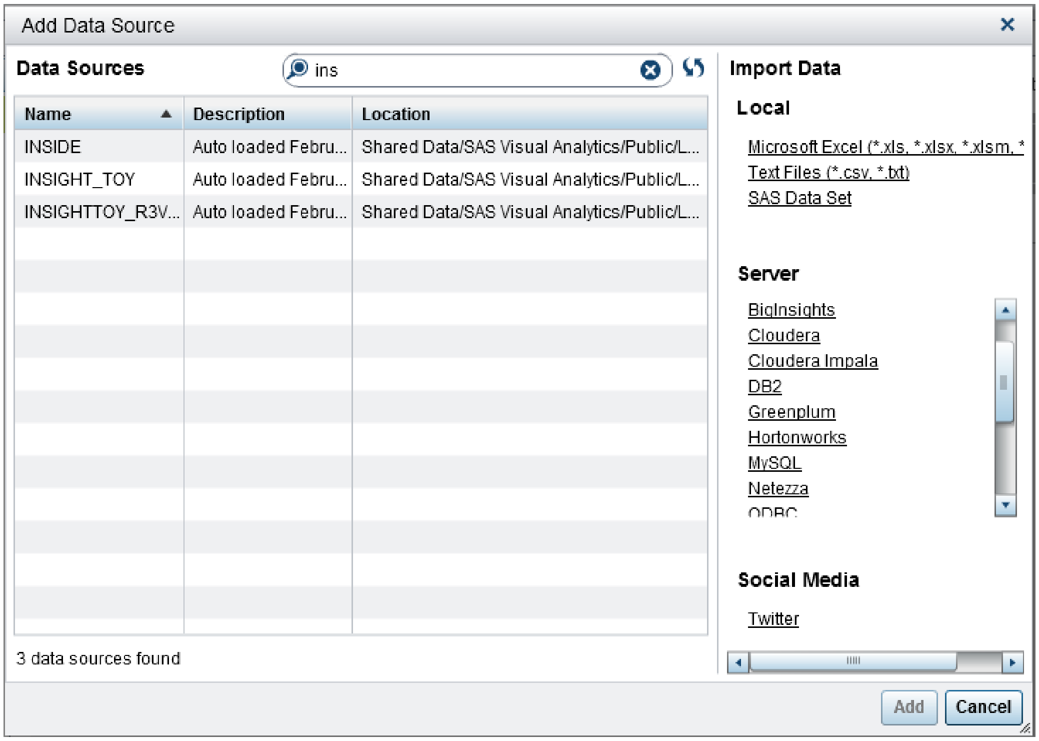
Close

Figure 3. Import Data Dialog Showing a Choice of Supported File Formats

* The upcoming release of SAS Visual Analytics will add Facebook and Google Analytics and more Hadoop versions to the list of available data sources.

AD HOC DATA ACCESS IN SAS® VISUAL ANALYTICS DESIGNER OR SAS® VISUAL ANALYTICS EXPLORER

In both SAS Visual Analytics Designer and SAS Visual Analytics Explorer, you can add data whenever necessary via the **Add Data Source** window:



ins

Name	Description	Location
INSIDE	Auto loaded Febru...	Shared Data/SAS Visual Analytics/Public/L...
INSIGHT_TOY	Auto loaded Febru...	Shared Data/SAS Visual Analytics/Public/L...
INSIGHTTOY_R3V...	Auto loaded Febru...	Shared Data/SAS Visual Analytics/Public/L...

3 data sources found

Figure 4. Add Data Source Window in SAS Visual Analytics Designer with Accessible Data Sources

The range of data sources that analysts want to explore or include in their reports has already broadened in recent years, and it can be expected that it will further grow in the future. SAS Visual Analytics supports this need by enabling the user to access multi-structured data for advanced analytics.

The user interface groups the distinctive data sources as Local, Server, and Social Media.

- **Local** is everything available on a user's PC or laptop, where you use SAS Visual Analytics.
- **Server** is the machine where data sources reside. Depending on your file format, it might be required to enter credential information to connect to the respective data source. For example, if you want to load a database table coming from an RDBMS system like Oracle, the user interface will present the option to provide database access user credentials and data table to access. Server also lists the available Hadoop sources that can be accessed via SAS/ACCESS® Interface to Hadoop, which comes with SAS Visual Analytics, if you are using Hadoop as your data persistence layer. SAS/ACCESS Interface to Hadoop lets you interact with your data using SQL constructs through Hive and HiveServer2. It also lets you access data directly from the underlying data storage layer, the Hadoop Distributed File System (HDFS). This differs from the traditional SAS/ACCESS engine behavior, which exclusively uses database SQL to read and write data.
- **Social Media** provides the option to create a connection to Twitter.com with a valid Twitter ID.

If your data is stored locally, it must be uploaded to the SAS LASR Analytic Server (in memory) to be further accessible inside SAS Visual Analytics. Because uploading these data files might take some time depending on file size and the bandwidth of network connection, an administrator can limit the size of files that can be uploaded to avoid user frustration. Nevertheless, files of a reasonable size (up to couple of megabytes) will be uploaded quickly.

SELF-SERVICE DATA ACCESS VIA THE AUTO LOAD MECHANISM

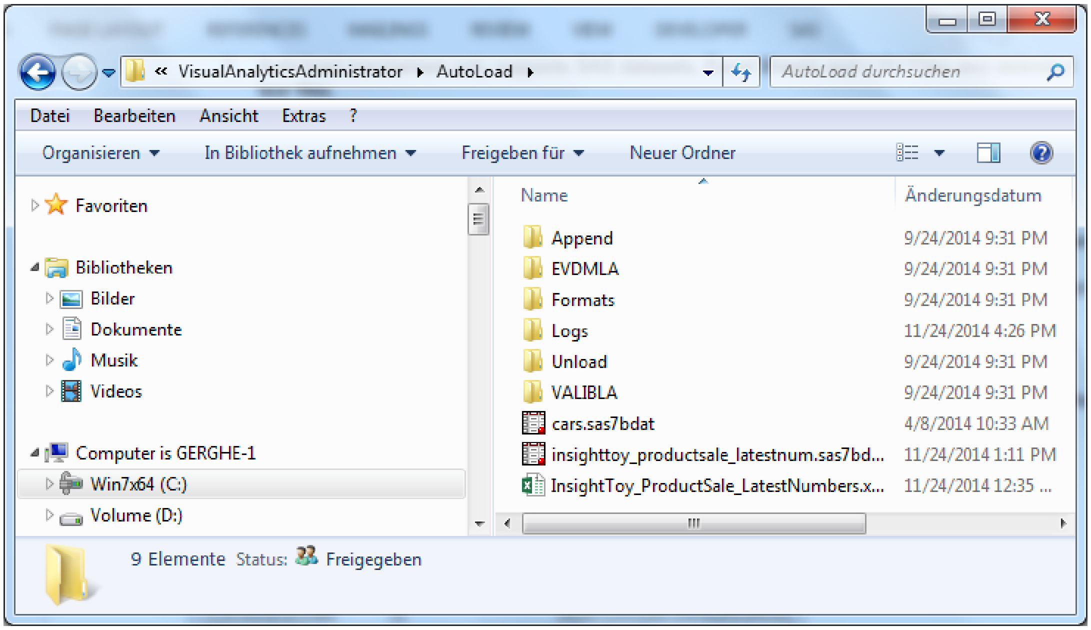
Another option of self-service data loading is implemented in SAS Visual Analytics via the auto load functionality. The principle is quite simple: An administrator defines a location where the users can drop the data files that they need to be loaded in memory for analysis. A system service checks periodically if data has been dropped in that area, and, if available, the data file gets uploaded automatically into memory.

To enable this degree of automation in the data loading process, there are some rules to follow:

1. The auto load mechanism supports only SAS data sets, Excel sheets, CSV files, and delimited text files.

2. When Excel sheets or CSV files are being loaded, variable names have to be in the first row of the document.

The auto-loading mechanism also supports appending new data to existing data tables in memory. There is a special Append folder that handles delta loading, and is configured during the auto load configuration. You can even unload unwanted data by placing it in the Unload folder. Auto load is always tied to a specific library.

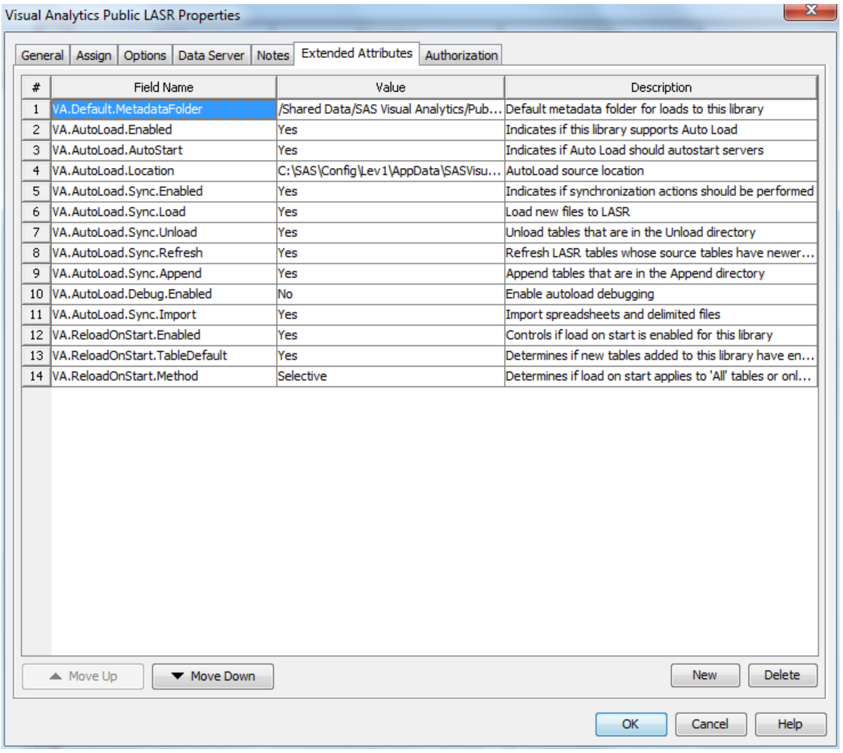


Name	Änderungsdatum
Append	9/24/2014 9:31 PM
EVDMLA	9/24/2014 9:31 PM
Formats	9/24/2014 9:31 PM
Logs	11/24/2014 4:26 PM
Unload	9/24/2014 9:31 PM
VALIBLA	9/24/2014 9:31 PM
cars.sas7bdat	4/8/2014 10:33 AM
insighttoy_productsale_latestnum.sas7bd...	11/24/2014 1:11 PM
InsightToy_ProductSale_LatestNumbers.x...	11/24/2014 12:35 ...

9 Elemente Status: Freigegeben

Figure 5. Default Structure of the AutoLoad Directory

By default, the Visual Analytics Public LASR library is enabled for auto loading. Individual behavior of the auto load mechanism can be controlled via a couple of extended attributes:



#	Field Name	Value	Description
1	VA.Default.MetadataFolder	/Shared Data/SAS Visual Analytics/Pub...	Default metadata folder for loads to this library
2	VA.AutoLoad.Enabled	Yes	Indicates if this library supports Auto Load
3	VA.AutoLoad.AutoStart	Yes	Indicates if Auto Load should autostart servers
4	VA.AutoLoad.Location	C:\SAS\Config\Lev1\AppData\SASVisu...	AutoLoad source location
5	VA.AutoLoad.Sync.Enabled	Yes	Indicates if synchronization actions should be performed
6	VA.AutoLoad.Sync.Load	Yes	Load new files to LASR
7	VA.AutoLoad.Sync.Unload	Yes	Unload tables that are in the Unload directory
8	VA.AutoLoad.Sync.Refresh	Yes	Refresh LASR tables whose source tables have newer...
9	VA.AutoLoad.Sync.Append	Yes	Append tables that are in the Append directory
10	VA.AutoLoad.Debug.Enabled	No	Enable autoload debugging
11	VA.AutoLoad.Sync.Import	Yes	Import spreadsheets and delimited files
12	VA.ReloadOnStart.Enabled	Yes	Controls if load on start is enabled for this library
13	VA.ReloadOnStart.TableDefault	Yes	Determines if new tables added to this library have en...
14	VA.ReloadOnStart.Method	Selective	Determines if load on start applies to 'All' tables or onl...

Figure 6. Extended Attributes of the Visual Analytics Public LASR Library

Auto load mechanism is useful when you want to load larger data files to SAS LASR Analytic Server and want to avoid the browser limitations of the size of data files when importing data using the user interface. If security requirements do not allow one generic auto load location for every department you can set up several auto load instances. Thus, you can also set individual thresholds for the memory consumptions of those auto load instances by assigning a separate LASR Analytic Server for each.

*In an upcoming release of SAS Visual Analytics, there will be two important enhancements to auto-load:

- An administrator can enable data compression when loading data into memory
- Text files with a delimiter will be supported

ENHANCING DATA

CALCULATE NEW DATA ITEMS

As users start using the centrally provided data sources, or even bring their own additional data source, there are business needs to build logical calculations on the fly using these data sources. In traditional BI systems with a complete IT-governed strategy, a user would have to request IT to build the new required calculations into the data source, which would take its own time based on priorities and data management schedules.

With self-service approaches supported by new data visualization solutions, users can leverage the built-in capability of defining new variables on the fly, and build their own business logic to test the outcomes in some cases as a playpen setup.

SAS Visual Analytics as a self-service application provides several capabilities to create new data items on the fly. These can be classified into four main categories:

1. **Calculated items on detail data**. Here the calculation will be done before any aggregation happens.

Example: X = Sum (A/B)

2. **Calculate aggregated measures**. In this case, data gets aggregated first before the calculation is done.

   Example: X = Sum (A) / Sum (B)

3. **Parameterized calculations** either on detail or aggregated level. Here, the resulting value of a calculation depends on the value of the parameter passed to the calculation expression.

   Example: Parameter Y = 1,2,3,4….

   X = A + B * Y

4. **Creating custom categories**. This enables users to combine the values of a variable and create new variables for data presentaion.

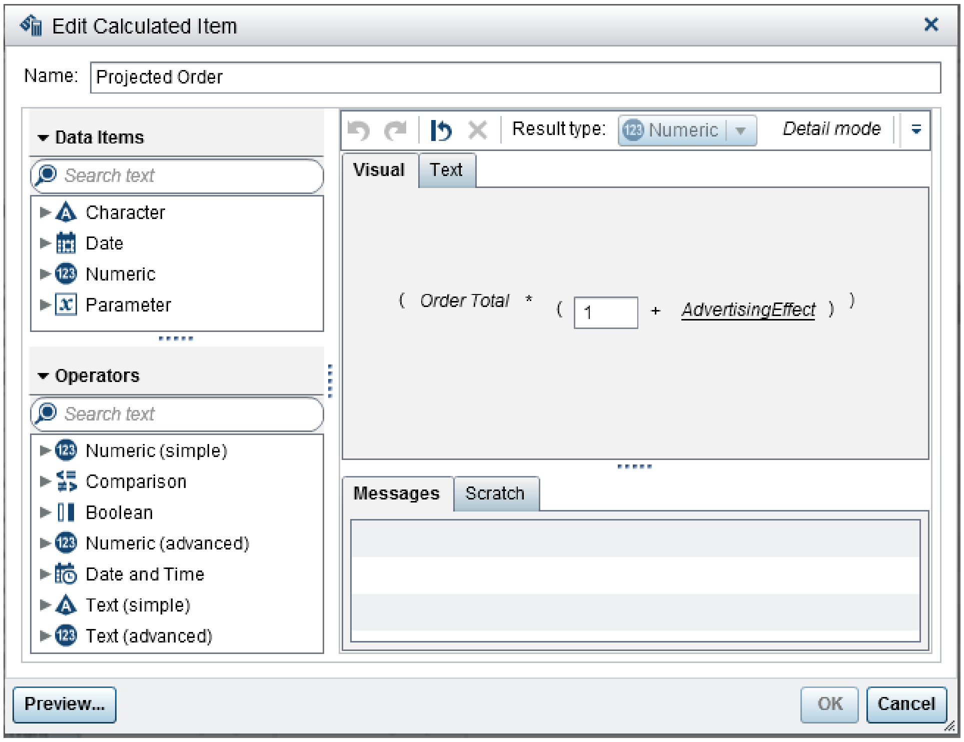
   Example: Creating a new region as "Nordics," which combines the values for regions = Norway, Denmark, Sweden, and Finland all together

Here are some important things to notice:

- The result of a calculated item on the detail level can be of any data type (character, numeric, date, time, or datetime).
- The result of an aggregated calculation is always numeric.
- Parameterized calculations can be done both on the detail or aggregated level.

The combination of the capabilities described above makes it easy for someone designing a report or analyzing data in SAS Visual Analytics to create new data items as required. Particularly of importance is the use of parameterized calculations, which not only provides an advanced way to interact with data, but also helps build scenario-driven analysis for the report users.

As shown in Figure 7 below, you can choose from a wide variety of operators ranging from simple arithmetic computations to advanced text manipulations. This capability makes SAS Visual Analytics attractive for users with different skill levels to create new calculations of varying complex business logics to be created using simple interactive drag-and-drop actions and without the need of special programming skills.



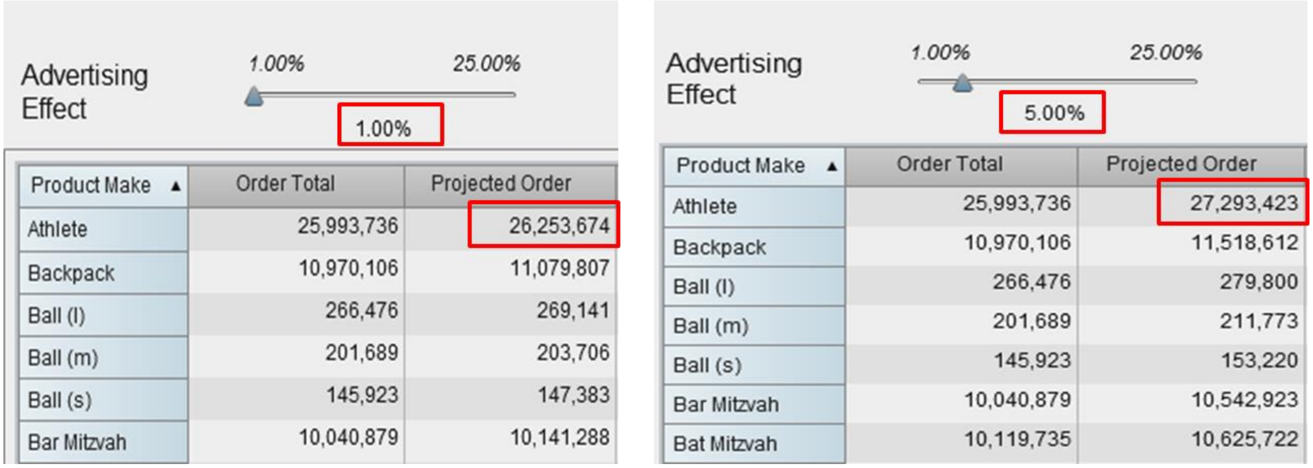
Name: Projected Order

Result type: Numeric

(Order Total * (1 + AdvertisingEffect))

Figure 7. New Aggregated Measure Using Parameter for Dynamic Computation

Figure 8 shows how the parameter-driven calculation “Projected Order” (as calculated in Figure 7) is used in a report, enabling the user to run a what-if analysis using the varying values of “Advertising Effect.”



Advertising Effect: 1.00%

Product Make ▲	Order Total	Projected Order
Athlete	25,993,736	26,253,674
Backpack	10,970,106	11,079,807
Ball (l)	266,476	269,141
Ball (m)	201,689	203,706
Ball (s)	145,923	147,383
Bar Mitzvah	10,040,879	10,141,288

Advertising Effect: 5.00%

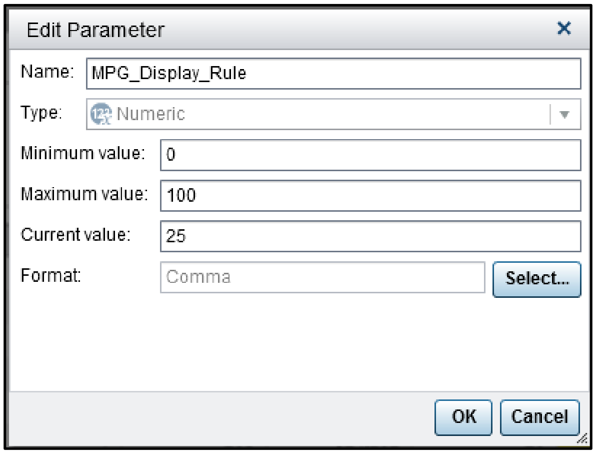
Product Make ▲	Order Total	Projected Order
Athlete	25,993,736	27,293,423
Backpack	10,970,106	11,518,612
Ball (l)	266,476	279,800
Ball (m)	201,689	211,773
Ball (s)	145,923	153,220
Bar Mitzvah	10,040,879	10,542,923
Bat Mitzvah	10,119,735	10,625,722

Figure 8. Using a New Calculated Measure in a What-If Style Scenario Controlled via a Slider

USING PARAMETERS TO MAKE REPORTS DYNAMIC

In SAS Visual Analytics 7.1, parameters were introduced. You can use parameters in calculations, display rules, filters, and ranks in SAS Visual Analytics Designer. Parameters enable a report designer to fulfill complex report design requirements that can give the report consumer a more dynamic reporting experience. Before going into detail on how parameters can be used in a report, let us first look at a few basic facts about parameters:

- Parameter types can be numeric or character. Date type parameters are not yet supported.
- Numeric parameters require a current value, a maximum value, and a minimum value during creation. Specifying a format is optional.
- Character parameters do not require a value when created.
- Parameters can be edited and deleted from the data pane.
- You cannot delete a parameter that is being used in any object.



Edit Parameter

Name: MPG_Display_Rule

Type: Numeric

Minimum value: 0

Maximum value: 100

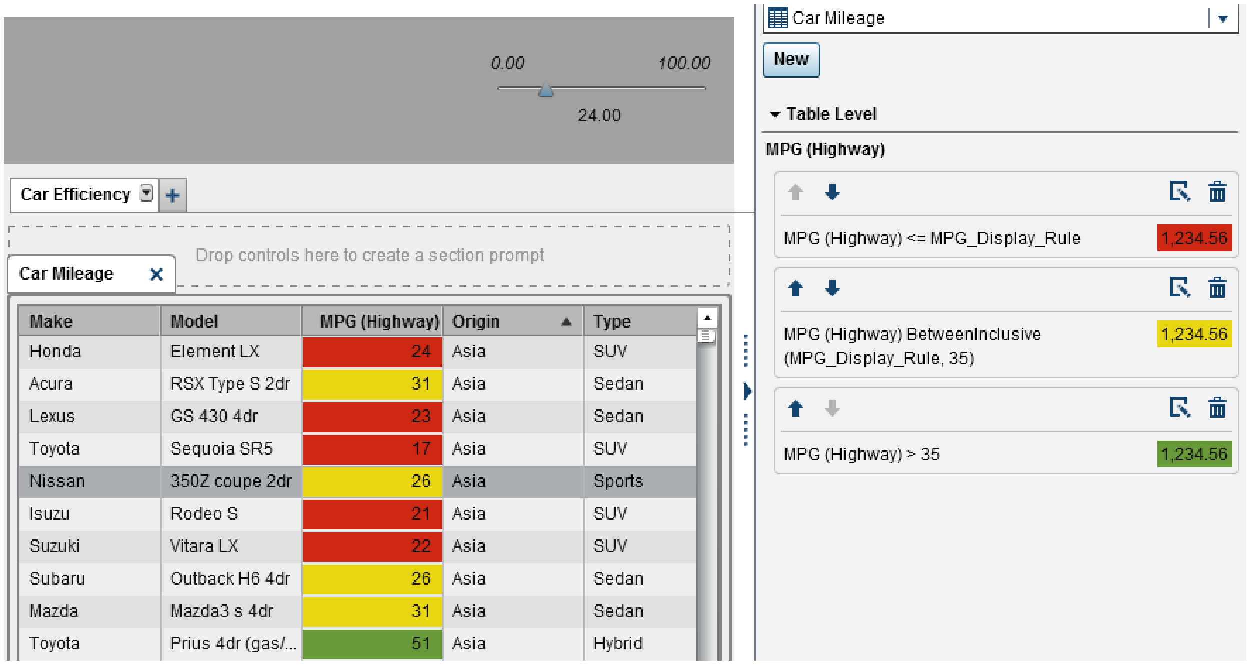
Current value: 25

Format: Comma

Figure 9. Edit Parameter Window Showing a Numeric Parameter

Listed below are some possible scenarios where the use of parameters can be beneficial:

1. Parameters can be used to dynamically influence the result of a calculation as already shown in Figure 8. This is of great value for an information consumer as he or she can play around with these values and instantly see the effect in a graph or tabular object.
2. Display rules usually have fixed thresholds. When you use parameters within the definition of a display rule, you can now modify the coloring of values dynamically as shown in Figure 10. Another possible scenario is when you display a sum in a crosstab that contains a hierarchy. Different hierarchy levels require different thresholds for coloring the values and can be easily achieved using rules based on parameter-driven variables.
3. Adding flexibility for advanced filtering: Filtering high cardinality categories often requires wildcards, because the report consumer might not know the exact value of a data item or declines to enter the complete value when entering only a part of it can deliver the same filter result. Use of parameters allow for creation on more complex filter conditions.
4. Cascading prompts for section-level or report-level filters: Cascading prompts can be generated in SAS Visual Analytics Designer out-of-the-box, if the filter objects are part of the report body and when using interactive links between filter controls. With parameter-driven values in filter controls, a similar cascading behavior can be achieved when using filter controls for section- and report-level prompts.
5. Ranking: With the help of parameters, we can allow users to enter the value for Top N/Bottom N dynamically. If you have a visual showing Top 3 (as a graph or list table), it might be interesting from a business perspective to see who the next best competitor is.



Make	Model	MPG (Highway)	Origin	Type
Honda	Element LX	24	Asia	SUV
Acura	RSX Type S 2dr	31	Asia	Sedan
Lexus	GS 430 4dr	23	Asia	Sedan
Toyota	Sequoia SR5	17	Asia	SUV
Nissan	350Z coupe 2dr	26	Asia	Sports
Isuzu	Rodeo S	21	Asia	SUV
Suzuki	Vitara LX	22	Asia	SUV
Subaru	Outback H6 4dr	26	Asia	Sedan
Mazda	Mazda3 s 4dr	31	Asia	Sedan
Toyota	Prius 4dr (gas/...	51	Asia	Hybrid

MPG (Highway)

MPG (Highway) <= MPG_Display_Rule 1,234.56

MPG (Highway) BetweenInclusive (MPG_Display_Rule, 35) 1,234.56

MPG (Highway) > 35 1,234.56

Figure 10. Using a Parameter to Modify Display Rules via Slider

These are only a few suggestions on how parameters can be used to meet complex report design requirements. I am sure there are a lot more. If you are encountering a problem during design of your report, check if using parameters might solve your problem.

DATA PREPARATION IN SAS® VISUAL DATA BUILDER

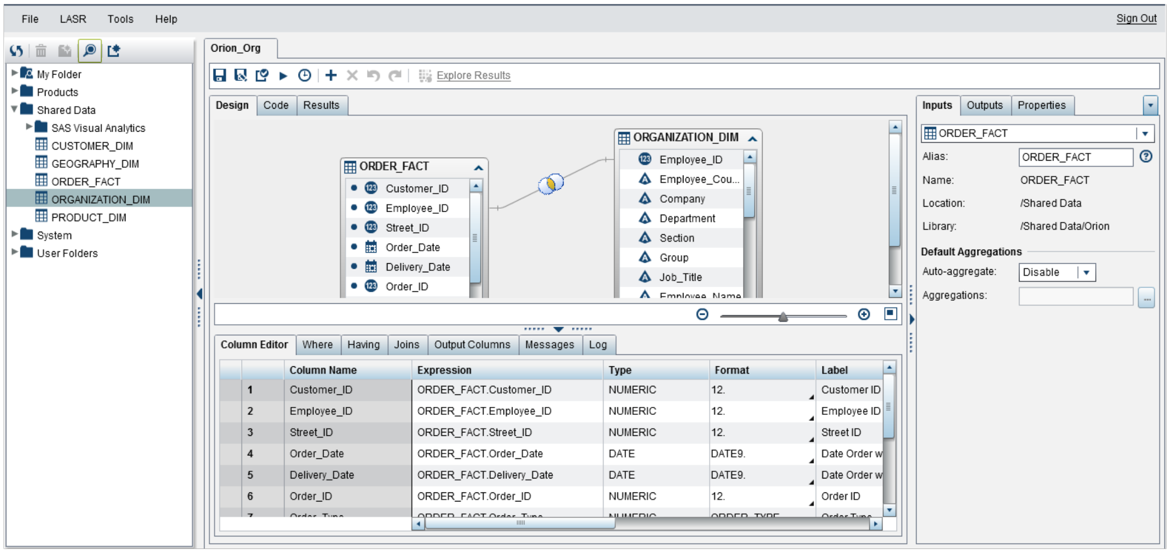
So far, we have only considered loading data into SAS Visual Analytics and enhancing it from within the interface. But there might be cases where an advanced business user with appropriate skills would want to join several data sources before loading into memory. The interface that supports these types of activities in SAS Visual Analytics is called SAS Visual Data Builder. It has been designed to fulfill basic data preparation requirements to bring the data into a shape that is effective in analytical data discovery and reporting. Behind the scenes, SAS Visual Data Builder generates SAS® code that can be executed in a separate SAS session.

In SAS Visual Data Builder, you have three main choices:

1. **Upload data as is**: This is identical to what has been described in the self-service data loading section.

2. **Create a data query**: This allows you to join several files and compute additional columns if necessary. The resulting data set can be stored permanently to a database or loaded to SAS LASR Analytic Server in memory directly for instant access. As part of query builder, users can create new columns by using the advanced expression builder user interface that comes with a library of pre-defined functions and allows users to add or update business labels and formats for their database columns,that are useful down the line for users. (See Figure 9.)

3. **Create a Star Schema**: Those working with traditional BI systems would be familiar with Star Schema—which is essentially a view of data coming from a central fact table that is associated with multiple dimension tables. Using a star-schema approach can be beneficial in cases where you want to upload only a specific view of your entire data into memory. This approach can help save some memory space in comparison to loading the complete data tables into memory.



	Column Name	Expression	Type	Format	Label
1	Customer_ID	ORDER_FACT.Customer_ID	NUMERIC	12.	Customer ID
2	Employee_ID	ORDER_FACT.Employee_ID	NUMERIC	12.	Employee ID
3	Street_ID	ORDER_FACT.Street_ID	NUMERIC	12.	Street ID
4	Order_Date	ORDER_FACT.Order_Date	DATE	DATE9.	Date Order w
5	Delivery_Date	ORDER_FACT.Delivery_Date	DATE	DATE9.	Date Order w
6	Order_ID	ORDER_FACT.Order_ID	NUMERIC	12.	Order ID
7	Order_Type	ORDER_FACT.Order_Type	NUMERIC	ORDER_TYPE	Order Type

ORDER_FACT

Alias: ORDER_FACT

Name: ORDER_FACT

Location: /Shared Data

Library: /Shared Data/Orion

Default Aggregations

Auto-aggregate: Disable

Aggregations:

Figure 11. Creating a Query in SAS Visual Data Builder

When modeling your data in the form of a Star Schema, certain aspects need to be followed:

1. Only one fact table is allowed.
2. One variable in the dimension table defines the relation to one variable in the fact table.
3. Dimension keys must be unique.

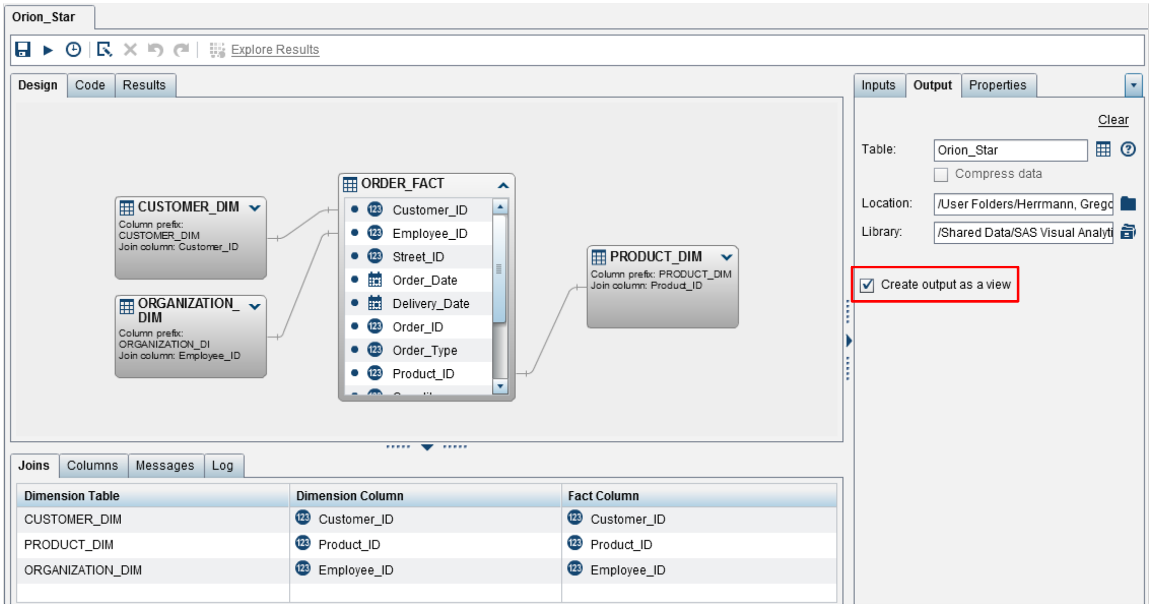


Table: Orion_Star

☐ Compress data

Location: /User Folders/Herrmann, Gregc

Library: /Shared Data/SAS Visual Analyti

☑ Create output as a view

Dimension Table	Dimension Column	Fact Column
CUSTOMER_DIM	Customer_ID	Customer_ID
PRODUCT_DIM	Product_ID	Product_ID
ORGANIZATION_DIM	Employee_ID	Employee_ID

Figure 12. Star Schema with One Fact Table and Three Dimension Tables

The default output format is a view. (See the checkbox ticked in Figure 12.) When you create a Star Schema with output format view, the fact and dimension tables will be loaded to the LASR Analytic Server. However, the result table will not be materialized immediately. The logical join will only be executed when the view is being used by a report or exploration, thus saving valuable memory space for other tables or actions.

PERFORMANCE CONSIDERATIONS

Data preparation and its management are critical activities for a successful running of any enterprise-class business intelligence system. To make the right choice and provide an optimal performing environment, some key factors must be taken into consideration. Some of these are more important from the system architecture perspective as well. Here are a few cases for your consideration:

1. **All tables being used are SAS data sets.** In this case, be aware that data preparation steps will be SAS programs running on the workspace server of the SAS Visual Analytics environment. This limits both computing and storage resources that are available to the environment and thus calls for appropriate sizing of the SAS Visual Analytics environment. Be aware that when you run joins on that environment, additional temporary space will be needed to complete these steps successfully. It is also required that before running the code, all necessary SAS data sets should be copied to the environment to avoid data movement across networks. This adversely affects performance.

2. **All tables are stored in one relational database.** In this case, the generated SQL code will be passed to the underlying database for processing (implicit pass-through) when the resulting table will also be stored in the database itself. An additional step is then required to load the resulting data table from the database to SAS LASR Analytic Server. If the database supports parallel load with the SAS Embedded Process, loading of the result table to SAS LASR Analytic Server (in memory) will be reasonably faster.

3. **A mixture of SAS data sets and tables in a relational database.** This is the most complex case. If possible, joins should be run on the database tables. This leverages the power of the database itself. The resulting data table can then be copied to the SAS workspace server to run remaining data preparation steps using SAS.

4. **All tables are stored in Hadoop.** In this case, the SAS/ACCESS Interface to Hadoop can be used to pass the processing to the Hadoop distributed environment. If you are running an asymmetric architecture (SAS LASR Analytic Server and Hadoop running on separate nodes) and the Hadoop version used supports the SAS Embedded Process, parallel loading can be used to make tables available in LASR. In this case, it is recommended to run data preparation completely in Hadoop especially if data volumes are big.

CONCLUSION

With the current shift from the use of centralized enterprise data warehouses, exclusively owned and operated by IT and supporting traditional static reporting to data discovery tools that provide self-service data management capabilities to the business, it is obvious that data preparation capabilities will be introduced on a broader scale across the userbase in maturing organizations. SAS Visual Analytics today already supports this shift and will extend these capabilities in future releases.

RECOMMENDED READING

Atassi, Sam, and Malcom Alexander. 2013. “Self-Service Data Management: SAS 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.

Mehler, Gary, and Donna Bennett. 2014. “Big Data Everywhere! Easily Loading and Managing Your Data in the SAS® LASR™ Analytic Server.” *Proceedings of the SAS Global Forum 2014 Conference.* Cary, NC. SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings14/SAS347-2014.pdf.

Rivers, Kerri, and Christopher Redpath. 2015. “Tips and Techniques for Efficiently Updating and Loading Data into SAS® Visual Analytics.” *Proceedings of the SAS Global Forum 2015 Conference.* Cary, NC. SAS Institute Inc.

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 at:

Gregor Herrmann
SAS Institute Inc.
SAS Campus Drive
Cary, NC, 27513
gregor.herrmann@sas.com

Anand Chitale
SAS Institute Inc.
SAS Campus Drive
Cary, NC, 27513
anand.chitale@sas.com