

Paper 215-30

Environments Needed for the Successful Deployment of a Data Mart/Warehouse

Mark Lucas, University of Saskatchewan, Saskatoon, Canada

ABSTRACT

The University of Saskatchewan, Office of Institutional Analysis (IA) has embarked on the development of an institutional data warehouse using the following components of the SAS 8.2 product suite: Data Warehouse Administrator, SAS IntrNet, SAS/ACCESS, SAS/CONNECT, Base SAS on Solaris Servers and Windows desktops.

INTRODUCTION

A fully functional Data Mart/Warehouse is more than databases and reports. It requires architected environments that provide Staging Areas, ETL environments and a WEB delivery environment. IA recognized the need for those environments and that the development of those environments were/are crucial to the successful deployment of our Data Warehouse. This paper will describe our key requirements for each of those environments, the structure of those environments and the role that SAS products played in the development of those environments.

ENVIRONMENTS NECESSARY FOR SUCCESSFUL DEVELOPMENT AND DEPLOYMENT

One of the most important aspects of building data marts or data warehouses is the requirement to build and support multiple environments which allow your developers and testers to:

- Make changes to the reports/views
- Add new subject areas
- Add or update dimensions
- Verify and test the accuracy of the data
- Experiment with joining/conforming subject areas

without effecting each other and/ or the production reports/data.

We believe that four environments are required:

- A development environment where data analysts, ETL specialists, and report writers could add new or enhance existing components of the warehouse and unit test their changes.
- An acceptance test environment which would be used to perform the final verification of data and reports prior to deploying into production.
- A production mirror environment. This environment mirrors production reports, WEB pages and data. At the U of S this environment serves two purposes:
 - the final staging area to verify that all the components (WEB pages, SAS reports and datasets) are properly synchronized prior to moving into the production environment. Production and Production mirror will be out of synch for short periods of time as changes are moved from the acceptance test environment, through prod mirror and into production.
 - a test environment to help diagnose and duplicate any problems that are reported in the production environments.
- Production environment

HOW THE ENVIRONMENTS ARE BUILT

We built the four environments using:

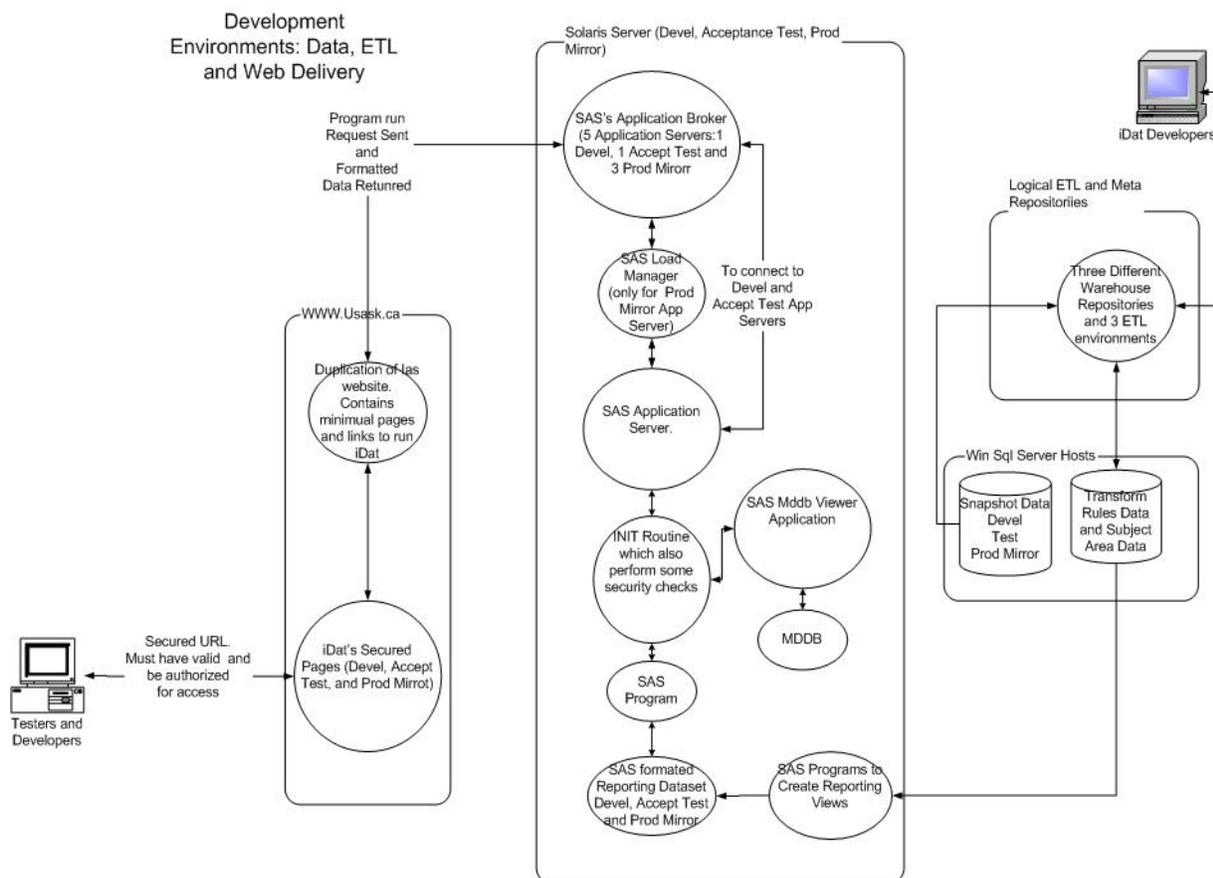
- Windows desktops
 - Base Sas
 - Data Warehouse Administrator
 - SAS/ACCESS (Sql Server ODBC)
 - SQL Server Enterprise Manager (Microsoft)
- Two Solaris Servers hosting (one for production, one for Development, Acceptance Test and Production Mirror)
 - a Apache WEB Server
 - a SAS IntrNet server
 - Application Dispatcher
 - Application Broker
 - Application Server (Type Socket)
 - Load Manager
 - MDDB Viewer
 - SAS Connect
 - the Warehouse Admin Repositories for each environment
 - the SAS ETL scripts created by SAS Warehouse Administrator.

- SAS datasets and SAS MDDDB objects which are accessed via the Web
- Samba Shares to aid in file transfers between environments. Three Windows NT SQL Server Database Servers (Development, Test, Production)
- all three servers contain all the data needed to run ETL scripts.

COMPONENTS OF THE STAGING AREA (FIGURE 1)

- Developer's interface to the four environments is controlled via a Microsoft Access form which contains a 'button' for each environment. Pushing a 'button' passes a parameter to a configuration file which then establishes the:
 - remote libraries
 - remote connections
 - warehouse repositories
 - local desktop ODBC connections required for each environment
 - starts up SAS Windows (Base SAS and Data Warehouse Admin). The SAS application window name is set to reflect the environment you have connected to. You can run one instance of SAS on your PC for each environment.
- The WEB presentation environment for Development, Acceptance Test and Production Mirror are accessed using three separate URLs. These URLs are secured and user authentication is provided by local group auth files which use a Kerberos authentication server.
- Development Solaris Server SAS IntrNet environment
 - One Broker which:
 - directly connects to one instance of a Development Application Server
 - directly connects to one instance of an Acceptance Test Application Server
 - connects via the Load Manager to one of three instances of the Production Mirror Application server. The Load Manager is set up on Production mirror as a duplicate of the Production environment.
 - Note, an Application Server can only respond to one request at a time and if performance is a concern you can run multiple instances of an Application Server and the Load Manager intelligently directs requests to non busy application servers.
- Production Solaris Server SAS IntrNet environment
 - Three Application Brokers
 - public access broker which handles requests for public information
 - private secured broker which handles requests for restricted/private information. Access to this broker is controlled via a local group auth file which uses a Kerberos authentication server.
 - Developers(debug) broker. This broker is secured with a local group auth file which uses a Kerberos authentication server. But this broker is configured to accept debug parameters which allows developers to see the SAS log associated with a request being run from the Web.
 - The three Application brokers connect to, via the Load Manager, 3 instances of the Production Application Server.
- The WEB presentation environment for Production is accessed by one public and one private URL.
- As more information is deployed and if demand warrants its, the local auth files used to control access to secured sites will be replace with Role Based Security model which will be built using SAS products.
- The snapshot files, data tables containing rules for transforms and the transformed data are stored on three separate Windows boxes running SQL Server; devel, acceptance test and prod mirror.
- SAS programs are used to extract data from the SQL server databases and create the final SAS datasets
- SAS/CONNECT and SAS/ACCESS are installed on both Solaris servers. SAS/CONNECT:
 - through Remote Compute Services allows us to submit programs, developed on Windows Desktops, to run on the Solaris servers
 - through Remote Library Services allows the SAS code running on Windows desktops to connect to datasets stored on the Solaris Servers.
- SAS/ACCESS allows SAS programs running on the Solaris servers to connect to and talk to the SQL Server Databases.
- Base Sas, ODS and Proc Template are used to write the SAS code, control the presentation format style of the SAS output.
- MDDDB Viewer is used to view the cubes in any of the environments.
- We have yet to pick a code management tool and hope to explore SAS code management product.

Figure 1: Development Environments



ETL ARCHITECTURE (FIGURE 2)

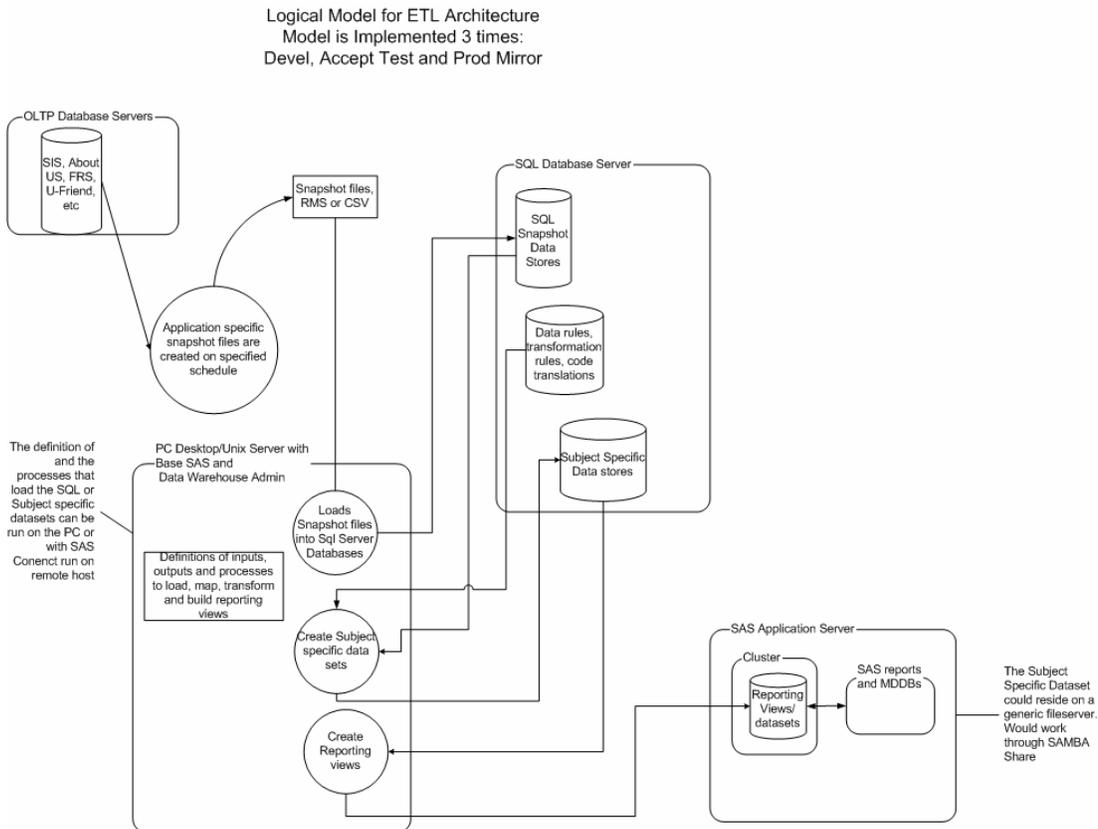
ETL environment had to meet the following requirements:

- Provide access to multiple sources of data in the same process
- Provide the ability to generate data stores in multiple formats
- Provide the ability to integrate with multiple environments, Devel, Acceptance Test and Production.
- Provide the ability to build cubes for data verification and Ad-Hoc Analysis.
- Provide the ability to create repeatable processes.

The SAS Data Warehouse Admin product combined with SAS/Access and SAS/Connect allowed us to:

- Create acceptance Test, Devel and Production repositories
- Connect to Text files, Oracle and SQL Server databases
- Run large cpu and I/O intensive jobs on remote Solaris Servers by exporting SAS code from the Data Warehouse Admin tool and scripting it to run on a Solaris Server
- Build MDDB cubes that improve the speed and reliability of data verification and data quality activities.
- Submit SAS code developed on PC's to run remotely on SOLARIS servers using the RSUBMIT command.

Figure 2: ETL Architecture



WIDA: WEB INFORMATION DELIVERY ARCHITECTURE (FIGURE 3)

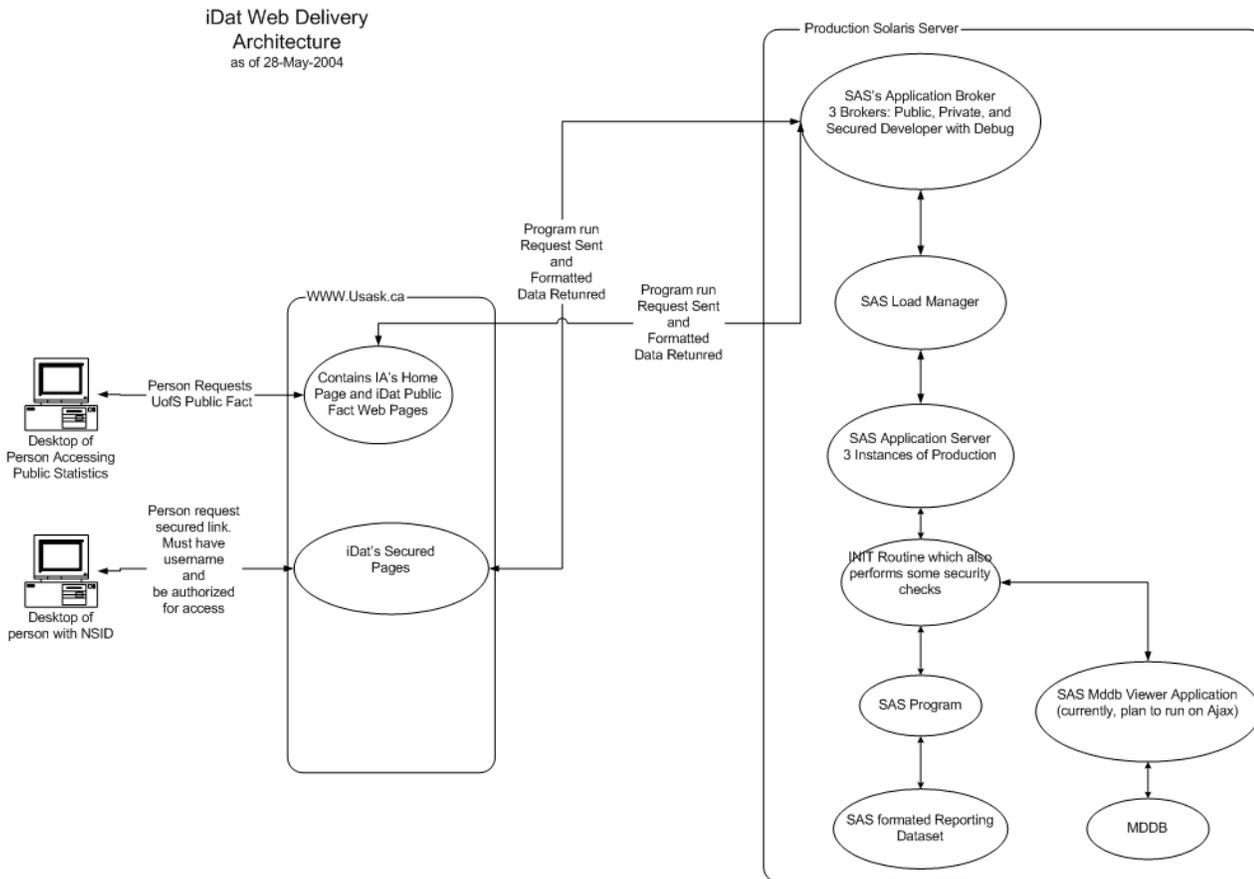
The WIDA environment had to meet five primary objectives:

- Display Public information about the University to IA's many clients
- Provide a Secured environment to control access to confidential information that is presented via Cubes, Web reports or Excel files.
- Handle concurrent requests and provide scalability.
- Provide information to IA's clients in formats that do not require additional licensed software, other than Acrobat reader on the clients desktop.
- Provide access to data which could be updated quickly to reflect new information as it becomes available.

The SAS IntrNet product set combined with SAS ODS met those 5 primary requirements.

- SAS IntrNet combined with ODS has allowed us to display/present information to our clients using PDF, HTML or Excel. The Public information, currently, is presented in PDF. We made use of the PROC Template to control foreground, background colors, margins, table line widths and font sizes. The ability to deliver information in PDF, HTML or Excel has allowed us to deliver information to our many and varied clients without requiring our clients to purchase any software.
- The combination of Web .htaccess files, a Kerberos Authentication server and SAS IntrNet allows use to create private and secured Websites. Currently, few people require access to the private sites. If the numbers of people needing access increase dramatically a role based security model could be built using SAS. There is at least one SUGI paper discussing the concept of such a model.
- The SAS Application Broker and SAS Load Manager can be configured with multiple instances of the application server. This allows the Application broker to respond to more than one concurrent request and provides the ability to scale.
- Version 8.2 of SAS provides a MDDB viewer application which allows people to view and manipulate an MDDB cube via a web browser without having to download or install any additional software.

FIGURE 3: Web Information Delivery Architecture



CONCLUSION

The SAS products mentioned in this paper and deployed as described above has provided the University of Saskatchewan with powerful, flexible and scaleable environments and the tools needed to build data marts/warehouses and distribute mission critical information to the people that need that information.

CONTACT INFORMATION

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

Mark Lucas
 Institutional Analysis
 University of Saskatchewan
 105 Maintenance Road
 Saskatoon SK S7N 5C4
 Canada
 Business: 306 966 7817
 Fax: 306 966 7819
 Email: Mark.Lucas@usask.ca
 Web: www.usask.ca/ia/

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.