

Fast Track to SAS: An Apache Armed with Axom

Matthew Pearce, Amadeus Software Limited



ABSTRACT

This paper takes a look at a set of SAS® Server tools called Axom, a simple, cost-effective way to access SAS® software reports and applications on the web.

Axom is an application server that can be used to control the SAS system in a unique way from any other software application. It uses a unique TCP/IP protocol to enable SAS software to be queried to perform tasks using other applications like Java, Perl, Visual Basic, or C/C++.

Axom remains true to the generic capabilities of SAS software, as it will function on most operating systems. Axom allows the user to run SAS software applications using a variety of different front-ends, including web pages (using HTML in tandem with a scripting language), Client-side applications or even Java applets.

The benefits of using Axom will be analysed in order to illustrate how Axom can provide a robust and dependable alternative to other SAS software communication methods. The configuration of Axom will be explained, in order to demonstrate the different ways in which Axom can function. Further features discussed here will include Axom Dynamic HTML, a choice of CGI scripting languages, Internet Server API support and a built-in web server.

INTRODUCTION

There are a variety of different methods of communicating between application front-ends and SAS® Software on the back-end. The communication method used generally determines which application server must be used to provide the go-between. SAS/Connect enables Client/Server processing, whilst SAS/IntrNet® software can link browser front-ends to SAS software via a web server. There are also communication methods that can accomplish both of these tasks and more besides; one such product is Axom, a suite of server tools from InterNext.

This paper will introduce Axom, with a brief explanation of what Axom is, how it works and where it fits in relation to SAS software. The features of Axom will be described here in order to show how Axom can provide a robust and dependable alternative to other SAS software communication methods.

A specific example will also be given of connecting a browser front-end to Axom via an Apache web server. This will illustrate the various configuration requirements of Axom the application server.

AXOM AND WHERE IT FITS

Axom is a set of server side tools that can harness CGI (Common Gateway Interface), ISAPI (Internet Server Application Protocol Interface) and Java technologies to enable users to analyse and report on data. Axom achieves this by allowing users to run SAS software programs and access SAS software reports and applications from the web or other environments.

Axom is essentially an application server, acting as the go-between for the application on the front end and the back-end processes. This relationship is illustrated in Appendix A at the end of this paper.

Note in particular the two Axom components, the Java dispatcher (configuration file dispatcher.cfg) which controls the Axom SAS drones (configuration file drone.sas), waiting to

process SAS software requests. We shall return to these files later in the section on Axom configuration.

BASIC FEATURES

RUNNING SAS SOFTWARE PROGRAMS

One of the benefits of Axom is the simplicity with which users can run SAS software programs. The design of Axom enables programmers to call SAS software programs by explicit path reference, i.e. using virtual directories combined with a program name and file extension. The example below shows one of the Axom test programs illustrating some sample ODS HTML output passed to the web browser by Axom.

Note: Axom test program available from the built-in web server at: <http://localhost:6500/Program+Files/InterNext/Axom/Demos/Test/test8.sas>

Macro Scope	Macro Variable Name	Offset into Macro Variable	Macro Variable Value
GLOBAL	_SQLOBS	0	0
GLOBAL	_AXOMSAVESESSION	0	Yes
GLOBAL	_SQLOOPS	0	0
GLOBAL	_AXOMSESSIONID	0	ZFQGDVSPDLUHZUNK
GLOBAL	_AXOMSESSIONID_	0	C
GLOBAL	_AXOMTOTALDRONES	0	1
GLOBAL	_IP	0	127.0.0.1
GLOBAL	_AXOMAPPENDLOG	0	No
GLOBAL	_BROWSE	0	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
GLOBAL	_AXOMCAPTURE	0	
GLOBAL	_AXOMCAPTUREDBY	0	
GLOBAL	_AXOMREQUESTSTARTED	0	1398810610.722
GLOBAL	_AXOMPROGRAMAFTR	0	
Request: Program=C:\Program			

Programs can also be called from within the web page, using a technique similar to that used in SAS/IntrNet® software. This is done by passing the `_program` parameter as a hidden value within an HTML form. The program must be referenced by one of the libnames defined in the `drone.sas` configuration file in the Axom install. Notice also that the program extension is also required:

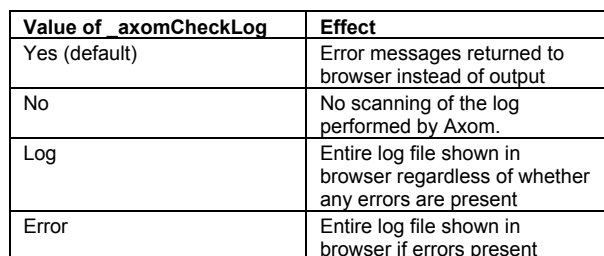
```
<FORM ACTION="/cgi-bin/broker.exe"
METHOD="post" NAME="myform">
<INPUT TYPE="hidden" NAME="_program"
VALUE="mylib.analyse.sas">
```

This can also be done for SCL programs (these must be pre-compiled)

```
<INPUT type="hidden" name="_program"
value="mylib.mycat.myscl.scl">
```

DEBUGGING SAS SOFTWARE PROGRAMS

The SAS software log is automatically scanned by Axom after a request has been processed, with the errant log returned to the browser instead of the output by default. This is controlled by the Axom macro variable `_axomCheckLog`, with four possible values:



The screenshot shows a web browser window with the address bar displaying `http://localhost:4500/Program-Files/InterNet/Usion/Demos/Test/test1.asp`. The main content area displays the SAS Log output, which includes the following text:

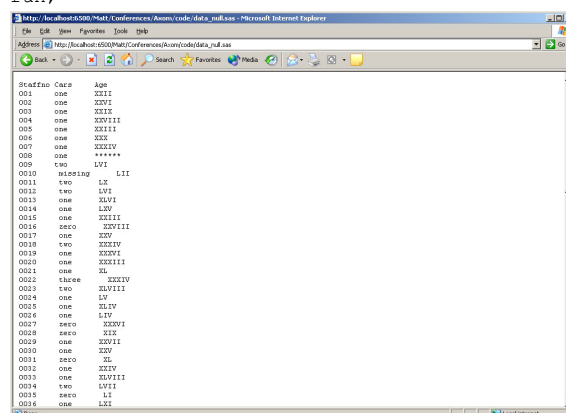
```
NOTE: SAS Version 1.3 Engine 31082004
NOTE: ASMA Request: Program-Files/InterNet/Usion/Demos/Test/test1.asp
NOTE: ASMA HostID: 76020707716122
121      list _asmaSasLogLog;
```

There are a variety of different methods that can be utilised to write SAS software output back to the browser. These include methods similar to those used by other Application Servers in addition to those specific to Axom.

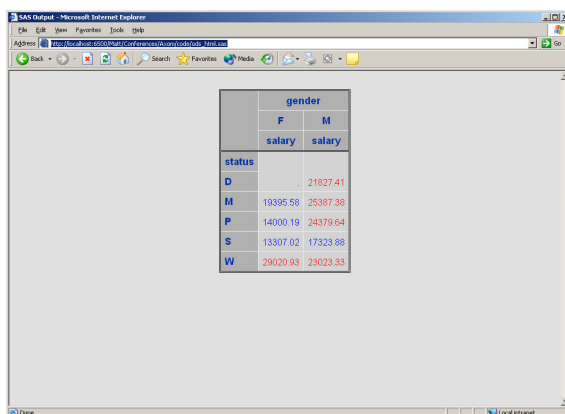
There are two filenames which enable the user to write output back to the web browser via Axom; `_webout` to write to files and `_webouta` to append to it. These two file names can be written to by various methods including:

-

- ```
data null ;
 file _webout;
 set amadeus.demog;
 put staffno age;
run;
```



- Page 2 of 2



|        | gender   |          |
|--------|----------|----------|
|        | F        | M        |
|        | salary   | salary   |
| status |          |          |
| D      |          | 21827.41 |
| M      | 10395.58 | 25387.35 |
| P      | 14000.19 | 24379.64 |
| S      | 13307.02 | 17323.88 |
| W      | 29020.93 | 23023.33 |

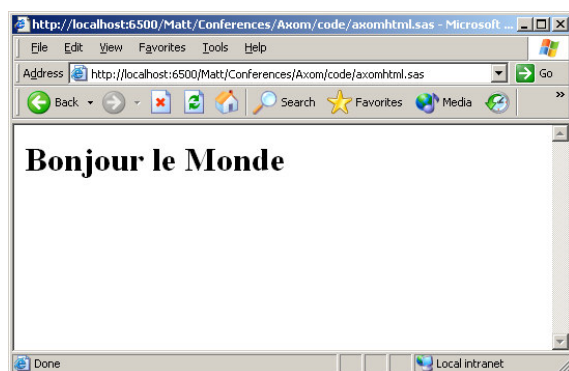
Note that the advantage of using ODS HTML is that the Axom header macro call is not required.

### %AXOMHTML MACRO

This is one of the built-in macros for Axom that allows a user to append HTML code directly into the \_webout filename to be passed back to the browser. This is an alternative to using ODS HTML; in this case you can actually write the html yourself.

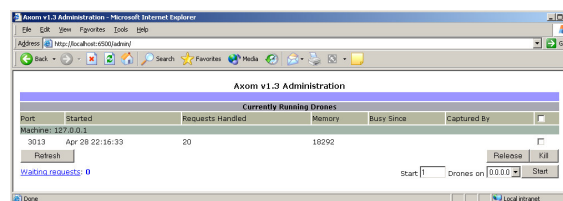
```
%axomHTML(
 <HTML>
 <BODY>
 <H1>
 Bonjour le Monde
 </H1>
 </BODY>
 </HTML>
)
```

Note that since this is essentially just a macro (although a pre-defined one) then you will require the usual quoting functions for special characters. This produces the following html page as seen below:



## AXOM ADMINISTRATION

Controlling the application dispatcher for Axom can be performed via the web browser using the simple Axom Administration screen. The address of this is specified in the configuration file for the Java application dispatcher (as Adminurl in the dispatcher.cfg file). This screen is shown below, used with localhost at <http://localhost:6500/admin/>.



| Axom v1.3 Administration  |                 |                  |        |            |
|---------------------------|-----------------|------------------|--------|------------|
| Currently Running Drones: |                 |                  |        |            |
| Port                      | Started         | Requests Handled | Memory | Busy Since |
| Machine: 127.0.0.1        |                 |                  |        |            |
| 3013                      | Apr 28 22:16:33 | 20               | 10292  |            |

From this screen the SAS software “drones” can be started, which each represent one SAS software session. The admin tool enables drones to be stopped, captured or released via the Java application dispatcher. Drones execute requests from the browser for SAS to perform an action, i.e. submit code. Capturing a SAS software drone means that a particular session will be retained exclusively by the Application Dispatcher for use by one drone.

## FLUSHING OUTPUT TO THE CLIENT

An important consideration when passing requests to a web browser is the amount of time it takes to receive the output from the request. This becomes more of an issue if there are several components to the output, since the user will have to wait for all of the output to be created before any will be displayed in the browser. To combat this issue, Axom provides the ability to “flush” output back to the client as it is created. This is done by including the following line comment within a SAS software program:

```
* Axom flush webout ;
```

The output buffer will be sent to the browser each time Axom encounters this line of code.

One drawback with this method is that it will not work when contained within a SAS macro, since the macro code resolution is treated as one block. So if procedures are dynamically created like the example below then the output is sent back as one block:

```
%macro tabulate;
%do i=1 %to 12;
proc tabulate data=work.month&i;
class gender status;
var salary;
table gender,status*salary;
run;
* Axom flush webout ;
%end;
%mend;
%tabulate;
```

## ADHTML BASE SAS SOFTWARE PROGRAMMING

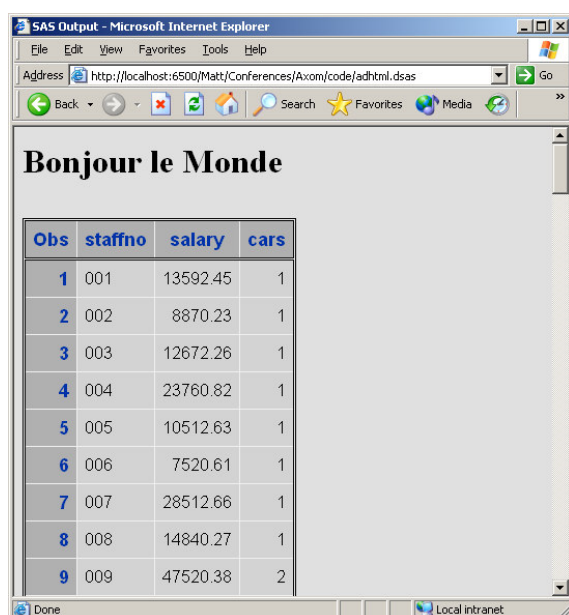
In the features discussed so far, the HTML screens and the SAS software code referenced by those screens has been kept apart. However, there is a method of allowing raw SAS software code to be embedded within the web page itself. This part of the Axom tool set is called Axom Dynamic HTML. These pages have their own special file extension, defined in the Axom SAS software configuration file (drone.sas). The default used is the .dsas extension, so that every file with this extension is assumed to contain Base SAS or SCL code.

The following code was saved to the file adhtml.dsas:

```
<HTML>
<BODY>
<H1>
Bonjour le Monde
</H1>
<%
 options nocenter;
 ods html file=_webouta;
 proc print data=amadeus.demog;
 var staffno salary cars;
 run;

 ods html close;
%>
</BODY>
</HTML>
```

Note the use of the `<%` and `%>` tags to denote where the SAS software code is referenced.



Obs	staffno	salary	cars
1	001	13592.45	1
2	002	8870.23	1
3	003	12672.26	1
4	004	23760.82	1
5	005	10512.63	1
6	006	7520.61	1
7	007	28512.66	1
8	008	14840.27	1
9	009	47520.38	2

## AN APACHE ARMED WITH AXOM

The example configuration explained here was setup with both Apache web server and Axom V1.4 installed on a Linux machine running Redhat 8. Axom was setup to mimic the behaviour of SAS/IntrNet® software, running SAS software programs via Apache and Axom using a CGI (Common Gateway Interface) executable file.

### APACHE WEB SERVER CONFIGURATION

The first task was to configure Apache to permit CGI execution in order to run the Axom broker file. This was controlled via the use of the `ScriptAlias` directive in the Apache configuration text file `httpd.conf`.

The `ScriptAlias` directive tells Apache that a particular directory is set aside for CGI programs. Apache will then regard every file in this directory as a CGI executable program, to be run when requested by a client.

The general syntax for the `ScriptAlias` directive is as follows:

```
ScriptAlias virtual_directory
 "physical_directory"
```

So for example, on Linux this was defined as:

```
ScriptAlias /web/cgi-bin
 "/usr/local/web/cgi-bin"
```

The `Alias` directive allows documents to be stored in the local file system other than under the `DocumentRoot` directive (the default location from which Apache will serve files). However, unlike `ScriptAlias`, executable files cannot be run from this location; so this was used to store any website pages, in order to make them accessible to outside requests.

The general syntax for the `Alias` directive is as follows:

```
Alias virtual_directory
 "physical_directory"
```

So for example, on Linux this was defined as:

```
ScriptAlias /web
 "/usr/local/web"
```

As previously mentioned, the Apache setup file is simply a text file so can be edited via a basic text editing tool. However, an alternative method on UNIX platforms is to install the webmin utility, an online application which enables administrators to modify web server configurations through the web browser. Further information on this useful item of freeware can be found at: <http://www.webmin.com/>.

Note: further information on the configuration of Apache web server can be found at the following address: <http://httpd.apache.org/docs/configuring.html>

## AXOM CONFIGURATION

At this stage Apache web server is configured to pass Axom requests onto the Axom broker file. The next step is to configure Axom to setup the SAS software environment that Axom will control.

### APPLICATION DISPATCHER

Axom SAS software sessions are controlled by the Java Application Dispatcher. This processes user requests, passing them on to an appropriate SAS software session (known as an Axom SAS software drone) and also manages the web-based administration utility for Axom. This is all configured via a configuration file called `dispatcher.cfg`, located in the `/Java` directory of the Axom installation.

The following key items are configured within this file, specific to each installation of Axom:

1. **SessionDirectory** - The Session directory for the Axom installation. When creating a new installation of Axom, this directory will have to be created within the Axom root directory. This will contain all data for a particular SAS software session and is assigned the "session" libname whilst a user is in a particular session. For example:

```
sessionDirectory=/var/axom/sessions
```

- AdminURL. Address of the Axom Administration utility for the installation. This admin URL can then be accessed by administrators to monitor the Application Dispatcher. For example:

```
adminUrl = /admin/
```

So to access from a web browser, user would go to:  
*http://servername:Port\_number/admin/*

- ServerAddress – IP address of the web server.
- ServerPort - A free server port number for Axom to use to handle requests. Ensure that this is not being used by anything else; Axom port numbers usually start from 6500. For example:

```
serverPort=6501
```

- startDroneCommand. This should point to the SAS software drone setup file for this installation (drone.sas).
- File extensions to process as ASAP Requests for integrated web server. The extension "exe" was added to mimic the behaviour of SAS/IntrNet® software. For example:

```
webServerExtensions=sas,scl,dsas,source
macro,exe
```

## SAS CONFIGURATION FILE

Axom SAS configuration is determined by the Axom root/SAS/drone.sas file, which sets up the SAS software environment using SAS macro variables.

The following items were edited in this file:

- Axom libname statement to point to the axom\_install\SAS. For example:  

```
libname axom '/usr/local/Axom/SAS';
```
- Session directory (same as defined in the Application Dispatcher). For example:  

```
sessionDirectory=/var/axom/sessions
```
- Dispatcher Address – IP address of the web server. (same as defined in the Application Dispatcher), only with Port Number added. For example:  

```
%let dispatcherAddr=10.13.2.40:6501;
```
- User defined libraries to reference from web pages – for example from \_program parameter submitted in an HTML form. Also add these libraries to the "list of permitted libnames" macro variable. For example:  

```
libname mylib "/usr/local/Programs";
%let libnamesAllowed=mylib;
```
- Axom license key – unique license key for your IP address and SAS software site license only.
- Location of Axom SAS software log files for the installation. For example:

```
%let errorLog
=/var/axom/SAS_errorLog.txt;
```

## WEB PAGE CONFIGURATION

The system setup was designed to mimic the behaviour of SAS/Intrnet software, utilising the \_program parameter submitted within a form and with the Axom broker file called as the form action (using the script alias referenced in the Apache web server config file httpd.conf).

```
<FORM ACTION="/web/cgi-bin/broker.exe"
METHOD="post" TARGET="" NAME="myForm")
<INPUT TYPE="HIDDEN" NAME="_program"
VALUE="mylib.myprog.sas">
```

Other hidden parameters are then passed through for use in the program "myprog.sas" as macro variables.

## CONCLUSION

Over the last few years, OLAP (Online Analytical Processing) has increased greatly in usage and therefore importance to a business. Users require access to dynamically driven data and reports that are up-to-date and accessible across the globe at any time.

As the need for this type of processing increases, the demand for this software tools such as Axom which are capable of enabling distributed browser based connections to SAS software for functions such as OLAP processing.

As a relatively new product, Axom has been designed to include compatibility with features found in more established communication tools such as SAS/Intrnet software. This is in addition to providing certain key features of its own, such as the ability to call programs directly by entering the virtual path in the browser; also to embed SAS software code directly within a web page.

New functionality is provided whilst providing compatibility to older, more established methodologies. This flexibility means that Axom SAS Server Tools can provide an attractive option to application administrators looking for an efficient and reliable communication tool.

## ACKNOWLEDGMENTS

Axom SAS Server Tools are owned and distributed by Internext, Inc. Further information can be found at [www.axomsoftware.com](http://www.axomsoftware.com)

## CONTACT INFORMATION

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

Author Name:	Matthew Pearce
Company:	Amadeus Software Limited
Address:	Orchard Farm, Witney Lane, Leafield, Oxon OX29 9PG
Phone:	01993 878287
Fax:	01993 878042
Email:	<a href="mailto:matthew.pearce@amadeus.co.uk">matthew.pearce@amadeus.co.uk</a>
Web:	<a href="http://www.amadeus.co.uk">www.amadeus.co.uk</a>



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 trademarks of their respective companies.

## APPENDIX A: AXOM SAS SERVER TOOLS – WHERE THEY FIT

