

JMP® Data Server for Multi-User Organization with Diverse Data Sources

Neal Verzwyvelt

Atmel Corporation, Colorado Springs, Colorado 80906



Introduction

JMP® has proven its ability to add value to an organization through powerful data analysis capabilities. A problem faced by many organizations is that of getting data into JMP® for analysis. Even with a rich set of import utilities, users often do not know where to find a given data set or which import method is best suited to the particular data source. In addition, the native ODBC (Open Database Connectivity) interface in JMP®, if used to access databases, requires that each potential user's PC have the necessary ODBC configuration setup and maintained, potentially creating a not insignificant workload for an organization's User Services department. This had been the case at Atmel Corporation, where data required for maximizing engineer productivity was spread across Several RDBMS Systems (Oracle®, PostgreSQL, SQL Server®) as well as flat files. By using JMP® JSL, the Apache Tomcat Servlet Container, and Java programming language, a simple system of queries has been deployed that allow for flexible and fast data import into JMP®.

Methods

JMP® Scripting Language (JSL)

JSL is used to extend and/or customize JMP® usage¹. The key JSL function in this poster is the Open function. This function is normally used to open a JMP® data table, CSV file, or other existing data set². Open can also be used, however, to open a website URL, and this is key to the JMP® Data Server Concept. Using the Tomcat Servlet Container and Java servlets, queries are built as URL strings and data is returned in CSV format to JMP®.

Tomcat Servlet Container

From the website <http://tomcat.apache.org>, “Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed under the Java Community Process”. Tomcat is free and compatible with Microsoft Windows® and Linux Distributions. For use as a JMP® Data Server, configuration is straightforward and can typically be used “out-of-the-box”. Need for security can increase configuration complexity, but the Data Server described here is for read-only access to data sources.

Java

The well-known Java programming language is now owned by Oracle® Corporation. It is a workhorse tool across many different industries and applications. For the purpose of the JMP® Data Server, Java's role is in the form of Servlets, small programs that take input in the form of a URL, like a web page, and return some type of text. Often, Servlets will return HTML or XML to be rendered by a Browser program, such as Internet Explorer or Firefox. The servlets referred to in this poster will return delimited text data (i.e. CSV) or JSI code.

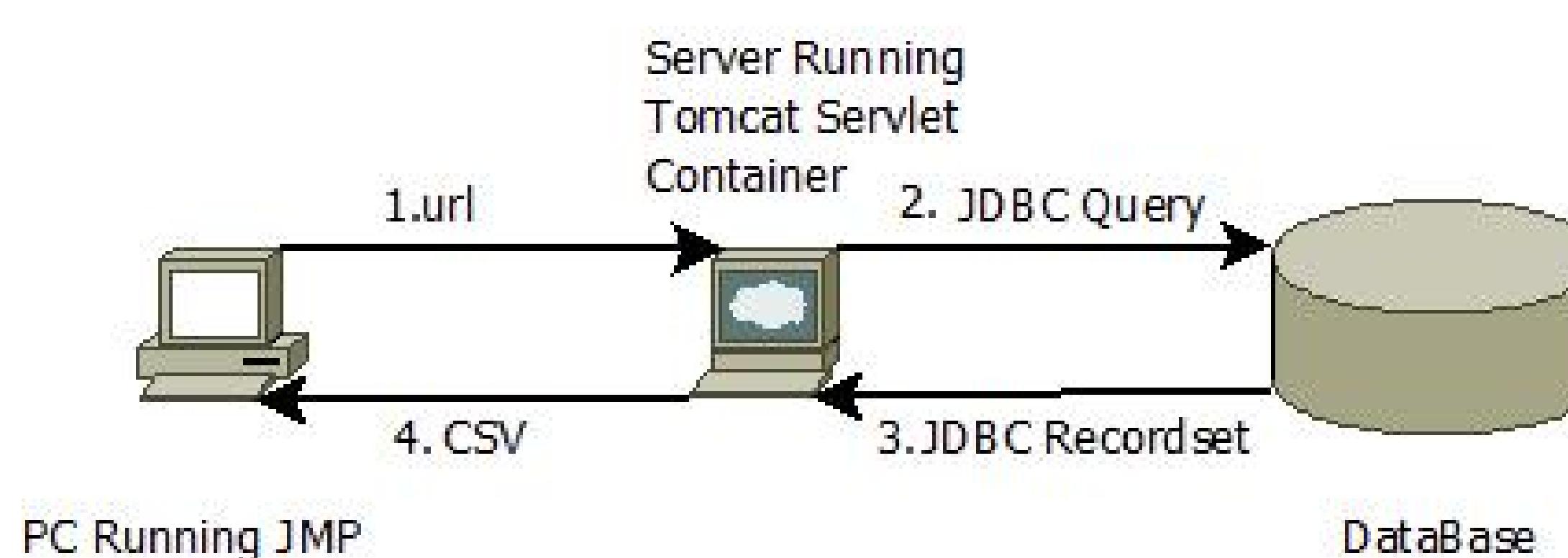


Figure 1. Typical flow of Data Server Usage

Results

Data queries are accessed by users via a dashboard modeled from the example JMPStarter.jsl supplied with the JMP® install. Queries are grouped into logical arrangements. Once a user installs the menu selection, all queries are available as well as any future additions. There is no need for a user to ever update their install as the main driver script resides on a server.

Data Queries Dashboards CSO Fndry Prd Eng Custom Qrys

These Queries are run against the ACE Database.

ACEQueries	Etested Lots	Select Etested Lots	Help
EDE Queries	Etest Lot Data	Get Etest Lot Data For Lot List	Help
HistoryQueries	Etested Wafer Data	Get Etest Wafer Data For Lot List	Help
SPACE	Sorted Lots	Select Sorted Lots	Help Notes
General Tools	Sort Lot Data	Get Lot Yield Data For Lot List	Help
Special Queries	Sort Wafer Data	Get Lot Yield Data For Wafer List	Help

"[http://jetjaguar.corp.atmel.com:8080/jmpDataServer/SortLotList?
strSDate=07/14/2011&strEDate=07/22/2011&strTestType=CP1&strProduct=82002165-001,AT19104-00K,AT19645-002](http://jetjaguar.corp.atmel.com:8080/jmpDataServer/SortLotList?strSDate=07/14/2011&strEDate=07/22/2011&strTestType=CP1&strProduct=82002165-001,AT19104-00K,AT19645-002)"

Select Start and End Date

Start Date End Date

Select By Product
 Select By Test Program

Select

354V0C12W	Add >>	82002165-001
354V3Q11W		AT19104-00K
354V4E11W		AT19645-002
354V5G11W		
354V8H11W		
354VAN11W		
355G1C11W		
359F1E15W		
82000550-003		
82000751-009		
82002165-001		
AT19104-00K		
AT19645-002		
AT19651-009		
AT19654-006		
AT19711-003		
AT19733-003		
AT19762-002		
AT19765-005		
AT19772-004		

[Enter Name for new table and run query](#)

[Go](#)

```
graph TD; A[Sorted Lots] --> B[Select Product]; B --> C["http://jetjaguar.corp.atmel.com:8080/jmpDataServer/SortLotList?strSDate=07/14/2011&strEDate=07/22/2011&strTestType=CP1&strProduct=82002165-001,AT19104-00K,AT19645-002"]
```

After a query is selected, user is presented with a simple form to collect the query parameters. These parameters are then placed into a URL via JSL.

Discussion

- This project has resulted in highly efficient data queries at Atmel Corporation. Users do not need to know the underlying database schemas, nor is any custom configuration necessary on the client computers other than install of a single JMP® menu item.
 - Data queries can be made from any RDBMS system that is supported by JDBC³. Java, however, is not required. A similar data server can be developed using Python, Perl, PHP, or any language that supports CGI.
 - New queries or applications can easily be added to the main Dashboard and are immediately available to users by simply re-opening the dashboard.
 - The set of queries and applications has been enhanced via the use of Java servlets and Apache Velocity templates. Items for a given JMP® dashboard can be stored in on the server and the template determines presentation. This makes adding new functionality less prone to errors.
 - Java routines can be implemented to reshape or pre-process data prior to sending to JMP®.

Reference

1. See especially SAS Institute Inc. 2009. *JMP® 8 Scripting Guide, Second Edition*. Cary, NC: SAS Institute Inc. Pages 11 - 27
 2. ibid. page 522
 3. System has been used at Atmel with Microsoft Sql Server® (2000 and 2005), Oracle® (8, 10, 11), and PostgreSQL 8.0.

Acknowledgement

The Author would like to thank Atmel Corporation and Joseph Conelman for the support of this project