SAS/IntrNet Software™: JConnect and JTunnel Dances with SAS/CONNECT® Software

Faith Reneé Sloan
FRS Associates,LLC San Francisco, California

ABSTRACT

The new JConnect Java classes give you the capability of accessing data via Structured Query Language (SQL). You can use either socket-based communication or the JTunnel feature. In general, Java applets are allowed to open socket connections only to the machine from which they have been downloaded. This means the SAS/CONNECT session and the Web server that provide the Java classes must run on the same machine. This restriction is implemented via the Java applet security model and does not apply to Java applications.

The new JTunnel feature bypasses this Java applet security 'feature', enabling you to host your Web server and your SAS/CONNECT session on different machines!

A Hot Dark Drink?

Networks aren't the same since the Java programming language and the concept of Intranets and Extranets came onto the scene. Just a few years ago, all Java meant to most of us was something to quench our thirst or boost our energy while conversating at an out door cafe.

Today, Java is fueling a period of tremendous growth and enthusiasm for networked, platform independent, client/server computing.

Applets vs. Applications

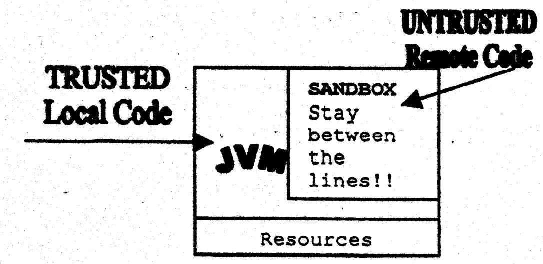
The first thing we must do is to distinguish between a *Java applet* and a *Java application*.

A Java program running in a browser is an applet. The security model inherent in the client browser is rather restrictive. This in turn, requires that your SAS server be installed on the same machine (or host) as your Web server resides.

A Java program which runs on some client machine in some Java Virtual Machine (JVM), but not in a browser is a Java application. By default the security manager lets an application do whatever it wants so there are no limitations except those enforced by the language itself, and even those can be bypassed by using native methods.

A Java application is essentially a stand-alone application. Java applets are designed to be used within the framework of an HTML document and accessible via a Java-enabled browser.

Figure 1: JAVA 'SandBox' Security Model



Java Security Considerations

So why is Java considered such a good match for network applications when it apparently has such a restrictive limitation?

Consider that an applet might be downloaded cross a firewall. The applet is now running in a machine on a secure Intranet. If the applet is allowed to open connections to anything it wants, it can now scan any machines it can reach on that intranet for ftp, telnet, http, smtp or any other possible entry points. If it sees any http (web server) ports, it can start downloading pages from http servers, and upload them to some malicious server out beyond the firewall. The applet is now stealing sensitive corporate data, or sensitive personal information .

This is indeed a legitimate rationale for the implementation of the Java applet security model or the 'Sandbox Model'. See Figure 1.

Java was developed to prevent applets from snooping or changing files on the client file system, or from using network connections to circumvent file protections or individual's expectations of privacy.

What this means for us developers is that your applet will only be able to read to and write from a file on the server that the applet was served from.

The applet security limitations have caused more than a few programmers to scream out in pain. The original 'Sandbox Model' has indeed evolved with Sun Microsystem, Inc.'s introduction of the Java Development Kit (JDK) 1.1. To date 'signed' and 'trusted' applets have been implemented. Still under development are:

- ***fine-grained access control***
- ***configurable security policy***
- ***enabling Java applets and Java applications to be treated uniformly***

What if you have an Intranet or extranet and you truly **want** your Java applets to be able to communicate with a server outside of the realm of the applet's home? What if you can't justify the expense of requiring that the SAS Software be installed on the tens or hundreds of sales representatives' personal computers? What if you want your sales representatives to access timely SAS data via a Web browser while they're in the field?

It appears that many vendors are providing workarounds to the security restrictions of Java applets. SAS Institute Inc is listening and being proactive in this regard.

The SAS Institute Inc Solution

SAS Institute Inc. has implemented its own solution in version 1.1 of the SAS/IntrNet Software.

JConnect is a compute services component of SAS/IntrNet. It is a driver or a set of Java class libraries which provides the Application Programming Interface (API) that allows Java applications or applets to connect directly to SAS servers. This is done via the SAS/Connect server.

The advantage of this service is that we are now able to create Java applets which can securely start and connect to a SAS session as well as execute SAS code including SQL statements and display results from the session.

JTunnel is a 'feature' that can be used with Java applets created using JConnect. It uses HTTP tunneling techniques to enable Java applets to take advantage of SAS data and computing services by eliminating almost all of the restrictions associated with where the SAS software is executed.

The icing on the cake is that our clients, staff, field sales representatives, and external partners do not need a copy of the SAS Software on their computers. Mission critical SAS data is available through a common interface - the web browser.

System Configuration

SAS/Connect Server Machine Configuration

The SAS/Connect Server is configured as follow:

- **Windows/NT 4.0 Operating System**
- **Dell PowerEdge 4100/200 (Pentium 200)**
- **256MB RAM**
- **The SAS Software System, Release 6.12 stored in D:\SAS directory**
- **SAS/IntrNet 1.1 including JConnect stored in D:\SAS\Intrnet**
- **The SAS/CONNECT Spawner program for Windows, tspawner.exe running as a Windows NT service for TCP/IP communications access method (advantages: allows the NT machine to service SAS/CONNECT remote requests without requiring the user to first log onto the Windows NT PC and execute the spawner program and the spawner honors the access rights of the Windows NT registry with respect to the requesting user id.)**
- **Clinical Trial RAW SAS Data sets**

After installing SAS/IntrNet 1.1, it was necessary to delete the old SAS spawner service, spawner.exe in favor of the new SAS spawner service, tspawner.exe. This was done as follow:

1. **From the Control Panel, double-click on the Services icon.**
2. **Find the SAS Spawner service in the list of services**
3. **Stop the SAS Spawner service**
4. **Go to the DOS command prompt.**
5. **CD to the directory containing spawner.exe. yourdrive:\SAS\CONNECT\SASEXE**
6. **Delete the old spawner service →Type: spawner -delete**
7. **Install the new spawner service →Type: tspawner -i comamid tcp -file d:\sas\sas.exe**

Refer to the Communications Access Methods SAS/CONNECT and SAS/SHARE Software Version 6, First Edition to gain an understanding of the spawner program and its many parameters.

The final step necessary to perform on the SAS/CONNECT server was to create an Autoexec.SAS file in the SAS executable directory. A libname statement was added in order to ensure that the SAS clinical data sets were accessible via the web browser.

The sample code in Figure 2 was submitted remotely by a third computer which has the SAS Software installed. It was used to test whether the spawner was correctly installed and started.

NOTE: No libname was specified as an Autoexec.SAS file exists on the SAS/CONNECT server, which contains all necessary libnames.

```
options comamid=tcp remote=bottom;
signon;
  rsubmit;
   proc download data=clinical.antres;
   run;
  endrsubmit;
signoff;
```

Figure 2: TESTSPAWN.SAS

Web Server Machine Configuration

The Web Server is configured as follow:

- **Windows/NT 4.0 Operating System**
- **Pentium 120 server**
- **96MB RAM**
- **Netscape Enterprise Server 3.0 installed in D:\Netscape\Suitespot directory**
- **JTunnel executable cgi programs (shrcgi.exe and shrproc.exe) stored in a subdirectory of the web server's cgi directory**

- **The uncompressed zipped Modified Java Tools for JDK 1.02 file stored in a subdirectory of the web server directory.**
- **Custom Version of SAS Institute's JTunnelViewer.html which uses the JViewer Java applet – PROJECT.html**

First, the Java archives in zipped format was copied to the D:\Netscape\SuiteSpot\docs\applets directory on the web server machine. Using a Java archive can significantly improve download time by reducing the number of HTTP connections required to fetch Java applet code.

The next series of steps are very crucial to ensuring that the tunneling feature is operative.

1. **Copy shrcgi.cfg, shrcgi.exe, and shrproc.exe to the cgi-bin (or wherever cgi scripts are stored on your web server machine) directory of the web server.**
2. **Modify the Message Router configuration file, shrcgi.cfg according to your unique site specifications.** See Figure 3
3. **Take special care to correctly specify the tunneling parameter, routerUrl. It is used by the TunnelConnectClient applet to determine the location of the Message Router.**

```
TIMEOUT=60

# The SASHOST parameter is the name of the SAS/CONNECT server
# which the JConnect applets will communicate with.

SASHOST=bottom

# The SASPORT parameter is port 23 which is the telnet port
# where SAS/CONNECT server is listening on. No other ports will
# be allowed access from the applets.

ALLOW_SASPORT=23

# The following parameters are used by JConnect. ONLY the
# command specified by the alias mySasCommand will be allowed
# for JConnect applets. This is the command to start SAS on
# the host, bottom. NO other commands will be allowed.

mySasCommand = d:\\sas\\sas.exe
ALLOW RESPONSE 3=$mySasCommand
```

Figure 3: SHRCGI.CFG Message Router Configuration File

The TelnetConnectClient applet communciates with the SAS/CONNECT spawner program (tspawner.exe) via port 23 (the telnet port) on the SAS/CONNECT server machine. The following HTML parameter tag will be used in Project.html below:

```
<param name=host value="bottom">
<param name=port value="23">
<param name=prompt1 value="Hello>">
<param name=response1 value="$mySasCommand">
<param name=routerUrl
value="http://masochist:81/cgi/jtunnel/shrcgi.exe">
```

A SAS session will be started by the TelnetConnectClient using the value of $mySasCommand defined in SHRCGI.CFG.

The TelnetConnectClient object does not create a socket connection to the remote SAS/CONNECT server. "Tunneling" is used which means that it sends an HTTP request to the CGI program, SHRCGI.EXE (the SAS Message Router), who then passes that request to SHRPROC.EXE (the SAS Protocol Interpreter). The SAS Protocol Interpreter is always 'awake' as it is a persistent process but the SAS Message Router is not persistent and is only 'awaken' when an HTTP request is sent from the TelnetConnectClient object.

The next thing we need to do on the web server machine is to create the web page which will enable us to remotely access the clinical trials SAS data files on the SAS/CONNECT server.

HyperText Markup Language(HTML) Web Page

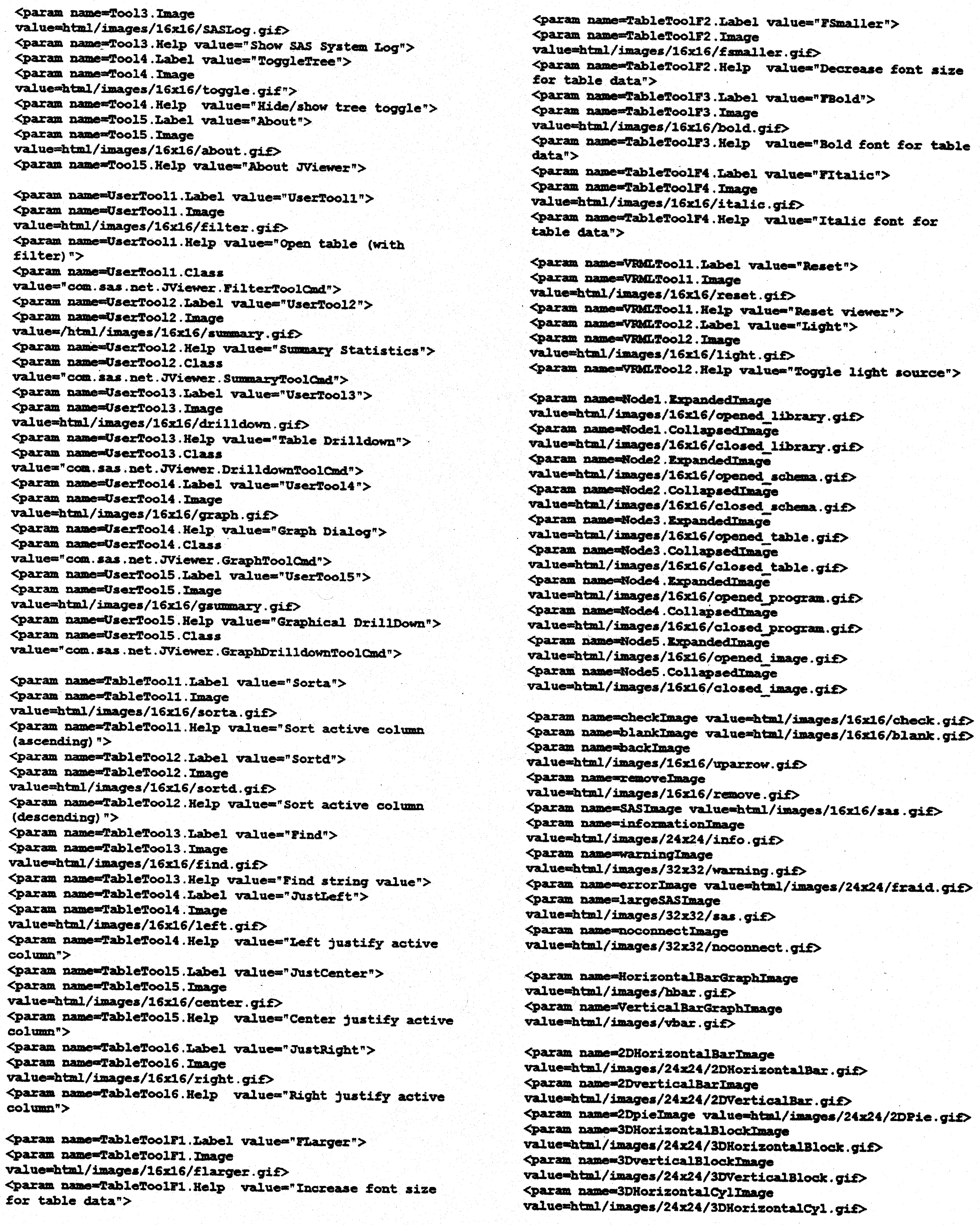
The HTML code to generate the main web page, PROJECT.HTML, was adapted from the code contained within the SAS Institute's web site and is as follow:

```
<HTML>
<HEAD>
<TITLE>Analyzing Clinical Trial SAS Data Remotely using
the JViewer Applet</TITLE>
</HEAD>
<body text="black" bgcolor="white" link="purple"
vlink="teal">
<table width=100%>
<tr>
<td bgcolor="black" align="middle">
<img src="images/frslogo.gif" height="127" width="277">
</td>
</tr>
</table>
<br>
<center>
<font size="+2">Remote Data Review System - RDRS</font>
</center>
<br>
*Remember that all data are preliminary to-date.
<br>
<font color="red" size="+1">
<center>
Phase I/II Trial of Safety and Immunogenicity of T Cell
Receptor Peptide Vaccines in Patients With Multiple
Sclerosis
</center>
</font>
<br>
<applet archive="js102mod.zip"
code="com.sas.net.JViewer.JViewerApplet.class"
codebase="../" width=700 height=350>
<param name=host value="bottom">
<param name=port value="23">
<param name=prompt1 value="Hello>">
<param name=response1 value="$mySasCommand">
<param name=routerUrl
value="http://masochist:81/cgi/jtunnel/shrcgi.exe">

<!-- The rest of the parameters do not require
modification unless you change the location of your
images or have created new user tool java applets and
added new tool bar buttons to the JViewer java applet-->

<param name=smallImageDimension value=16,16>
<param name=mediumImageDimension value=24,24>
<param name=largeImageDimension value=32,32>
<param name=fontImageDimension value=100,36>
<param name=DefaultFixedWidthFont value=Courier-bold-14>
<param name=viewerBackgroundColor value=white>

<param name=Tool1.Label value="Open">
<param name=Tool1.Image value=html/images/16x16/open.gif>
<param name=Tool1.Help value="Open table (no filter)">
<param name=Tool2.Label value="Properties">
<param name=Tool2.Image
value=html/images/16x16/properties.gif">
<param name=Tool2.Help value="Show table properties">
<param name=Tool3.Label value="SASLog">
```



```
<param name=Tool3.Image
value=html/images/16x16/SASLog.gif>
<param name=Tool3.Help value="Show SAS System Log">
<param name=Tool4.Label value="ToggleTree">
<param name=Tool4.Image
value=html/images/16x16/toggle.gif">
<param name=Tool4.Help  value="Hide/show tree toggle">
<param name=Tool5.Label value="About">
<param name=Tool5.Image
value=html/images/16x16/about.gif>
<param name=Tool5.Help value="About JViewer">

<param name=UserTool1.Label value="UserTool1">
<param name=UserTool1.Image
value=html/images/16x16/filter.gif>
<param name=UserTool1.Help value="Open table (with
filter)">
<param name=UserTool1.Class
value="com.sas.net.JViewer.FilterToolCmd">
<param name=UserTool2.Label value="UserTool2">
<param name=UserTool2.Image
value=/html/images/16x16/summary.gif>
<param name=UserTool2.Help value="Summary Statistics">
<param name=UserTool2.Class
value="com.sas.net.JViewer.SummaryToolCmd">
<param name=UserTool3.Label value="UserTool3">
<param name=UserTool3.Image
value=html/images/16x16/drilldown.gif>
<param name=UserTool3.Help value="Table Drilldown">
<param name=UserTool3.Class
value="com.sas.net.JViewer.DrilldownToolCmd">
<param name=UserTool4.Label value="UserTool4">
<param name=UserTool4.Image
value=html/images/16x16/graph.gif>
<param name=UserTool4.Help value="Graph Dialog">
<param name=UserTool4.Class
value="com.sas.net.JViewer.GraphToolCmd">
<param name=UserTool5.Label value="UserTool5">
<param name=UserTool5.Image
value=html/images/16x16/gsummary.gif>
<param name=UserTool5.Help value="Graphical DrillDown">
<param name=UserTool5.Class
value="com.sas.net.JViewer.GraphDrilldownToolCmd">

<param name=TableTool1.Label value="Sorta">
<param name=TableTool1.Image
value=html/images/16x16/sorta.gif>
<param name=TableTool1.Help value="Sort active column
(ascending)">
<param name=TableTool2.Label value="Sortd">
<param name=TableTool2.Image
value=html/images/16x16/sortd.gif>
<param name=TableTool2.Help value="Sort active column
(descending)">
<param name=TableTool3.Label value="Find">
<param name=TableTool3.Image
value=html/images/16x16/find.gif>
<param name=TableTool3.Help value="Find string value">
<param name=TableTool4.Label value="JustLeft">
<param name=TableTool4.Image
value=html/images/16x16/left.gif>
<param name=TableTool4.Help  value="Left justify active
column">
<param name=TableTool5.Label value="JustCenter">
<param name=TableTool5.Image
value=html/images/16x16/center.gif>
<param name=TableTool5.Help  value="Center justify active
column">
<param name=TableTool6.Label value="JustRight">
<param name=TableTool6.Image
value=html/images/16x16/right.gif>
<param name=TableTool6.Help  value="Right justify active
column">

<param name=TableToolF1.Label value="FLarger">
<param name=TableToolF1.Image
value=html/images/16x16/flarger.gif>
<param name=TableToolF1.Help  value="Increase font size
for table data">
```

```
<param name=TableToolF2.Label value="FSmaller">
<param name=TableToolF2.Image
value=html/images/16x16/fsmaller.gif>
<param name=TableToolF2.Help  value="Decrease font size
for table data">
<param name=TableToolF3.Label value="FBold">
<param name=TableToolF3.Image
value=html/images/16x16/bold.gif>
<param name=TableToolF3.Help  value="Bold font for table
data">
<param name=TableToolF4.Label value="FItalic">
<param name=TableToolF4.Image
value=html/images/16x16/italic.gif>
<param name=TableToolF4.Help  value="Italic font for
table data">

<param name=VRMLTool1.Label value="Reset">
<param name=VRMLTool1.Image
value=html/images/16x16/reset.gif>
<param name=VRMLTool1.Help value="Reset viewer">
<param name=VRMLTool2.Label value="Light">
<param name=VRMLTool2.Image
value=html/images/16x16/light.gif>
<param name=VRMLTool2.Help value="Toggle light source">

<param name=Node1.ExpandedImage
value=html/images/16x16/opened_library.gif>
<param name=Node1.CollapsedImage
value=html/images/16x16/closed_library.gif>
<param name=Node2.ExpandedImage
value=html/images/16x16/opened_schema.gif>
<param name=Node2.CollapsedImage
value=html/images/16x16/closed_schema.gif>
<param name=Node3.ExpandedImage
value=html/images/16x16/opened_table.gif>
<param name=Node3.CollapsedImage
value=html/images/16x16/closed_table.gif>
<param name=Node4.ExpandedImage
value=html/images/16x16/opened_program.gif>
<param name=Node4.CollapsedImage
value=html/images/16x16/closed_program.gif>
<param name=Node5.ExpandedImage
value=html/images/16x16/opened_image.gif>
<param name=Node5.CollapsedImage
value=html/images/16x16/closed_image.gif>

<param name=checkImage value=html/images/16x16/check.gif>
<param name=blankImage value=html/images/16x16/blank.gif>
<param name=backImage
value=html/images/16x16/uparrow.gif>
<param name=removeImage
value=html/images/16x16/remove.gif>
<param name=SASImage value=html/images/16x16/sas.gif>
<param name=informationImage
value=html/images/24x24/info.gif>
<param name=warningImage
value=html/images/32x32/warning.gif>
<param name=errorImage value=html/images/24x24/fraid.gif>
<param name=largeSASImage
value=html/images/32x32/sas.gif>
<param name=noconnectImage
value=html/images/32x32/noconnect.gif>

<param name=HorizontalBarGraphImage
value=html/images/hbar.gif>
<param name=VerticalBarGraphImage
value=html/images/vbar.gif>

<param name=2DHorizontalBarImage
value=html/images/24x24/2DHorizontalBar.gif>
<param name=2DverticalBarImage
value=html/images/24x24/2DVerticalBar.gif>
<param name=2DpieImage value=html/images/24x24/2DPie.gif>
<param name=3DHorizontalBlockImage
value=html/images/24x24/3DHorizontalBlock.gif>
<param name=3DverticalBlockImage
value=html/images/24x24/3DVerticalBlock.gif>
<param name=3DHorizontalCylImage
value=html/images/24x24/3DHorizontalCyl.gif>
```

```
<param name=3DVerticalCylImage
value=html/images/24x24/3DVerticalCyl.gif>
<param name=3DHorizontalHexImage
value=html/images/24x24/3DHorizontalHex.gif>
<param name=3DVerticalHexImage
value=html/images/24x24/3DVerticalHex.gif>
<param name=3DHorizontalPrismImage
value=html/images/24x24/3DHorizontalPrism.gif>
<param name=3DVerticalPrismImage
value=html/images/24x24/3DVerticalPrism.gif>
<param name=3DPieImage value=html/images/24x24/3DPie.gif>
<param name=swissFontImage value=html/images/swiss.gif>
<param name=swissLFontImage value=html/images/swissl.gif>
<param name=swissBFontImage value=html/images/swissb.gif>
<param name=swissBIFontImage
value=html/images/swissbi.gif>
<param name=simplexFontImage
value=html/images/simplex.gif>
<param name=italicFontImage value=html/images/italic.gif>
<param name=centBFontImage value=html/images/centb.gif>
<param name=centBIFontImage value=html/images/centbi.gif>
<param name=defaultFontImage
value=html/images/default.gif>

<param name=useLabels value="true">
</applet>
<br>
<a href=http://www.frsa.com>FRS Associates,LLC</a>
<br>
<IMG SRC="images/saslstat.gif">
</BODY>
</HTML>
```

On the Netscape Enterprise web server machine, the directory structure is as follow:

- **Docs**
  - **ClinicalStudies**
    - **Html**
      - **Images**
    - **com**
      - **sas**
        - **net**
          - **JViewer**

Project.html is stored in the D:\Netscape\SuiteSpot\Docs\ClinicalStudies\Html sub-directory. The JViewer applets (class files) are stored in the D:\Netscape\SuiteSpot\Docs\ClinicalStudies\com\sas\net\Jviewer sub-directory.

Finally, all GIF images are stored in the D:\Netscape\SuiteSpot\Docs\ClinicalStudies\Html\Images sub-directory.

By glancing at Project.html source, you can see where the HTML <Applet> tag is used.

```
<applet archive="js102mod.zip"
code="com.sas.net.JViewer.JViewerApplet.class"
codebase="../" width=700 height=350>
<param name=host value="bottom">
```

What happens here is that the "archive" attribute of the <APPLET> tag allows the file js102mod.zip to be downloaded to the client's disk and will search it for the Jviewer class and the classes it requires. This zip file is found relative to the codebase path, and must not be compressed. Classes not in the zip archive will still be searched for via the old mechanism if indeed requried.

The advantage of the archive approach is that it minimizes the number of HTTP connections needed everytime the Jviewer applet requires another Java applet. If we chose not to use the archive, then each request for an applet would require yet another request to the HTTP server.

The 'codebase' is defined as being in the parent directory of the HTML directory (../). The parent directory is the ClinicalStudies directory. The Jviewer applet, is found in the **"../com/sas/net/Jviewer/JViewerApplet.class"** directory.

Using the PARAM element, you can pass general purpose parameters to an element. The first parameter passed to the applet is the 'host' parameter with a value of "bottom". "Bottom" is the name of the SAS/CONNECT server machine.

Applets access the parameters set in the <PARAM> tag using the getParameter() method.

Check out the SAS Institute's web site referenced in the References section of this paper for a detailed discussion of the various parameters utilized by the Jviewer applet class.

Following is a discussion of a few of the exciting functions the JViewer applet class has to offer.

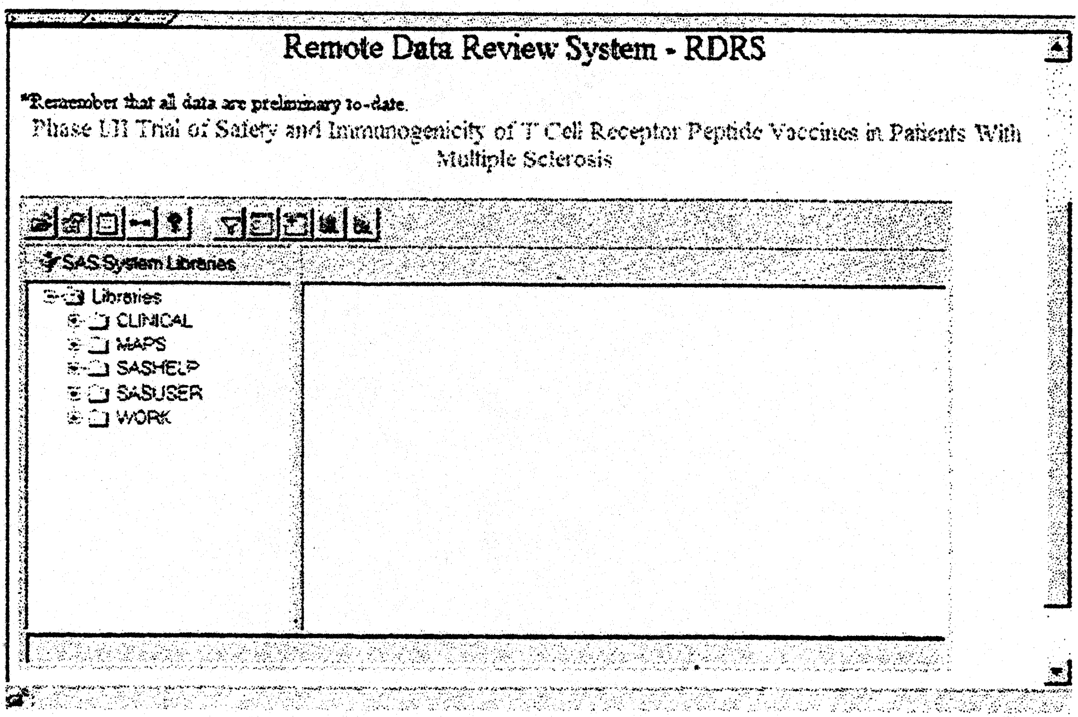
Let The Dance Begin!

When the web page is first accessed, it will take a few seconds for the JViewer applet to connect to the SAS/CONNECT server and to display the available SAS libraries.

Figure 4 illustrates the appearance of the web page, Project.html when it is first accessed via the browser. The SAS data library tree structure displays each of the available libraries which were defined to the SAS/CONNECT server via Config.sas and Autoexec.SAS. These are the only libraries which the JViewer applet is allowed to access. The plus sign (+) indicates that the SAS data sets or views are accessible.

Here we will access the CLINICAL SAS data library. By double clicking on CLINICAL or double clicking on the folder icon or clicking once on the plus sign, the CLINICAL folder is transformed from 'collapsed' format to 'expanded' format as illustrated in Figure 5

Figure 4: PROJECT.HTML upon first access

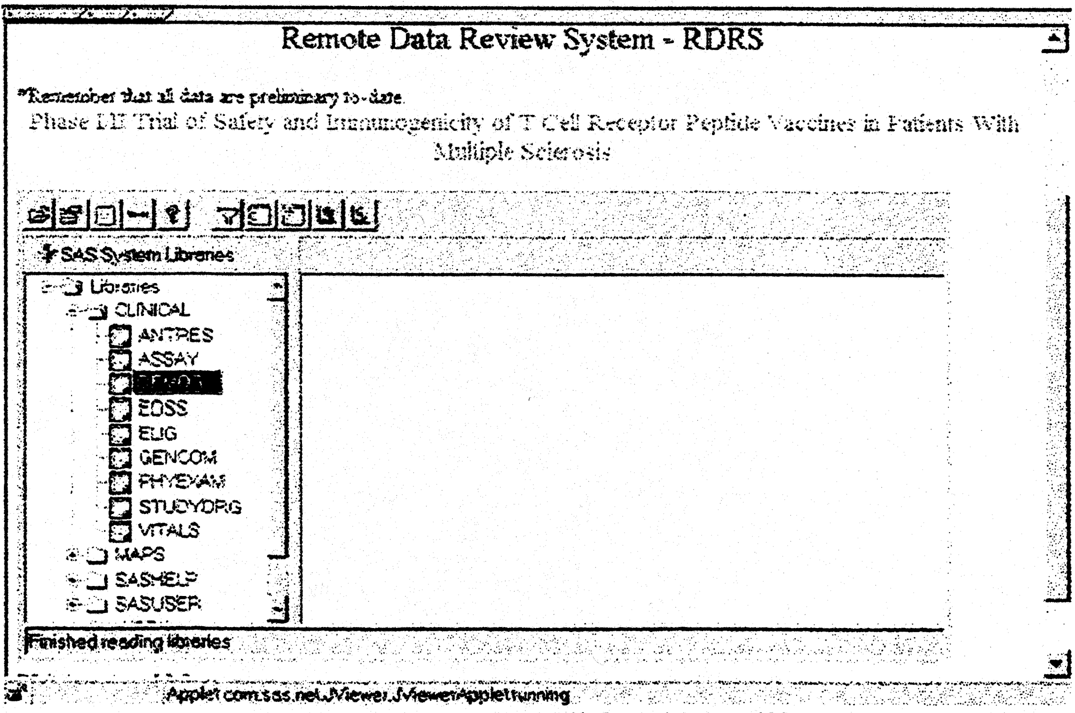


Remote Data Review System - RDRS

*Remember that all data are preliminary to-date.

Phase I/II Trial of Safety and Immunogenicity of T Cell Receptor Peptide Vaccines in Patients With Multiple Sclerosis

Figure 5: Expanded Tree Structure for CLINICAL SAS Data



Remote Data Review System - RDRS

*Remember that all data are preliminary to-date.

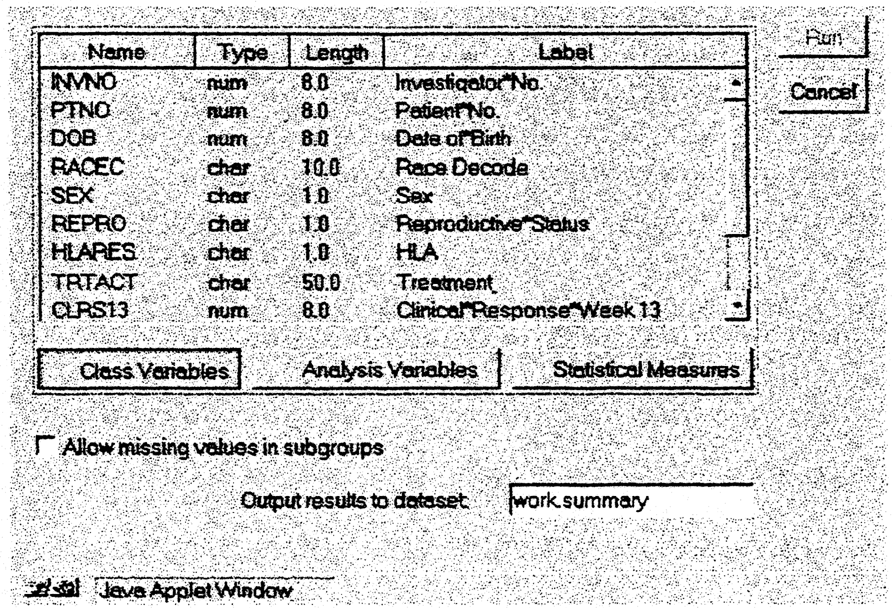
Phase I/II Trial of Safety and Immunogenicity of T Cell Receptor Peptide Vaccines in Patients With Multiple Sclerosis

Finished reading libraries

Applet com.sas.net.JViewer.JViewerApplet running

By clicking once on the DEMOG SAS data set, and then clicking on the Summary Statistics toolbutton, on the toolbar, a Java applet window (dialog box) will appear as illustrated in Figure 6.

Figure 6: SUMMARY STATISTICS Tool Button Java Dialog Window For DEMOG SAS Data Set



Name	Type	Length	Label
INVNO	num	8.0	Investigator*No.
PTNO	num	8.0	Patient*No.
DOB	num	8.0	Date of Birth
RACEC	char	10.0	Race Decode
SEX	char	1.0	Sex
REPRO	char	1.0	Reproductive*Status
HLARES	char	1.0	HLA
TRTACT	char	50.0	Treatment
CLRS13	num	8.0	Clinical*Response*Week 13

□ Allow missing values in subgroups

Output results to dataset: work.summary

Java Applet Window

From this window, you can select one or more class variables, analysis variables, and statistical measures such as number of observations, mean, maximum, minimum, corrected sum of squares, standard deviation, etc. A rather robust tool indeed.

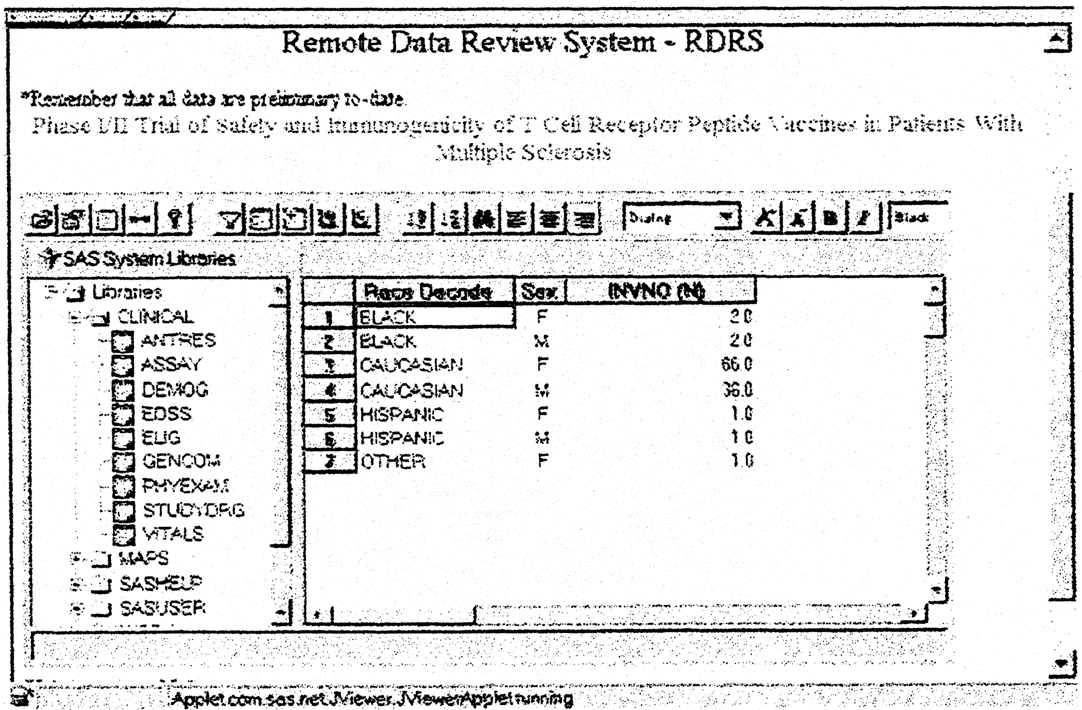
The following parameters were selected:

- **Class Variables**: RACEC (Race Decode) and SEX (Sex)
- **Analysis Variables**: INVNO (Investigator Number)
- **Statistical Measures**: Number of Observations,

The resulting output data set will be stored in WORK.SUMMARY. This data set name can also be typed in from the Java Dialog Window.

The JVIEWER applet class calls the applet which 'remote submits' SAS Procedure code for data summarization. In Figure 7 there is a summarization of the number of investigators categorized by the race and sex variables.

Figure 7: Results of SAS Procedure Code



Remote Data Review System - RDRS

*Remember that all data are preliminary to-date.

Phase I/II Trial of Safety and Immunogenicity of T Cell Receptor Peptide Vaccines in Patients With Multiple Sclerosis

	Race Decode	Sex	INVNO (N)
1	BLACK	F	2.0
2	BLACK	M	2.0
3	CAUCASIAN	F	66.0
4	CAUCASIAN	M	36.0
5	HISPANIC	F	1.0
6	HISPANIC	M	1.0
7	OTHER	F	1.0

Applet com.sas.net.JViewer.JViewerApplet running

Conclusion

The capabilities demonstrated in this paper does not come close to demonstrating the power of the SAS Software System! The JViewer package can equip a remote user with the ability to subset or filter data, sort data sets, perform summary data drill-downs, graphics drill-down, view SAS logs, format output dynamically and more.

In addition, there exist a facility where a Java developer can add additional functionality to the JViewer applet.

The SAS Institute has brought the realm of thin client computing to the forefront of where data analysis and presentation should be today--platform independent client/server computing.

References

http://java.sun.com/sfaq/information Frequently Asked Questions - Java Security.

http://www.sas.com/rnd/web/java/ Java tools and documentation from the SAS Institute.

http://www.sas.com/rnd/web/java/tshoot.html Frequently Asked Questions - Java Security.

http://www.frsa.com/ Click on the online applications link to access the JConnect and JTunnel demonstration in this paper. Click on the papers link to access this paper online.

SAS Institute Inc., (1997), *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

Walters,B., (1997), "Exploiting Java^{TM} Technology with the SAS® Software," *Proceedings of the Twenty Second Annual SAS Users Group International Conference*, 22, 865-869.

SPECIAL NOTES

Client Platform LIMITED Testing:

Check out the SAS Institute Java Tools - Troubleshooting Guide set forth in the References section above.

Netscape Navigator 3.03 - SUCCESSFUL

Netscape Communicator 4.01a - FAILURE

```
Applet exception: nanos > 999999999 or < 0
java lang IllegalArgumentException: nanos >999999999 or < 0
Applet com.sas.net.Jviewer.JviewerAapplet nanos >999999999 or < 0
```

Netscape Communicator 4.02 - SUCCESSFUL

Netscape Navigator(standalone) 4.04 - SUCCESSFUL

Netscape Communicator 4.04 - SUCCESSFUL

Microsoft Internet Explorer 4.0 with default Java JIT Compiler turned on - FAILURE

Microsoft Internet Explorer 4.0 with default Java JIT Compiler turned off (and reboot) - SUCCESSFUL

AUTHOR CONTACT:

Faith Reneé Sloan
FRS Associates,LLC
2750 Market Street, Suite 101
San Francisco, CA 94114-1987
(415) 626-9796
http://www.frsa.com/
faith@frsa.com