

JMP® and SAS®: One Completes The Other

Philip Brown, Predictum Inc, Potomac, MD
Wayne Levin, Predictum Inc, Toronto, ON

ABSTRACT

Integrating JMP® with SAS® creates a formidable data management and analytical system. JMP® takes the role of the client, providing interactivity and visualization and, as a result, makes advanced statistical methods accessible to analysts of all sorts. The limitation of JMP® is that it requires data to be in memory. That limits the size of the data available for analysis and also requires the transfer of data from its sources into RAM; a slow process for large data sets. SAS® on the server can manage and quickly manipulate data on a large scale and integrate JMP® with other proprietary databases. This paper will review options in extending the added value of JMP® by extending its capabilities with SAS® and discuss the optimal distribution of labor between JMP® and SAS®.

INTRODUCTION

JMP® has grown to be the analysis software of choice for scientists and engineers not only for its wide array of statistical functions, but also because it's easy to setup and allows users to get to the point of exploring, analyzing and visualizing their data very quickly. With JMP®'s scripting language (JSL), one can extend JMP®'s functionality by creating interactive applications that automate analytical tasks. In the roughly 14 years that JSL has been in existence, we have seen many innovative scripts and more recently, JMP® Add-ins, produced with various levels of complexity. Within large corporations, there have been numerous success stories of JMP® and JSL script usage as is evident from material presented at JMP® Discovery Summits over the years[1]. However, as both the analysis needs and the number of scripts developed increase, it can become difficult to manage and deploy JSL scripts. It is not uncommon for JSL applications to have thousands of lines of code spread across many files. Also, as the user base increases, the need to administer user access and permissions becomes increasingly important. Usually, IT department involvement is required to create group policies and network folders for the distributed analytics environment. IT and engineering staff often have differing opinions on the best way to do things, and cycle time to get a particular task done can be lengthy.

SAS®, on the other hand, is built with the enterprise in mind. Not only does it have an extensive library of statistical procedures of its own, but also can provide a self-contained environment for administering users and managing data, thereby reducing the dependence on IT department resources.

We will explore three key areas: User Administration, Analytics Management, Data Management, where difficulties often arise when attempting to expand a JMP®-centric environment to the enterprise and how SAS® can play a major role in mitigating some of these challenges.

I. User Administration

In environments where both the user base and the number of JMP® scripts are growing, there may be a need to separate users into groups with different levels of access. For example, we could imagine three groups in a typical organization: Admin, Engineering, Production. The Admin group would have all access with permission to modify analysis scripts and configuration files. Analysts would not see resources related to system configuration, but may only have permission to run the analyses while the Production group may only have permission to view reports created by the Analysts. JMP® doesn't offer such facility natively, although one can create hybrid solutions by integrating other technologies such as .NET/ Java. However, this requires knowledge of these languages and often means working collaboratively with the IT department.

SAS® provides a layer of administration that helps to reduce the need for IT involvement. The layer is managed through SAS® Metadata Server. The Metadata Server essentially establishes a "shell" that interfaces with the IT

infrastructure on the outside, while providing full user rights/access control on all activity within the “shell.” This means that analysts now have the ability to control their environment as they see fit. Users and groups can be created at will, for exclusive access by SAS® clients such as JMP® and SAS® Enterprise Guide, among others.

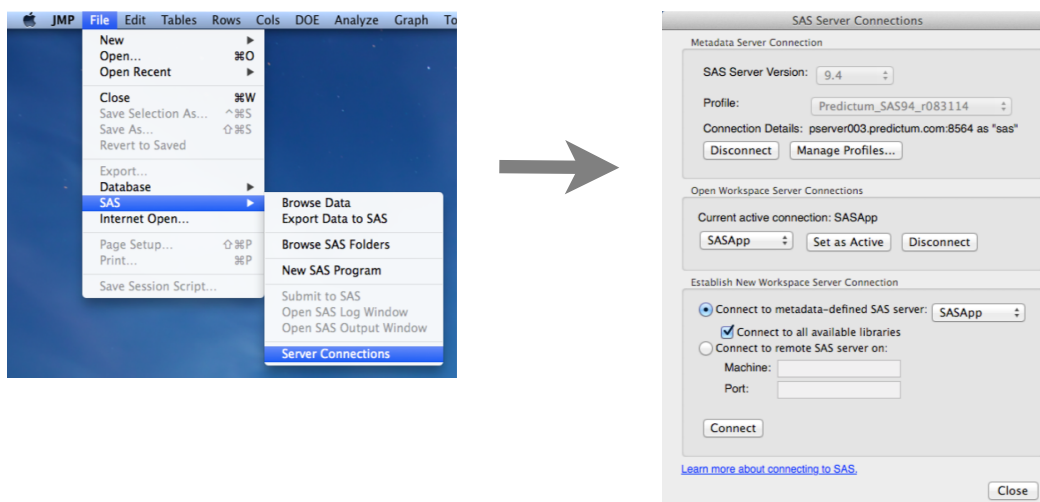


Figure 1: Making a SAS® Server connection from within JMP®

In the case of JMP®, once a user is authenticated by the Metadata Server, the JMP® menu options to **Browse Data** and **Browse SAS Folders**, become available. Depending on the permission settings, one can control what objects are actually seen in each of these. In this way, again analysts have control of their environment.

Browse Data presents an interface to browse SAS® data libraries on the server, The SAS® data library is a logical container defined in the Metadata Server, through which SAS® data sets can be accessed. With the interface, the user can look at table attributes and decide which tables to download to JMP®.

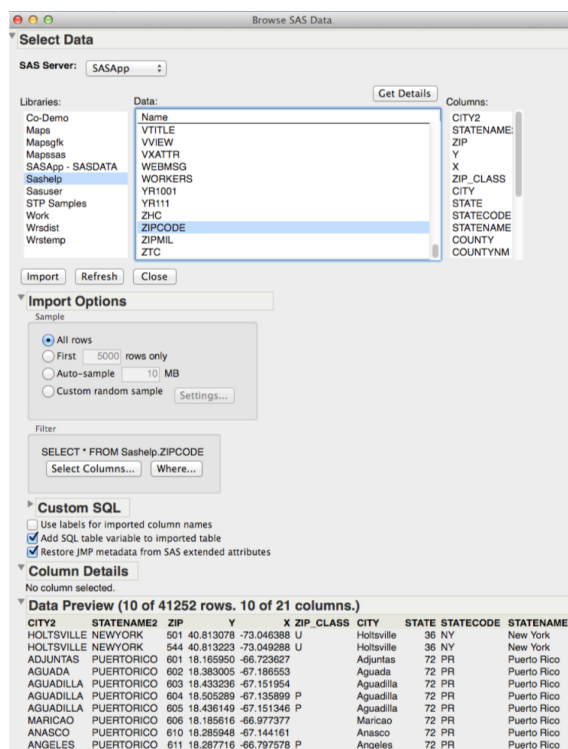


Figure 2: JMP® Interface for browsing SAS® Data Libraries

This provides a simple, seamless way of accessing centralized data. Since the structure is defined in the Metadata Server, fine-grained permissions all the way down to the table column level can be set.

Browse SAS Folders provides access to SAS' internal network folders. This too is a Metadata Server construct and is fully configurable. Contents of these folders can range from individual SAS® code files and data sets, to reports generated by SAS® WebReport Studio. In SAS® clients such as Enterprise Guide, the user's home folder on the server is accessible as are, individual code and data files. In JMP®, however, only SAS stored processes and datasets appear to be accessible.

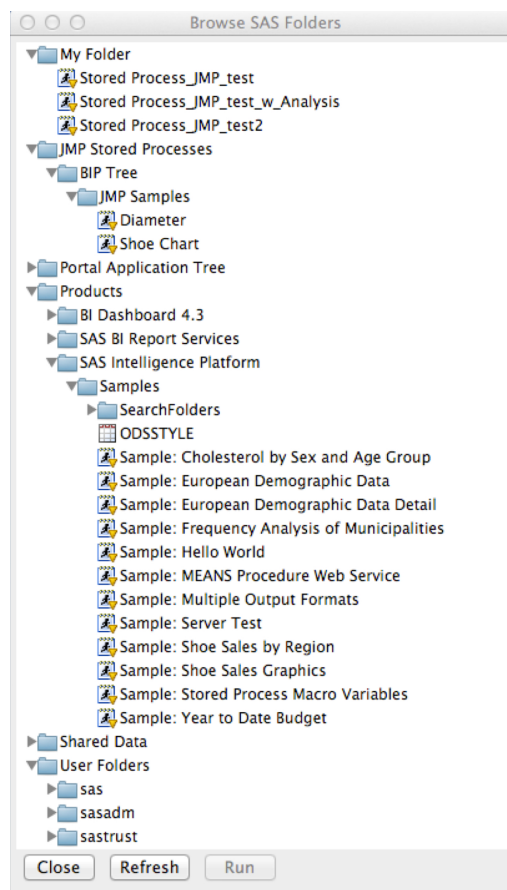


Figure 3: Browse SAS Folders (Stored Processes)

II. Analytics Management

In the case where there are numerous JSL scripts and JMP® Add-ins to manage, there is typically difficulty in efficiently deploying and maintaining them. Even though it is possible to create JMP® Add-ins and JMP® Applications, automating the deployment of updates and installation of new Add-ins is often desired. Hybrid solutions have been pursued where shared drives, network folders and web portals are setup by the IT department to allow users to access latest versions of scripts from a central location. Add-in Managers have been discussed and prototypes are in circulation[2]. However, SAS® is well suited to be the hub for both SAS® code and JSL scripts, through use of the SAS® Stored Process.

The SAS® Stored Process is arguably the most important SAS® object that is available to JMP®. A stored process is essentially a collection of source code, prompts and other instructions that have been bundled together into a single package that can be invoked by a number of SAS® client applications. Often stored processes are created such that they can be called with arguments, allowing for function parameters to be arbitrarily set at runtime. Output of a stored process can be of two forms, either a Stream or a Package. Stream output is built for use with SAS® Web

Infrastructure and is not available to JMP®. The Package form can be output in formats such as HTML, PDF, RTF, etc that is handled by SAS® Output Delivery Systems (ODS) system. JMP® provides an additional option for ODS output: "JMP® Report" which can be set either via JSL or from within JMP® Preferences.

In addition to the standard stored process, there is an additional, less explored option that appears to be only available in SAS® Enterprise Guide. Users have the capability to create what is called a JMP® Stored Process Package. This is essentially a standard stored process that is specially configured to return and run a user defined JSL script when the stored process is called from JMP®. This makes SAS® Enterprise Guide a vital companion to JMP®.

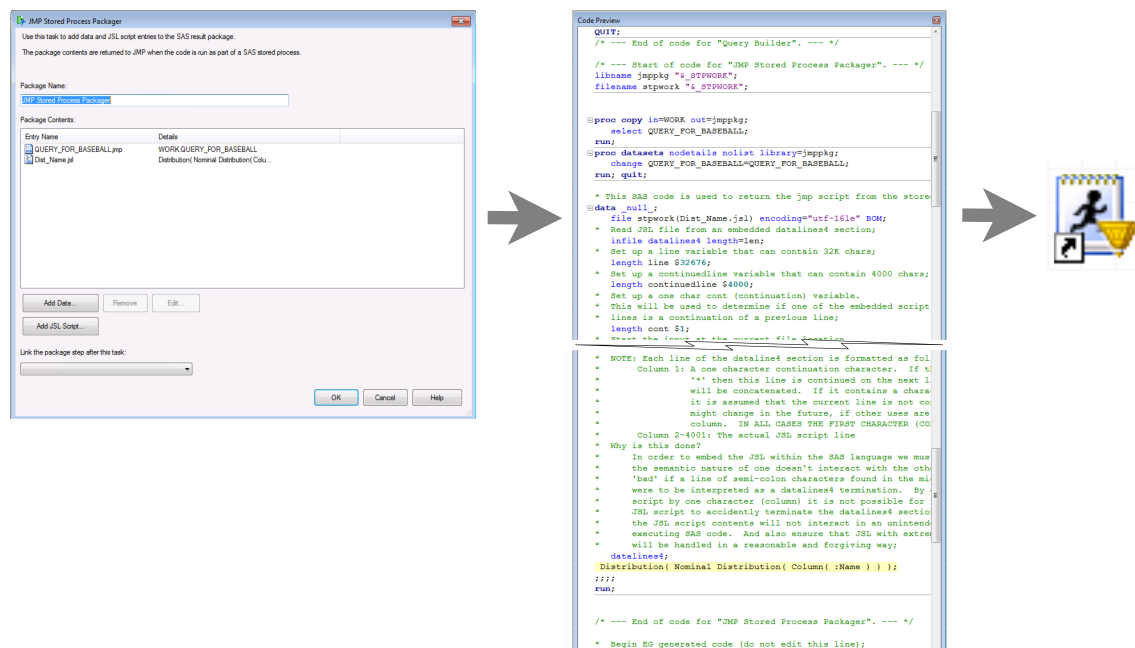


Figure 4: Creating a JMP® Stored Process in SAS® Enterprise Guide

This presents tremendous possibilities for integrating SAS® code and JSL, one supplementing the other depending on the particular need. Tasks that are typically time consuming and cumbersome to perform in JMP®, for example, re typically time consuming and cumbersome to do in JMP® such as summarization and complex merges of large data sets, can be coded in SAS® with a higher degree of efficiency. A stored process can then be created and bundled together with a JSL script that carries out some predefined set of analyses.

Another important feature that SAS® brings, is the ability to schedule code and stored processes as jobs. These jobs can not only be set to run at a specified time, but also they can also be sequenced with conditional blocks, in SAS®' Job Editor.

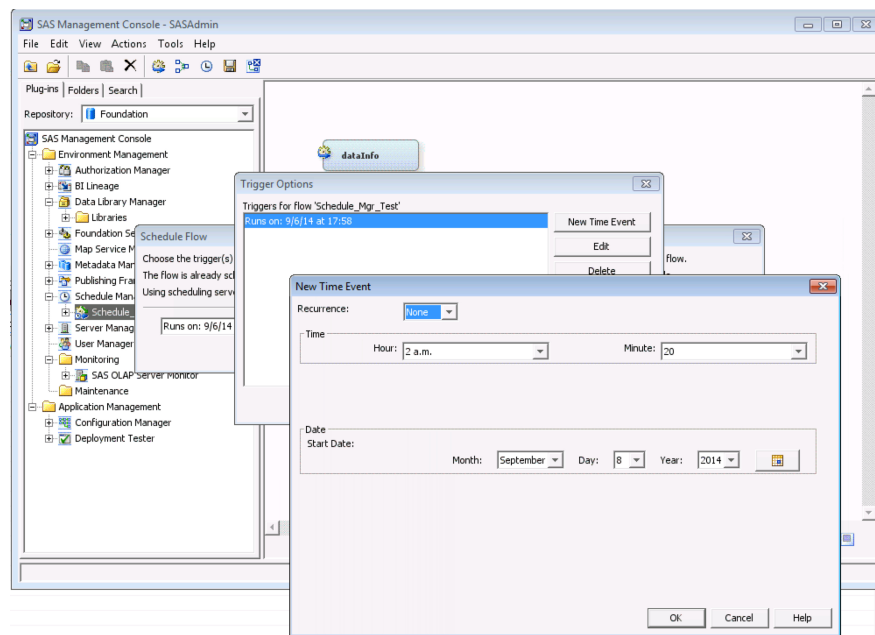


Figure 5: Schedule Manager plug-in - SAS® Management Console

Of course, execution of SAS® code directly from JMP® is possible as well. Included with JMP® is a SAS® Program Editor which gives the user a similar experience to using Base SAS®. The JSL language also includes a library of SAS® specific functions, which does allow a certain level of hybridization of the two languages. JMP®'s Fit Model and Neural Network platforms both offer the capability of exporting code as a SAS® DATA Step. It could be that one uses JMP® to prototype and refine models, exports them to SAS® code, and then safely tucks them in a SAS® Stored Process. The end user sees only the stored process object, and is unable to view or change the underlying code. Since stored processes are defined in the Metadata Server, access can be controlled through the same methods described in Section I and interesting options such as audit trail and change control management can be configured as well.

III. Data Management

Getting the data in the desired format is a common problem in most analysis communities. In some cases, access to several different data sources is required in order to pull together what the analyst needs. These can include databases, instrumentation/equipment, individual Excel and CSV files. Usually IT department involvement is necessary to get database access and to perform the tasks for creating the required views and tables. For the loose Excel and CSV files, shared folders are sometimes created. Having to maintain these various resources can be difficult to manage. There are many cases where IT and analysts actually work well enough together at building hybrid applications that allow a large number of users to efficiently access the data they need. However, there are many more cases where it takes multiple iterations over some extended period of time to get it right.

SAS® offers specialized data connectors called ACCESS Engines. These allow for seamless access to over 60 different data sources, including MS SQL Server, Oracle, and Hadoop, to name a few. To the end user, the database tables appear as regular SAS® data sets. These datasets can be organized into logical containers called Data Libraries. As previously mentioned, JMP® provides a means to connect to and browse SAS® Data Libraries, extending to the JMP® user this seamless access to multiple data sources. Since the libraries are administered by the Metadata Server, the same security features previously discussed are also available.

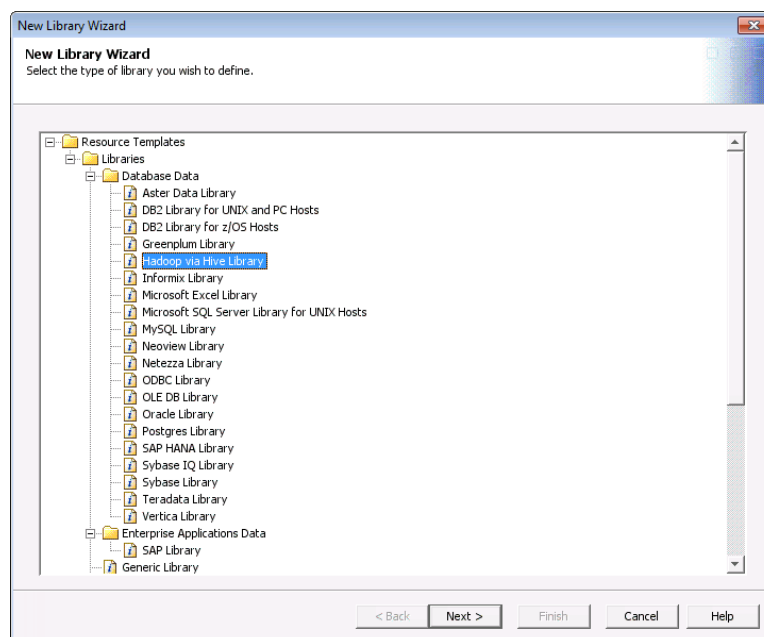


Figure 6: Data library types - SAS® Management Console

Conclusion

In large organizations where there is mature JMP® usage, the need for enterprise level integration to support user administration, analytics and data management usually leads to heavy dependence on resources of IT departments or contractors to develop hybrid solutions. It is proposed that many of these needs can be filled by integrating JMP® with SAS®. SAS® is well suited for handling complex data integration and automated analysis tasks on enormous datasets. This coupled with JMP®'s strength in data visualization opens up many possibilities for hybrid JMP® Applications. The security features of the SAS® Metadata Server, allow JSL programmers to create applications with user authentication and data access control. The JMP® Stored Process Packager allows for rapid deployment of verified analysis procedures, and allows for centralized administration of these packages. The ability to blend SAS® code and JSL in these stored processes gives programmer more options to work with. Finally, with SAS® Data libraries, data can be managed from a central location and yet readily available in JMP®. With the SAS®/ACCESS Engines, data connectors are available for almost every major database, expanding the range of data sources accessible directly from JMP®.

REFERENCES

- [1] Burkhardt, C., Brown D. P. (2010). A Virtual Laboratory System for Tire Compounding. *JMP Discovery Summit 2010*.
- [2] Haslam M., Erb J., Brown D. P., Levin J. W. (2013), Best Practices in JMP® Add-in Management. *JMP® White Paper*. <http://j.mp/1jhZdoE>
- [3] Predictum (2013), Master Add-in Manager© http://predictum.com/solutions/jmp_addins_jmp_scripting/master-add-in-manager/
- [4] Weibe P. (2012), JMP® Add-in Manager, *JMP® File Exchange*. http://support.sas.com/demosdownloads/downarea_t4.jsp?productID=111561&jmpflag=N

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